POLITECHNIKA GDAŃSKA

The author of the doctoral dissertation: Daniel Korzekwa
Scientific discipline: Technical Informatics and Telecommunications

DOCTORAL DISSERTATION

Title of doctoral dissertation: Automated detection of pronunciation errors in non-native English speech employing deep learning

Title of doctoral dissertation (in Polish): Automatyczna detekcja błędów wymowy z wykorzystaniem głębokiego uczenia maszynowego w celu wsparcia nauki języka

Supervisor *signature*	Second supervisor *signature*
Prof. Bożena Kostek (Ph.D., D.Sc., Eng.)	<Title, degree, first name and surname>
Auxiliary supervisor *signature*	Cosupervisor *signature*
<Title, degree, first name and surname>	<Title, degree, first name and surname>

Gdańsk, year 2022

POLITECHNIKA GDAŃSKA

OPIS ROZPRAWY DOKTORSKIEJ

Autor rozprawy doktorskiej: Daniel Korzekwa

Tytuł rozprawy doktorskiej w języku polskim: Automatyczna detekcja błędów wymowy z wykorzystaniem głębokiego uczenia maszynowego w celu wsparcia nauki języka

Tytuł rozprawy w języku angielskim: Automated detection of pronunciation errors in non-native English speech employing deep learning

Język rozprawy doktorskiej: angielski

Promotor rozprawy doktorskiej: prof. dr hab. inż. Bożena Kostek

~~**Drugi promotor rozprawy doktorskiej***: <imię, nazwisko>~~

~~**Promotor pomocniczy rozprawy doktorskiej***: <imię, nazwisko>~~

~~**Kopromotor rozprawy doktorskiej***: <imię, nazwisko>~~

Data obrony:

Słowa kluczowe rozprawy doktorskiej w języku polskim: nauka wymowy wspomagana komputerowo, automatyczna detekcja błędów wymowy, synteza mowy, konwersja mowy, mowa dyzartryczna, głębokie uczenie maszynowe

Słowa kluczowe rozprawy doktorskiej w języku angielskim: computer-assisted pronunciation training, automated pronunciation error detection, speech synthesis, voice conversion, dysarthric speech, deep learning

Streszczenie rozprawy w języku polskim:

Pomimo znacznego postępu, jaki dokonał się w ostatnich latach, istniejące metody wspomaganego komputerowo treningu wymowy CAPT (ang. Computer-Assisted Pronunciation Training) wykrywają błędy wymowy ze stosunkowo niską dokładnością (precyzja rzędu 60% przy wskaźniku czułości 40%-80%). W niniejszej pracy doktorskiej zaproponowano nowe techniki głębokiego uczenia do wykrywania błędów wymowy w nierodzimej (L2) mowie angielskiej, przewyższając aktualny stan wiedzy w metryce wskaźnika pola AUC (Area under the Curve) o 41%, tj. z 0.528 do 0.749. Ze względu na małą dostępność baz adnotowanej mowy z błędami wymowy, potrzebnych do wiarygodnego treningu modeli głębokich, problem wykrywania błędów wymowy został przeformułowany na zadanie generowania syntetycznej mowy z błędami (L2) wymowy. W ten sposób w procesie syntezy mowy tworzone są dane treningowe do efektywnej detekcji błędów wymowy. Ponadto, aby wyeliminować potrzebę transkrypcji mowy nierodzimej na poziomie fonetycznym, zaproponowano nowatorską technikę wielozadaniową typu end-to-end do bezpośredniego wykrywania błędów wymowy. Opracowane modele zostały zastosowane w firmie Amazon do automatycznego wykrywania błędów wymowy w mowie syntetycznej w celu przyspieszenia badań nad nowymi technikami syntezy mowy. Pokazano, że zastosowane metody uczenia głębokiego aplikują się w zadaniach wykrywania i rekonstrukcji mowy dyzartrycznej.

POLITECHNIKA GDAŃSKA

Streszczenie rozprawy w języku angielskim:

Despite significant advances in recent years, the existing Computer-Assisted Pronunciation Training (CAPT) methods detect pronunciation errors with a relatively low accuracy (precision of 60% at 40%-80% recall). This Ph.D. work proposes novel deep learning methods for detecting pronunciation errors in non-native (L2) English speech, outperforming the state-of-the-art method in AUC metric (Area under the Curve) by 41%, i.e., from 0.528 to 0.749. One of the problems with existing CAPT methods is the low availability of annotated mispronounced speech needed for reliable training of pronunciation error detection models. Therefore, the detection of pronunciation errors is reformulated to the task of generating synthetic mispronounced speech. Intuitively, if we could mimic mispronounced speech and produce any amount of training data, detecting pronunciation errors would be more effective. Furthermore, to eliminate the need to align canonical and recognized phonemes, a novel end-to-end multi-task technique to directly detect pronunciation errors was proposed. The pronunciation error detection models have been used at Amazon to automatically detect pronunciation errors in synthetic speech to accelerate the research into new speech synthesis methods. It was demonstrated that the proposed deep learning methods are applicable in the tasks of detecting and reconstructing dysarthric speech.

** niepotrzebne skreślić*

*** dotyczy rozpraw doktorskich napisanych w innych językach, niż polski lub angielski*

GDAŃSK UNIVERSITY OF TECHNOLOGY

DESCRIPTION OF DOCTORAL DISSERTATION

The Author of the doctoral dissertation: Daniel Korzekwa

Title of doctoral dissertation: Automated detection of pronunciation errors in non-native English speech employing deep learning

Title of doctoral dissertation in Polish: Automatyczna detekcja błędów wymowy z wykorzystaniem głębokiego uczenia maszynowego w celu wsparcia nauki języka

Language of doctoral dissertation: English

Supervisor: Prof. Bożena Kostek (Ph.D., D.Sc., Eng.)

~~**Second supervisor***: <first name, surname >~~

~~**Auxiliary supervisior***: <first name, surname >~~

~~**Cosupervisior***: <first name, surname >~~

Date of doctoral defense:

Keywords of doctoral dissertation in Polish: nauka wymowy wspomagana komputerowo, automatyczna detekcja błędów wymowy, synteza mowy, konwersja mowy, mowa dyzartryczna, głębokie uczenie maszynowe

Keywords of doctoral dissertation in English: computer-assisted pronunciation training, automated pronunciation error detection, speech synthesis, voice conversion, dysarthric speech, deep learning

Summary of doctoral dissertation in Polish:

Pomimo znacznego postępu, jaki dokonał się w ostatnich latach, istniejące metody wspomaganego komputerowo treningu wymowy CAPT (ang. Computer-Assisted Pronunciation Training) wykrywają błędy wymowy ze stosunkowo niską dokładnością (precyzja rzędu 60% przy wskaźniku czułości 40%-80%). W niniejszej pracy doktorskiej zaproponowano nowe techniki głębokiego uczenia do wykrywania błędów wymowy w nierodzimej (L2) mowie angielskiej, przewyższając aktualny stan wiedzy w metryce wskaźnika pola AUC (Area under the Curve) o 41%, tj. z 0.528 do 0.749. Ze względu na małą dostępność baz adnotowanej mowy z błędami wymowy, potrzebnych do wiarygodnego treningu modeli głębokich, problem wykrywania błędów wymowy został przeformułowany na zadanie generowania syntetycznej mowy z błędami (L2) wymowy. W ten sposób w procesie syntezy mowy tworzone są dane treningowe do efektywnej detekcji błędów wymowy. Ponadto, aby wyeliminować potrzebę transkrypcji mowy nierodzimej na poziomie fonetycznym, zaproponowano nowatorską technikę wielozadaniową typu end-to-end do bezpośredniego wykrywania błędów wymowy. Opracowane modele zostały zastosowane w firmie Amazon do automatycznego wykrywania błędów wymowy w mowie syntetycznej w celu przyspieszenia badań nad nowymi technikami syntezy mowy. Pokazano, że zastosowane metody uczenia głębokiego aplikują się w zadaniach wykrywania i rekonstrukcji mowy dyzartrycznej.

GDAŃSK UNIVERSITY OF TECHNOLOGY

Summary of doctoral dissertation in English:

Despite significant advances in recent years, the existing Computer-Assisted Pronunciation Training (CAPT) methods detect pronunciation errors with a relatively low accuracy (precision of 60% at 40%-80% recall). This Ph.D. work proposes novel deep learning methods for detecting pronunciation errors in non-native (L2) English speech, outperforming the state-of-the-art method in AUC metric (Area under the Curve) by 41%, i.e., from 0.528 to 0.749. One of the problems with existing CAPT methods is the low availability of annotated mispronounced speech needed for reliable training of pronunciation error detection models. Therefore, the detection of pronunciation errors is reformulated to the task of generating synthetic mispronounced speech. Intuitively, if we could mimic mispronounced speech and produce any amount of training data, detecting pronunciation errors would be more effective. Furthermore, to eliminate the need to align canonical and recognized phonemes, a novel end-to-end multi-task technique to directly detect pronunciation errors was proposed. The pronunciation error detection models have been used at Amazon to automatically detect pronunciation errors in synthetic speech to accelerate the research into new speech synthesis methods. It was demonstrated that the proposed deep learning methods are applicable in the tasks of detecting and reconstructing dysarthric speech.

**delete where appropriate*

***applies to doctoral dissertations written in other languages, than Polish or English*

GDAŃSK UNIVERSITY OF TECHNOLOGY

THE "IMPLEMENTATION DOCTORATE" PROGRAM OF THE MINISTER OF EDUCATION AND SCIENCE AT GDANSK UNIVERSITY OF TECHNOLOGY

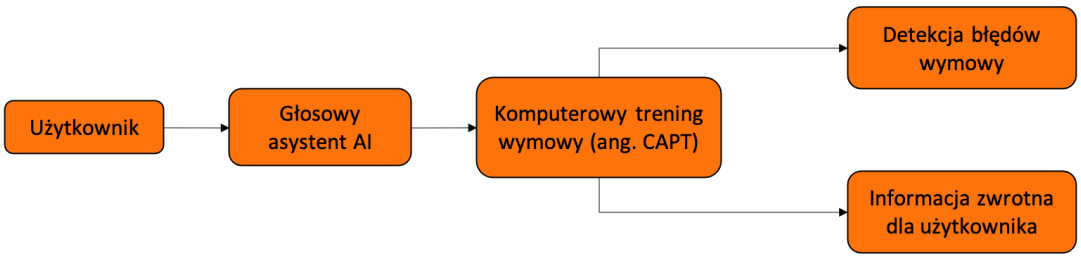
This Ph.D. was conducted within the framework of the "Implementation Doctorate" Program of the Ministry of Education and Science at Gdansk University of Technology, under the supervision of prof. Bożena Kostek (Gdańsk University of Technology) and Roberto Barra-Chicote, Ph.D. (TTS Research, Amazon).

Rozszerzone streszczenie w j. polskim

Cel pracy doktorskiej

Język odgrywa kluczową rolę w edukacji, dając ludziom dostęp do dużej ilości informacji zawartych w książkach, notatkach i pamiętnikach spisywanych na przestrzeni wieków. Niestety edukacja nie jest dostępna jednakowo dla wszystkich ludzi. Według raportu UNESCO 40% światowej populacji nie ma dostępu do edukacji w języku, który rozumieją (UNESCO, 2016). Jeszcze trudniejszy wydaje się przypadek nauki języka obcego, bowiem, w tym przypadku działa zasada: „jeśli nie rozumiesz, jak możesz się uczyć?". Nauka języka wspomagana komputerowo (ang. Computer-Assisted Language Learning (CALL)) (Asrifan et al., 2020) jest jednym z możliwych rozwiązań, które mogą poprawić znajomość języka angielskiego w różnych regionach świata. CALL opiera się na narzędziach komputerowych, które są wykorzystywane przez uczniów do ćwiczenia języka, zwykle języka obcego (w mowie nierodzimej).

Niniejsza praca doktorska poświęcona jest zagadnieniom związanym z wykrywaniem błędów w wymowie i treningiem wymowy przez osoby uczące się języka angielskiego (Computer-Assisted Pronunciation Training; CAPT) (Fouz-González, 2015) - jest to element systemu CALL. System CAPT składa się z dwóch części: modułu automatycznej oceny wymowy i modułu informacji zwrotnej, jak pokazano na rysunku 1. Moduł automatycznej oceny wymowy jest odpowiedzialny za wykrywanie błędów wymowy, na przykład za wykrywanie niepoprawnie wymawianych fonemów lub słów. Moduł informacji zwrotnej informuje użytkownika o błędnie wymawianych słowach i podpowiada, jak je poprawnie wymówić.



RYSUNEK 1: Ogólny schemat komputerowego systemu do nauki wymowy (ang. CAPT).

W szczególności, niniejsza rozprawa koncentruje się na automatycznej ocenie

wymowy. Pomimo badań poświęconych automatycznej ocenie wymowy prowadzonych intensywnie przez kilka ostatnich dekad, nadal istnieje duży potencjał w kontekście poprawy dokładności automatycznego wykrywania błędów wymowy. Istniejące metody wykrywają błędy wymowy ze stosunkowo niską dokładnością (precyzja rzędu 60% przy wskaźniku czułości 40%-80%) (Leung et al., 2019; Korzekwa, Lorenzo-Trueba, Zaporowski, et al., 2021; Z. Zhang et al., 2021). Wskazywanie poprawnie wymawianych słów jako błędów wymowy przez narzędzie CAPT może zdemotywować osobę uczącą się języka i wpłynąć na jakość nauki. Z kolei, pomijanie błędów wymowy może spowolnić proces uczenia się.

Tezy i tło badawcze

W odpowiedzi na cel badawczy, jakim jest poprawa dokładności wykrywania błędów wymowy w nierodzimej (L2) mowie angielskiej, sformułowano podstawową tezę badawczą:

Możliwe jest zwiększenie dokładności metod uczenia głębokiego do wykrywania błędów wymowy w nierodzimej mowie angielskiej poprzez zastosowanie syntetycznego generowania mowy i bezpośredniej detekcji błędów typu end-to-end, które zmniejszają zapotrzebowanie na nagrania i fonetyczną transkrypcję mowy.

Oprócz podstawowej tezy badawczej, w celu zbadania możliwości uogólniania zaproponowanych metod wykrywania błędów wymowy w pokrewnym obszarze mowy dyzartrycznej, sformułowana została druga teza badawcza.

Metody uczenia głębokiego służące do wykrywania błędów wymowy w nierodzimej mowie angielskiej można przenieść na pokrewne zadania wykrywania i rekonstrukcji mowy dyzartrycznej.

Wykrywanie błędów wymowy w nierodzimej mowie

Błąd wymowy w mowie można zdefiniować jako przypadek, kiedy osoba wymawia słowo lub zdanie inaczej niż wymowa oczekiwana według kanonicznej transkrypcji fonetycznej (Witt et al., 2000). Błędy w wymowie mogą odnosić się np. do błędnie wymawianych fonemów, np. błędne wymówienie fonemu /eh/ jako /ey/ w zdaniu "I said" /ay s eh d/. Błąd akcentu leksykalnego (Ferrer et al., 2015) to inny rodzaj błędu wymowy, który pojawia się, gdy osoba podkreśla nieprawidłową sylabę w słowie, na przykład, niepoprawne podkreślenie pierwszej sylaby w słowie „remind” /r iy1 m ay0 n d/. Błędy wymowy mogą występować na różnych poziomach szczegółowości, na przykład, na poziomie fonemów (Leung et al., 2019), słów (Korzekwa, Lorenzo-Trueba, Drugman, Calamaro, et al., 2021), lub zdań (Gong et al., 2022).

Wydaje się, że wykrycie błędu wymowy na poziomie fonemów powinno być dobrym rozwiązaniem dla osoby uczącej się, ale system tego typu może okazać się zbyt skomplikowany. Rzadko kiedy osoby uczące się języka znają pojęcie fonemu. Ponadto, automatyczne rozpoznanie wymawianych fonemów nie jest proste (Z. Zhang et al., 2021). Dlatego, nauczyciel języka nie zawsze przekazuje informację zwrotną na poziomie fonemów, zamiast tego, wskazuje źle wymówione słowo i pokazuje, jak je poprawnie wymówić. Podobnie, Asystent CAPT oparty na sztucznej inteligencji może przekazywać użytkownikowi informację zwrotną za pomocą syntetycznego głosu. Dodatkowo, w ten sposób użytkownik może ćwiczyć umiejętność wymowy za pośrednictwem interfejsu głosowego.

W ramach pracy doktorskiej zaproponowano różne modele do wykrywania zarówno błędnie wymawianych fonemów (Beringer et al., 2020; Korzekwa, Lorenzo-Trueba, Zaporowski, et al., 2021; Korzekwa, Lorenzo-Trueba, Drugman, Calamaro, et al., 2021; Korzekwa, Lorenzo-Trueba, Drugman, and Kostek, 2022), jak i błędów akcentu leksykalnego (Korzekwa, Barra-Chicote, Zaporowski, et al., 2021), na poziomie fonemów i słów. Jednak kierunek w którym te modele ewoluują – w kierunku wykrywania błędów wymowy na poziomie słowa – jest motywowany przypadkiem użycia ćwiczenia umiejętności wymowy z wykorzystaniem interfejsu asystenta głosowego opartego na sztucznej inteligencji, jak pokazano na rysunku 1.

Na podstawie literatury można zauważyć, że istniejące metody wykrywania błędów wymowy nie sprawdzają się w różnym kontekście. Obserwacje te prowadzą do nowych modeli głębokiego uczenia się w celu poprawy dokładności wykrywania błędów wymowy i usprawnienia działania narzędzi CAPT:

1. Transkrypcja mowy obcej jest trudna i kosztowna

   Końcowym wynikiem modelu wykrywania błędów wymowy jest prawdopodobieństwo błędu wymowy na poziomie segmentu, takiego jak fonem lub słowo. Stworzenie modelu, który nie wymaga rozpoznania wypowiedzianych fonemów i bezpośrednio (ang. end-2-end) szacuje to prawdopodobieństwo, może sprawić, że transkrypcje fonetyczne mowy obcej staną się niepotrzebne (Z. Zhang et al., 2021; Korzekwa, Lorenzo-Trueba, Drugman, Calamaro, et al., 2021).

2. Dokładne dopasowanie kanonicznych i rozpoznanych fonemów jest skomplikowane

   Aby wykryć błędy wymowy, istniejące metody CAPT rozpoznają wymawiane fonemy, a następnie porównują je z oczekiwaną (kanoniczną) wymową osoby mówiącej w języku rodzimym (Witt et al., 2000; K. Li, Qian, and Meng, 2016; Sudhakara, Ramanathi, Yarra, and Ghosh, 2019; Leung et al., 2019). Wykrywanie błędów wymowy bezpośrednio przez model (ang. end-to-end) może wyeliminować proces dopasowania fonemów jako potencjalne źródło błędów negatywnie wpływających na dokładność wykrywania błędów wymowy.

3. Nie wszystkie błędy wymowy są tak samo istotne dla osoby uczącej się języka

   Niektóre błędy wymowy są bardziej istotne niż inne. Kategoryzacja błędów wymowy według poziomu istotności pozwala zgłaszać osobie uczącej się tylko poważniejsze błędy i zmniejsza ryzyko wykrycia poprawnie wymawianego tekstu jako błędu wymowy (Yan, M.-C. Wu, et al., 2020; Korzekwa, Lorenzo-Trueba, Drugman, Calamaro, et al., 2021).

4. Zdanie można wymówić poprawnie na wiele różnych sposobów

   Osoby mówiące w języku rodzimym mogą wymawiać ten sam tekst poprawie na wiele sposobów. Model wykrywania błędów wymowy powinien brać to pod uwagę. Uwzględnienie zmienności w wymowie zmniejszy prawdopodobieństwo zgłaszania użytkownikowi fałszywych alarmów dotyczących jego wymowy (Qian et al., 2010; Korzekwa, Lorenzo-Trueba, Zaporowski, et al., 2021).

5. Ćwiczenie akcentu leksykalnego jest ważną częścią CAPT

   Istniejące metody CAPT koncentrują się na ćwiczeniu wymowy fonemów (Witt et al., 2000; Leung et al., 2019; Korzekwa, Lorenzo-Trueba, Zaporowski, et al., 2021). Niemniej jednak wykazano, że ćwiczenie akcentu leksykalnego poprawia zrozumiałość nierodzimej mowy w języku angielskim (Field, 2005; Lepage et al., 2014). Dobre modele uczenia głębokiego powinny być w stanie wykryć zarówno błędy w wymawianych fonemach, jak i błędy akcentu leksykalnego.

6. Dostępność mowy nierodzimej z błędami wymowy jest ograniczona

   Modele uczenia głębokiego działają bardzo dobrze, gdy ilość danych treningowych jest duża (Shah et al., 2021). Istnieją dowody w pokrewnej dziedzinie wizji komputerowej, że generowanie obrazów syntetycznych poprawia dokładność modeli klasyfikacyjnych (Wong et al., 2016). Dlatego, podobna technika może poprawić dokładność wykrywania błędów wymowy w nierodzimej mowie. Powielanie ilości danych (ang. data augmentation) (Badenhorst et al., 2017) i generowanie danych (ang. data generation) (A. Lee et al., 2016) to dwie techniki, które pomagają tworzyć syntetyczne błędy wymowy w celu uwzględnienia ograniczonej dostępności nierodzimej mowy z błędami wymowy. Ostatnie postępy w syntezie mowy (Fazel et al., 2021) i konwersji głosu (Shah et al., 2021) otwierają możliwość generowania mowy syntetycznej, która będzie w stanie doskonale naśladować nierodzimą mowę i pozwoli na trenowanie modeli wykrywania błędów wymowy tylko na danych syntetycznych.

7. Wielozadaniowe uczenie maszynowe (ang. multi-tasking) jako podejście do walki z nadmiernym dopasowaniem (ang. overfitting) w metodach głębokiego uczenia się

W wielozadaniowym uczeniu maszynowym, oprócz podstawowego zadania wykrywania błędów wymowy w sygnale mowy, można dodać zadanie drugorzędne, takie jak rozpoznawanie wymawianych fonemów (Z. Zhang et al., 2021; Korzekwa, Lorenzo-Trueba, Drugman, Calamaro, et al., 2021). Oba zadania będą ze sobą współdziałać, dzięki czemu model będzie mniej podatny na nadmierne dopasowanie.

Wykrywanie i rekonstrukcja mowy dyzartrycznej

Pożądanymi cechami metod uczenia maszynowego są możliwość łatwego uogólnienia oraz skalowalność w kontekście innych powiązanych problemów. Druga teza badawcza ma na celu zbadanie, czy metody głębokiego uczenia można stosować w zadaniach wykrywania i rekonstrukcji mowy dyzartrycznej.

Dyzartria jest motorycznym zaburzeniem mowy, które wynika z zaburzeń neurologicznych, takich jak porażenie mózgowe, udar mózgu/afazja, otępienie i torbiel mózgu (M. Cuny et al., 2017; Banovic, L. J. Zunic, et al., 2018). Z powodu uszkodzenia układu nerwowego, połączenia pomiędzy mózgiem a narządem mowy i ich mięśniami ulegają osłabieniu, co skutkuje zniekształceniem mowy (ASHA, 2022). W porównaniu z normalną mową, mowa dyzartryczna jest szorstka i zawiera zwiększoną ilość oddechów, zawiera błędy w wymowie, ma spłaszczoną intonację i zmniejszoną prędkość mówienia.

Można postawić hipotezę, że modele uczenia głębokiego używane do automatycznego wykrywania błędów wymowy w nierodzimej mowie mogą zostać przeniesione do zadania wykrywania mowy dyzartrycznej, lub szerzej, upośledzonej mowy, takiej jak w chorobie Parkinsona (PD) (Korzekwa, Barra-Chicote, Kostek, et al., 2019; Romana et al., 2021). Zarówno w nierodzimej, jak i dyzartrycznej mowie, można zaobserwować podobne zniekształcenia mowy, takie jak błędna wymowa fonemów i nieprawidłowe wzorce prozodii. Dlatego wydaje się zasadne postawienie hipotezy badawczej, że podobne modele uczenia głębokiego mogą mieć zastosowanie w obu obszarach.

Osoby z dyzartrią mają trudności z porozumiewaniem się z innymi ludźmi, ponieważ ich mowa jest zniekształcona i mniej zrozumiała. Istnieją podobieństwa pomiędzy generowaniem mowy syntetycznej imitującej nierodzimą mowę w celu poprawy dokładności wykrywania błędów wymowy a rekonstrukcją mowy dyzartrycznej w celu uczynienia mowy bardziej zrozumiałą. W scenariuszu detekcji błędów wymowy, system konwersji mowy na mowę (ang. speech-to-speech) służy do 'niszczenia' poprawnie wymawianej mowy poprzez wprowadzanie błędów wymowy, albo poprzez podmianę fonemów lub poprzez wprowadzenie niepoprawnego wzorca stresu leksykalnego. W scenariuszu mowy dyzartrycznej, mowa ta jest 'naprawiana' tak aby była bardziej płynna, np., poprzez automatyczne usunięcie niepotrzebnych przerw między fonemami i sylabami, oraz aby wypowiedziane fonemy były bardziej zrozumiałe. Można postawić hipotezę, że podobne techniki uczenia głębokiego powinny być skuteczne w obu scenariuszach.

Publikacje i wkład naukowy

W ramach prowadzonych badań powstało sześć publikacji, w których autor rozprawy jest głównym autorem (Korzekwa, Lorenzo-Trueba, Drugman, and Kostek, 2022; Korzekwa, Lorenzo-Trueba, Drugman, Calamaro, et al., 2021; Korzekwa, Lorenzo-Trueba, Zaporowski, et al., 2021; Korzekwa, Barra-Chicote, Zaporowski, et al., 2021; Korzekwa and Kostek, 2019; Korzekwa, Barra-Chicote, Kostek, et al., 2019). Publikacje te są bezpośrednio związane z tezami badawczymi przedstawionymi w rozdziale 1.3 i stanowią główny wkład naukowy rozprawy doktorskiej.

Dodatkowo, z tematem rozprawy doktorskiej wiąże się dziewięć publikacji, których współautorem jest Daniel Korzekwa. Pierwsze dwie publikacje poświęcone są tematyce automatycznej detekcji błędów wymowy w nierodzimej mowie (D. Zhang et al., 2022; Beringer et al., 2020). Kolejne sześć publikacji dotyczy syntezy mowy i konwersji głosu, które kładą podwaliny pod generowanie syntetycznych błędów wymowy i rekonstrukcję mowy dyzartrycznej (Bilinski et al., 2022; Merritt, Ezzerg, et al., 2022; Jiao et al., 2021; Gabryś et al., 2021; Shah et al., 2021; Ezzerg et al., 2021). Dziewiąta publikacja dotyczy kolekcji nienatywnego korpusu mowy, który został wykorzystany do oceny modeli wykrywania błędów wymowy (Weber et al., 2020).

Wkład naukowy

W ramach pracy doktorskiej zaproponowano oraz opracowano wiele nowatorskich metod uczenia głębokiego do wykrywania błędów wymowy w nierodzimej mowie angielskiej, podsumowanych poniżej.

1. Wykonywanie transkrypcji fonetycznej nierodzimej mowy jest czasochłonne, a niekiedy transkrypcja ta jest niemożliwa ze względu na różnice pomiędzy językami mówionymi. Zaproponowano nową metodę do bezpośredniego (ang. end-2-end) wykrywania błędów wymowy, o nazwie WEAKLY-S (Weakly-supervised), pokazanej na rysunku 2. Ze względu na bezpośrednią detekcję błędów wymowy, metoda ta nie wymaga transkrypcji fonetycznej nierodzimej mowy (Korzekwa, Lorenzo-Trueba, Drugman, Calamaro, et al., 2021).

2. Istniejące metody wykrywania błędów wymowy dopasowują kanoniczne i rozpoznane sekwencje fonemów w celu identyfikacji błędnie wymawianych segmentów mowy, takich jak fonemy i słowa. Wszelkie niedokładności wprowadzone w procesie dopasowania obniżyłyby dokładność wykrywania błędów wymowy. Zaproponowana metoda WEAKLY-S do bezpośredniego wykrywania błędów wymowy nie wymaga dopasowywania kanonicznych i rozpoznawanych sekwencji fonemów (Korzekwa, Lorenzo-Trueba, Drugman, Calamaro, et al., 2021). Metoda ta zwiększa dokładność wykrywania błędów wymowy w metryce AUC (Area under the Curve) nawet o 30% w porównaniu do istniejących metod w literaturze.

3. Istnieją dwa czynniki, które mogą wpływać na dokładność wykrywania błędów wymowy. Po pierwsze, to samo zdanie można wymówić na wiele poprawnych sposobów, co nie powinno powodować detekcji błędu wymowy. Po drugie, dokładne rozpoznanie fonemów wymawianych przez osobę uczącą się jest trudne i należy wziąć to pod uwagę. W tym celu zaproponowano nową metodę uwzględniającą: i) wiele poprawnych sposobów wymawiania tego samego zdania oraz ii) niepewność rozpoznawania fonemów (Korzekwa, Lorenzo-Trueba, Zaporowski, et al., 2021). Zaproponowana metoda zwiększa precyzję wykrywania błędów w wymowie nawet o 18% w porównaniu z istniejącym podejściem.

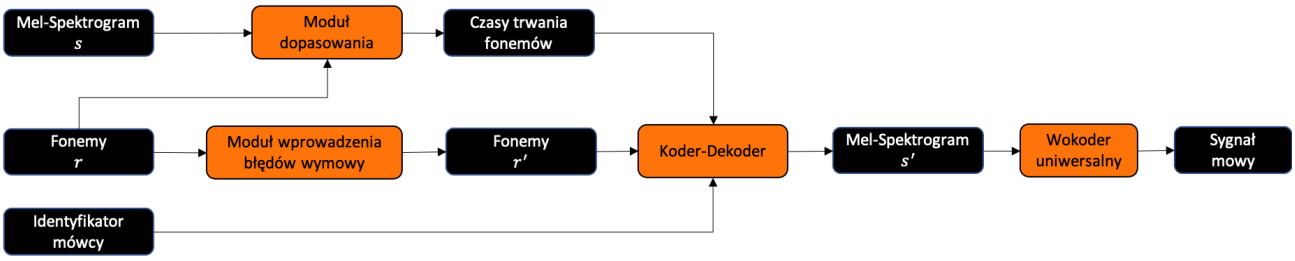
4. Istniejące metody wykrywania błędów wymowy często opierają się na ekstrahowaniu cech mowy, takich jak f0, energia, dopasowanie fonemów do sygnału mowy, itd. W pracy zaproponowano nową metodę wykrywania błędów wymowy opartą na mechanizmie uwagi (ang. attention mechanism) do automatycznego wyodrębniania optymalnych cech mowy (Korzekwa, Barra-Chicote, Zaporowski, et al., 2021). Mechanizm uwagi odgrywa istotną rolę w proponowanych modelach głębokiego uczenia stosowanych do wykrywania niepoprawnie wymówionych fonemów i błędów akcentu leksykalnego.

5. Mechanizm uwagi pomaga rozłożyć model głębokiego uczenia się na wiele zależnych składników w procesie zwanym faktoryzacją. Faktoryzacja prowadzi do lepszej interpretacji modelu głębokiego uczenia, na przykład, wizualizacji modelu do wykrywania błędów akcentu leksykalnego w celu lepszego zrozumienia, jak działa taki model i w jaki sposób podejmuje decyzje (Korzekwa, Barra-Chicote, Zaporowski, et al., 2021). Uczenie wielozadaniowe to rodzaj faktoryzacji modelu, który może sprawić, że model głębokiego uczenia będzie bardziej niezawodny i mniej podatny na nadmierne dopasowanie (Korzekwa, Lorenzo-Trueba, Drugman, Calamaro, et al., 2021). Zaproponowano wielozadaniowy model wykrywania błędów wymowy WEAKLY-S z dwoma zadaniami, a) rozpoznawanie fonemów i b) wykrywanie błędów wymowy, co zwiększa dokładność tego modelu. Faktoryzacja może również przybrać formę interpretowalnej warstwy ukrytej w modelu głębokiego uczenia (ang. latent space or hidden space), która może być wykorzystana do modyfikacji określonych cech sygnału. Na przykład, może uczynić mowę dyzartryczną bardziej płynną i zrozumiałą (Korzekwa, Barra-Chicote, Kostek, et al., 2019).

6. Dostępność nierodzimej mowy jest ograniczona, a jej nagranie/zbieranie i wykonanie transkrypcji fonetycznej jest czasochłonne. Odwołując się do teorii prawdopodobieństwa i reguły Bayesa, problem wykrywania błędów wymowy został przeformułowany jako zadanie generowania mowy, jak pokazano na rysunku 3. Intuicyjnie, w przypadku nieograniczonej ilości mowy syntetycznej, która mogłaby naśladować nierodzimą mowę, modele uczenia głębokiego

do wykrywania błędów wymowy byłyby mniej podatne na nadmierne dopasowanie. Najlepsza zaproponowana metoda generowania nierodzimej mowy (ang. speech-to-speech) zwiększa dokładność wykrywania błędów wymowy w metryce AUC o 41%, z 0.528 do 0.749, w porównaniu z istniejącym podejściem (Korzekwa, Lorenzo-Trueba, Drugman, and Kostek, 2022).

RYSUNEK 2: Architektura modelu WEAKLY-S do wykrywania błędów wymowy na poziomie wyrazów w konfiguracji wielozadaniowej. Zadanie 1 - wykrycie błędów w wymowie e. Zadanie 2 - rozpoznawanie fonemów r_o.



RYSUNEK 3: Architektura modelu zamiany mowy na mowę (ang. speech-to-speech) do generowania błędnie wymawianej mowy syntetycznej przy zachowaniu prozodii i barwy głosu mowy wejściowej. Czarne prostokąty reprezentują dane (tensory), a pomarańczowe prostokąty reprezentują bloki przetwarzania.

Zastosowanie

Modele wykrywania błędów wymowy

Zaproponowane modele CAPT do wykrywania błędów wymowy w nierodzimej mowie angielskiej zastosowano do automatycznego wykrywania błędów wymowy w mowie syntetycznej w dwóch scenariuszach: 1) podczas wnioskowania (ang. during inference) i 2) podczas treningu modeli syntezy mowy. Celem modelu CAPT podczas wnioskowania jest automatyczna ocena jakości mowy generowanej przez modele syntezy mowy, to znaczy czy mowa jest zrozumiała i nie zawiera błędów wymowy. Po wytrenowaniu modelu syntezy mowy, duża liczba wypowiedzi jest syntetyzowana i automatycznie przetwarzana przez model wykrywania błędów

wymowy. Automatyczne wykrywanie błędów wymowy umożliwia ocenę głosów syntetycznych na dużą skalę i znacznie zmniejsza liczbę testów odsłuchowych, które są przeprowadzane przez słuchaczy. Podczas trenowania, model wykrywania błędów wymowy jest używany do pomiaru czy wygenerowana mowa zawiera odpowiednie fonemy, dzięki czemu model syntezy mowy wygeneruje bardziej zrozumiałą mowę.

Synteza i konwersja mowy

Systemy syntezy mowy i konwersji głosu składają się z dwóch części, modułu generowania kontekstu, który generuje spektrogram w skali melowej z wejściowego tekstu i/lub wejściowego sygnału mowy, oraz modułu wokodera, który wytwarza surowy sygnał mowy w dziedzinie czasu na podstawie spektrogramu w skali melowej. Oba komponenty zostały wdrożone na urządzeniach Alexa i obsługują miliony użytkowników Amazon na całym świecie. Ponadto, mowa syntetyczna generowana przez modele syntezy mowy i konwersji głosu została wykorzystana podczas trenowania modeli CAPT do wykrywania błędów wymowy, poprawiając ich dokładność.

Wnioski

W ramach badań związanych z doktoratem opracowano nowatorskie metody głębokiego uczenia w celu automatycznego wykrywania błędów wymowy w nierodzimej (drugi język - L2) mowie angielskiej. Przeprowadzono rozległe eksperymenty, aby zmierzyć skuteczność proponowanych metod w CAPT. Do oceny zaproponowanych metod wykorzystano nierodzimą mowę angielską osób głównie posługujących się rodzimym językiem niemieckim, włoskim i polskim. Zarejestrowano dwa korpusy nierodzimej mowy angielskiej osób z rodzimym językiem słowiańskim i bałtyckim (Weber et al., 2020). Najlepsza zaproponowana metoda poprawia dokładność wykrywania błędów wymowy w metryce AUC o 41%, z 0.528 do 0.749, w porównaniu z istniejącym podejściem (Korzekwa, Lorenzo-Trueba, Drugman, and Kostek, 2022). Odpowiada to 80.45% w metryce precyzji i 40.12% w metryce czułości. Biorąc pod uwagę tylko poważne błędy wymowy, według subiektywnej oceny osób natywnie posługujących się językiem angielskim, AUC wzrasta z 0.749 do 0.834, co odpowiada 93.54% precyzji i 40.15% czułości. Dwie najważniejsze techniki zastosowane w tej metodzie to: 1) bezpośrednia detekcja błędów wymowy (ang. end-to-end) oraz 2) wykorzystanie techniki zamiany mowy na mowę (ang. speech-to-speech) do generowania syntetycznej mowy z błędami wymowy. Obie techniki zmniejszają zapotrzebowanie na nagrania i fonetyczną transkrypcję mowy, która jest potrzebna do trenowania modeli CAPT. Osiągnięcia te pozwoliły na udowodnienie pierwszej (głównej) tezy badawczej.

Aby zbadać możliwości uogólniania, opracowane techniki uczenia głębokiego do wykrywania błędów wymowy zostały z powodzeniem zastosowane w pokrewnym obszarze wykrywania i rekonstrukcji mowy dyzartrycznej (Korzekwa, Barra-Chicote,

Kostek, et al., 2019). Zaproponowano model autoenkodera (ang. autoencoder; uczenie nienadzorowane), aby przekodować cechy mowy dyzartrycznej na przestrzeń utajoną (ang. latent space). Kontrolując utajoną reprezentację, można poprawić płynność mowy, np., poprzez automatyczne usunięcie niepotrzebnych przerw pomiędzy fonemami i sylabami. Kontrola ta polega na automatycznym znalezieniu wektora przesunięcia w przestrzeni utajonej, który sprawi, że mowa stanie się bardziej płynna przy jednoczesnym zachowaniu innych parametrów mowy takich jak barwa głosu czy wypowiedziane fonemy. Utajoną reprezentację można wykorzystać do wykrywania mowy dyzartrycznej na poziomie słów z precyzją na poziomie 83.1% i czułością na poziomie 91.1%. Nowe techniki głębokiego uczenia zostały z powodzeniem zastosowane w temacie mowy dyzartrycznej, potwierdzając walidację drugiej tezy badawczej.

Plan na przyszłość

W trakcie pracy doktorskiej wyłoniło się wiele interesujących kierunków badawczych. Najbardziej przyszłościowym pomysłem jest kontynuacja badania nad przeformułowaniem problemu wykrywania błędów wymowy jako zadania generowania mowy (Korzekwa, Lorenzo-Trueba, Drugman, and Kostek, 2022). Zaproponowana metoda zamiany mowy na mowę (ang. speech-to-speech, S2S) może generować syntetyczną błędną wymowę, ale nie jest w stanie w pełni naśladować mowę nierodzimą. Aby udoskonalić metodę S2S, należy stworzyć uniwersalny model, aby generować dowolny rodzaj mowy. Model ten powinien być w stanie przekształcić rodzimą mowę w nierodzimą mowę, odzwierciedlając tożsamość głosu, prozodię, styl mówienia i wymowę w nierodzimej mowie. Takie podejście może sprawić, że bazy danych mowy typu L2 (mowa nierodzima) będą zbędne, ponieważ model wykrywania błędów wymowy będzie trenowany tylko na danych syntetycznych.

Innymi interesującymi kierunkami badawczymi jest zbadanie nienadzorowanych reprezentacji mowy, takich jak Wav2vec (Peng et al., 2021), oraz przeprowadzenie wielomodalnego (ang. multi-modal) wykrywania błędów wymowy poprzez wykorzystanie z audiowizualnych korpusów mowy (Czyzewski et al., 2017; Oneata et al., 2022).

Przyszłe prace skoncentrują się również na opracowaniu kompletnego systemu CAPT opartego na sztucznej inteligencji w celu podniesienia znajomości języków obcych na świecie, nie tylko języka angielskiego. W tym celu powinien zostać utworzony agent konwersacyjny oparty na sztucznej inteligencji. Agent ten będzie składał się z dwóch elementów: modułu wykrywania błędów wymowy oraz modułu informacji zwrotnej. Moduł wykrywania błędów wymowy będzie oparty na wynikach badań zawartych w rozprawie doktorskiej, natomiast moduł informacji zwrotnej będzie wymagał dodatkowych badań. System CAPT będzie kontrolowany tylko za pomocą interfejsu głosowego, a uczeń będzie miał wrażenie zajęć prowadzonych przez nauczyciela języka obcego.

Acknowledgements

I would like to thank many people who left their mark on this dissertation as well as on me as a person. Prof. Bożena Kostek, I could always count on your advice and you were always available to me no matter what time of day or day of the week. Roberto Barra-Chicote, I still remember our discussion in Cambridge on affective computing and empathetic AI, it was 2018. This discussion prompted me to do my Ph.D. research, and you, Roberto, have been with me all this time. Jaime Lorenzo-Trueba, thank you for our frequent discussions about the results of scientific experiments and research plans for the future. Thomas Drugman, thanks to you, my writing in English has improved significantly and the clarity of my publications is now much better. Szymon Zaporowski, you recorded the corpus of non-native speech, which gave me the data to evaluate my pronunciation error detection models. Shira Calamaro, you helped with the linguistic and phonetic parts of the research, and with understanding the nature of pronunciation errors made by non-native speakers. Grzegorz Beringer, your internship at Amazon on pronunciation error detection and our numerous discussions motivated me to take up this topic in my Ph.D. research. Alicja Serafinowicz, you gave me a unique perspective of an English teacher on computer-assisted pronunciation training, and created a list of words that are often mispronounced by your students. Jasha Droppo, you have advised me how important it is to keep the big picture of the research in mind. I still remember your remark about moving not only forward but in the right long-term direction. Gary Cook, Andrew Breen, Mateusz Łajszczak, Adam Nadolski, Jonah Rohnke, Viacheslav Klimkov, and Kayoko Yanagisava thank you for reviewing my publications and thesis, and providing many constructive comments. Thank you to Amazon company for allowing me to use the Amazon EC2 cloud to conduct research experiments and for 5 weeks to work on the final mile of my Ph.D. thesis. I would like to thank my beloved wife Luiza and my sons Kacper, Mateusz, and Tymoteusz. There are no words to express how grateful I am to them. Only they know how much support and dedication they gave me. Finally, I would like to thank my parents who persistently raised me in difficult times.

List of Figures

1.1 Overview of the Computer-Assisted Pronunciation Training System . 3
1.2 Overview of the Dysarthric Speech Detection System 9
1.3 The use of a pronunciation error detection model to evaluate speech synthesis 17
1.4 The use of pronunciation error detection at the training time of a speech synthesis model 18
1.5 A single universal vocoder serving multiple text-to-speech requests across multiple voices 18
1.6 A single universal vocoder serving the generation of mispronounced speech across multiple speakers 18

2.1 Arpabet phonetic alphabet (Arpabet, 2022) 20
2.2 Organs involved in the process of speech production (*University physics Volume 1* 2016) 21
2.3 Probabilistic Graphical Model (PGM) for the coin game, including two random variables; x - whether the coin is biased (comes from a pirate), y - the outcome of a single coin toss. The PGM image was created with the SamIam - a tool for modeling and reasoning in Bayesian Networks (Darwiche, 2009) 27
2.4 Probabilistic Graphical Model (PGM) for the coin game, including three random variables. x - whether the coin is biased (comes from a pirate), $y1, y2$ - the outcomes of two coin tosses. The PGM image was created with the SamIam - a tool for modeling and reasoning in Bayesian networks (Darwiche, 2009) 28
2.5 Posterior plots for different probabilistic model architectures from Figure 2.6: a) Naive Bayes - a single latent variable x estimated from multiple independent observations $\{y_1, y_2, ..., y_N\}$, b) Hidden Markov Model - a latent variable x_i conditioned on the local context of two neighboring variables x_{i-1} and x_{i+1}, c) Gaussian Process with the RBF kernel - a model with an infinite number of latent variables $\{x_1, x_2, ..., x_N\}$ conditioned on independent observations $\{y_1, y_2, ..., y_N\}$, and d) Gaussian Process with the Linear kernel - a model with an infinite number of latent variables. Each plot contains observations from the training set, the predicted mean values, and the corresponding 95% confidence interval 30

2.6 Graphical models for different probabilistic model architectures: a) Naive Bayes - a single latent variable x estimated from multiple independent observations $\{y_1, y_2, ..., y_N\}$, b) Hidden Markov Model - a latent variable x_i conditioned on the local context of two neighboring variables x_{i-1} and x_{i+1}, and c) Gaussian Process - a model with infinite number of latent variables $\{x_1, x_2, ..., x_N\}$ conditioned on independent observations $\{y_1, y_2, ..., y_N\}$ 31
2.7 Covariance plots for different probabilistic model architectures from Figure 2.6: a) Naive Bayes - a single latent variable x estimated from multiple independent observations $\{y_1, y_2, ..., y_n\}$, b) Hidden Markov Model - a latent variable x_i conditioned on the local context of two neighboring variables x_{i-1} and x_{i+1}, c) Gaussian Process with the RBF kernel - a model with infinite number of latent variables $\{x_1, x_2, ..., x_n\}$ conditioned on independent observations $\{y_1, y_2, ..., y_n\}$, and d) Gaussian Process with the Linear kernel - a model with infinite number of latent variables. The covariance function, also known as a kernel or covariance matrix, is computed with $cov(x, x')$ for all possible combinations of latent variables $\{x_1, x_2, ..., x_n\}$. The form of a $cov()$ function depends on the probabilistic model architecture 38
2.8 Neural network architectures based on the perceptron and a dense layer components: a) neural network with input vector $\mathbf{x}$ and scalar output y_1, known as the perceptron, b) neural network with input vector $\mathbf{x}$, one dense layer $\mathbf{z}$, and scalar output y_1, c) neural network with input vector $\mathbf{x}$, one dense layer $\mathbf{z}$, and vector-based output $\mathbf{y}$ . . . 41
2.9 An operation in a convolutional neural block that maps between a single z_{ij} value in the $\mathbf{z}$ output tensor (layer) and the $\mathbf{x}$ input layer. A 3x3 convolutional kernel (filter) is multiplied element-wise by the corresponding region of the $\mathbf{x}$ input layer, followed by the max-pooling operation 42
2.10 Recurrent neural network architectures. a) Recurrent network without autoregressive loop. All x_i inputs must be available in advance to the model. b) Autoregressive recurrent neural network. Only the first x_0 element must be available to the model. In general, the x_i element is computed based on the value of the previous output y_{i-1} 43
2.11 The attention mechanism illustrated by the example of computing a single element of the output sequence $\mathbf{y}$ from the input sequence $\mathbf{x}$. Q - query, K - keys, V - values 44
2.12 Architecture of the Variational Auto-Encoder (VAE) model. a) Neural network representation of the VAE model, b) Bayesian network representation of the VAE model 45

3.1 Neural network architecture of the WEAKLY-S model for word-level pronunciation error detection 57
3.2 Details of the neural network architecture of the WEAKLY-S model for word-level pronunciation error detection 57
3.3 Precision-recall curves for the WEAKLY-S and baseline models, PR-PM and PR, (a) tested on Isle Corpus of German and Italian speakers and (b) GUT Isle Corpus of Polish speakers. (c) Ablation study on the GUT Isle corpus. (d) Analysis of mispronunciation severity levels 60
3.4 Architecture of the system for detecting mispronounced words in a spoken sentence 68
3.5 Architecture of the PR, PM, and PED subsystems. l_s - the size of the phoneme set 68
3.6 Precision-recall curves for the evaluated systems 72
3.7 Attention-based Deep Learning model for the detection of lexical stress errors 74
3.8 Top: forced-alignment mapping between phonemes and frames for the word 'garage'. Middle: Frame-to-syllable attention weights matrix. Bottom: (Sub)Phoneme-to-syllable attention weights matrix 77
3.9 Precision-recall curves for evaluated systems 80
3.10 Probabilistic graphical models for three methods to generate pronunciation errors: P2P, T2S and S2S. Empty circles represent hidden (latent) variables, while filled (blue) circles represent observed variables. $\mathbf{s}$ - the speech signal, $\mathbf{r}$ - the sequence of phonemes that the user is trying to pronounce (canonical pronunciation), the superscript $'$ represents a variable with generated mispronunciations 92
3.11 Architecture of the S2S model to generate mispronounced synthetic speech while maintaining prosody and voice timbre of the input speech. The black rectangles represent the data (tensors) and the orange boxes represent processing blocks. This color notation is used in all machine learning model diagrams throughout the article 94
3.12 Architecture of the WEAKLY-S model for word-level pronunciation error detection trained in the multi-task setup. Task 1 - to detect pronunciation errors e. Task 2 - to recognize phonemes r_o 97
3.13 Precision-recall curve for the ablation study on the GUT Isle corpus, illustrating the effect of using synthetic pronunciation errors generated by the P2P method 98
3.14 Architecture of the system for detecting mispronounced words in a spoken sentence based on the native speech pronunciation model . . . 102

3.15 Precision-recall curves for the evaluated systems to measure the effect of using the PM model in detecting pronunciation errors. PR-PM - full model with the PM enabled. PR-LIK - the PR-PM model with the PM disabled. PR-NOLIK - non-probabilistic variant of the PR-LIK model proposed by Leung et al. (Leung et al., 2019) 103
3.16 Attention-based model for the detection of lexical stress errors 105
3.17 Precision-recall curves for lexical stress error detection models 106

4.1 Architecture of deep learning model for detection and reconstruction of dysarthric speech 112
4.2 Unsupervised learning. Top row: Separation between dysarthric and control speakers in the latent space on a speaker (left) and word (right) level. Bottom row: Correlation between both dimensions of the latent space and the intelligibility scores 117
4.3 Supervised learning. As in Figure 4.2 118
4.4 MUSHRA results for the fluency of speech for 5 reconstructions and one recorded speech. Rank order (left) and the median score on the scale from 0 to 100 (right) 119
4.5 Reconstruction of dysarthric speech ('command' word) 119

List of Tables

1.1 Description of EF English Proficiency Index (EPI) (EF-Education-First, 2020) 2
1.2 English proficiency by region (EF-Education-First, 2020) 2

2.1 The PMF function for the x variable presented in a tabular form (color of the ball selected from the container) 25
2.2 The Conditional Probability Table (CPT) for the variable x - whether the coin is biased (comes from a pirate) or not. 26
2.3 The Conditional Probability Table (CPT) for the variable y (the outcome of the coin) conditioned on the variable x (the coin comes from a pirate or not). 26

3.1 Summary of speech corpora used in experiments. * - audiobooks read by volunteers from all over the world (Zen et al., 2019) 61
3.2 Accuracy metrics of detecting word-level pronunciation errors. WEAKLY-S vs baseline models. 62
3.3 Ablation study for the GUT Isle corpus. 64
3.4 Severity of mispronunciation by inter-tester agreement for the GUT Isle Corpus. 1 - MINOR, 2 - MEDIUM, 3 - MAJOR. 64
3.5 Accuracy metrics for different severity levels of mispronunciation for the GUT Isle Corpus. 65
3.6 The summary of speech corpora used by the PR. 71
3.7 Precision and recall of detecting word-level mispronunciations. CI - Confidence Interval. 72
3.8 Train and test sets details. 78
3.9 Precision and recall [%, 95% Confidence Interval] of detecting lexical stress errors, at around 50% recall. * - Ferrer et al. model has been evaluated on the data with 46.4% of lexical stress errors, compared to 9.4% of errors on our data set. This data point indicates that our proposed model AttTTS should outperform Ferrr et al. model if both were evaluated exactly in the same conditions. 81
3.10 Summary of human speech corpora used in the pronunciation error detection experiments. * - audiobooks read by volunteers from all over the world (Zen et al., 2019) 95

3.11 Details of the training and test sets for the lexical stress error detection model. 96
3.12 Ablation study for the GUT Isle corpus to show the effect of using synthetic data and other elements of the WEAKLY-S model. Pr. - Precision, Re. - Recall 98
3.13 Accuracy metrics of detecting word-level pronunciation errors. WEAKLY-S vs. baseline models. 99
3.14 Comparison of the P2P, T2S and S2S methods in the task of pronunciation error detection assessed on the GUT Isle corpus. 100
3.15 Comparison of the P2P, T2S and S2S methods in the task of pronunciation error detection assessed on the GUT Isle corpus only for major pronunciation errors. 100
3.16 Accuracy (AUC) in detecting pronunciation errors assessed in synthetic speech at different severity levels of mispronunciation for the best S2S method. 101
3.17 Precision and recall of detecting word-level mispronunciations. CI - Confidence Interval. PR-PM - full model with the PM enabled. PR-LIK - the PR-PM model with the PM disabled. 104
3.18 AUC, precision and recall [%, 95% Confidence Interval] metrics for lexical stress error detection models. Att. - Model with attention. Syn. - Synthetic mispronunciations. 106

4.1 Configuration of the neural network blocks. 113
4.2 Accuracy of dysarthria detection including 95% CI. Classifier task - target mel-spectrogram (ML) is not observed during training. Multitask - both targets ML and dysarthric labels are observed 116

List of Abbreviations

Language Learning and Speech Processing

CALL	**C**omputer-**A**ssisted **L**anguage **L**earning
CAPT	**C**omputer-**A**ssisted **P**ronunciation **T**raining
EPI	**E**nglish **P**roficiency **I**ndex
ESL	**E**nglish as a **S**econd **L**anguage
GOP	**G**oodness **o**f **P**ronunciation
L1	Native language
L2	Non-native language
MCD	**M**el **C**epstral **D**istortion
MDN	**M**ispronunciations **D**etection **N**etwork
MDD	**M**ispronunciation **D**etection and **D**iagnosis
MOS	**M**ean **O**pinion **S**core
MUSHRA	**MU**ltiple **S**timuli with **H**idden **R**eference and **A**nchor
PM	**P**ronunciation **M**odel
PR	**P**honeme **R**ecognizer
PRN	**P**honeme **R**ecognition **N**etwork
TTS	**T**ext-**T**o-**S**peech

Math, Stats, and Machine Learning

AUC	**A**rea **U**nder the RO**C** Curve
CDF	**C**umulative **D**ensity **F**unction
CPT	**C**onditional **P**robability **T**able
XOR	Exclusive **OR**
EM	Expectation **M**aximization
FAR	**F**alse **A**cceptance **R**ate
FNR	**F**alse **N**egative **R**ate
FP	**F**alse **P**ositives
FPR	**F**alse **P**ositive **R**ate
FN	**F**alse **N**egatives
FRR	**F**alse **R**ejection **R**ate
GMM	**G**aussian **M**ixture **M**odel
GP	**G**aussian **P**rocess
HMM	**H**idden **M**arkov **M**odel

iid	**i**ndependently and **i**dentically **d**istributed
KLD	**K**ullback–**L**eibler **D**ivergence
ML	**M**achine **L**earning
PDF	**P**robability **D**ensity **F**unction
PGM	**P**robabilistic **G**raphical **M**odel
PMF	**P**robability **M**ass **F**unction
RBF	**R**adial **B**asis **F**unction
TN	**T**rue **N**egatives
TP	**T**rue **P**ositives
TPR	**T**rue **P**ositive **R**ate

Deep Learning

A-RNN	**A**ttention-based **R**ecurrent **N**eural **N**etwork
CNN	**C**onvolutional **N**eural **N**etwork
CTC	**C**onnectionist **T**emporal **C**lassification
DGP	**D**eep **G**aussian **P**rocesses
DNN	**D**eep **N**eural **N**etworks
MLP	**M**ulti-**L**ayer **P**erceptron
NF	**N**ormalizing **F**low
RCNN	**R**ecurrent **C**onvolutional **N**eural **N**etwork
ReLU	**R**ectified **L**inear **U**nit
RNN	**R**ecurrent **N**eural **N**etwork
VAE	**V**ariational **A**uto**E**ncoders
VQ-VAE	**V**ector-**Q**uantized **V**ariational-**A**uto-**E**ncoder

Other terms

GUT	**G**dansk **U**niversity of **T**echnology
ITU-R	**I**nternational **T**elecommunication **U**nion – **R**adiocommunication **S**ector
UNESCO	**U**nited **N**ations **E**ducational, **S**cientific and **C**ultural **O**rganization

List of Symbols

x, y	local variables used across different chapter of the Ph.D. Thesis
$\mathbf{x}, \mathbf{y}$	variables in bold indicate vectors or a list of variables
$f : x \mapsto y$	a function mapping from x to y
$y = f(x)$	a function mapping from x to y
$\{1, 2, 3\}$	a set of three numbers
$(0, 1)$	an open interval from 0 to 1, excluding the values of 0 and 1
$[0, 1]$	a closed interval from 0 to 1, including the values of 0 and 1
$x \in \mathcal{R}^D$	a variable x is a member of a set of D-dimensional real numbers
$x \in [0, 1]$	a variable x is a member of a closed interval between 0 and 1
$\{1..N\} \backslash i$	a set excluding the ith element
x^T	the transpose of the variable
$x \odot y$	element-wise matrix multiplication
$(f_1 \circ f_2)(x)$	$f_1(f_2(x))$
$\lvert x - y \rvert$	Euclidean distance
$\lvert K \rvert$	determinant of matrix K
$p(x)$	a probability distribution of the variable x
$p(x, y)$	a joint probability distribution of the variables x and y
$p(x \lvert y)$	a conditional probability distribution (x conditioned on y)
$x \sim p(x \lvert y)$	a variable follows the probability distribution
$p(y) \sim \int p(x, y) dx$	a marginal distribution over the variable y
$p(x \lvert y) \propto p(x) p(y \lvert x)$	the probability distribution p(x | y) is proportional to
μ	the mean value of a probability distribution
σ	standard deviation of a probability distribution
$\tilde{x}, \tilde{\mu}, \tilde{\sigma_2}$	a variable with tilde corresponds to the posterior value of that variable
σ^2	variance of a probability distribution
$\mathcal{N}(\mu, \sigma^2)$	Normal (Gaussian) probability distribution
$\mathcal{L}(\theta)$	the likelihood function parametrized by θ
$\mathcal{I}$	identity matrix
$p - value$	the probability value in the null-hypothesis statistical test
$t - test$	a statistical test checking for a significant difference in the two mean values
tp	the number of true positives, $tp \in \mathbb{Z}$
tn	the number of true negatives, $tn \in \mathbb{Z}$

fp	the number of false positives, $fp \in \mathbb{Z}$
fn	the number of false negatives, $fn \in \mathbb{Z}$
e	the probability of pronunciation error, $e \in (0, 1)$
t	a threshold value
θ	trainable parameters of a machine learning model
κ	an activation function in neural networks

Contents

Acknowledgements **xvii**

List of Figures **xxii**

List of Tables **xxiv**

Abbreviations **xxv**

Symbols **xxvii**

Contents **xxxii**

1 Introduction **1**
1.1 Problem statement 1
1.2 Aim of the thesis 2
1.3 Research theses and background 4
1.3.1 Pronunciation error detection in non-native speech 5
1.3.2 Detection and reconstruction of dysarthric speech 8
1.4 Publications and scientific contribution 9
1.4.1 Contributions from primary author publications 11
1.4.2 Contributions from additional co-authored publications 13
1.5 Applicability 16
1.5.1 Pronunciation error detection 17
1.5.2 Speech synthesis and voice conversion 17

2 Research methodology **19**
2.1 Speech production 19
2.1.1 Articulation 20
2.1.2 Prosody 22
2.2 Machine learning techniques 23
2.2.1 Probability theory 24
2.2.1.1 Probability distribution 24
2.2.1.2 Conditional probability distribution 26
2.2.1.3 Bayesian networks 27
2.2.2 Probabilistic machine learning 28
2.2.2.1 Naive Bayes 29
2.2.2.2 Hidden markov model 32

2.2.2.3 Non-parametric Gaussian processes 35
2.2.2.4 Summary of probabilistic machine learning 40
2.2.3 Deep learning 40
2.2.3.1 Perceptron, dense layer and multi-Layer perceptron . 40
2.2.3.2 Convolutional neural networks 41
2.2.3.3 Recurrent neural networks 42
2.2.3.4 Attention 43
2.2.4 Deep learning – probabilistic perspective 44
2.3 Performance metrics 46
2.3.1 Metrics for the detection of pronunciation errors 47
2.3.1.1 Key metrics 47
2.3.1.2 Discussion 48
2.3.2 Metrics for the generation of speech 50
2.3.2.1 Metrics for the generation of synthetic pronunciation errors 50
2.3.2.2 Metrics for speech reconstruction 51

3 Pronunciation error detection 55
3.1 Introduction 55
3.2 Weakly-supervised word-level pronunciation error detection in non-native English speech 56
3.2.1 Introduction 57
3.2.2 Proposed model 59
3.2.2.1 Model definition 59
3.2.2.2 Neural network details 59
3.2.3 Experiments 60
3.2.3.1 Speech corpora and metrics 61
3.2.3.2 Comparison with state-of-the-art 62
3.2.3.3 Ablation study 63
3.2.3.4 Severity of mispronunciation 63
3.2.4 Conclusions and future work 64
3.3 The role of uncertainty modeling 65
3.3.1 Introduction 66
3.3.2 Related work 66
3.3.3 Proposed model 67
3.3.3.1 Phoneme recognizer 68
3.3.3.2 Pronunciation model 68
3.3.3.3 Pronunciation error detector 69
3.3.4 Experiments and discussion 69
3.3.4.1 Model details 70
3.3.4.2 Speech corpora 70
3.3.4.3 Experimental results 70

3.3.5 Conclusion and future work 72
3.4 Detection of Lexical Stress Errors in Non-Native (L2) English with Data Augmentation and Attention 73
3.4.1 Introduction 73
3.4.2 Related work 74
3.4.3 Proposed model 75
3.4.3.1 Feature extractor 76
3.4.3.2 Attention-based classification model 76
3.4.3.3 Training of the classification model 77
3.4.3.4 Lexical stress error detector 78
3.4.4 Speech corpus 78
3.4.4.1 Human speech 79
3.4.4.2 Synthetic speech 79
3.4.4.3 Lexical stress annotations 79
3.4.5 Experiments 80
3.4.5.1 Experimental results 80
3.4.6 Conclusion and future work 81
3.5 Speech synthesis is almost all you need 82
3.5.1 Introduction 82
3.5.2 Related work 85
3.5.2.1 Pronunciation error detection 85
3.5.2.2 Lexical stress error detection 88
3.5.2.3 Synthetic speech generation for pronunciation error detection 89
3.5.3 Methods of generating pronunciation errors 90
3.5.3.1 P2P method 91
3.5.3.2 T2S method 92
3.5.3.3 S2S method 93
3.5.3.4 Summary of mispronounced speech generation 94
3.5.4 Speech corpora 95
3.5.4.1 Corpora of continuous speech 95
3.5.4.2 Corpora of isolated words 95
3.5.5 Experiments 96
3.5.5.1 Generation of mispronounced speech 96
3.5.5.2 Model of native speech pronunciation 101
3.5.5.3 Lexical stress error detection 103
3.5.6 Conclusions 105

4 Generalization of deep learning methods for pronunciation error detection 109
4.1 Introduction 110
4.2 Related work 110
4.2.1 Dysarthria detection 110

4.2.2 Speech reconstruction 111
4.3 Proposed model 112
4.3.1 Mel-spectrogram and text encoders 113
4.3.2 Spectrogram decoder and dysarthria detector 114
4.4 Experiments 114
4.4.1 Dysarthric speech database 114
4.4.2 Automatic detection of dysarthria 115
4.4.3 Interpretable modeling of dysarthric patterns 115
4.4.4 Reconstruction of dysarthric speech 116
4.5 Conclusions 119
4.6 Acknowledgements 120

5 Conclusions 121
5.1 Summary 121
5.2 Novelty 122
5.3 Applicability 124
5.4 Future work 124

A Declaration of authorship 127

B List of publications of the author of the doctoral dissertation 135

C Primary author publications in the original format 137

D Co-authored publication on pronunciation error detection prior to Ph.D. research 183

References 189

Chapter 1

Introduction

1.1 Problem statement

Language is a way of communication between people. Currently, about 7,139 languages are spoken in the world, English being the most dominant one with 1.348 billion speakers (Eberhard et al., 2021). English has its written and spoken versions (Denham et al., 2012). Written language is based on words and sentences made out of symbols called letters. Spoken language enables people to communicate verbally by producing a stream of sounds called phones that represent spoken words and sentences.

Language plays a key role in education, giving people access to a large amount of information contained in books, notes, and diaries written down through the ages. Thanks to spoken language, people can participate in interactive discussions with teachers and engage in lively brainstorming with other people. In the age of the Internet and online education, people can access books, articles, video lectures, and even get a university degree from almost anywhere in the world.

Unfortunately, education is not equally accessible to everybody. Regarding the UNESCO report, 40% of the global population does not have access to education in a language they understand (UNESCO, 2016). 'If you don't understand, how can you learn?' the report says. This situation could be improved by popularizing education in the student's mother (native) language, but with more than seven thousand languages in the world, it may not be possible to increase the access to education significantly.

Another approach to increasing access to education is to ensure that people learn at least one foreign language, such as English. Learning multiple languages has benefits beyond having access to better education. It has been reported that multilingualism can boost economic growth (WorldEconomicForum, 2018), help find a better job (EF-Education-First, 2020), and protect against cognitive decline (Kroll et al., 2017).

However, learning a foreign language seems easier than it is. The study by EF Education First (EF-Education-First, 2020) shows a large disproportion in English proficiency in different countries and continents. The lowest English proficiency is in the Middle East region that falls into the 'very low' category of language proficiency. People falling into this category are not able to navigate an English-speaking country

or understand a simple email from a colleague. The description of the English proficiency categories is shown in Table 1.1, while Table 1.2 shows English proficiency by region.

TABLE 1.1: Description of EF English Proficiency Index (EPI) (EF-Education-First, 2020)

Proficiency Index (EPI)	Sample tasks
Very High Netherlands Singapore Sweden	Use nuanced and appropriate language in social situations Read advanced texts with ease Negotiate a contract with a native English speaker
High Hungary Kenya Philippines	Make a presentation at work Understand TV shows Read a newspaper
Moderate China Costa Rica Italy	Participate in meetings in one's area of expertise Understand song lyrics Write professional emails on familiar subjects
Low Dominican Republic Pakistan Turkey	Navigate an English-speaking country as a tourist Engage in small talk with colleagues Understand simple emails from colleagues
Very low Cambodia Tajikistan United Arab Emirates	Introduce oneself simply Understand simple signs Give basic directions to a foreign visitor

TABLE 1.2: English proficiency by region (EF-Education-First, 2020)

Region	English Proficiency Index (EPI)
Europe	high
Asia	low
Africa	low
Latin America	low
Middle East	very low

1.2 Aim of the thesis

Computer-Assisted Language Learning (CALL) (Asrifan et al., 2020) is a possible solution to improve English proficiency in different regions. CALL is based on self-service computer-based tools that are used by students to practice a language, usually a foreign (non-native) language. In CALL, students can practice multiple aspects of

the language including grammar, vocabulary, writing, reading, and speaking. CALL can complement traditional language learning provided by teachers. It also has a chance to democratize second-language learning in places where traditional ways of learning languages are not possible due to the costs of learning or the lack of access to foreign language teachers.

This Ph.D. thesis has been completed within the "Implementation doctorate" program, carried out by the Gdańsk University of Technology, and written in agreement with the Amazon company employing the Ph.D. candidate. It is devoted to CALL in the task of learning pronunciation skills by non-native speakers of English, also known as Computer-Assisted Pronunciation Training (CAPT) (Fouz-González, 2015). In general, a CAPT system consists of two components: an automated pronunciation assessment component and a feedback component. The automated pronunciation assessment component is responsible for detecting pronunciation errors in the pronounced speech, for example, for detecting phonemes or words pronounced by the speaker incorrectly. The feedback component informs the speaker about mispronounced words and advises on how to pronounce them correctly. In the future, the CAPT system may be integrated into a voice-enabled AI assistant to let people practice pronunciation skills using a voice interface, as illustrated in Figure 1.1.

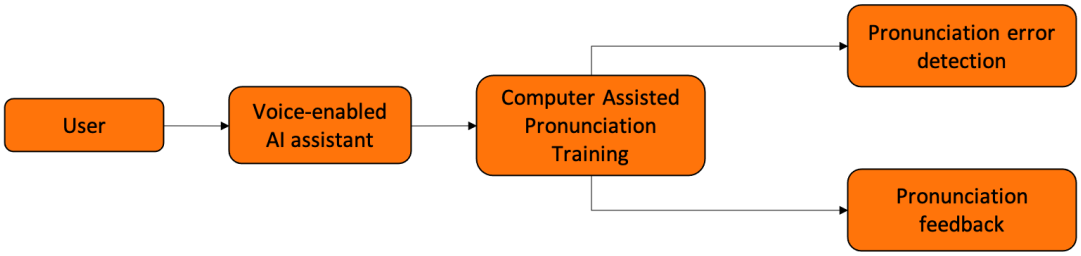


FIGURE 1.1: Overview of the Computer-Assisted Pronunciation Training System.

In particular, this dissertation focuses on the automated pronunciation assessment in CAPT. Despite decades of work in the scientific community devoted to automated pronunciation assessment, there is still a great potential to improve the accuracy to detect pronunciation errors in speech automatically. State-of-the-art methods detect pronunciation errors with a relatively low accuracy of 60% precision at 40%-80% recall (Leung et al., 2019; Korzekwa, Lorenzo-Trueba, Zaporowski, et al., 2021; Z. Zhang et al., 2021). Highlighting correctly pronounced words as pronunciation errors by a CAPT tool can demotivate the language learner and affect the quality of learning. In contrast, missing pronunciation errors can slow down the learning process. The ultimate motivation behind the doctoral dissertation is twofold:

1. **To raise foreign language proficiency in the global population by improving the accuracy of automated pronunciation assessment in CAPT.**

2. **To apply the results obtained in the doctoral dissertation at Amazon company as the thesis was realized within the "Implementation doctorate" program, carried out by the Gdańsk University of Technology.**

This thesis is organized as follows:

- Chapter 1: Introduction - The Ph.D. thesis begins with presenting the problem statement, motivation, and the aim of the thesis. Next, the subject of the dissertation is translated into research theses as well as the research background is presented. At the end of the chapter, a summary of the Ph.D. scientific contribution is included in the form of the most important co-authored publications presented at international conferences and in scientific journals.
- Chapter 2: Research methodology - This chapter covers the fundamental topics related to the Ph.D. research, including speech generation process, the probability theory, probabilistic machine learning, deep learning, and evaluation metrics. This material lays the foundations for deep learning methods in automated detection of pronunciation errors, presented in the next chapter.
- Chapter 3: Pronunciation error detection - This chapter constitutes the main scientific part of 1the doctoral dissertation. Original deep learning methods for detecting pronunciation and lexical stress errors in non-native English speech are presented.
- Chapter 4: Generalization of deep learning methods for pronunciation error detection - This chapter explores the generalization capabilities of deep learning methods for detecting pronunciation errors in two related tasks: detection and reconstruction of dysarthric speech.
- Chapter 5: Summary and Conclusions - The final chapter summarizes the doctoral dissertation, presents the main conclusions, and draws a plan for the future.
- References and Appendices.

1.3 Research theses and background

To address the research goal, which is to improve the accuracy of detecting pronunciation errors in non-native English speech, the primary research thesis is formulated. The primary aim of this Ph.D. work is to establish a new state-of-the-art deep learning method for the detection of pronunciation errors in non-native English, so the thesis is formulated as follows:

1. **It is possible to improve the accuracy of deep learning methods for detecting pronunciation errors in non-native English by employing synthetic speech generation and end-to-end modeling techniques that reduce the need for phonetically transcribed mispronounced speech.**

In addition to the primary research thesis, the secondary research thesis is formulated to investigate the generalization capabilities of the invented methods of pronunciation error detection in the related area of dysarthric speech.

2. Deep learning methods for the detection of pronunciation errors in non-native speech are transferable to the related tasks of detection and reconstruction of dysarthric speech.

1.3.1 Pronunciation error detection in non-native speech

What are pronunciation errors and pronunciation error detection?

A pronunciation error in speech occurs when a speaker pronounces a word or a sentence differently from the expected pronunciation provided by the canonical phonetic transcription (Witt et al., 2000). Mispronunciations may refer to incorrectly pronounced phonemes, e.g., mispronouncing the phoneme /eh/ as /ey/ in the English sentence 'I said' /ay s eh d/.

Phonemes are abstract symbols that correspond to the mental representation of the pronunciation of a word. Phonemes are related to phones that correspond to specific sounds made by a speaker. The way a word is pronounced is determined by its phonetic transcription. For example, the word 'cat' is transcribed as [k ae t] and the word 'cell' is transcribed as [s eh l]. Phoneme transcription is represented with slashes //, e.g., /s eh l/ as opposed to using brackets [] for phones. A more detailed description of a speech production process is presented in Section 2.1.

Lexical stress error (Ferrer et al., 2015) is another type of pronunciation error that occurs when a speaker stresses an incorrect syllable in a word, e.g., incorrectly stressing the first syllable in the word 'remind' /r iy1 m ay0 n d/. Pronunciation errors can exist at different levels of granularity, for example, at the level of phonemes (Leung et al., 2019), words (Korzekwa, Lorenzo-Trueba, Drugman, Calamaro, et al., 2021) and utterances (Gong et al., 2022).

Apparently, detecting a pronunciation error at the phoneme level provides a user with the most informative feedback, but it is more complicated. Not all language learners are familiar with the concept of a phoneme; secondly, sometimes, it may be very difficult to recognize the phoneme pronounced by a user (Z. Zhang et al., 2021). Therefore, language teachers do not always provide users with the phoneme-level feedback. Instead, they simply point out a mispronounced word and use their voice to show how to pronounce it correctly. AI-based CAPT assistants can provide similar verbal feedback to a user using their synthetic voices. In this way, a user can practice pronunciation skills just from the comfort of the couch via the voice interface.

Within the Ph.D. thesis, various models were built to detect both mispronounced phonemes (Beringer et al., 2020; Korzekwa, Lorenzo-Trueba, Zaporowski, et al., 2021; Korzekwa, Lorenzo-Trueba, Drugman, Calamaro, et al., 2021; Korzekwa, Lorenzo-Trueba, Drugman, and Kostek, 2022) and lexical stress errors (Korzekwa, Barra-Chicote, Zaporowski, et al., 2021), at the phoneme and word levels. However, the

direction in which these models are evolving - towards detecting pronunciation errors at the word level - is motivated by the use case of practicing pronunciation skills based on AI-based voice assistant interface, as shown in Figure 1.1.

How deep learning methods to detect pronunciation errors may be improved?

Deep learning is often considered a universal machine that can automatically solve any problem if sufficient training data are available. However, deep learning models are generally data-hungry (Lake et al., 2015; Marcus, 2018). They work well for speech processing tasks but require a large amount of training data to generalize to unseen data (Shah et al., 2021). In pronunciation error detection, existing deep learning methods detect pronunciation errors with a relatively low accuracy of 60% precision at 40%-80% recall (Leung et al., 2019; Korzekwa, Lorenzo-Trueba, Zaporowski, et al., 2021; Z. Zhang et al., 2021). Many interesting statements can be made about existing methods of detecting pronunciation errors. These statements can lead to new designs of deep learning models to improve the accuracy of pronunciation error detection models and ultimately improve the CAPT user experience.

These statements that constitute the background of this Ph.D. work are as follows:

1. Transcription of non-native speech is a difficult and costly process

   The end result of the pronunciation error detection model is the probability of a pronunciation error at the segment level, such as a phoneme or a word. Creating an end-2-end model (Z. Zhang et al., 2021) that directly estimates this probability could make phonetic transcriptions of non-native speech redundant (Korzekwa, Lorenzo-Trueba, Drugman, Calamaro, et al., 2021).

2. Aligning canonical and recognized phonemes accurately is challenging

   To detect pronunciation errors, existing methods recognize pronounced phonemes and then compare them with the expected (canonical) pronunciation of a native speaker (Witt et al., 2000; K. Li, Qian, and Meng, 2016; Sudhakara, Ramanathi, Yarra, and Ghosh, 2019; Leung et al., 2019). Detecting pronunciation errors directly by an end-to-end model could eliminate the alignment as a potential source of errors affecting the accuracy of detecting pronunciation errors.

3. Not all pronunciation errors are the same

   Some pronunciation errors are more severe than others. Categorizing pronunciation errors by severity level allows reporting only more severe errors to the user and reduces the risk of correctly pronounced text being detected as a pronunciation error (Yan, M.-C. Wu, et al., 2020; Korzekwa, Lorenzo-Trueba, Drugman, Calamaro, et al., 2021)

4. A sentence can be pronounced correctly in multiple different ways

   Native speakers can pronounce the same text in many correct ways. The pronunciation error detection model should take this observation into account

and allow a language learner to pronounce the same text in different ways. Taking into account the variability of pronunciation will reduce the likelihood of reporting false pronunciation alarms to the user (Qian et al., 2010; Korzekwa, Lorenzo-Trueba, Zaporowski, et al., 2021)

5. Practicing lexical stress is an important part of CAPT

   Existing CAPT methods concentrate on practicing the pronunciation of phonemes (Witt et al., 2000; Leung et al., 2019; Korzekwa, Lorenzo-Trueba, Zaporowski, et al., 2021). Nevertheless, it has been shown that practicing lexical stress improves the intelligibility of non-native English speech (Field, 2005; Lepage et al., 2014). Good deep learning models in CAPT should be capable of detecting both pronunciation and lexical stress errors.

6. The availability of non-native speech with pronunciation errors is limited

   Deep learning models work very well when the amount of training data is large (Shah et al., 2021). There is evidence in the related field of computer vision that generating synthetic images improves the accuracy of classification models (Wong et al., 2016). Therefore, a similar technique may improve the accuracy of detecting pronunciation errors in non-native speech. Data augmentation (Badenhorst et al., 2017; Fu, Gao, et al., 2022) and data generation (A. Lee et al., 2016) are two techniques that can create synthetic pronunciation errors to account for the limited availability of non-native speech with pronunciation errors. Recent advances in speech synthesis (Fazel et al., 2021) and voice conversion (Shah et al., 2021) open the door to the generation of synthetic speech, which eventually may be able to mimic non-native human speech perfectly and enable training pronunciation error detection models only on synthetic data.

7. Multi-task learning as an approach to tackling overfitting in deep learning methods

   In multi-tasking, in addition to the primary task of detecting pronunciation errors in a speech signal, a secondary task can be added, such as recognizing pronounced phonemes (Z. Zhang et al., 2021; Korzekwa, Lorenzo-Trueba, Drugman, Calamaro, et al., 2021). Both tasks will interact, making the model less prone to overfitting.

To summarize the research thesis on pronunciation error detection, this Ph.D. research explores various deep learning methods related to probabilistic machine learning, multi-tasking, and data generation techniques. It has been hypothesized that by using these techniques, it should be possible to improve the accuracy of the state-of-the-art methods of detecting pronunciation errors. The proposed models for detecting pronunciation errors are evaluated on the non-native speech of multiple nationalities, including German, Italian, and Polish speakers, including a new corpus of non-native

speech (Weber et al., 2020) recorded at the Gdańsk University of Technology (GUT) to facilitate these evaluations.

1.3.2 Detection and reconstruction of dysarthric speech

Good machine learning methods should be generic and scale to other related problems. The secondary research thesis aims to investigate whether deep learning methods can be transferred to the tasks of detecting and reconstructing dysarthric speech.

Detection of dysarthric speech

Speech production begins in the brain, where the mental representation of a message is formed as a sequence of abstract symbols called phonemes. The brain then controls the speech organs to generate a spoken message. The lungs generate air that flows through the larynx, and the oral and nasal cavities, generating speech. Multiple muscles are involved in this process, such as lips, throat (pharynx), and jaw (Trujillo, 2006).

Dysarthria is a motor speech disorder that results from neurological disorders such as cerebral palsy, brain stroke/aphasia, dementia, and brain cyst (M. Cuny et al., 2017; Banovic, L. J. Zunic, et al., 2018). Due to damage to the nervous system, the connections between the brain and the speech organs and their muscles are weakened, resulting in distorted speech (ASHA, 2022). Compared to normal speech, dysarthric speech is harsh and breathy, contains mispronunciations, has flattened intonation, and has a lower speech rate.

It can be hypothesized that deep learning models used to automatically detect pronunciation errors in non-native speech can be transferred to the dysarthric speech detection task, or more broadly, impaired speech, such as in Parkinson's disease (PD) (Korzekwa, Barra-Chicote, Kostek, et al., 2019; Romana et al., 2021). In both non-native and dysarthric speech, similar distortions of speech, such as mispronunciations and incorrect prosody patterns, can be observed. Therefore, similar deep-learning models should apply in both areas.

In Figure 1.1, it was shown that a voice-enabled AI assistant could be used to build a system for detecting pronunciation errors and providing feedback to a user. Such design can be adopted to create a health assistant system that can detect dysarthric speech and provide advice to a user to visit a consultant, as illustrated in Figure 1.2.

Reconstruction of dysarthric speech

People with dysarthria have difficulty communicating with other people because their speech is distorted and less intelligible. Speech therapy is one way to improve spoken communication skills; for example, when dysarthria results from a brain stroke causing an aphasia condition (Farrajota et al., 2012; Koyuncu et al., 2016; Brady et al., 2016). In cases where speech therapy is not effective, it may still be

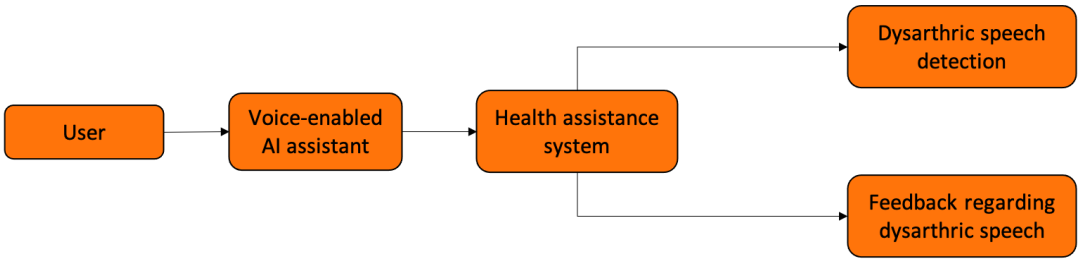


FIGURE 1.2: Overview of the Dysarthric Speech Detection System.

possible to help people communicate by reconstructing their speech using a speech-to-speech approach (Korzekwa, Barra-Chicote, Kostek, et al., 2019; W.-C. Huang et al., 2021). The input to a speech-to-speech system is distorted speech spoken by a person with dysarthria, and the output is the reconstructed speech with improved intelligibility. Similar techniques can be applied to non-native speech. Radzikowski et al. (Radzikowski et al., 2016) use Hidden Markov Models (HMM) to make corrections in non-native speech so that students and teachers can communicate more easily during lectures.

There are similarities between the generation of synthetic speech errors in non-native speech for the detection of pronunciation errors and the reconstruction of dysarthric speech. In the synthetic speech scenario, the speech-to-speech system is used to 'destroy' correctly pronounced speech by introducing pronunciation errors. In the dysarthric speech scenario, speech is processed the other way round to improve the intelligibility of distorted speech. It can be hypothesized that a similar deep learning technique should be effective in both scenarios.

1.4 Publications and scientific contribution

In this Section, first, the articles co-authored by Daniel Korzekwa are listed, and then the main scientific contributions are presented in more detail in the following subsections.

Six articles were published or accepted for publication with Daniel Korzekwa as the primary author. The declaration of authorship is included in Appendix A. These publications are directly related to the research theses presented in Section 1.3 and constitute the main scientific contribution of the doctoral dissertation:

- Computer-assisted Pronunciation Training - Speech synthesis is almost all you need; accepted for publication in Speech Communication Journal on June 17 '2022, in print (Korzekwa, Lorenzo-Trueba, Drugman, and Kostek, 2022)

- Weakly-supervised word-level pronunciation error detection in non-native English speech, Interspeech, 2021 (Korzekwa, Lorenzo-Trueba, Drugman, Calamaro, et al., 2021)

- Mispronunciation Detection in Non-native (L2) English with Uncertainty Modeling, ICASSP, 2021 (Korzekwa, Lorenzo-Trueba, Zaporowski, et al., 2021)
- Detection of Lexical Stress Errors in Non-native (L2) English with Data Augmentation and Attention, Interspeech, 2021 (Korzekwa, Barra-Chicote, Zaporowski, et al., 2021)
- Deep learning model for automated assessment of lexical stress of non-native English speakers, The Journal of the Acoustical Society of America, 2019 (Korzekwa and Kostek, 2019)
- Interpretable deep learning model for the detection and reconstruction of dysarthric speech, Interspeech, 2019 (Korzekwa, Barra-Chicote, Kostek, et al., 2019)

Additionally, nine publications co-authored by Daniel Korzekwa are devoted to topics related to the doctoral dissertation. Two publications are devoted to pronunciation error detection in non-native English. Six publications concern speech synthesis and voice conversion, which lay the foundations for generating synthetic pronunciation errors and the reconstruction of dysarthric speech. The ninth publication concerns the collection of non-native speech corpus that was used to evaluate the pronunciation error detection models:

- L2-GEN: A Neural Phoneme Paraphrasing Approach to L2 Speech Synthesis for Mispronunciation Diagnosis, accepted to Interspeech, 2022 (D. Zhang et al., 2022)
- Creating New Voices using Normalizing Flows, accepted to Interspeech, 2022 (Bilinski et al., 2022)
- Text-free non-parallel many-to-many voice conversion using normalizing flows, ICASSP, 2022 (Merritt, Ezzerg, et al., 2022)
- Universal neural vocoding with parallel wavenet, ICASSP, 2021 (Jiao et al., 2021)
- Improving the expressiveness of neural vocoding with non-affine Normalizing Flows, Interspeech, 2021 (Gabryś et al., 2021)
- Non-Autoregressive TTS with Explicit Duration Modelling for Low-Resource Highly Expressive Speech, ISCA Speech Synthesis Workshop – a satellite event at Interspeech, 2021 (Shah et al., 2021)
- Enhancing audio quality for expressive Neural Text-to-Speech, ISCA Speech Synthesis Workshop – a satellite event at Interspeech, 2021 (Ezzerg et al., 2021)
- Constructing a dataset of speech recordings with Lombard effect, IEEE SPA 2020 (Weber et al., 2020)
- Extending Goodness of Pronunciation to generate mispronunciation hypotheses for pronunciation assessment in L2-English, AMLC, 2020 (Beringer et al., 2020)

1.4.1 Contributions from primary author publications

Three publications are devoted to the automated detection of pronunciation errors (incorrectly pronounced phonemes) in non-native speech.

Korzekwa, D., J. Lorenzo-Trueba, T. Drugman, and B. Kostek (2022). "Computer- assisted Pronunciation Training - Speech synthesis is almost all you need". In: accepted for publication in Speech Communication Journal on June 17 '2022, in print.

Novelty: The research community has long studied computer-assisted pronunciation training (CAPT) methods in non-native speech. Researchers focused on studying various model architectures, such as Bayesian networks and deep learning methods, as well as on the analysis of different representations of the speech signal. Despite significant progress in recent years, existing CAPT methods are not able to detect pronunciation errors with high accuracy (only 60% precision at 40%-80% recall). One of the key problems is the low availability of mispronounced speech that is needed for the reliable training of pronunciation error detection models. If we had a generative model that could mimic non-native speech and produce any amount of training data, then the task of detecting pronunciation errors would be much easier. We present three innovative techniques based on phoneme-to-phoneme (P2P), text-to-speech (T2S), and speech-to-speech (S2S) conversion to generate correctly pronounced and mispronounced synthetic speech. We show that these techniques not only improve the accuracy of three machine learning models for detecting pronunciation errors but also help establish a new state-of-the-art in the field. Earlier studies have used simple speech generation techniques such as P2P conversion, but only as an additional mechanism to improve the accuracy of pronunciation error detection. We, on the other hand, consider speech generation to be the first-class method of detecting pronunciation errors. The effectiveness of these techniques is assessed in the tasks of detecting pronunciation and lexical stress errors. Non-native English speech corpora of German, Italian, and Polish speakers are used in the evaluations. The best proposed S2S technique improves the accuracy of detecting pronunciation errors in AUC metric by 41% from 0.528 to 0.749 compared to the state-of-the-art approach.

Korzekwa, D., J. Lorenzo-Trueba, T. Drugman, S. Calamaro, and B. Kostek (2021). "Weakly-Supervised Word-Level Pronunciation Error Detection in Non-Native English Speech". In: Proc. Interspeech 2021, pp. 4408–4412. DOI: 10.21437/Interspeech.2021-38.

Novelty: We propose a weakly-supervised model for word-level mispronunciation detection in non-native (L2) English speech. To train this model, phonetically transcribed L2 speech is not required and we only need to mark mispronounced words. The lack of phonetic transcriptions for L2 speech means that the model has to learn only from a weak signal of word-level mispronunciations. Because of that and due to the limited amount of mispronounced L2 speech, the model is more likely to overfit. To limit this risk, we train it in a multi-task setup. In the first task,

we estimate the probabilities of word-level mispronunciation. For the second task, we use a phoneme recognizer trained on phonetically transcribed L1 speech that is easily accessible and can be automatically annotated. Compared to state-of-the-art approaches, we improved the accuracy of detecting word-level pronunciation errors in AUC metric by 30% on the GUT Isle Corpus of L2 Polish speakers and by 21.5% on the Isle Corpus of L2 German and Italian speakers.

Korzekwa, D., J. Lorenzo-Trueba, S. Zaporowski, S. Calamaro, T. Drugman, and B. Kostek (2021). "Mispronunciation Detection in Non-Native (L2) English with Uncertainty Modeling". In: ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, pp. 7738–7742. DOI: 10.1109/ICASSP39728.2021.9413953

Novelty: A common approach to the automatic detection of mispronunciation in language learning is to recognize the phonemes produced by a student and compare them to the expected pronunciation of a native speaker. This approach makes two simplifying assumptions: a) phonemes can be recognized from speech with high accuracy, b) there is a single correct way for a sentence to be pronounced. These assumptions do not always hold, which can result in a significant amount of false mispronunciation alarms. We propose a novel approach to overcome this problem based on two principles: a) taking into account uncertainty in the automatic phoneme recognition step, b) accounting for the fact that there may be multiple valid pronunciations. We evaluate the model on non-native (L2) English speech of German, Italian and Polish speakers, where it is shown to increase the precision of detecting mispronunciations by up to 18% (relative) compared to the common approach.

Two publications relate to the detection of lexical stress errors. Preliminary work was first presented in the Journal of the Acoustical Society of America in 2019. The final results were published at the Interspeech 2021 conference.

Korzekwa, D. and B. Kostek (2019). "Deep learning model for automated assessment of lexical stress of non-native English speakers". In: The Journal of the Acoustical Society of America 146.4, pp. 2956–2957. DOI: 10.1121/1.5137270

Korzekwa, D., R. Barra-Chicote, S. Zaporowski, G. Beringer, J. Lorenzo-Trueba, A. Serafinowicz, J. Droppo, T. Drugman, and B. Kostek (2021). "Detection of Lexical Stress Errors in Non-Native (L2) English with Data Augmentation and Attention". In: Proc. Interspeech 2021, pp. 3915–3919. DOI: 10.21437/Interspeech.2021-86

Novelty: We describe two novel complementary techniques that improve the detection of lexical stress errors in non-native (L2) English speech: attention-based feature extraction and data augmentation based on Neural Text-To-Speech (TTS). In a classical approach, audio features are usually extracted from fixed regions of speech, such as the syllable nucleus. We propose an attention-based deep learning model

that automatically derives optimal syllable-level representation from frame-level and phoneme-level audio features. Training this model is challenging because of the limited amount of incorrect stress patterns. To solve this problem, we propose to augment the training set with incorrectly stressed words generated with Neural TTS. Combining both techniques achieves 94.8% precision and 49.2% recall for the detection of incorrectly stressed words in L2 English speech of Slavic and Baltic speakers.

One publication deals with the detection and reconstruction of dysarthric speech - a topic of the secondary research thesis.

Korzekwa, D., R. Barra-Chicote, B. Kostek, T. Drugman, and M. Lajszczak (2019). "Interpretable Deep Learning Model for the Detection and Reconstruction of Dysarthric Speech". In: Proc. Interspeech 2019, pp. 3890–3894. DOI: 10.21437/Interspeech.2019-1206

Novelty: We present a novel deep learning model for the detection and reconstruction of dysarthric speech. We train the model with a multi-task learning technique to jointly solve dysarthria detection and speech reconstruction tasks. The model key feature is a low-dimensional latent space that is meant to encode the properties of dysarthric speech. It is commonly believed that neural networks are "black boxes" that solve problems but do not provide interpretable outputs. On the contrary, we show that this latent space successfully encodes interpretable characteristics of dysarthria, is effective at detecting dysarthria, and that manipulation of the latent space allows the model to reconstruct healthy speech from dysarthric speech. This work can help patients and speech pathologists to improve their understanding of the condition, lead to more accurate diagnoses, and aid in reconstructing healthy speech for afflicted patients.

1.4.2 Contributions from additional co-authored publications

Publications related to pronunciation error detection:

Zhang, D., A. Ganesan, S. Campbell, and D. Korzekwa (2022). "L2-GEN: A Neural Phoneme Paraphrasing Approach to L2 Speech Synthesis for Mispronunciation Diagnosis". In: accepted to Interspeech 2022.

Novelty: In this paper, we study the problem of generating mispronounced speech mimicking non-native (L2) speakers learning English as a Second Language (ESL) for the mispronunciation detection and diagnosis (MDD) task. The paper is motivated by the widely observed yet not well addressed data sparsity issue in MDD research where both L2 speech audio and its fine-grained phonetic annotations are difficult to obtain, leading to unsatisfactory mispronunciation feedback accuracy. We propose L2-GEN, a new data augmentation framework to generate L2 phoneme sequences that capture realistic mispronunciation patterns by devising an unique

machine translation-based sequence paraphrasing model. A novel diversified and preference-aware decoding algorithm is proposed to generalize L2-GEN to handle both unseen words and new learner population with very limited L2 training data. A contrastive augmentation technique is further designed to optimize MDD performance improvements with the generated synthetic L2 data. We evaluate L2-GEN on public L2-ARCTIC and SpeechOcean762 datasets. The results have shown that L2-GEN leads to up to 3.9%, and 5.0% MDD F1-score improvements in in-domain and out-of-domain scenarios respectively.

Beringer, G., D. Korzekwa, A. Sanchez, B.Wang, and J. Lorenzo-Trueba (2020). "Extending Goodness of Pronunciation to generate mispronunciationhypotheses for pronunciation assessment in L2-English". In: Amazon Machine Learning Conference, Seattle.

Novelty: We propose a method to extend Goodness of Pronunciation (GOP), a commonly used pronunciation scoring metric, to generate mispronunciation hypotheses, which are then used to find what the speaker has actually uttered. We show that this allows to alleviate GOP's problem of being over-dependant on phone boundaries computed by force-alignment, leading to an improvement in mispronunciation detection and diagnosis. We also argue that introducing hypothesis prior could be used to improve the model in the context of pronunciation teaching, where high precision is required. We demonstrate that a method of increasing the prior of canonical hypothesis by a factor can enable us to have control over precision-recall trade-off. For our experiments, we use a dataset of isolated words, which contain recordings of 23 Polish-based speakers.

Six co-authored publications are devoted to the topic of speech synthesis and voice conversion. These methods are used in two areas of the Ph.D. thesis: generation of mispronounced non-native speech and reconstruction of dysarthric speech. In addition, speech synthesis technology is used in Alexa devices, serving millions of people worldwide.

A modern speech synthesis and voice conversion systems consist of two components: a context generator and a vocoder. The context generator creates a mel-spectrogram from the input text (text-to-speech mode) (Y. Wang, R. Skerry-Ryan, et al., 2017). Alternatively, it can process the mel-spectrogram extracted from another speech signal (speech-to-speech mode) (Jia et al., 2019). The mel-spectrogram created by the context generator is processed by a vocoder to generate the raw audio signal (Oord et al., 2018; Lorenzo-Trueba et al., 2018).

Publications related to context generation:

Bilinski, P., T. Merritt, A. Ezzerg, K. Pokora, S. Cygert, K. Yanagisawa, R. Barra Chicote, and D. Korzekwa (2022). "Creating New Voices using Normalizing Flows". In: accepted to Interspeech 2022.

Novelty: Creating realistic and natural-sounding synthetic speech remains a big challenge for voice identities unseen during training. As there is growing interest in synthesizing voices of new speakers, here we investigate the ability of normalizing flows in text-to-speech (TTS) and voice conversion (VC) modes to extrapolate from speakers observed during training to create unseen speaker identities. Firstly, we create an approach for TTS and VC, and then we comprehensively evaluate our methods and baselines in terms of intelligibility, naturalness, speaker similarity, and ability to create new voices. We use both objective and subjective metrics to benchmark our techniques on 2 evaluation tasks: zero-shot and new voice speech synthesis. The goal of the former task is to measure the precision of the conversion to an unseen voice. The goal of the latter is to measure the ability to create new voices. Extensive evaluations demonstrate that the proposed approach systematically allows to obtain state-of-the-art performance in zero-shot speech synthesis and creates various new voices, unobserved in the training set. We consider this work to be the first attempt to synthesize new voices based on mel-spectrograms and normalizing flows, along with a comprehensive analysis and comparison of the TTS and VC modes.

Shah, R., K. Pokora, A. Ezzerg, V. Klimkov, G. Huybrechts, B. Putrycz, D. Korzekwa, and T. Merritt (2021). "Non Autoregressive TTS with Explicit Duration Modelling for Low-Resource Highly Expressive Speech". In: Proc. 11th ISCA Speech Synthesis Workshop (SSW 11), pp. 96–101. DOI: 10.21437/SSW.2021-17

Ezzerg, A., A. Gabryś, B. Putrycz, D. Korzekwa, D. Saez Trigueros, D. McHardy, K. Pokora, J. Lachowicz, J. Lorenzo-Trueba, and V. Klimkov (2021). "Enhancing audio quality for expressive Neural Text-to-Speech". In: Proc. 11th ISCA Speech Synthesis Workshop (SSW 11), pp. 78–83. DOI: 10.21437/SSW.2021-14

Novelty: These two publications propose context generation models based on deep learning techniques, including VAE (J. Chorowski, Weiss, et al., 2019; Van Den Oord et al., 2017), attention mechanism (Vaswani, Shazeer, Parmar, Uszkoreit, Jones, Gomez, Ł. Kaiser, et al., 2017), sequence-to-sequence models (Sutskever et al., 2014), and controllable speech synthesis (Ren et al., 2019). The main novelties are: improving signal quality and stability of speech synthesis, and creating TTS voices of speakers with a limited amount of speech recordings. The proposed TTS models lay the foundations for the generation of mispronounced non-native speech (Section 3.5) and improved intelligibility of dysarthric speech (Chapter 4).

Merritt, T., A. Ezzerg, P. Biliński, M. Proszewska, K. Pokora, R. Barra-Chicote, and D. Korzekwa (2022). "Text-free non parallel many-to-many voice conversion using normalising flows". In: Acoustics, Speech and Signal Processing (ICASSP). DOI: 10.1109/ICASSP43922.2022.9746368

Novelty: One of the trends deeply explored in the Ph.D. thesis concerns the use of speech conversion to generate synthetic pronunciation errors (Korzekwa,

Lorenzo-Trueba, Drugman, Calamaro, et al., 2021). This task requires a speech-to-speech (S2S) technique that takes correctly pronounced native speech and converts it to mispronounced speech. This publication proposes a novel voice conversion technique that enables speech conversion without relying on phonetic transcriptions. Collecting phonetic transcriptions is very time-consuming and this method makes the process redundant, paving the way to much more efficient ways of generating mispronounced speech. Second, this method can convert any input speaker to any output speaker, which is useful for generating a diverse range of speakers.

Publications related to neural speech vocoding:

Jiao, Y., A. Gabryś, G. Tinchev, B. Putrycz, D. Korzekwa, and V. Klimkov (2021). "Universal neural vocoding with parallel wavenet". In: ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, pp. 6044–6048. DOI: 10.1109/ICASSP39728.2021.9414444

Gabryś, A., Y. Jiao, V. Klimkov, D. Korzekwa, and R. Barra-Chicote (2021). "Improving the Expressiveness of Neural Vocoding with Non-Affine Normalizing Flows". In: Proc. Interspeech 2021, pp. 1679–1683. DOI: 10.21437/Interspeech. 2021-1555

Novelty: In the dissertation-based research, the speech-to-speech technique is used to generate pronunciation errors for hundreds of voices. The context generator creates a mispronounced speech spectrogram that is converted into a raw speech signal by the vocoder. Typically, a dedicated vocoder would have to be trained for each unique voice, but that approach would not scale here. These two articles propose a universal neural vocoder that can transform any speaker's mel-spectrogram into a raw speech signal. The universal vocoder is a key component, enabling the generation of mispronounced speech for many speakers at scale.

In addition to four publications on speech synthesis, there is one publication on the non-native speech corpus collection.

Weber, D., S. Zaporowski, and D. Korzekwa (2020). "Constructing a Dataset of Speech Recordings with Lombard Effect". In: 24th IEEE SPA. DOI: 10.23919/ SPA50552.2020.9241266

Novelty: The speech corpus of non-native speech was collected and used to evaluate the proposed pronunciation error detection models.

1.5 Applicability

Note: Due to confidentiality reasons, only selected use-cases are provided.

The results of the doctoral dissertation are widely applicable at Amazon in many use cases. The pronunciation error detection models are used to detect pronunciation

errors in speech synthesis automatically. The speech synthesis and voice conversion models are used in Alexa devices to serve millions of Amazon customers around the world. In addition, speech synthesis and voice conversion are used as a data augmentation technique to improve the accuracy of the pronunciation error detection models.

1.5.1 Pronunciation error detection

Scientists who work on new speech synthesis models often explore many research hypotheses and train many machine learning models for speech synthesis. Being able to get quick feedback on the quality of the generated speech is crucial for rapid progress in research. Traditionally, the quality of speech generated by speech synthesis models is evaluated by humans. Humans listen to synthesized utterances and evaluate them in terms of naturalness and speech intelligibility. Typically, multiple voices are scored within a single evaluation to understand which of the voices perform best with respect to certain aspects of speech quality. This manual perceptual evaluation process is a bottleneck that is slowing down research into new models of speech synthesis.

The pronunciation error detection model (Korzekwa, Lorenzo-Trueba, Drugman, Calamaro, et al., 2021) can complement manual perceptual evaluation, speeding up the research work on speech synthesis at Amazon in four different languages (detailed locations cannot be provided for confidentiality reasons). Many utterances can be synthesized and automatically checked for pronunciation errors, as illustrated in Figure 1.3. Scientists are researching new models of speech synthesis working in a closed-loop cycle. They conduct perceptual evaluations and use their results to design and create new speech synthesis models.

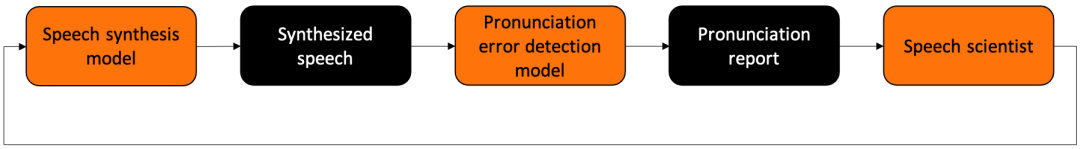


FIGURE 1.3: The use of a pronunciation error detection model to evaluate speech synthesis.

The pronunciation error detection model (Korzekwa, Lorenzo-Trueba, Zaporowski, et al., 2021) can be used at the training time of speech synthesis models. Traditionally, a speech synthesis model is trained in a supervised way by minimizing the mean square error between the synthesized and target speech signals, as shown in Figure 1.4. Adding another loss, which minimizes the probability of pronunciation errors, improves the stability of the synthesized speech.

1.5.2 Speech synthesis and voice conversion

The created speech synthesis and voice conversion models (Merritt, Ezzerg, et al., 2022; Jiao et al., 2021; Gabryś et al., 2021; Shah et al., 2021; Ezzerg et al., 2021) serve

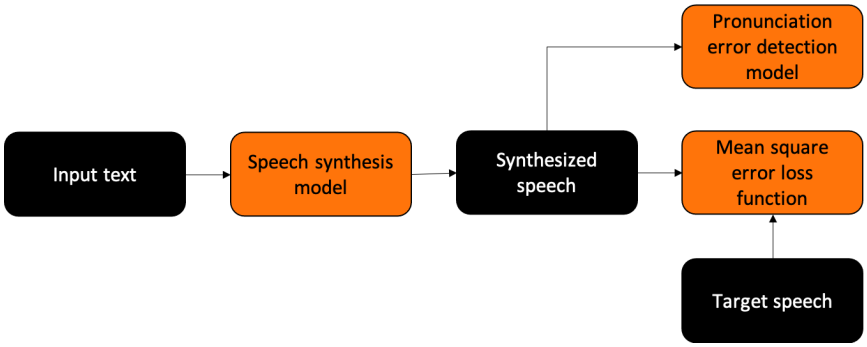


FIGURE 1.4: The use of pronunciation error detection at the training time of a speech synthesis model.

two purposes. Firstly, many of them are used by Alexa devices to generate synthetic speech and communicate with people, but the second important application of these methods from the point of view of the Ph.D. thesis is the generation of synthetic mispronounced speech, which improves the accuracy of pronunciation error detection models.

Universal vocoder (UV) (Jiao et al., 2021; Gabryś et al., 2021) is a model that converts a mel-spectrogram to a raw speech signal. A mel-spectrogram is generated based on the input text (Shah et al., 2021; Ezzerg et al., 2021). The vocoder is universal because it supports all speakers and speaking styles, eliminating the need to train a dedicated vocoder for each speaker. The universal nature of the vocoder makes it much easier for the Alexa device to speak with multiple voices, as shown in Figure 1.5. In addition, UV allows generating mispronounced speech for hundreds of speakers, which is used for training pronunciation error detection models (Korzekwa, Lorenzo-Trueba, Drugman, Calamaro, et al., 2021), as shown in Figure 1.6.

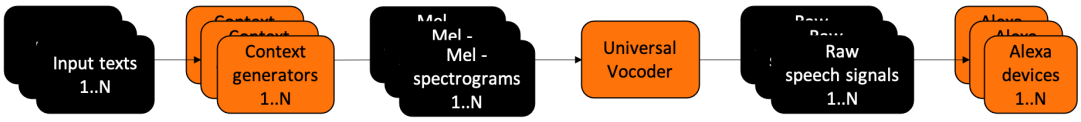


FIGURE 1.5: A single universal vocoder serving multiple text-to-speech requests across multiple voices.

FIGURE 1.6: A single universal vocoder serving the generation of mispronounced speech across multiple speakers.

Chapter 2

Research methodology

This chapter provides the basis for the topics of speech production, machine learning, and performance metrics to lay the groundwork for a detailed description of the research work and obtained dissertation results. Much of the material presented in this chapter is elaborated in the following chapters in relation to the doctoral dissertation.

2.1 Speech production

Spoken languages date back to 2000 B.C. The first spoken languages, Sumerian, Chinese, Mayan, used symbols to represent whole words (Jurafsky et al., 2009). Modern spoken languages represent different parts of words with symbols. Japanese hiragana is a syllabic language in which one symbol corresponds to one syllable. In contrast, Roman languages, such as English, use an alphabet of letters to represent different words. In the Ph.D. thesis, the focus is on English, as this is important from the product applicability point of interest for Amazon's Alexa. To recall, this thesis is realized within the "Implementation doctorate" program, carried out by the Gdańsk University of Technology, and written in agreement with the Amazon company employing the Ph.D. candidate.

The English alphabet consists of letters. Letters are the basic units of written words and then sentences. However, it is not enough to look at the letters to understand how to pronounce a word. The letter 'c' may be pronounced differently in the words 'cat' and 'cell' and a similar observation applies to other letters. The phonetic alphabet is made of phones. Each phone corresponds to a specific sound made by a speaker. The way a word is pronounced is determined by its phonetic transcription. For example, the word 'cat' is transcribed as [k ae t] and the word 'cell' is transcribed as [s eh l]. Phonemes are abstract symbols that correspond to the mental representation of the pronunciation of a word. Phoneme transcription is represented with slashes //, e.g. /s eh l/ as opposed to using brackets [] for phones. Two popular phonetic alphabets are International Phonetic Alphabet (IPA) and Arpabet (Jurafsky et al., 2009), as shown in Figure 2.1. In this Ph.D. thesis, the Arpabet representation is used. Different ways a phoneme can be pronounced (i.e., phonetic variations of a phoneme that do not change spoken word meaning) are called allophones (Piotrowska et al., 2021).

Vowels[2]

ARPABET 1-letter	ARPABET 2-letter	IPA	Example(s)
a	AA	ɑ	balm, bot
@	AE	æ	bat
A	AH	ʌ	butt
c	AO	ɔ	story
W	AW	aʊ	bout
x	AX	ə	comma
N/A	AXR[3]	ɚ	letter
Y	AY	aɪ	bite
E	EH	ɛ	bet
R	ER	ɝ	bird
e	EY	eɪ	bait
I	IH	ɪ	bit
X	IX	ɨ	roses, rabbit
i	IY	i	beat
o	OW	oʊ	boat
O	OY	ɔɪ	boy
U	UH	ʊ	book
u	UW	u	boot
N/A	UX[3]	ʉ	dude

Consonants[2]

ARPABET 1-letter	ARPABET 2-letter	IPA	Example
b	B	b	buy
C	CH	tʃ	China
d	D	d	die
D	DH	ð	thy
F	DX	ɾ	butter
L	EL	ḷ	bottle
M	EM	m̩	rhythm
N	EN	n̩	button
f	F	f	fight
g	G	g	guy
h	HH *or* H[3]	h	high
J	JH	dʒ	jive
k	K	k	kite
l	L	l	lie
m	M	m	my
n	N	n	nigh
G	NX *or* NG[3]	ŋ	sing
N/A	NX[3]	ɾ̃	winner
p	P	p	pie

Consonants[2]

ARPABET 1-letter	ARPABET 2-letter	IPA	Example
Q	Q	ʔ	uh-oh
r	R	ɹ	rye
s	S	s	sigh
S	SH	ʃ	shy
t	T	t	tie
T	TH	θ	thigh
v	V	v	vie
w	W	w	wise
H	WH	ʍ	why
y	Y	j	yacht
z	Z	z	zoo
Z	ZH	ʒ	pleasure

FIGURE 2.1: Arpabet phonetic alphabet (Arpabet, 2022).

Speech begins in the brain. The mental picture of the message is formed and represented by a phoneme sequence. The nervous system initiates the flow of air in the lungs. Air passes through the trachea, larynx, and then leaves the human body through the mouth and nose (Jurafsky et al., 2009). All key parts of the human body involved in speech production are presented in Figure 2.2. The flow of air caries energy in the form of fluctuations in air molecules oscillating at specific frequencies, creating sound waves. Many sound waves that oscillate in parallel at certain frequencies over time and carry certain energy are called speech. Various unique speech sounds with different energy at different frequencies over time are called phones. Simply speaking, the human brain of the listener receives the incoming flow of air through the ears and decodes the message, creating its mental representation on the listener's side.

2.1.1 Articulation

There are two types of phones, voiced and unvoiced sounds. Voiced sounds are created by introducing vibrations into the vocal folds located in the larynx organ. In unvoiced sounds, the vocal folds do not vibrate.

Phones are split into consonants and vowels. Consonants can be voiced, e.g. [b], [d], and unvoiced, e.g. [p], [t], while vowels, such as [a], [o], are generally voiced. In some languages, for example, Japanese, vowels can be unvoiced in a certain context. In English, all whispered vowels can be considered as unvoiced.

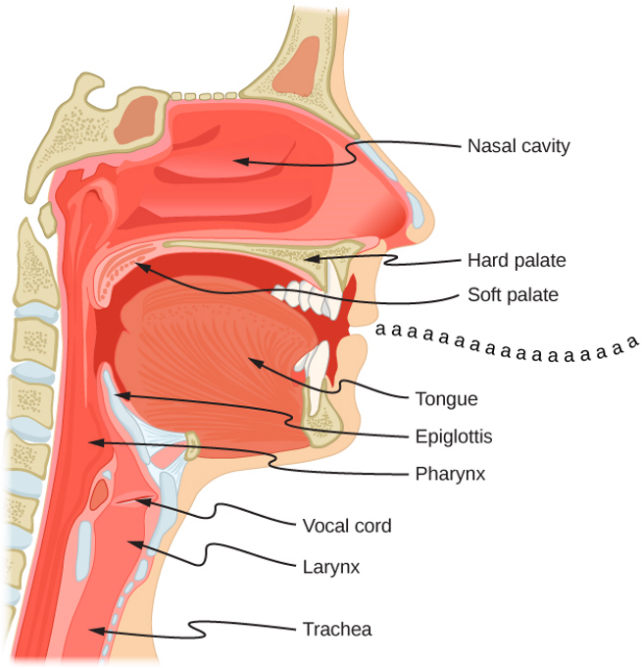


FIGURE 2.2: Organs involved in the process of speech production (*University physics Volume 1* 2016).

Consonants are formed by controlling different parts of the vocal tract. The vocal tract is the area above the trachea, which consists of the larynx and oral and nasal cavities. Depending on the location of the vocal tract that imposes the biggest restriction on airflow, consonants can be divided into labial, dental, alveolar, palatal, velar, and glottal categories. For example, [p] and [b] phones are called labial because they are generated by restricting airflow by putting lips together. Additionally, constants can be divided into stop, nasal, fricatives, sibilants, approximant, and tap categories, depending on the type of air restriction. For example, stop consonants, such as [b], [d], [p], require that the airflow be completely blocked for a short time 2.2.

Vowels are formed similarly to consonants by changing the position of the articulators, mainly of the tongue and lips. The tongue can be higher or lower in the nasal cavity, it can be moved forward to get closer to the lips or moved backward. Depending on the position of the tongue, vowels can be divided into front/back and high/medium/low categories. For example, in the word 'beet' [b iy t], the tongue is placed high and forward, whereas, in the word 'bat' [b ae t], the tongue is placed low and front. Changing the shape of lips is another important way to create vowels. For example, the phone [uw] requires the lips to be rounded off, as shown by the word 'tulip' [t uw l ix p]. Some vowels involve a change in the position of the articulators during the formation of a vowel, which corresponds to the production of two vowels immediately following each other. Such vowels are called diphthongs. For example, the diphthong [ow] in the word 'lotus' [l ow dx ax s].

Consonants and vowels make up syllables. Each syllable usually consists of one vowel and at least one consonant. The vowel part of a syllable is called the nucleus.

Syllables form words and words form sentences. Words can consist of one, two, three, etc., syllables. For example, the word 'napkin' [n a p - k ax n] has two syllables.

Non-native English speakers can make pronunciation mistakes for a number of reasons. They can incorrectly map the written word into a phoneme representation. Even if the phonemes are pronounced correctly, meaning that the correct phones are produced, the word will be mispronounced due to a mismatch between the expected (canonical) phoneme representation and the corresponding phoneme representation made in the human brain. Correctly encoding a word in the human brain does not mean it is pronounced correctly. Different languages have different phone sets, therefore, people may not be able to pronounce all the phones in the non-native language correctly. An example of this is the phone [th] in English; this phone does not exist in the Polish phone set. People may also skip phones while speaking or not be able to pronounce them because of various health problems such as dysarthria.

2.1.2 Prosody

Prosody is related to features of speech consisting of F0, energy, and duration. F0 is the fundamental frequency at which vocal folds vibrate. Energy is defined as the variance of a speech signal. Energy is usually expressed in decibels (dB), reflecting more human perception than raw energy values. The energy in dB is called loudness. Duration determines how long various sounds last, such as phones and silence between words and sentences (Jurafsky et al., 2009).

Prosody emphasizes different parts of speech, which usually corresponds to raising F0, increasing loudness and extending the duration of speech sounds (Jurafsky et al., 2009). Emphasizing syllables corresponds to lexical stress. English dictionary contains rules (lexical stress) that define which syllables in different words should be stressed. Sometimes, placing lexical stress on an incorrect syllable may change the meaning of the word, for example, the word 'produce' has two forms, the verb is stressed on the second syllable and the noun is stressed on the first syllable. In compound nouns, one word can be emphasized while the other is not. A compound noun is a noun that consists of two parts, two nouns or an adjective followed by a noun. For example, in the compound noun 'bulldog' the first noun is stressed.

Prosody is used to distinguish vocal patterns, e.g., to indicate whether a sentence is a question or not. Yes-no questions in English have raised intonation at the end, for example, 'Can we meet tomorrow?'. On the contrary, in declarative sentences such as 'We will meet tomorrow', the intonation falls down at the end of the sentence. Intonation can also be used to separate different words in enumerations. For example, the intonation slightly raises after each comma in the sentence 'One, two, three, start!'.

Non-native speakers can make prosodic mistakes in speech, for example, because they are not familiar with the rules defined in the language dictionary, such as which syllable to emphasize. Multiple studies have shown that correct prosody improves the intelligibility of speech (Field, 2005; Lepage et al., 2014), therefore, practicing the prosodic aspects of speech is an important part of CAPT.

2.2 Machine learning techniques

The doctoral thesis focuses on the application of deep learning techniques in automated pronunciation assessment. Deep learning is a branch of machine learning (LeCun et al., 2015). In general, machine learning is the process in which a machine automatically learns how to perform a specific task. For example, learn to classify images into two categories of cats and dogs. In the context of the doctoral dissertation, it is about learning to detect pronunciation errors in speech. Deep learning is a multidisciplinary field rooted in multiple related fields, including machine learning, probability theory, statistics, and mathematics. To explain deep learning, there are other areas that need to be discussed first, notably the probability theory, machine learning, and probabilistic machine learning (Bishop, 2006; Murphy, 2012).

In the simplest scenario, both machine learning and its probabilistic variant aim to learn the function $y = f(x)$. The variable x may represent an image, whereas the variable y may represent a decision whether the image represents a dog or a cat. The function $f()$ represents the mapping between both variables.

In machine learning, the variables x and y are vectors $x \in \mathcal{R}^{D_X}$ and $y \in \mathcal{R}^{D_Y}$in multidimensional spaces D_X and D_y, and the function f() can take any form. While in probabilistic machine learning, the variables x and y are constrained to the form of certain probability distributions, denoted as $x \sim p(x)$, $y \sim p(y)$, and $y = f(x) \sim p(y|x)$. The main idea behind probabilistic modeling is to represent variables and dependencies with probability distributions, as opposed to using only scalar or vector variables. Intuitively, probabilistic models account for the uncertainty by looking at all possible values of the input and output variables, whereas non-probabilistic methods only consider input and output variables as point estimates. Representing variables as probability distributions helps to overcome the problem of overfitting in which a machine learning model does not generalize well to unseen data, e.g., the inability to correctly classify unseen images into the categories of dogs and cats.

Interestingly, there are many similarities between both non-probabilistic and probabilistic machine learning. For example, a machine learning technique called dropout introduces random noise in the training process and consequently makes the input and output variables more probabilistic. In recent years, there has been a trend of mixing the concepts of probabilistic and non-probabilistic machine learning, gradually blurring the lines between the two types of machine learning. Good examples of such models are the Variational Auto-Encoder (VAE) (Van Den Oord et al., 2017) and Normalizing Flows (NF) (Kobyzev et al., 2020). To understand existing modern machine learning architectures and design new ones, it is important to explore both non-probabilistic and probabilistic views on machine learning.

Deep learning differs from machine learning in the way the function $y = f(x)$ is defined. In deep learning, this function has multiple levels of nesting: $y = f(x) = f_1 \circ f_2 \circ ... \circ f_n$, whereas in the non-deep variant there is just one function mapping from x to y. In the simplest possible scenario of two nested levels, the deep learning model is

defined as $y = f_1(f_2(x))$. Deep learning is often equated with Deep Neural Networks (DNN) that have multiple hidden functions (neural network layers). However, there are other types of deep learning models, such as Deep Gaussian Processes (DGP) (Damianou et al., 2013). Therefore, the term 'deep learning' should be considered more broadly.

The following sections will introduce in detail various concepts of machine learning in that are used in the Ph.D. thesis, including the probability theory, probabilistic machine learning, deep learning, and the probabilistic perspective on deep learning.

2.2.1 Probability theory

The probability theory provides a mathematical framework that enables modeling random events. A **random event**, also known as a **random variable**, represents an event with an unknown outcome. Imagine you are selecting a ball from a container with two balls, one red and one blue. This is an example of a random variable x with two possible outcomes $x \in \{red, blue\}$. If both balls are identical except for the color, the chances of blindly selecting red and blue balls will be the same. If there were three reds and one blue ball, the chances of choosing a red ball will be higher respectively. The chance that an event would lead to a certain outcome is also known as **likelihood** or **probability**.

The origins of the probability theory go back to the 16th century when Gerolamo Cardano studied games of chance such as roulette and dice, in which the outcome depends on random events (Ore, 2017). In the next century, Blaise Pascal made his first attempts to formulate the concept of expected value by studying a game of chance called 'problem of points' (Todhunter, 2014). The expected value, also know as 'expectation', is an important part of the probability theory (Bishop, 2006). In the 18th and 19th centuries, Thomas Bayes and Pierre Laplace formulated the probability theory as we know it today (Bishop, 2006).

The core of probabilistic machine learning is the probability theory, and in particular, its two concepts are very important: probability distribution and Bayes' theorem. Both concepts are described in this section, whereas a comprehensive look at the probability theory and probabilistic machine learning is presented in these three excellent books written in recent years. 'Pattern Recognition and Machine Learning' by Christopher Bishop (Bishop, 2006), 'Probabilistic Graphical Models: Principles and Techniques' by Daphne Koller (Koller et al., 2009), and 'Machine Learning: a Probabilistic Perspective' by Kevin P. Murphy(Murphy, 2012).

2.2.1.1 Probability distribution

Probability, also known as likelihood, or more colloquially a chance, is denoted as $p(x) \in [0, 1]$. The probability of a random event (random variable) x can be 0 - the event cannot take place, it can be higher than zero but less than 1 - the event may happen, or it can be exactly 1 - the event will always happen. A random variable

can have multiple outcomes, for example, selecting a ball from three possible colors with the value $p(x = red) = 0.2$ means that the probability of selecting the red ball is 20%. A random variable can be discrete or continuous. Selecting a ball out of a finite set of possible colors $x \in \{red, blue, green\}$ is an example of a discrete random variable, while selecting a number from a set of real numbers $x \in \mathcal{R}$ corresponds to a continuous random variable.

The function that defines the probabilities for all possible outcomes of a random variable is called a **probability distribution**, denoted as $x \sim p(x)$. The probability distribution of a discrete random variable is called a **Probability Mass Function (PMF)**, whereas a **Probability Density Function (PDF)** defines the probability distribution of a continuous random variable.

The PMF function can be presented in a tabular form (Table 2.1).

TABLE 2.1: The PMF function for the x variable presented in a tabular form (color of the ball selected from the container) .

x	**p(x)**
red	0.75
blue	0.25

The PDF function is usually represented by a mathematical equation, as illustrated by a random variable following the Normal probability distribution (Eq. 2.1). The Normal distribution, also known as Gaussian, is one of the commonly used representations of random variables due to its simple form that makes mathematical computations possible in closed form (Bishop, 2006). In practice, other continuous probability distributions are also used, such as Beta and Gama distributions (Murphy, 2012; Bishop, 2006).

$$p(x) \sim \mathcal{N}(\mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-0.5(\frac{x-\mu}{\sigma})^2} \tag{2.1}$$

The PMF and PDF functions must satisfy two conditions. First, the probability value must be greater or equal to 0:

$$p(x) \geq 0 \tag{2.2}$$

Second, the sum of the probabilities for all possible outcomes of the event must be 1, represented as a sum function (Eq. 2.3) and an integral function (Eq. 2.4) for discrete and continuous random variables, respectively.

$$\sum_x p(x) = 1 \tag{2.3}$$

$$\int p(x)dx = 1 \tag{2.4}$$

2.2.1.2 Conditional probability distribution

In the real world, multiple random variables can interact with each other. The probability distribution of one random variable may depend on the outcome of another variable. This concept is illustrated in the coin game, where the goal is to guess whether a coin will land on heads or tails. The coin can be fair, resulting in equal probabilities for both possible outcomes. However, the coin may be biased with one outcome more likely than the other, e.g., the coin is made by a pirate who wants to win the game by cheating. This scenario can be modeled using two random variables. The variable $y \in \{heads, tails\}$ represents the two possible coin outcomes, and the variable $x \in \{true, false\}$, with the value of $true$ if the coin comes from a pirate, $false$ otherwise.

The PMF functions for both x and y variables can be represented in tabular form, also known as a Conditional Probability Table (CPT) (Koller et al., 2009). Suppose that the probabilities of the x variable are the same for both outcomes, which means that there are equal chances that the coin can come from a pirate or not. The CPT for the x variable is shown in Table 2.2. In addition, let's assume that the probabilities of the y variable depend on the x variable - if the coin comes from a pirate, it is more likely to land on heads than on tails (the CPT is shown in 2.3). Both assumptions are known as **prior probabilities** or prior knowledge, giving information about the environment that is modeled with random variables. The probability distribution of the random variable y, denoted as $y \sim p(y|x)$, is called **conditional probability distribution** because it is conditioned on the variable x.

TABLE 2.2: The Conditional Probability Table (CPT) for the variable x - whether the coin is biased (comes from a pirate) or not.

x	p(x)
false	0.5
true	0.5

TABLE 2.3: The Conditional Probability Table (CPT) for the variable y (the outcome of the coin) conditioned on the variable x (the coin comes from a pirate or not).

x	y	p(y | x)
false	heads	0.5
false	tails	0.5
true	heads	0.6
true	tails	0.4

2.2.1.3 Bayesian networks

Random variables and their dependencies can be represented graphically using the framework of Probabilistic Graphical Models (PGM) (Darwiche, 2009; Koller et al., 2009). The PGM graph for the coin game is depicted in Figure 2.3. In PGM notation, circles represent random variables, whereas directed arrows represent dependencies between variables (conditional probability distributions). Each random variable can have many children and parent variables. PGM containing only directed arrows and no directed cycles is called Bayesian Network (Darwiche, 2009). The variant of PGM with unidirectional arrows is called Markov Network.

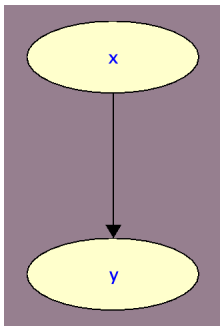


FIGURE 2.3: Probabilistic Graphical Model (PGM) for the coin game, including two random variables; x - whether the coin is biased (comes from a pirate), y - the outcome of a single coin toss. The PGM image was created with the SamIam - a tool for modeling and reasoning in Bayesian Networks (Darwiche, 2009)

Random variables can be multiplied with each other, the concept is known as **product rule** (Bishop, 2006). The product of multiple random variables results in the **joined probability distribution** shown in Eq. 2.5.

$$p(x, y) = p(x)p(y|x) = p(y)(x|y) \tag{2.5}$$

The random variable can be integrated out of the joined probability distribution, resulting in the **marginal probability distribution** over the remaining random variables. This process is known as the **sum rule** and is shown in Eq. 2.6.

$$p(x) = \int p(x, y)dy \tag{2.6}$$

Both the sum and product rules can be combined to form the Bayes's theorem, also known as the Bayes rule, as shown in Eq. 2.7.

$$p(x|y) = \frac{p(x)p(y|x)}{p(y)} \tag{2.7}$$

The sum rule, product rule, and Bayes rule provide a powerful framework for reasoning and making decisions under uncertainty. Reasoning, also known as **inference**, is the process of estimating the state of a random variable based on evidence provided

by other dependent random variables. For example, the conditional probability $p(x|y)$ that the coin is biased given it has landed on heads can be estimated using the Bayes rule in Eq. 2.7. This new state of the random variable given evidence is known as **posterior probability** or posterior probability distribution. The posterior probability contrasts with **prior probability** that represents the belief about the random variable before observing the outcomes of dependent variables. An unobserved variable is referred to as a **hidden variable** or a **latent variable**. The marginal probability of a latent variable can be estimated by integrating other latent variables using the sum rule in Eq. 2.6, the process also known as marginalization. For example, the probability that a coin will land heads is given by $p(y) = \int p(x,y)dx$.

In this coin game example, there is only one coin toss represented by a single random variable. To estimate the probability that the coin is biased, given it has landed on heads twice, another random variable is added to the PGM graph, as illustrated in Figure 2.4. The Figure shows the posterior probability of the x variable (whether the coin is biased) given it has landed heads twice.

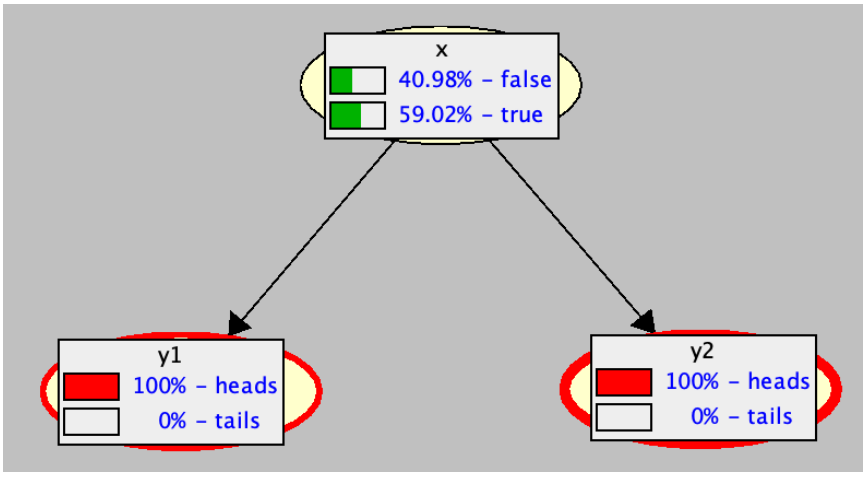


FIGURE 2.4: Probabilistic Graphical Model (PGM) for the coin game, including three random variables. x - whether the coin is biased (comes from a pirate), $y1, y2$ - the outcomes of two coin tosses. The PGM image was created with the SamIam - a tool for modeling and reasoning in Bayesian networks (Darwiche, 2009)

The presented foundations of the probability theory form the basis of both probabilistic and non-probabilistic machine learning. The concepts of graphical models and reasoning (inference) about latent variables enable the creation of different types of machine learning models for tasks such as prediction, detection, and classification.

2.2.2 Probabilistic machine learning

Probabilistic machine learning is based on probability theory. The basic principle of modeling real problems with random variables and inferring the state of a hidden variable given some evidence can be applied to many practical problems. Different problems can be solved with different model architectures such as Hidden Markov

Models, Gaussian Mixture Models, Kalman Filters, Gaussian Processes, and Variational Auto Encoders (Bishop, 2006; Goodfellow et al., 2016). Each model can include different sets of latent variables and their dependencies (conditional probability distributions). Latent variables can have different probability distributions, discrete (Binomial, Multi-modal) and continuous (Gaussian, Beta, Gamma). Probabilistic models can be trained with the help of many algorithms such as Belief Propagation, Expectation Maximization, Expectation Propagation, and Variational Inference (Bishop, 2006). All of these considerations on how to apply in practice the basic principles of random variables and Bayes-rule define what probabilistic machine learning is about.

To see probabilistic machine learning in action, consider the problem of estimating temperature values from noisy observations. The training data consist of N temperature measurements y_i, where $i = 1..N$, collected at different t_i time locations in the $(0, 10)$ range. The task is to estimate the temperature values at unseen time locations within the range of the training data (interpolation task) and outside this range (extrapolation task).

Figure 2.5 shows the estimated (predicted) mean temperature values for four different probabilistic model architectures. The mean values are accompanied by the corresponding 95% confidence intervals. Probabilistic machine learning gives confidence intervals 'for free' by modeling latent variables with probability distributions. Probabilistic distributions are usually parametrized by the mean μ and variance σ_2 parameters that can be converted to confidence intervals using the corresponding CDF function (Bishop, 2006). The corresponding probabilistic model architectures, Naive Bayes, Hidden Markov Model, and two variants of Non-parametric Gaussian Process, are presented in Figure 2.6. All models represent the latent temperature variables with the Gaussian distribution $p(x_i) = \mathcal{N}(\mu_{x_i}, \sigma^2_{x_i})$. The noisy temperature observations follow the conditional Gaussian distribution $p(y_i|x_i) = \mathcal{N}(x_i, \sigma^2_{y_i})$. However, these models differ in how the latent variable x_t changes and correlates across time locations, leading to different abilities in interpolation and extrapolation tasks.

2.2.2.1 Naive Bayes

The simplest model, known as Naive Bayes (Murphy, 2012), has one latent variable x across all observations y_i (Figure 2.6a). Consequently, the model estimates a single temperature value across all time locations, as shown in Figure 2.5a).

Posterior estimation

The posterior value of the variable x is defined by:

$$p(x|y_1, y_2, .., y_N) \propto p(x) \prod_{i=1}^{N} p(y_i|x) \tag{2.8}$$

with the prior and conditional probability distributions $p(x)$ and $p(y_i|x)$ defined by:

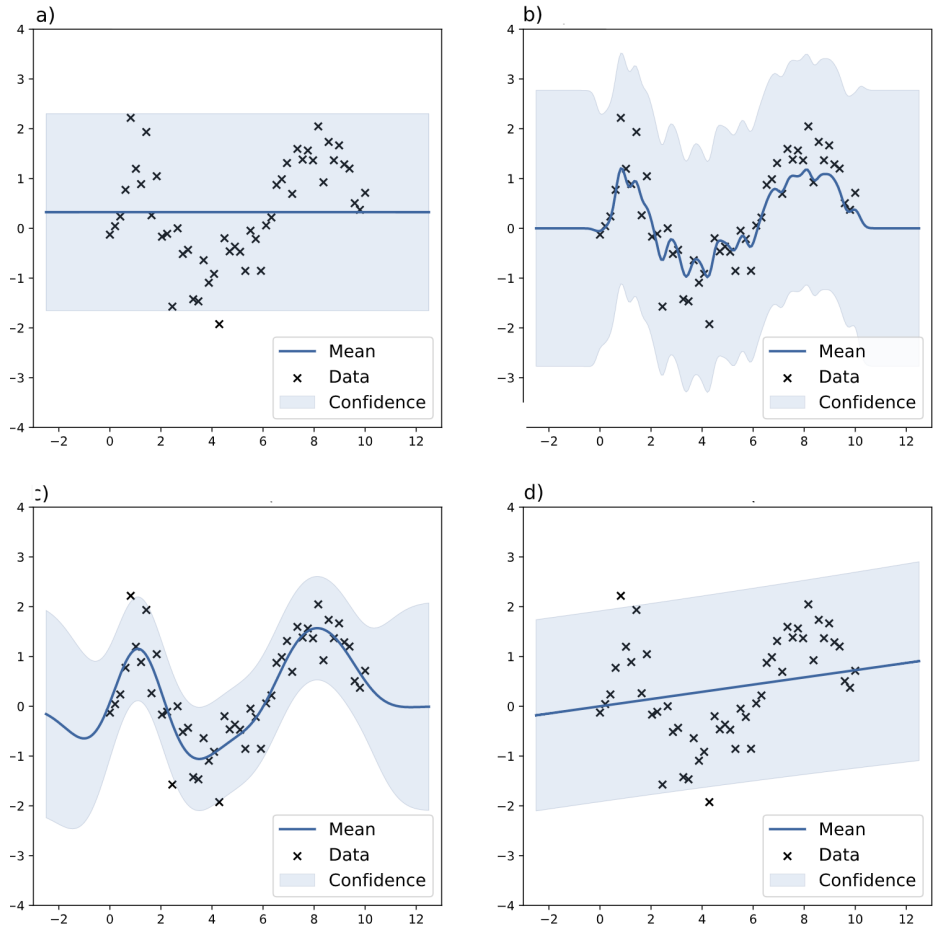


FIGURE 2.5: Posterior plots for different probabilistic model architectures from Figure 2.6: a) Naive Bayes - a single latent variable x estimated from multiple independent observations $\{y_1, y_2, ..., y_N\}$, b) Hidden Markov Model - a latent variable x_i conditioned on the local context of two neighboring variables x_{i-1} and x_{i+1}, c) Gaussian Process with the RBF kernel - a model with an infinite number of latent variables $\{x_1, x_2, ..., x_N\}$ conditioned on independent observations $\{y_1, y_2, ..., y_N\}$, and d) Gaussian Process with the Linear kernel - a model with an infinite number of latent variables. Each plot contains observations from the training set, the predicted mean values, and the corresponding 95% confidence interval.

$$p(x) = \mathcal{N}(\mu_x, \sigma_x^2) \tag{2.9}$$

$$p(y_i|x) = \mathcal{N}(x, \sigma_y^2) \tag{2.10}$$

Since all the terms in Eq. 2.8 follow Gaussian distributions, the posterior over x can be estimated analytically:

$$p(x|y_1, y_2, .., y_N) = \mathcal{N}(\tilde{\mu}_x, \tilde{\sigma}_x^2) \tag{2.11}$$

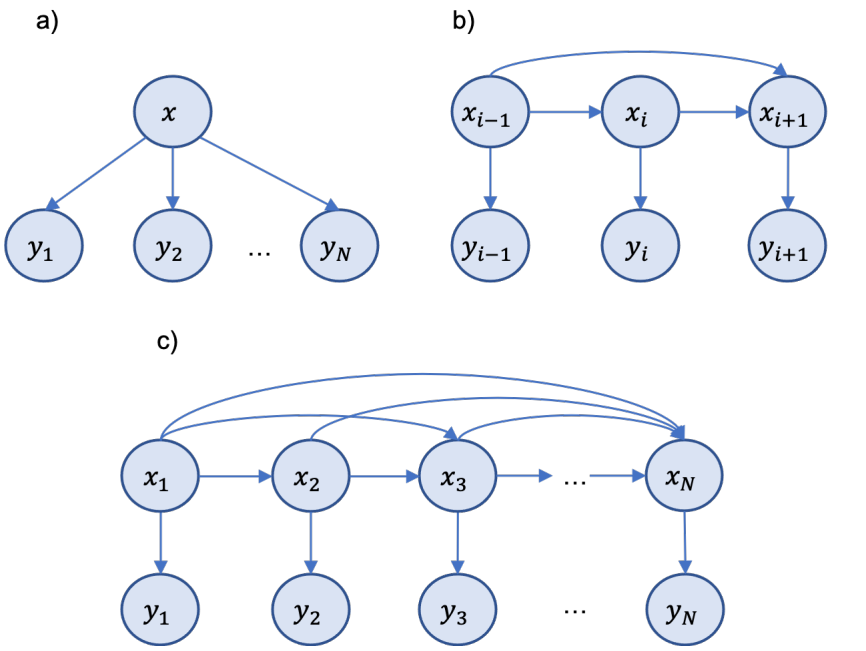


FIGURE 2.6: Graphical models for different probabilistic model architectures: a) Naive Bayes - a single latent variable x estimated from multiple independent observations $\{y_1, y_2, ..., y_N\}$, b) Hidden Markov Model - a latent variable x_i conditioned on the local context of two neighboring variables x_{i-1} and x_{i+1}, and c) Gaussian Process - a model with infinite number of latent variables $\{x_1, x_2, ..., x_N\}$ conditioned on independent observations $\{y_1, y_2, ..., y_N\}$.

, where:

$$\tilde{\mu}_x = \frac{\sigma_y^2}{N\sigma_x^2 + \sigma_y^2}\mu_x + \frac{N\sigma_x^2}{N\sigma_x^2 + \sigma_y^2}\mu_{y_{ml}} \tag{2.12}$$

$$\tilde{\sigma_x}^2 = \frac{1}{\frac{1}{\sigma_x^2} + \frac{N}{\sigma_y^2}} \tag{2.13}$$

, where $\mu_{y_{ml}}$ is estimated with the maximum likelihood approach (Bishop, 2006) given by:

$$\mu_{y_{ml}} = \frac{1}{N}\sum_{n=1}^{N} y_i \tag{2.14}$$

Model training

The Naive Bayes model presented in Figure 2.6a is parametrized with the parameters of the prior and conditional probability distributions (Equations 2.9 and 2.10). The model parameters can be estimated in two ways. First, by extending Eq. 2.8 to include the prior variables over the parameters of the model. For example, to learn the mean μ_x parameter of the prior distribution $p(x)$, a latent variable $p(\mu_x)$ can be added to the equation:

$$p(x|y_1, y_2, .., y_N) \propto \int p(\mu_x)p(x|\mu_x)\prod_{i=1}^{N} p(y_i|x)d\mu_x \tag{2.15}$$

Interestingly, it can be seen that there is not much difference between inferring (estimating) the value of the latent variable of interest x (Eq. 2.8) and inferring (learning) and then integrating out the parameters of the model (Eq. 2.15). Both tasks, estimating and learning, use the same basic rules of the probability theory: the sum rule, product rule, and Bayes rule.

The second way to estimate the parameters of the model is to use a standard optimization technique such as gradient descent (Bishop, 2006), in which the parameters of the model $\theta = \{\theta_x, \theta_y\}$ are estimated by finding the maximum of the likelihood function $\arg\max_\theta \mathcal{L}(\theta)$. The likelihood function is defined by:

$$\mathcal{L}(\theta) = p(y_1, y_2, .., y_N|\theta) \propto \int p(x|\theta_x)\prod_{i=1}^{N} p(y_i|x, \theta_y)dx \tag{2.16}$$

Due to numerical instabilities, the log-likelihood function is minimized in practice. The likelihood function can be computed analytically for probabilistic models with both prior and conditional probability distributions represented by the Gaussian distribution (Bishop, 2006). For other distributions, approximation techniques such as Monte Carlo sampling (Koller et al., 2009) and Variational Inference (Bishop, 2006) are often used.

Summary of the Naive Bayes model

The example of the Naive Bayes model recalled above illustrates the general mechanism of using the probability theory to design probabilistic machine learning models. Probabilistic models differ in architecture. In some cases, the inference process is analytically tractable, but in others, optimization-based techniques are used. Some models have more, and some have fewer random variables. However, regardless of the model architecture, all models can be derived using the same probability theory. The following sections present more advanced probabilistic models for the temperature estimation task, for which the inference process has no analytical solution and requires optimization-based techniques.

2.2.2.2 Hidden markov model

The Naive Bayes model described in the previous section estimates only the average temperature value across all time locations. The model introduced in this section, known as Hidden Markov Model (HMM) (Bishop, 2006), addresses this limitation by modeling local time dependencies between latent variables x_i, x_{i-1}, and x_{i+1}. Note that the vanilla HMM model only includes a dependency on the past variable $p(x_i|x_{i-1})$. Here, a slightly modified version of the model is presented, which takes into account both the past and future time dependencies $p(x_i|x_{i-1}, x_{i+1})$. The model

architecture is presented in Figure 2.6b. The estimated temperature values by the model are shown in Figure 2.5b. The model interpolates well but is not capable of reasoning beyond the range of the training data. Modeling only the local context does not capture long-term dependencies in the data, which results in poor performance in the extrapolation task.

Probabilistic models based on local context dependencies have long been studied (Särkkä, 2013). Most often, these models belong to the class of models known as Markov Chains (Bishop, 2006). The Markov Chain, or Markov Process, is a stochastic process in which the state of the latent variable x_i depends only on the state of the latent variable x_{i-1} at the previous time. In other words, the future and the past are independent of each other given the current state is known. Kalman Filter and Exponential Moving Average (EMA) are two examples of Markov Chain-based models.

Posterior estimation

The posterior of the ith temperature latent variable is defined by:

$$p(x_i|y_1, y_2, .., y_N) \propto \int p(x_1) \prod_{i=2}^{N} p(x_i|x_{i-1}, x_{i+1}) \prod_{i=1}^{N} p(y_i|x_i) dx_{1..N\setminus i} \tag{2.17}$$

Similarly to the Naive Bayes model in the previous section, a posterior variable $p(x_i|y_1, y_2, .., y_N)$ can be calculated analytically for certain forms of conditional probability distributions, such as Gaussian. However, this process is computationally expensive for long sequences.

Belief propagation, also known as 'message passing', is a popular algorithm that can efficiently compute posterior values for multiple latent variables x_i (Koller et al., 2009; Bishop, 2006). In a nutshell, posteriors for latent variables x_i are computed iteratively using the current best posterior estimates of the other dependent variables. Once the posterior value for one variable is estimated, its state is sent as a message to other dependent variables in the PGM graph. Hence, the name of this algorithm is 'message passing'.

Let us consider a simplified model of three latent variables defined by $p(x_0, x_1, x_2) = p(x_0)p(x_1|x_0)p(x_2|x_1)$. To estimate the posterior value of the variable x_1 conditioned on the observed variable x_2, two incoming messages are needed from both neighboring variables, $m_{0->1}$ and $m_{2->1}$. The message $m_{0->1}$ is defined by:

$$m_{0->1} = \int p(x_0)p(x_1|x_0)dx_0 \tag{2.18}$$

whereas the message $m_{2->1}$ is defined by:

$$m_{2->1} = p(x_2|x_1) \tag{2.19}$$

then the posterior of x_1 is defined as the product of both messages:

$$p(x_1|x_2) = m_{0->1} m_{2->1} \tag{2.20}$$

Messages are sent between the variables of the PGM graph till convergence, i.e., the delta between two consecutive posterior estimates is lower than a certain threshold. If the PGM graph is a tree, i.e., there are no loops between the variables and all messages can exactly be computed, i.e., no approximations are used to estimate any message, then the messages in the graph need to be passed only twice. This variant of Belief propagation is known as the forward-backward message passing algorithm (Koller et al., 2009). If the PGM is a graph, i.e., there are loops between the variables, and all messages are computed exactly, the messages in the graph usually have to be passed more than twice to reach the convergence point. This variant is called Loopy Belief Propagation (Koller et al., 2009). In addition, if the messages are based on approximated probability distributions, then Loopy Belief Propagation is known as the Expectation Propagation algorithm (Minka, 2013).

Model training

Conceptually, the HMM model can be trained in the same way as the simpler Naive Bayes model from the previous section. That is, either by introducing latent variables representing the parameters of the model $\theta = \{\theta_x, \theta_y\}$ or by directly optimizing the likelihood function of the data. However, due to the complicated forms of the posterior distribution (Eq. 2.17) and the likelihood function (Eq. 2.21), these techniques are often computationally intractable.

$$\mathcal{L}(\theta) = p(y_1, y_2, .., y_N|\theta) \propto \int p(x_1|\theta_x) \prod_{i=2}^{N} p(x_i|x_{i-1}, x_{i+1}, \theta_x) \prod_{i=1}^{N} p(y_i|x_i, \theta_y) dx \tag{2.21}$$

Expectation Maximization (EM) is an iterative algorithm that enables the training of complex probabilistic models (Moon, 1996). The Baum-Welch algorithm is a popular variant of EM-based methods of estimating the parameters of latent variables for more advanced probabilistic models (Welch, 2003). The algorithm decomposes a complex task of computing and optimizing the likelihood function into two simpler steps.

The Maximization step maximizes the likelihood function in Eq. 2.21 with respect to the model parameters. The calculation of the likelihood function is complicated due to the latent variables that have to be integrated out. If there were no latent variables, the likelihood function could be factorized into the product of independent likelihood terms and be much easier to estimate:

$$\mathcal{L}_{\mathcal{EM}}(\theta) = p(y_1, y_2, .., y_N|\theta) \propto \prod_{i=1}^{N} p(y_i|x_i, \theta) \tag{2.22}$$

The redefined likelihood function $L_{EM}(\theta)$ is called the expected likelihood function because it depends on the estimates (expectations) of the latent variables. However, the model latent variables $x_1, .., x_N$ are not observed. To overcome this problem, the posteriors of the latent variables are computed based on the current best estimates of the model θ parameters using Eq. 2.17 - this is the Expectation step.

The EM algorithm is a chicken and egg problem. To compute and maximize the likelihood function $L_{EM}(\theta)$ in the Maximization step, the posteriors of the latent variables $x_1, .., x_N$ have to be known in advance. To estimate the latent variables during the Expectation step, the model parameters θ are needed. The EM algorithm interchangeably iterates between the Expectation and Maximization steps till the model converges, that is, until the posteriors of the latent variables and the model parameters fall below a certain threshold.

Summary of the HMM model

While discussing the HMM model, two important concepts were introduced. First, it has been shown that the exact estimation of the posteriors of latent variables in more complex probabilistic models is not always feasible. In theory, probabilistic machine learning attracts with the beauty of its basic principles based on the probability theory. However, in practice, optimization-based algorithms such as message-passing, Belief Propagation, and Expectation Propagation are required to compute the posteriors of latent variables.

The EM algorithm is another important concept introduced in this section. There are many machine learning algorithms that have their roots in the EM method, such as k-means clustering, EM clustering, Auto-Encoders, Variational-Auto-Encoders, and Variational Inference (Bishop, 2006; Goodfellow et al., 2016). Studying the similarities between different algorithms strengthens understanding of machine learning in general and makes it easier to invent new machine learning techniques to solve new problems.

2.2.2.3 Non-parametric Gaussian processes

The two previously described Naive Bayes and HMM models show that adding more latent variables increases the accuracy of the temperature estimation. While this is generally true, it comes at the cost of increasing the complexity of the model, making it more likely to overfit the training data.

Interesting things happen when the model complexity grows to the point where there are an infinite number of latent variables and dependencies between them. Suddenly, the prior over the latent variables and their conditional dependencies can be computed using a relatively simple function parametrized with a few parameters only. Such a model is capable of representing complex distributions without overfitting to the training data. An example of such a model is the Gaussian Process (GP) model (Williams et al., 2006).

Definition of the Gaussian Process model

In general form, GP is simply a multivariate Gaussian distribution over latent variables $\mathbf{x} = \{x_1, ..., x_N\}$ conditioned on observations $\mathbf{y} = \{y_1, .., y_N\}$:

$$p(\mathbf{x}, \mathbf{y}) = p(\mathbf{x})p(\mathbf{y}|\mathbf{x}) \tag{2.23}$$

where the latent variable $\mathbf{x}$ follows the Multivariate Normal distribution $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ parametrized with the mean $\boldsymbol{\mu}$ and the covariance matrix $\boldsymbol{\Sigma}$ parameters. The covariance matrix $\boldsymbol{\Sigma}$ is computed using the covariance function, also known as the kernel function, or just the kernel. The ij-th element of the covariance matrix $\boldsymbol{\Sigma}$ is defined by:

$$\boldsymbol{\Sigma}_{ij} = cov(\mathbf{x}_i, \mathbf{x}_j) \tag{2.24}$$

where $\mathbf{x}_i$ and $\mathbf{x}_j$ are the i-th and j-th elements of the latent variable $\mathbf{x}$.

The covariance function defines the form of the function that can be modeled by the latent variable $\mathbf{x}$. For example, the underlying function can be smooth, periodic, linear, or it can model both global and local temporal dependencies.

The likelihood function $p(\mathbf{y}|\mathbf{x})$ is also a Gaussian function conditioned on the latent variable $\mathbf{x}$. In its basic form, the likelihood function assumes that an individual observation y_i is conditioned only on the corresponding latent variable x_i, defined by:

$$p(y_i|\mathbf{x}_i) = \mathcal{N}(\mathbf{x}_i, \sigma_y^2) \tag{2.25}$$

where σ_y^2 is the noise related to imperfect observation y_i of the latent variable $\mathbf{x}_i$.

Gaussian Process for the temperature estimation problem

A graphical representation of the GP model for the temperature estimation problem is shown in Figure 2.6c. The temperature observations are represented by the variable $\mathbf{y}$, whereas the latent variable $\mathbf{x}$ represents the temperature values over time that are estimated from noisy observations $\mathbf{y}$. The estimated temperature values for the GP model in both interpolation and extrapolation tasks are presented in Figures 2.5c and2.5d for the Radial Basis Function (RBF) kernel and the linear kernel respectively.

The RBF kernel (Duvenaud, 2014), also known as Gaussian or Squared Exponential Kernel, is defined by:

$$cov_{rbf}(\mathbf{x}_i, \mathbf{x}_j) = \sigma^2 \exp(-\frac{|\mathbf{f}(\mathbf{x})_i - \mathbf{f}(\mathbf{x})_j|^2}{2l^2}) \tag{2.26}$$

The RBF kernel imposes the constraint that the represented function is smooth, which means that points close to each other have more similar values than points falling more apart. The function $f(\mathbf{x})$ returns the feature vector for the latent variable $\mathbf{x}$. In the case of the temperature problem, the feature vector corresponds to time information, e.g., the number of seconds since 1970-01-01 00:00:00, but it can contain

any multi-dimensional data that are supported by Euclidean distance. The variance parameter σ_2 tells how much the function values can differ from the mean value of the function. The length scale l^2 parameter indicates how many different variables $\mathbf{x}_i$ depend on each other over time. The higher the value, the stronger the temporal dependency.

The RBF kernel performs well for both interpolation and extrapolation tasks. Its behavior in the extrapolation task is especially noteworthy. The function estimated with the RBF kernel can maintain its trend outside the regions of the training data, as shown in Figure 2.5c, while using the confidence score to reflect the increasing uncertainty of the estimated values.

The Kernel Cookbook (Duvenaud, 2014) presents different types of kernels such as Rational Quadratic Kernel, Periodic Kernel, Locally Periodic Kernel, and Linear Kernel. Different kernels can be combined to form new kernels by using the multiplication or addition functions. For example, Linear times Periodic kernel or RBF plus Linear kernel.

To get a better intuition on how different kernels perform in the temperature estimation problem, the GP model with a linear kernel is evaluated, with the results presented in Figure 2.5d. The linear kernel (Eq. 2.27) corresponds to Bayesian linear regression (Williams et al., 2006), having the ability to model only linear functions with respect to the feature vector $f(\mathbf{x})$ (Williams et al., 2006).

$$cov_{linear}(\mathbf{x}_i, \mathbf{x}_j) = \sigma_b^2 + \sigma_v^2 (f(\mathbf{x})_i - c)(f(\mathbf{x})_j - c) \tag{2.27}$$

GP covariance matrices can be presented graphically, providing some insights into how the latent variables $\mathbf{x}_i$ are correlated with each other. Figures 2.7c and 2.7d show the covariance matrices for the RBF and linear kernels, respectively. Interestingly, two previously described models, the Naive Bayes and HMM, can be seen as special cases of GP with particular forms of the kernel function. Figure 2.7a shows the covariance matrix for the Naive Bayes model, whereas the HMM model is presented in Figure 2.7b.

Posterior estimation

Consider the task of estimating the temperature value $\tilde{x}_* \sim \mathcal{N}(\tilde{\mu}_*, \tilde{\sigma}_*^2)$ at the location x_*. The variable $\mathbf{y}$ represents the observed temperature values, and $\mathbf{x}$ denotes the corresponding latent variable. The posterior mean of x_* is defined by:

$$\tilde{\mu}_* = k(x_*, \mathbf{x})(k(\mathbf{x}, \mathbf{x}) + \sigma_y^2 I)^{-1}\mathbf{y} \tag{2.28}$$

whereas the posterior variance is given by:

$$\tilde{\sigma}_*^2 = k(x_*, x_*) - k(x_*, \mathbf{x})(k(\mathbf{x}, \mathbf{x}) + \sigma_y^2 I)^{-1} k(\mathbf{x}, x_*) \tag{2.29}$$

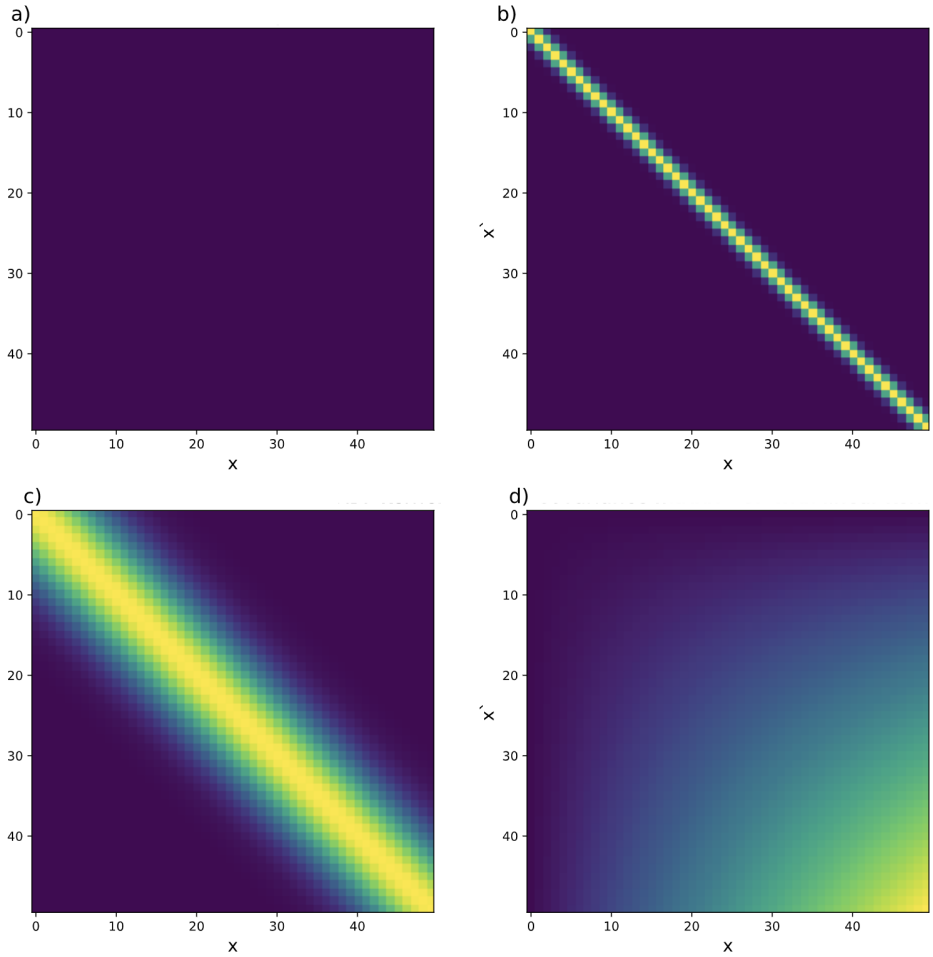


FIGURE 2.7: Covariance plots for different probabilistic model architectures from Figure 2.6: a) Naive Bayes - a single latent variable x estimated from multiple independent observations $\{y_1, y_2, ..., y_n\}$, b) Hidden Markov Model - a latent variable x_i conditioned on the local context of two neighboring variables x_{i-1} and x_{i+1}, c) Gaussian Process with the RBF kernel - a model with infinite number of latent variables $\{x_1, x_2, ..., x_n\}$ conditioned on independent observations $\{y_1, y_2, ..., y_n\}$, and d) Gaussian Process with the Linear kernel - a model with infinite number of latent variables. The covariance function, also known as a kernel or covariance matrix, is computed with $cov(x, x')$ for all possible combinations of latent variables $\{x_1, x_2, ..., x_n\}$. The form of a $cov()$ function depends on the probabilistic model architecture.

$k(\mathbf{x}, \mathbf{x})$ is a shortcut for the covariance function $cov(\mathbf{x}, \mathbf{x})$. The variance σ_y^2 is the independent Gaussian noise of the likelihood function from Eq. 2.25.

The equations for $\tilde{\mu}_*$ and $\tilde{\sigma}_*^2$ are computationally expensive, with cubic runtime complexity $O(N^3)$ and quadratic space complexity $O(N^2)$, where N is the number of observations (temperature measurement) in the training data. The key operation is to compute the inverse of the covariance function $(k(\mathbf{x}, \mathbf{x}) + \sigma_y^2 I)^{-1}$, where the dimensionality of $\mathbf{x}$ is N. One way to overcome high computational complexity is to use inducing points, which will lower the dimensionality of the covariance matrix

from NxN to NxM, where M is the number of inducing points (Williams et al., 2006). The inducing points can be selected directly from the training data by random selection, clustering the training data into clusters, or creating 'virtual' inducing points during optimization of the likelihood function.

Model training

The model is trained with a gradient decent-based algorithm by optimizing the marginal likelihood function defined in Equations 2.30 and 2.31. The runtime and space complexity are the same as for the case of posterior estimation presented in the previous section: $O(N^3)$ and $O(N^2)$, respectively. A similar technique based on inducing points can be used to scale training to larger datasets.

$$p(\mathbf{y}|\boldsymbol{\theta}) = \int p(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta}) p(\mathbf{x}, \boldsymbol{\theta}) d\mathbf{x} \tag{2.30}$$

$$log\mathcal{L}(\boldsymbol{\theta}) = logp(\mathbf{y}|\boldsymbol{\theta}) = -0.5\mathbf{y}^T(k(\mathbf{x},\mathbf{x}) + \sigma_n^2 I)^{-1}\mathbf{y} - 0.5log|k(\mathbf{x},\mathbf{x}) + \sigma_y^2 I| - \frac{n}{2}log2\pi \tag{2.31}$$

where $\boldsymbol{\theta}$ represents trainable model parameters.

Summary of Gaussian Processes

Gaussian Processes (GPs) provide a powerful framework for creating probabilistic machine learning models. With the use of a covariance function, many model architectures can be created, each taking into account different prior assumptions. Depending on the choice of the covariance function, GPs can capture both short-term and long-term temporal dependencies in the training data. GPs perform very well when the model has to make decisions under uncertainty with relatively little training data available.

GPs have some weaknesses, despite their solid mathematical foundations and the ability to generalize to multiple different modeling use cases. First, GPs are computationally expensive, and it is difficult to scale this method to millions of training examples. Second, GP is a shallow machine learning model, which means that it cannot easily discover deep dependencies in the data - something that deep neural networks (Goodfellow et al., 2016) and decision trees (Ali et al., 2012) can do. There is a deep learning model called Deep Gaussian Processes (Damianou et al., 2013) that can include multiple GP layers stacked on top of each other, but this model is computationally expensive. Finally, GP models make Gaussian assumptions about the prior probability distribution and the likelihood function, which can lead to less accurate posterior estimates in applications such as vision and speech.

2.2.2.4 Summary of probabilistic machine learning

Probabilistic machine learning models provide an elegant framework for creating generative models that can reason under uncertainty. However, probabilistic models make strong assumptions about the generative process behind the training data, often modeling latent variables with the Gaussian distribution. The Gaussian distribution is used not because it represents the underlying process well, but because the mathematics behind it becomes simpler. One alternative to probabilistic models are deep learning techniques such as deep neural networks. Deep neural networks can more accurately represent the underlying generative process without making Gaussian assumptions, leading to more precise models. In addition, deep neural networks can incorporate elements of probabilistic machine learning to create models that are both precise and can reason under uncertainty. The following two sections present deep neural networks and their probabilistic perspective in more detail.

2.2.3 Deep learning

Deep learning generally refers to any machine learning model that can learn data representation at multiple levels. Such models consist of multiple layers processing the input signal through a series of transformations to generate the output signal. Each layer can take inputs from multiple layers and generate new data that represent specific signal characteristics. Deep Neural Networks (DNN), the most popular class of deep learning, the task is to estimate the variable $y = f(x)$, where the output y and the input x variables can be scalars, vectors, or tensors, and the dependencies between the variables are represented by computational blocks such as Feed-forward Layer (Goodfellow et al., 2016), Convolutional Neural Network (CNN) (Gu et al., 2018), and Recurrent Neural Network (RNN) (Sutskever et al., 2014).

One of the first commercially deployed deep learning models is the speaker verification system based on multi-layer neural networks (Heck et al., 2000). Deep learning is commonly identified with neural networks, but there are other types of deep learning models, such as Deep Gaussian Processes (Damianou et al., 2013). This section focuses on deep learning techniques that are used in the thesis to create various models for detecting pronunciation errors in non-native speech.

2.2.3.1 Perceptron, dense layer and multi-Layer perceptron

The perceptron is a basic building block of deep neural networks (Rosenblatt, 1960). Let $\mathbf{x}$ be a $1 \times n$ input vector, $\mathbf{w}$ be $1 \times n$ vector of trainable parameters, w_0 be a trainable scalar parameter, and κ be a non-linear transform function. The output scalar value y_1 is computed as follows:

$$y_1 = \kappa(\mathbf{x}\mathbf{w}^{\mathbf{T}} + w_0) \tag{2.32}$$

A graphical representation of the perceptron is shown in Figure 2.8a. The non-linear transform κ is known as the activation function. Popular variants of the activation function include the sigmoid, TanH and ReLU (Rectified Linear Unit) functions(Goodfellow et al., 2016). The perceptron can be used as a binary classification model, but only for patterns that can be linearly separated. Exclusive OR (XOR) is a classic non-linear function $f : X \mapsto y$, where $X \in \mathcal{R}^2$ and $y \in \{0, 1\}$, which cannot be separated linearly into two binary categories.

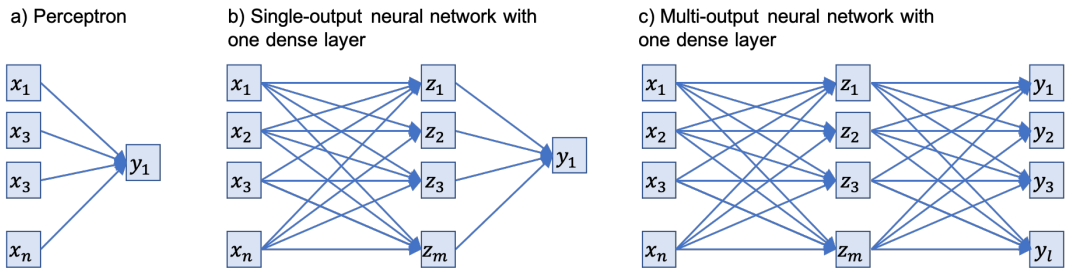


FIGURE 2.8: Neural network architectures based on the perceptron and a dense layer components: a) neural network with input vector $\mathbf{x}$ and scalar output y_1, known as the perceptron, b) neural network with input vector $\mathbf{x}$, one dense layer $\mathbf{z}$, and scalar output y_1, c) neural network with input vector $\mathbf{x}$, one dense layer $\mathbf{z}$, and vector-based output $\mathbf{y}$.

The perceptron can be generalized by stacking multiple layers, also known as dense layers, on top of each other. Such a model is called a Multi-Layer Perceptron (Goodfellow et al., 2016). MLP is shown in Figure 2.8b. By stacking multiple layers, the model is able to separate non-linear multi-dimensional spaces such as the XOR function, but only if the κ activation function is non-linear. Stacking multiple layers followed by linear activation functions does not make the model non-linear. In addition, MLP can support multi-output functions by producing a vector-based output $\mathbf{y}$ as shown in Figure 2.8c.

2.2.3.2 Convolutional neural networks

Convolution Neural Networks (CNN) (Goodfellow et al., 2016) are designed to detect patterns in highly-dimensional unstructured data such as images, video, and speech. The basic idea is based on the observation that the same processing block can be applied to different parts of the input signal. With this approach, fewer trainable parameters are needed and the network is less likely to overfit. Compared to CNN, the MLP network requires orders of magnitude more network parameters because of having to map between all elements of the input and output layers.

Let $\mathbf{x}$ be $n \times m$ dimensional input tensor and $\mathbf{z}$ be $n \times m$ dimensional output tensor. Let i and j be the indices of a single cell in the tensor, e.g. z_{01}, where $i = 0$ and $j = 1$, corresponds to the second element in the first row of the tensor $\mathbf{z}$ as shown in Figure 2.9. The value of a single z_{ij} element is calculated by multiplying (element-wise) the

kernel tensor $\mathbf{k}$, for brevity called 'kernel', by the corresponding region $\mathbf{x_k}$ of the input tensor $\mathbf{x}$. The result of the element-wise multiplication is passed through the max function, producing a single value z_{ij}. The complete operation to compute z_{ij} is defined as follows:

$$z_{ij} = \max\left(\mathbf{x_k} \odot \mathbf{k}\right) \tag{2.33}$$

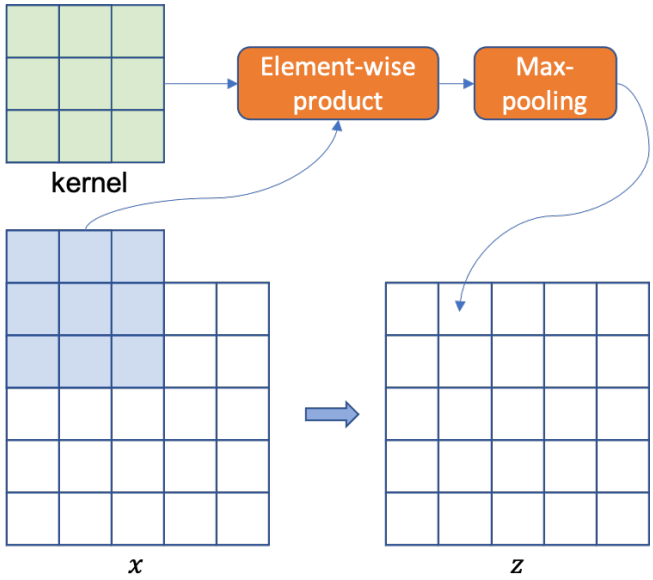


FIGURE 2.9: An operation in a convolutional neural block that maps between a single z_{ij} value in the $\mathbf{z}$ output tensor (layer) and the $\mathbf{x}$ input layer. A 3x3 convolutional kernel (filter) is multiplied element-wise by the corresponding region of the $\mathbf{x}$ input layer, followed by the max-pooling operation.

In a generic case, multiple kernels can be applied to the input tensor $\mathbf{x}$, which results in the output tensor $\mathbf{z}$ of shape $n \times m \times l$, where l is the number of kernels. Multiple convolutional blocks can be stacked on top of each other to extract features at different levels of abstraction. The dimensionality of the input $\mathbf{x}$ and output $\mathbf{z}$ kernels do not need to match, and the max function can be replaced with other options such as the *average* function.

2.2.3.3 Recurrent neural networks

Recurrent Neural Networks (RNNs) (Goodfellow et al., 2016) are suitable for modeling sequential data, such as a speech signal, where future values depend on past values. RNNs compute and maintain the z_i latent state by sequentially processing the x_i elements of the $\mathbf{x}$ input sequence to generate the $\mathbf{y}$ output sequence. RNNs can be used to process a signal known in advance to the model, such as recorded speech, as shown in Figure 2.10a. Alternatively, RNNs can generate new sequential data, such as a speech signal. In this scenario, the value of x_i input depends on the value of the y_{i-1} output that was generated previously.

Gated Recurrent Unit (GRU) and Long Short-Term Memory (LSTM) are the most popular variants of the blocks that compute the z_i latent space (Goodfellow et al., 2016). In a nutshell, the GRU and LSTM blocks track the latent state z_i based on previously processed inputs x_{i-1} and z_{i-1}, and they can update the z_i with new information or forget its state.

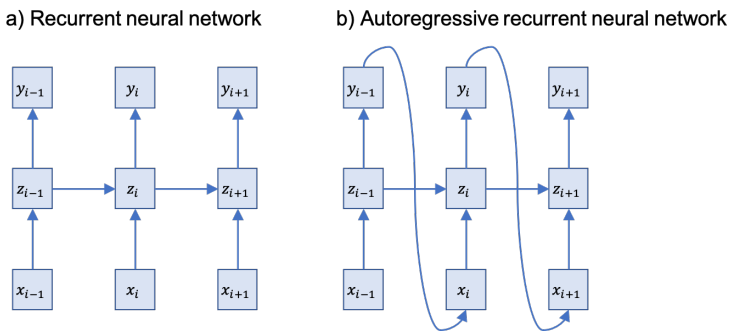


FIGURE 2.10: Recurrent neural network architectures. a) Recurrent network without autoregressive loop. All x_i inputs must be available in advance to the model. b) Autoregressive recurrent neural network. Only the first x_0 element must be available to the model. In general, the x_i element is computed based on the value of the previous output y_{i-1}.

2.2.3.4 Attention

The attention mechanism (Vaswani, Shazeer, Parmar, Uszkoreit, Jones, Gomez, Ł. Kaiser, et al., 2017) maps the **x** input sequence to the **y** output sequence. Each y_i element in the output sequence is computed from all elements of the input sequence, with the attention mechanism, telling which elements of the input sequence should be used when computing the output value. In other words, to which elements of the input sequence the y_i element should attend to. Hence, the name of this mechanism is attention.

The attention mechanism has three inputs: query **Q**, values **V**, and keys **K**, as illustrated in Figure 2.11. The values **V** represent the **x** input sequence. The query **Q** corresponds to the element y_i in the output sequence **y**. The keys **K** are derived from the **x** input sequence, which tells how much each x_i element should be included in the computation of y_i. The softmax function of the dot-product of the query **Q** and the keys **K** results in the vector of attention weights (probabilities). The dot-product between the attention weights and the values **V** returns the y_i output. The attention equation is defined by:

$$Attention(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = softmax(\frac{\mathbf{Q}\mathbf{K}^t}{\sqrt{d_k}})\mathbf{V} \tag{2.34}$$

In Eq. 2.34, the dot-product between the query **Q** and the keys **K** is used to calculate attention weights, but there are other options available. Almost any type of neural

network can be used to compute attention weights. Chaudhari et al. present a comprehensive review of various attention mechanisms (Chaudhari et al., 2021). The attention mechanism is suitable for tracking very long dependencies because it can attend to all elements in the input data.

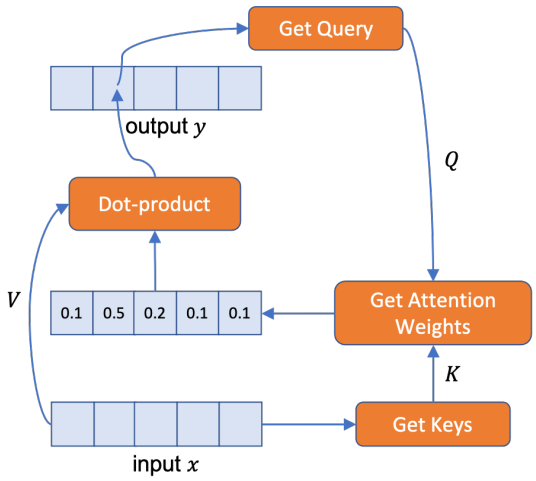


FIGURE 2.11: The attention mechanism illustrated by the example of computing a single element of the output sequence **y** from the input sequence **x**. Q - query, K - keys, V - values.

2.2.4 Deep learning – probabilistic perspective

Understanding the probability theory and the Bayesian rule concept is essential in getting to the origins of various neural network architectures. Many neural networks and other machine learning models have probabilistic counterparts. Linear regression, one of the simplest regression models, can be implemented as a probabilistic model known as Bayesian linear regression. Linear regression can be generalized as the Gaussian Process, and the Gaussian Process can be implemented as a neural network with one hidden layer with an infinite number of layers. Dropout and L2 regularization in neural networks are related to the concept of a prior variable in Bayesian networks. There are endless examples of machine learning models with neural networks and probabilistic counterparts, many of which are presented in two excellent books on probabilistic machine learning by Christopher Bishop (Bishop, 2006) and Kevin Murphy (Murphy, 2012).

To illustrate the relationship between the probability theory and neural networks, this section explains how the Variational Auto-Encoder (VAE) neural network can be derived with the use of the probability theory. VAE is an auto-encoder neural network that maps from the **x** input to the **x** output via the $\tilde{\mathbf{z}}$ bottleneck layer, as shown in Figure 2.12a.

During training, the sum of the two loses is minimized:

$$logp(\mathbf{x}) = logp(\mathbf{x}|\mathbf{z}) + D_{KL}(p(\mathbf{z}|\mathbf{x})||p(\mathbf{z})) \tag{2.35}$$

where $\tilde{\mathbf{z}} = p(\mathbf{z}|\mathbf{x})$ is the posterior probability of the variable $\mathbf{z}$. The first term is the reconstruction loss that minimizes the distance between the $\mathbf{x}$ input and the $\mathbf{x}$ output variables. The second term is the Kullback–Leibler Divergence (KLD) distance between the $\tilde{\mathbf{z}}$ bottleneck layer and the $\mathbf{z}$ Gaussian prior variable.

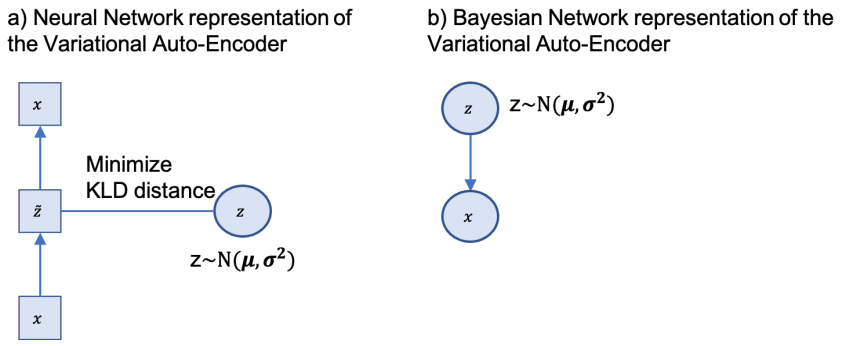


FIGURE 2.12: Architecture of the Variational Auto-Encoder (VAE) model. a) Neural network representation of the VAE model, b) Bayesian network representation of the VAE model.

At first sight, the motivation for adding the KLD loss is difficult to explain, but it becomes more apparent when we consider the probabilistic variant of the model. Consider a Bayesian network shown in Figure 2.12b with two variables $\mathbf{x}$ and $\mathbf{z}$. This network takes into account the prior belief that the observed $\mathbf{x}$ variable depends on the variable $\mathbf{z}$ that is unobserved (latent). To train this model, the latent variable has to be integrated out:

$$logp(\mathbf{x}) = log \int p(\mathbf{x}, \mathbf{z}) d\mathbf{z} \tag{2.36}$$

The integral in Eq. 2.36 can be approximated using the framework of variational inference (Jordan et al., 1999) as shown in Eq. 2.44:

$$\begin{aligned}
logp(\mathbf{x}) &= log \int p(\mathbf{x}, \mathbf{z}) d\mathbf{z} && (2.37)\\
&= log \int p(\mathbf{z}|\mathbf{x}) \frac{p(\mathbf{x}, \mathbf{z})}{p(\mathbf{z}|\mathbf{x})} d\mathbf{z} && (2.38)\\
&\geq E_{p(z|x)}[log \frac{p(\mathbf{x}, \mathbf{z})}{p(\mathbf{z}|\mathbf{x})}] && (2.39)\\
&= E_{p(z|x)}[log \frac{p(\mathbf{x}|\mathbf{z})p(\mathbf{z})}{p(\mathbf{z}|\mathbf{x})}] && (2.40)\\
&= E_{p(z|x)}[logp(\mathbf{x}|\mathbf{z}) + log \frac{p(\mathbf{z})}{p(\mathbf{z}|\mathbf{x})}] && (2.41)\\
&= E_{p(z|x)}[logp(\mathbf{x}|\mathbf{z})] + E_{p(z|x)}[log \frac{p(\mathbf{z})}{p(\mathbf{z}|\mathbf{x})}] && (2.42)\\
&= E_{p(z|x)}[logp(\mathbf{x}|\mathbf{z})] + D_{KL}(p(\mathbf{z}|\mathbf{x})||p(\mathbf{z})) && (2.43)\\
& && (2.44)
\end{aligned}$$

The final derivation is as follows:

$$logp(\mathbf{x}) \geq E_{p(z|x)}[logp(\mathbf{x}|\mathbf{z})] + D_{KL}(p(\mathbf{z}|\mathbf{x})||p(\mathbf{z})) \tag{2.45}$$

The first term is the VAE neural network reconstruction loss described earlier in Eq. 2.35, while the second term is the KLD loss. Both VAE representations based on neural networks and Bayesian networks are equivalent. The Bayesian representation made it possible to derive the VAE neural network architecture using an elegant mathematical framework of the probability theory. A similar approach can be used to derive other neural network architectures.

2.3 Performance metrics

In machine learning, performance metrics are used to evaluate different models to select the one that performs the best in the real-world scenario (Hossin et al., 2015; Botchkarev, 2018). Generally, a performance metric is defined by a function that takes two arguments: the ground-truth value for a target variable and the estimated (predicted) value from a machine learning model. The metric function usually outputs a real-value number that indicates the overall performance of the model averaged out over all examples in the test data.

As an intuitive example, let us consider a binary classification problem of classifying images into two classes, e.g., apples and oranges. One possible performance metric is 'accuracy', defined as the ratio of correctly classified images. However, there are other possible options such as precision, recall, AUC, log-likelihood (Hossin et al., 2015; Sofaer et al., 2019). The choice depends on the machine learning task.

In this section, a review of performance metrics used in the Ph.D. thesis is given, and the choices compared to other possible options are justified. This discussion is divided into two parts dedicated to different types of machine learning problems that require different types of metrics:

- Detection of pronunciation errors (mispronounced phones and incorrect lexical stress errors) - this is a classification machine learning problem in which the task is to estimate the probability of a speech error at the word or the syllable level.

- Generation of synthetic pronunciation errors in non-native speech and reconstruction of dysarthric speech - this is a regression problem with a goal of generating speech of desired characteristics such as including mispronunciations (non-native speech) or improving the intelligibility of speech (dysarthric speech).

2.3.1 Metrics for the detection of pronunciation errors

Performance metrics for detecting pronunciation errors are designed to ensure the optimal user experience of using a CAPT tool. Foremost, the tool should correctly identify mispronunciations. A user might get demotivated and eventually abandon using CAPT if the tool often provides incorrect feedback. Second, even if the tool is always correct while providing feedback, it should not miss too many mispronunciations made by the user. Otherwise, the user will be consolidating bad pronunciation habits and language learning will be less efficient. To summarize, a good CAPT tool should aim to: 1) not provide incorrect feedback, 2) not miss mispronunciations.

2.3.1.1 Key metrics

There are three key metrics to address the user experience requirements: precision, recall, and Area Under the Curve (AUC) (Hossin et al., 2015; Sofaer et al., 2019).

The precision metric reflects the requirement 'do not provide incorrect feedback'. It is defined as the proportion of raised mispronunciations that are identified correctly:

$$precision = \frac{TP}{TP + FP} \tag{2.46}$$

where TP (true positives) is the number of correctly detected mispronunciations and FP (false positives) is the number of incorrectly detected mispronunciations.

The recall metric addresses the requirement 'do not miss mispronunciations', and it is defined as the proportion of all mispronunciations that are identified correctly:

$$recall = \frac{TP}{TP + FN} \tag{2.47}$$

where FN (false negatives) is the number of missed mispronunciations.

In addition to the statistics TP, FP, and FN, there is also the TN (true negatives) quantity, which is the number of correctly identified good pronunciations. All four statistics, when summed up, give the total number of speech segments, e.g., words, for which the pronunciation error detection model is evaluated for. They serve as basic information for other more high-level metrics such as precision, recall, and AUC.

To compute the statistics TP, FP, TN, and FN, the test data with spoken sentences are first annotated to provide ground-truth information. Human listeners skilled in English listen to spoken sentences and label speech segments, e.g., words, with a binary label $e_g \in \{0, 1\}$, where the value of 1 means that the speech segment is mispronounced. The ground-truth label e_g is compared with the corresponding output of the pronunciation error detection model $\tilde{e} \in \{0, 1\}$. There are four possible combinations of each pair $\{e_g, \tilde{e}\}$, contributing to one of the statistics TP, FP, TN, and FN. For example, $\{e_g = 0, \tilde{e} = 1\}$ adds to the total number of FP.

Instead of directly producing a binary label $\tilde{e} \in \{0, 1\}$, the pronunciation error detection models proposed in the Ph.D. thesis estimate the probability of mispronunciation denoted as e. The variable e is modeled as a conditional Bernoulli distribution $e \sim p(e|speech + context)$, conditioned on the speech signal and additional context such as pronunciation of a native speaker. However, to compute the statistics TP, FP, TN, and FN, a binary output from the model is needed. To convert the probability of mispronunciation to a binary output, a threshold t is used as follows:

$$\tilde{e} = \begin{cases} 1 & \text{if } p(e) > t \\ 0 & \text{otherwise} \end{cases} \tag{2.48}$$

Changing the threshold t value allows for different trade-offs between precision and recall metrics. Increasing t, increases precision and decreases recall. Decreasing t, has the opposite effect. However, this controllability makes it difficult to estimate precision and recall metrics because it is unclear which threshold t value should be used. AUC metric overcomes the need for selecting the value of threshold t (Sofaer et al., 2019). Intuitively, AUC summarizes precision and recall metrics across all possible thresholds, producing a single score between 0 and 1. The value of 0 indicates that pronunciation errors are always detected incorrectly, and the value of 1 means the opposite. The value of 0.5 represents a model that detects pronunciation errors by random, assuming 50% of all speech segments are mispronounced. The AUC metric is defined as follows:

$$AUC = \int_0^1 precision(recall^{-1}(x))dx \tag{2.49}$$

where $recall^{-1}(x)$ returns the threshold t value for the recall value x. This function is the inverse of $x = recall(t)$ that returns the recall value for a given threshold. Graphically, the AUC metric can be visualized as the area under the curve on a precision-recall plot, with precision placed on the y-axis and recall on the x-axis. Precision-recall plots provide an intuitive view of how precision and recall change across different values of threshold t. For illustration, the examples of precision-recall plots with the corresponding AUC values are presented in Section 3.2.3.2.

To summarize, there are three key metrics used for the evaluation of pronunciation error detection: precision, recall, and AUC. Precision and recall reflect the two user experience requirements: 'do not provide incorrect feedback' and 'do not miss mispronunciations', respectively. The AUC metric provides a single-number performance metric, accounting for all possible trade-offs between precision and recall.

2.3.1.2 Discussion

The metrics of our choice, precision and recall, are already used in the field of pronunciation error detection (Leung et al., 2019; Z. Zhang et al., 2021; Yan and B.

Chen, 2021). They are especially useful when the data are imbalanced, with fewer positive (incorrect pronunciation) than negative (correct pronunciation) examples. Precision and recall do not depend on the statistic TN (the number of correctly identified good pronunciations), and therefore, they are unlikely to underestimate the negative impact of either missing mispronunciation or raising a false alarm.

FPR (False Positive Rate), also known as False Rejection Rate (FRR), is another popular metric (K. Li, Qian, and Meng, 2016; Leung et al., 2019; Z. Zhang et al., 2021). FPR is the ratio of good pronunciations that were incorrectly raised as mispronunciations, and in such a sense, it is similar to precision.

$$FPR = \frac{FP}{FP + TN} \tag{2.50}$$

However, contrary to precision, FPR may underestimate the negative effect of raising false pronunciation alarms. In the denominator of the FPR formula, there is the number of correctly identified good pronunciations (TN), which may outweigh the number of incorrectly raised mispronunciations (FP).

The recall metric is closely related to the False Negative Rate (FNR), also known as the False Acceptance Rate (FAR) (K. Li, Qian, and Meng, 2016; Leung et al., 2019; Z. Zhang et al., 2021). FNR is defined as the ratio of all mispronunciations that are identified as good pronunciations. There is no difference between using both metrics, except that recall should be maximized, and FNR minimized.

$$FNR = \frac{FN}{FN + TP} = 1 - recall \tag{2.51}$$

It is somewhat difficult to compare the different pronunciation error detection models using precision and recall metrics. One model may have higher precision, whereas the other model may be better in recall. AUC metric mitigates this problem by providing a single score based on precision and recall values (Eq. 2.49). F1-score is another single-score metric based on precision and recall, and it is widely used in other works on pronunciation error detection (Leung et al., 2019; Z. Zhang et al., 2021; Yan and B. Chen, 2021):

$$f_1 = 2 \cdot \frac{precision \cdot recall}{precision + recall} \tag{2.52}$$

Contrary to AUC, F1-score depends on precision and recall values computed for a specific value of threshold t (Eq. 2.48). This threshold is applied to the probability of mispronunciation used to compute the precision and recall values. Different pronunciation error detection models might perform differently for the same threshold, and it is hard to decide on its value in order to compare different models. AUC metric averages out over all possible values of threshold t, making it easier for model comparison.

In two works, the accuracy metric is used (Leung et al., 2019; Z. Zhang et al., 2021),

defined as the ratio of correctly classified speech segments, either as mispronunciations or good pronunciations:

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.53}$$

However, this metric is not used in the Ph.D. thesis because it does not work well with imbalanced data. For example, for the data set with 10% of mispronunciations, the model that never raises any mispronunciations would have an accuracy of 90%, which does not sound correct. On the other hand, both precision and recall values would equal 0, correctly indicating poor model performance.

Many discussed metrics have multiple names, making it harder to review and compare different models in the field. A good example is the recall metric, also known as True Positive Rate (TPR), Sensitivity, and Hit rate. In the Ph.D. thesis, the naming convention from the machine learning field is used with names, such as precision, recall, TPR, FPR and FNR.

2.3.2 Metrics for the generation of speech

There are two types of machine learning models for speech generation discussed in the Ph.D. thesis. First, the generation of synthetic pronunciation errors helps to improve the accuracy of detecting pronunciation errors in non-native speech. Thanks to improved accuracy, a person learning a foreign language receives a better user experience of using a CAPT tool. The second machine learning model performs the reconstruction of dysarthric speech that helps people with dysarthria disorder to better communicate with other people.

Both models are different in the way they influence the user experience. Synthetic pronunciation errors generated by the first model are not visible to language learners; they are used only to increase the size of the training data, improving accuracy of machine learning models. This is an example of an indirect impact on the user experience. Besides, speech reconstruction performed by the second model directly influences the user experience. Poor reconstruction may negatively influence the intelligibility and fluency of speech perceived by humans. The second model impacts the user experience directly. The difference between the direct and indirect impact on the user experience suggests that dedicated approaches to performance metrics should be used.

2.3.2.1 Metrics for the generation of synthetic pronunciation errors

Synthetic mispronounced speech is added to the training data to improve accuracy of pronunciation error detection. Intuitively, to help achieve better accuracy, synthetic speech should simulate as closely as possible real speech of non-native speakers. This intuition suggests that a good performance metric should reflect relevant aspects of a synthetic speech signal, such as the signal quality and the similarity to the

mispronounced speech of human speakers. However, what really matters to CAPT users are not the characteristics of a synthetic speech signal, but whether using synthetic pronunciation errors improves the accuracy of pronunciation error detection. Therefore, to measure the benefits of using synthetic pronunciation errors, the same performance metrics as for the detection of pronunciation errors are used (see Section 2.3.1).

To measure the effect of adding synthetic speech errors to the training data, two models for the detection of pronunciation errors are evaluated and compared with each other. For the first model, synthetic speech errors are added to the training data, whereas for the second model, they are not. Precision, recall, and AUC metrics are computed for both models, and their deltas are analyzed. Such investigation in which one aspect of the model is removed to understand its contribution to the overall model performance is known as an ablation study (Meyes et al., 2019).

2.3.2.2 Metrics for speech reconstruction

The goal of speech reconstruction is to make it easier for people with speech disorders to communicate with other people. Performance metrics should reflect human opinions about reconstructed speech. In a perceptual speech test, human listeners listen to multiple samples of speech and answer various questions, for example, 'please rate the naturalness of speech on the scale from 0 (the least natural) to 100 (the most natural)'. Ratings obtained from multiple listeners are aggregated into performance metrics, such as Mean Opinion Score (MOS) and MUltiple Stimuli with Hidden Reference and Anchor (MUSHRA), reflecting human opinions on certain aspect of speech (Merritt, Putrycz, et al., 2018; Wagner et al., 2019). By varying questions asked to listeners, multiple characteristics of speech may be assessed, such as naturalness, fluency, intelligibility, and similarity to other speech. Perceptual speech tests performed by human listeners are also known as subjective evaluation tests, because they reflect personal human opinions.

Human perceptual tests are laborious. They usually engage between 20 and 50 human listeners who have to listen to each audio sample and score it carefully. Automated perceptual evaluation tests are designed to simulate human perception and complement human-based evaluation (Valizada et al., 2021; Wagner et al., 2019). Some automated tests attempt to mimic directly human listeners, such as AutoMOS (Patton et al., 2016) that estimates the naturalness of speech on a scale from 1 (the most natural) to 5 (the least natural). In comparison, other automated models produce less interpretable metrics, such as the distance between generated and reference speech samples. Mel Cepstral Distortion (MCD) is an example of such distance based metrics (R. Skerry-Ryan et al., 2018; Valizada et al., 2021). The AutoMOS model does not require providing a reference audio signal, whereas, in MCD, this signal is required. Reference-free methods are more flexible, as they can be used to assess any generated speech sample, even if the reference signal is not available. While working on new machine learning models for speech generation, multiple evaluations have to be

conducted to assess the progress of work. Automated perceptual speech tests are often used in this research phase. Final evaluations of the speech generation models are usually conducted by human listeners.

In this Ph.D. thesis, MUSHRA is used as the primary metric to assess the performance of speech reconstruction. MUSHRA has been initially designed to evaluate the quality of audio coders in telecommunications (Series, 2014), but in recent years it has been successfully adopted in the field of speech synthesis (Rosenberg et al., 2017; Merritt, Putrycz, et al., 2018; Wagner et al., 2019; Mu et al., 2021), and music (Hines et al., 2015). In the MUSHRA test, listeners evaluate multiple systems, for example, different machine learning models for speech reconstruction. Various aspects of speech may be evaluated, such as signal quality, naturalness, and intelligibility. The goal of the test depends only on how the question is formulated, for example, 'please rate the naturalness of speech'. A listener is presented with audio samples, one sample for each system, and rates them on a scale from 0 (the lowest performance) to 100 (the highest performance). There are multiple rounds (screens) in which a listener listens to audio samples and scores them. Collected scores are aggregated across listeners into multiple statistics such as the mean, median, and rank values, and then statistical tests such as p-value and t-test are conducted to conclude the final outcome of the MUSHRA test.

Original MUSHRA specification created by International Telecomm. Union – Radio communication Sector (ITU-R) makes a few additional recommendations for the MUSHRA test construction (Series, 2014). On each MUSHRA screen, listeners are asked to rank one system with a score of 100 (upper anchor) and one system with a score of 0 (lower anchor). These anchors help calibrate the evaluated system on the 0-100 scale. In the field of speech synthesis, sometimes, only the upper anchor is employed, and the user is not forced to score one system as 100 (Merritt, Putrycz, et al., 2018).

Merrit et al. suggest using 50 listeners and assigning 40 screens to each listener to achieve repeatable and statistically significant results (Merritt, Putrycz, et al., 2018). However, measuring statistical significance in perceptual tests is a complex problem. In MUSHRA, standard p-value-based statistical tests are common. These tests cannot be reliably used because they rely on the assumption that listener responses are independently and identically distributed (iid), but this is not guaranteed. For example, one tester can strongly prefer audio samples generated by one system, whereas the second listener can have a strong preference for the second system. In this case, all scores within a listener will be correlated more than the scores between different listeners (Bishop, 2006). Effectively, in such situations, p-value-based tests provide an over-optimistic estimate of statistical significance. Due to violating the iid assumption, selecting the number of listeners, the number of unique texts for which audio samples are generated, and the number of screens per tester is often a trial and error process.

MOS is another popular metric for synthetic speech evaluation (Rosenberg et al.,

2017; Y. Wang, R. Skerry-Ryan, et al., 2017). Listeners listen to audio samples for multiple systems one sample at a time and rate them on a scale between 1 and 5, sometimes between 1 and 7. The average scores are computed for all systems and compared against each other. The statistical significance of the results is computed with the paired t-test. Calculations of the mean and p-value statistics used in the t-test assume that the input data are normally distributed; however, the MOS scale is ordinal, which violates this assumption (Rosenberg et al., 2017). On the other hand, the MUSHRA scale is more granular (0-100), making the scale closer to the continuous nature of the normal distribution. It must be noted that there exist statistical tests and statistics that do not require the data to be normally distributed, such as median (Y.-G. Lee et al., 2008) and Wilcoxon signed-rank test (Woolson, 2007). Another difference between both tests is that in MUSHRA, a listener is presented with audio samples for all systems at once and then rates them, whereas in MOS, a listener listens to audio samples and rates them one at a time. Thanks to presenting multiple systems at once, listeners can calibrate between different systems before providing their scores. Therefore, MUSHRA obtains statistically significant results faster than MOS (Wagner et al., 2019).

A preference test (Mu et al., 2021; Gabryś et al., 2021) is similar to a MUSHRA test. A listener listens to multiple systems in parallel and then rates them. One difference is that only two systems are evaluated in the preference test. Second, contrary to the fine-grained 0-100 scale in MUSHRA, a listener selects from the limited set of choices: system A is better, system B is better, and both systems are the same. Sometimes, the scale is extended with two additional options: system A or B is significantly better. The preference test is also known as the AB test. There exists a variant of the AB test called the ABX test (Mu et al., 2021). A listener is presented with the reference audio signal X and has to decide which of the A and B systems is closer to the reference signal. Statistical significance of AB tests is conducted with the Binomial test (Abdi, 2007). This test provides the p-value score that gives the probability that systems A and B are the same based on provided preference scores. If the p-value is low, e.g., <0.01, then it means that one of the systems has been scored higher by listeners; otherwise, it is assumed that the difference between the two systems is due to random sampling. The Bernoulli test assumes that all individual scores provided by listeners are iid. Because this assumption does not always hold, the p-value tends to be over-estimated (lower than it should be).

In conclusion, the MUSHRA test is used in the doctoral dissertation for the evaluation of speech reconstruction. MUSHRA enables listeners to listen to speech samples from multiple systems simultaneously and score them on a continuous scale from 0 to 100, providing more precise results on the quality of the speech being assessed.

Chapter 3

Pronunciation error detection

3.1 Introduction

This chapter constitutes the main scientific part of the doctoral dissertation. The aim is to explore the key research thesis to create new deep learning models for pronunciation error detection:

It is possible to improve the accuracy of deep learning methods for detecting pronunciation errors in non-native English by employing synthetic speech generation and end-to-end modeling techniques that reduce the need for phonetically transcribed mispronounced speech.

The results of this research have been published in scientific publications at major international speech conferences and scientific journals (Korzekwa, Lorenzo-Trueba, Drugman, and Kostek, 2022; Korzekwa, Lorenzo-Trueba, Drugman, Calamaro, et al., 2021; Korzekwa, Lorenzo-Trueba, Zaporowski, et al., 2021; Korzekwa, Barra-Chicote, Zaporowski, et al., 2021; Korzekwa and Kostek, 2019). These publications address the following challenges of the existing methods for detecting pronunciation errors in non-native speech, with respect to the research thesis. The research background on these challenges was presented in Section 1.3.

1. Transcription of non-native speech is a difficult and costly process

   Section 3.2 describes a new approach to pronunciation error detection that does not require phonetic transcriptions of non-native speech.

2. Aligning canonical and recognized phonemes accurately is challenging

   Section 3.2 describes an end-2-end model for detecting pronunciation errors that does not need to align between canonical and recognized phonemes.

3. Not all pronunciation errors are the same

   Section 3.2 describes a new approach to categorizing pronunciation errors by severity to further improve the accuracy of detecting pronunciation errors.

4. A sentence can be pronounced correctly in multiple different ways

Section 3.3 describes a probabilistic model that reduces the number of false mispronunciation alarms by accounting for multiple correct pronunciations of the same sentence.

5. Practicing lexical stress is an important part of CAPT

   Section 3.4 describes a new method for the detection of lexical stress errors based on synthetically generated lexical errors and the attention mechanism.

6. The availability of non-native speech with pronunciation errors is limited

   Section 3.5 describes a new approach to pronunciation error detection that reformulates the problem of detecting pronunciation errors as a speech generation task.

7. Multi-task learning as an approach to tackling overfitting in deep learning

   Section 3.2 presents the model that includes a phoneme recognizer as a secondary task to regularize the primary task of computing the probability of a pronunciation error at the word level.

3.2 Weakly-supervised word-level pronunciation error detection in non-native English speech

Daniel Korzekwa, Jaime Lorenzo-Trueba, Thomas Drugman, Shira Calamaro, Bozena Kostek, Weakly-supervised word-level pronunciation error detection in non-native English speech, Interspeech, 2021

Abstract

We propose a weakly-supervised model for word-level mispronunciation detection in non-native (L2) English speech. To train this model, phonetically transcribed L2 speech is not required and we only need to mark mispronounced words. The lack of phonetic transcriptions for L2 speech means that the model has to learn only from a weak signal of word-level mispronunciations. Because of that and due to the limited amount of mispronounced L2 speech, the model is more likely to overfit. To limit this risk, we train it in a multi-task setup. In the first task, we estimate the probabilities of word-level mispronunciation. For the second task, we use a phoneme recognizer trained on phonetically transcribed L1 speech that is easily accessible and can be automatically annotated. Compared to state-of-the-art approaches, we improve the accuracy of detecting word-level pronunciation errors in AUC metric by 30% on the GUT Isle Corpus of L2 Polish speakers, and by 21.5% on the Isle Corpus of L2 German and Italian speakers.

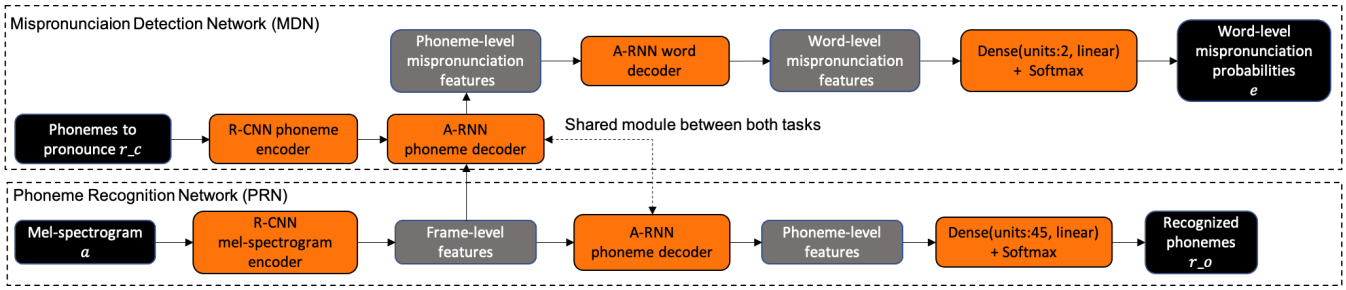


FIGURE 3.1: Neural network architecture of the WEAKLY-S model for word-level pronunciation error detection.

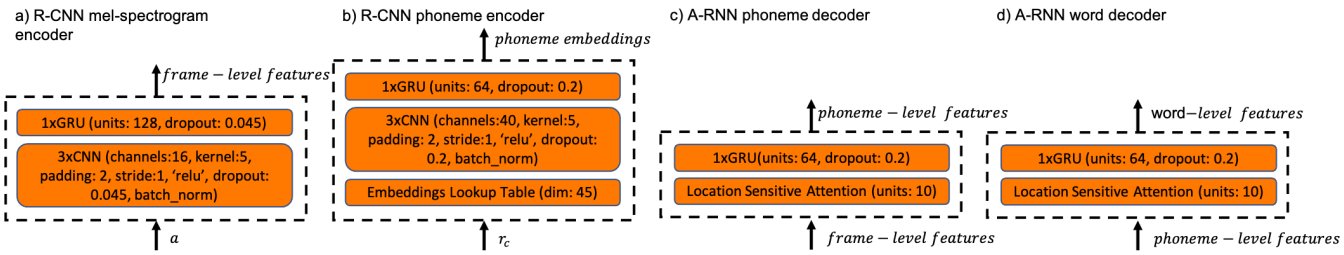


FIGURE 3.2: Details of the neural network architecture of the WEAKLY-S model for word-level pronunciation error detection.

3.2.1 Introduction

It has been shown that Computer-Assisted Pronunciation Training (CAPT) helps people practice and improve pronunciation skills (Neri et al., 2008; Tejedor-Garcıa et al., 2020). Despite significant progress over the last two decades, standard methods are still unable to detect mispronunciations with high accuracy. These methods can detect phoneme-level mispronunciations at about 60% precision and 40%-80% recall (Leung et al., 2019; Korzekwa, Lorenzo-Trueba, Zaporowski, et al., 2021; Z. Zhang et al., 2021). By further raising precision we can lower the risk of providing incorrect feedback, whereas with higher recall, we can detect more mispronunciation errors.

Standard methods aim at recognizing the phonemes pronounced by a speaker and compare them with expected (canonical) pronunciation of correctly pronounced speech. Any mismatch between recognized and canonical phonemes yields a pronunciation error at the phoneme level. Phoneme recognition-based approaches rely on phonetically transcribed speech labeled by human listeners. Human-based transcription is a laborious task, especially, in the case of L2 speech where listeners have to identify mispronunciations. Sometimes, it might be even impossible to transcribe L2 speech because different languages have different phoneme sets and it is unclear which phonemes were pronounced by the speaker.

Phoneme recognition-based approaches generally fall into two categories. The first category uses forced-alignment techniques (H. Li et al., 2011; K. Li, Qian, and Meng, 2016; Sudhakara, Ramanathi, Yarra, and Ghosh, 2019; Cheng et al., 2020) based on the work by Franco et al. (Franco et al., 1997) and the Goodness of Pronunciation

(GOP) method (Witt et al., 2000). The GOP uses Bayesian inference to find the most likely alignment between canonical phonemes and the corresponding audio signal (forced alignment). Then, the GOP uses the likelihoods of the aligned audio signal as an indicator for mispronounced phonemes. In the second category there are methods that recognize phonemes pronounced by a speaker purely from a speech signal, and only then align them with canonical phonemes (Minematsu, 2004; Harrison et al., 2009; A. Lee and Glass, 2013; Plantinga et al., 2019; Sudhakara, Ramanathi, Yarra, Das, et al., 2019). Techniques falling into both categories can be complemented with the use of a reference signal obtained either from a database of speech (Xiao et al., 2018; Nicolao, Beeston, et al., 2015; J. Wang et al., 2019) or generated from phonetic representation (Korzekwa, Lorenzo-Trueba, Zaporowski, et al., 2021; Qian et al., 2010).

There are two challenges for the phoneme recognition approaches. First, phonemes pronounced by a speaker have to be recognized accurately, which has been shown to be difficult (Z. Zhang et al., 2021; J. Chorowski, Bahdanau, et al., 2014; J. K. Chorowski et al., 2015; Bahdanau et al., 2016). Second, standard approaches expect only a single canonical pronunciation of a given text, but this assumption does not always hold true due to phonetic variability of speech. In (Korzekwa, Lorenzo-Trueba, Zaporowski, et al., 2021), we addressed these problems by modeling uncertainty in the model by incorporating a pronunciation model of L1 speech. Nonetheless, this approach still relies on phonetically transcribed L2 speech.

In this paper, we introduce a novel model (noted as WEAKLY-S) for the detection of word-level pronunciation errors that does not require phonetically transcribed L2 speech. The model produces the probabilities of mispronunciation for all words, conditioned on a spoken sentence and canonical phonemes. Mispronunciation error types include any of phoneme replacement, addition, deletion or unknown speech sound. During training, the model is weakly supervised, in the sense that we only mark mispronounced words in L2 speech and the data do not have to be phonetically transcribed. Due to the limited availability of L2 speech and the fact it is not phonetically transcribed, the model is more likely to overfit. To solve this problem, we train the model in a multi-task setup. In addition to a primary task of word-level mispronunciation detection, we use a phoneme recognizer trained on automatically transcribed L1 speech for the secondary task. Both tasks share common parts of the model, which makes the primary task less likely to overfit. Additionally, we address the overfitting problem with synthetically generated pronunciation errors that are derived from L1 speech.

Leung et al. (Leung et al., 2019) used a phoneme recognizer based on Connectionist Temporal Classification (CTC) for pronunciation error detection. Instead, we use an attention-based phoneme recognizer following Chorowski et al. (J. K. Chorowski et al., 2015) so that we can regularize the model by both tasks sharing a common component (attention). With a CTC-based phoneme recognizer it would not be possible because this technique does not use attention that could be shared

between both tasks. Zhang et al. (Z. Zhang et al., 2021) employed a multi-task model for pronunciation assessment, but with two important differences. First, they use a Needleman-Wunsch algorithm (Needleman et al., 1970) for aligning canonical and recognized sequences of phonemes, but this algorithm cannot be tuned towards sequences of phonemes. We use an attention mechanism that automatically maps the speech signal to the sequence of word-level pronunciation errors. Second, Zhang et al. detect pronunciation errors at the phoneme level and they expect L2 speech to be phonetically transcribed. This differs from our method of recognizing pronunciation errors at the word level with no need for phonetic transcriptions of L2 speech. To the best of our knowledge, this is the first approach to train word-level pronunciation error detection model that does not require phonetically transcribed L2 speech and can be optimized directly towards word-level mispronunciation detection.

3.2.2 Proposed model

3.2.2.1 Model definition

The model is made of two sub-networks: *i)* a word-level Mispronunciations Detection Network (MDN) detects word-level pronunciation errors $\mathbf{e}$ from the audio signal $\mathbf{a}$ and canonical phonemes $\mathbf{r_c}$, *ii)* a Phoneme Recognition Network (PRN) recognizes phonemes $\mathbf{r_o}$ pronounced by a speaker from the audio signal $\mathbf{a}$ (Fig. 3.1).

More formally, let us define the following variables: $\mathbf{a}$ - speech signal represented by a mel-spectrogram, $\mathbf{r_c}$ - canonical phonemes that the speaker was expected to pronounce, $\mathbf{r_o}$ - phonemes pronounced, and $\mathbf{e}$ - the probabilities of mispronouncing words in the spoken sentence. The model outputs the probabilities of word-level mispronunciation, denoted as $\mathbf{e} \sim p(\mathbf{e}|\mathbf{a}, \mathbf{r_c}, \boldsymbol{\theta})$, where $\boldsymbol{\theta}$ represent parameters of the model.

We train the WEAKLY-S model in a multi-task setup. In addition to the primary task $\mathbf{e}$, we use a phoneme recognizer denoted as $\mathbf{r_o} \sim p(\mathbf{r_o}|\mathbf{a}, \boldsymbol{\theta})$ for the secondary task. The parameters $\boldsymbol{\theta}$ are shared between both tasks, which makes the MDN less likely to overfit. We define the loss function as the sum of two losses: a word-level mispronunciation loss and a phoneme recognition loss. Its formulation for the *ith* training example is presented in Eq. 3.1. We train the model using two types of training data: phonetically transcribed L1 speech (both losses are used) and untranscribed L2 speech (only the mispronunciation loss is used). Having a separate loss for word-level mispronunciation lets us train the model from speech data that are not phonetically transcribed.

$$\mathcal{L}(\subseteq) = log(p(\mathbf{e}|\mathbf{a}, \mathbf{r_c}, \boldsymbol{\theta})) + log(p(\mathbf{r_o}|\mathbf{a}, \boldsymbol{\theta})) \tag{3.1}$$

3.2.2.2 Neural network details

Following Sutskever et al. (Sutskever et al., 2014), the MDN network encodes the mel-spectrogram $\mathbf{a}$ and the canonical phonemes $\mathbf{r_c}$ with Recurrent Convolutional Neural

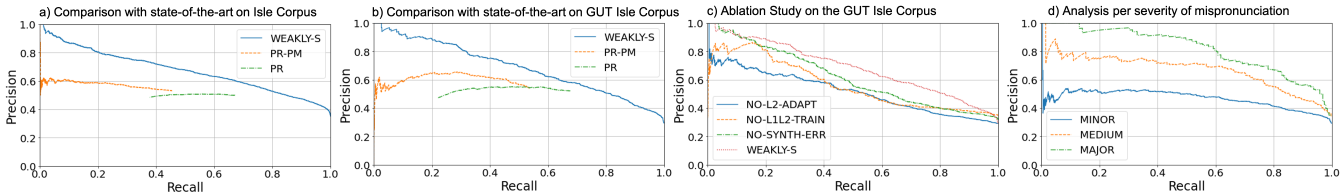


FIGURE 3.3: Precision-recall curves for the WEAKLY-S and baseline models, PR-PM and PR, (a) tested on Isle Corpus of German and Italian speakers and (b) GUT Isle Corpus of Polish speakers. (c) Ablation study on the GUT Isle corpus. (d) Analysis of mispronunciation severity levels.

Network (RCNN) encoders (Fig. 3.2a and Fig. 3.2b). These encoded representations are passed into an attention-based (Vaswani, Shazeer, Parmar, Uszkoreit, Jones, Gomez, Ł. Kaiser, et al., 2017) Recurrent Neural Network (A-RNN) decoder (Fig. 3.2c) that generates phoneme-level mispronunciation features. Phoneme-level features are transformed into word-level features (Fig. 3.2d) based on an attention mechanism and these finally are used for computing word-level mispronunciation probabilities $\mathbf{e}$.

The PRN recognizes phonemes $\mathbf{r_o}$ pronounced by the speaker. It is similar to the attention-based phoneme recognizer by Chorowski et al. (J. K. Chorowski et al., 2015). To generate phoneme-level features, it uses the same RCNN mel-spectrogram encoder and A-RNN decoder as the MDN. The only difference is that the A-RNN decoder is not conditioned on canonical phonemes. Phoneme-level features are transformed to the probabilities of pronounced phonemes. We added a phoneme recognition task due to the limited amount of L2 speech annotated with word-level mispronunciations. Without it, the MDN would be prone to overfitting if it was trained only on its own. By sharing common parts between both models, the PRN acts as a backbone for the MDN and makes it more robust.

The model was implemented in MxNet framework (T. e. a. Chen, 2015) and tuned for hyper-parameters with AutoGluon Bayesian optimization framework (Erickson et al., 2020). The model was first pretrained on L1 and L2 speech corpora and then the MDN part was fine-tuned only on L2 speech data. We used the Adam optimizer with learning rate 0.001 and gradient clipping 5. Training data were segmented into buckets with batch size 32, using GluonCV (Guo et al., 2020). The A-RNN phoneme and word decoders are based on Location Sensitive Attention by Chorowski et al. (J. K. Chorowski et al., 2015).

3.2.3 Experiments

We present three experiments. We start with comparing our model against state-of-the-art approaches in the task of word-level mispronunciation detection. In an ablation study we analyze which elements of the model contribute the most to its

performance. Finally, we analyze how the severity of pronunciation error affects the accuracy of the model.

3.2.3.1 Speech corpora and metrics

In our experiments, we use a combination of L1 and L2 English speech. L1 speech is obtained from TIMIT (Garofolo et al., 1993) and LibriTTS (Zen et al., 2019) corpora. L2 data come from the Isle (Atwell et al., 2003) corpus (German and Italian speakers) and the GUT Isle (Weber et al., 2020) corpus (Polish speakers). In total, we collected 102,812 utterances, summarized in Table 3.1. We split the data into training and test sets, holding out 28 L2 speakers (11 German, 11 Italian, and 6 Polish) only for testing the performance of the model.

The L2 corpus of Polish speakers was annotated for word-level pronunciation errors by 5 native English speakers. Annotators marked mispronounced words and indicated their severity levels using one of the three possible values: 1 - MINOR, 2 - MEDIUM, 3 - MAJOR. The Isle corpus of German and Italian speakers comes with phoneme level mispronunciations. Words with at least one mispronounced phoneme were automatically marked as mispronounced. The Isle corpus is not mapped to severity levels of mispronunciations. In total, there are 35,555 L2 words, including 8035 mispronounced words. All data were re-sampled to 16 kHz.

We extended the train set with 292,242 utterances of L1 speech with synthetically generated pronunciation errors. We use a simple approach of perturbing phonetic transcription for the corresponding speech audio. First, we sample these utterances with replacement from L1 corpora of human speech. Then, for each utterance, we replace phonemes with random phonemes with a probability of 0.2. In (Korzekwa, Barra-Chicote, Zaporowski, et al., 2021) we found that generating incorrectly stressed speech using Text-To-Speech (TTS) improves the accuracy of detecting lexical stress errors in L2 speech. Although, as opposed to using TTS, we create pronunciation errors by perturbing the text, we expect this simpler approach should still help recognizing word-level pronunciation errors.

TABLE 3.1: Summary of speech corpora used in experiments. * - audiobooks read by volunteers from all over the world (Zen et al., 2019)

Native Language	Hours	Speakers
English	90.47	640
Unknown*	19.91	285
German and Italian	13.41	46
Polish	1.49	12

To evaluate our model, we use three standard metrics: Area Under Curve (AUC), precision and recall. The AUC metric provides an overall performance of the model accounting for all possible trade offs between precision and recall. Precision-recall

plots illustrate relations between both metrics. Complementary, to analyze precision, in all our experiments we consistently fix recall at the value of 0.4 to be comparable with two baseline models that do not cover the whole range of recall values (see Section 3.2.3.2).

3.2.3.2 Comparison with state-of-the-art

We compare our proposed WEAKLY-S model against two state-of-the-art baselines. The phoneme recognizer (PR) model by Leung et al. (Leung et al., 2019) is our first baseline. The PR is based on CTC loss (Graves, 2012) and it outperforms multiple alternative approaches for pronunciation assessment. The original CTC-based model uses a hard likelihood threshold applied to recognized phonemes. To compare it with two other models, following our work in (Korzekwa, Lorenzo-Trueba, Zaporowski, et al., 2021), we replaced hard likelihood threshold with a soft threshold. The second baseline is the PR extended by a pronunciation model (PR-PM model (Korzekwa, Lorenzo-Trueba, Zaporowski, et al., 2021)). The pronunciation model accounts for phonetic variability of speech produced by native speakers, which results in higher precision of detecting pronunciation errors.

The results are presented in Fig. 3.3a, Fig. 3.3b and Table 3.2. The WEAKLY-S model turns out to outperform the second best model in AUC by 30% from 52.8 to 68.63 and in precision by 23% from 61.21 to 75.25 on the GUT Isle Corpus of Polish speakers. We observe similar improvements on the Isle Corpus of German and Italian speakers.

TABLE 3.2: Accuracy metrics of detecting word-level pronunciation errors. WEAKLY-S vs baseline models.

Model	**AUC [%]**	**Precision [%,95%CI]**	**Recall [%,95%CI]**
		Isle corpus (German and Italian)	
PR	55.52	49.39 (47.59-51.19)	40.20 (38.62-41.81)
PR-PM	48.00	54.20 (52.32-56.08)	40.20 (38.62-41.81)
WEAKLY-S	**67.47**	71.94 (69.96, 73.87)	40.14 (38.56, 41.75)
		GUT Isle corpus (Polish)	
PR	52.8	54.91 (50.53-59.24)	40.29 (36.66-44.02)
PR-PM	50.50	61.21 (56.63-65.65)	40.15 (36.51-43.87)
WEAKLY-S	**68.63**	75.25 (71.67-78.59)	40.38 (37.52-43.29)

One difference between our model and the two baselines is that they both use the Needleman-Wunsch algorithm (Needleman et al., 1970) for aligning canonical and recognized sequences of phonemes. This is a dynamic programming-based algorithm for comparing biological sequences and cannot be optimized for mispronunciation errors. Our model automatically finds the mapping between regions in the speech signal and the corresponding canonical phonemes, and then identifies word-level

mispronunciation errors. In this way, we eliminate the Needleman-Wunsch algorithm as a possible source of error.

The second difference is the use of phonetic transcriptions for L2 speech. Both baselines use automatic transcriptions provided by an Amazon-proprietary grapheme-to-phoneme model. In (Korzekwa, Lorenzo-Trueba, Zaporowski, et al., 2021) we found that for the PR and PR-PM models it is better to use automatically transcribed L2 speech for training a phoneme recognizer than not use L2 speech at all. Note that these automatic transcriptions will include phoneme mistakes for mispronounced speech. Our model does not use transcriptions of L2 speech, and instead it is guided by the word-level pronunciation errors of L2 speech in a weakly-supervised fashion.

3.2.3.3 Ablation study

We now investigate which elements of our new model contribute the most to its performance. Along with the WEAKLY-S model, we trained three additional variants, each with a certain feature removed. The NO-L2-ADAPT variant does not fine-tune the model on L2 speech, though it is still exposed to L2 speech while it is trained on a combined corpus of L1 and L2 speech. The NO-L1L2-TRAIN model is not trained on L1/L2 speech, and fine-tuning on L2 speech starts from scratch. It means that the model will not use a large amount of phonetically transcribed L1 speech data and ultimately the secondary task of the phoneme recognizer will not be used. In the NO-SYNTH-ERR model, we exclude synthetic samples of mispronounced L1 speech. It significantly reduces the amount of incorrectly pronounced words used during training from 1,129,839 to only 5,273 L2 words.

L2 Fine-tuning (NO-L2-ADAPT) is the most important factor that contributes to the performance of the model (Fig. 3.3c and Table 3.3), with an AUC of 51.72% compared to 68.63% for the full model. Training the model on both L2 and L1 speech together is not sufficient. We think it is because L2 speech accounts for less than 1% of the training data and the model naturally leans towards L1 speech. The second most important feature is training the model on a combined set of L1 and L2 speech (NO-L1L2-TRAIN), with AUC of 56.46%. L1 speech accounts for more than 99% of the training data. These data are also phonetically transcribed, and therefore can be used for the phoneme recognition task. The phoneme recognition task acts as a 'backbone' and reduces the effect of overfitting in the main task of detecting word pronunciation errors. Finally, excluding synthetically generated pronunciation errors (NO-SYNTH-ERR) reduces the AUC from 68.63% to 61.54%.

3.2.3.4 Severity of mispronunciation

When providing feedback to the L2 speaker about mispronounced words, we want to reflect the severity of mispronunciation, in order to focus on more severe errors and not report them all at once. We segment pronunciation errors into three categories:

TABLE 3.3: Ablation study for the GUT Isle corpus.

Model	AUC [%]	Precision [%]	Recall [%]
NO-L2-ADAPT	51.72	57.89	40.11
NO-L1L2-TRAIN	56.46	59.73	40.20
NO-SYNTH-ERR	61.54	67.22	40.38
WEAKLY-S	**68.63**	75.25	40.38

LOW, MEDIUM and HIGH, based on an inter-tester agreement of annotating sentences for word-level mispronunciations. Mispronounced words with less than 40% inter-tester agreement belong to the LOW category, between 40% and 80% to MIDDLE, and over 80% to HIGH. We validated that the proposed inter-tester agreement bands are well correlated with explicit listener opinions on the severity of mispronunciation, as shown in Table 3.4. This result shows that data on mispronunciation severity can be derived automatically, without the need to collect it.

TABLE 3.4: Severity of mispronunciation by inter-tester agreement for the GUT Isle Corpus. 1 - MINOR, 2 - MEDIUM, 3 - MAJOR.

Inter-tester agreement	Severity [mean and 95% CI]
LOW (Less than 40%)	1.32 (1.28-1.35)
MEDIUM (Between 40% and 80%)	1.58 (1.54-1.62)
HIGH(Higher than 80%)	2.08 (2.03-2.13)

We aim at detecting the words of HIGH inter-tester agreement with higher precision to provide more relevant feedback to L2 speakers. To make AUC, precision, and recall metrics comparable between different levels of inter-tester agreement, we enforce the ratio of mispronounced words across all categories to the same level of 29.2% by randomly down-sampling correctly pronounced words. This value is the proportion of mispronounced words across all inter-tester agreement levels in the GUT Isle Corpus. We observe that we can detect pronunciation errors of HIGH inter-tester agreement with 91.67% precision at 40.38% recall (Fig. 3.3d and Table 3.5). By segmenting pronunciation errors into three difference bands, we can report to a language learner only the errors of HIGH inter-tester agreement, and improve their learning experience.

3.2.4 Conclusions and future work

We proposed a model for detecting pronunciation errors in English that can be trained from L2 speech labeled only for word-level mispronunciations. The data do not have to be phonetically transcribed. The model outperforms state-of-the-art models in AUC metric on the GUT Isle Corpus of Polish speakers and the Isle Corpus of German and Italian speakers. The limited amount of L2 speech and the lack of phonetically

TABLE 3.5: Accuracy metrics for different severity levels of mispronunciation for the GUT Isle Corpus.

Inter-test agreement	AUC [%]	Precision [%]	Recall [%]
LOW	46.99	51.84	40.48
MEDIUM	66.90	71.89	40.80
HIGH	81.48	91.67	40.31

transcribed speech makes this model prone to overfitting. We overcame this issue by proposing a multi-task training with two tasks: a word-level pronunciation error detector trained on L1 and L2 speech, and a phoneme recognizer trained on L1 speech. The most important factors that contribute to the model accuracy are: *i)* fine-tuning on L2 speech, *ii)* pre-training on a joined corpus of L1 and L2 speech, and *iii)* use of synthetically generated pronunciation errors.

The level of inter-tester agreement in annotating pronunciation errors correlates with explicit human opinions about the severity of mispronunciation. By detecting pronunciation errors only for high inter-tester agreement, we may significantly lower the number of false positives reported to a language learner.

In the future, we will experiment with discrete representation of the latent phoneme space such as Vector-Quantized Variational-Auto-Encoder (VQ-VAE) (J. Chorowski, Weiss, et al., 2019; Van Den Oord et al., 2017), which should fit better to discrete nature of phonemes. We plan to generate synthetic mispronounced speech, which is motivated by our recent work on using speech synthesis for generating speech errors in the related task of lexical stress error detection (Korzekwa, Barra-Chicote, Zaporowski, et al., 2021).

3.3 The role of uncertainty modeling

Daniel Korzekwa, Jaime Lorenzo-Trueba, Szymon Zaporowski, Shira Calamaro, Thomas Drugman, Bozena Kostek, Mispronunciation Detection in Non-Native (L2) English with Uncertainty Modeling, IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2021

Abstract

A common approach to the automatic detection of mispronunciation in language learning is to recognize the phonemes produced by a student and compare it to the expected pronunciation of a native speaker. This approach makes two simplifying assumptions: a) phonemes can be recognized from speech with high accuracy, b) there is a single correct way for a sentence to be pronounced. These assumptions do not always hold, which can result in a significant amount of false mispronunciation alarms. We propose a novel approach to overcome this problem based on two principles: a) taking into account uncertainty in the automatic phoneme recognition

step, b) accounting for the fact that there may be multiple valid pronunciations. We evaluate the model on non-native (L2) English speech of German, Italian and Polish speakers, where it is shown to increase the precision of detecting mispronunciations by up to 18% (relative) compared to the common approach.

3.3.1 Introduction

In Computer Assisted Pronunciation Training (CAPT), students are presented with a text and asked to read it aloud. A computer informs students on mispronunciations in their speech, so that they can repeat it and improve. CAPT has been found to be an effective tool that helps non-native (L2) speakers of English to improve their pronunciation skills (Neri et al., 2008; Tejedor-Garcıa et al., 2020).

A common approach to CAPT is based on recognizing the phonemes produced by a student and comparing them with the expected (canonical) phonemes that a native speaker would pronounce (Witt et al., 2000; K. Li, Qian, and Meng, 2016; Sudhakara, Ramanathi, Yarra, and Ghosh, 2019; Leung et al., 2019). It makes two simplifying assumptions. First, it assumes that phonemes can be automatically recognized from speech with high accuracy. However, even in native (L1) speech, it is difficult to get the Phoneme Error Rate (PER) below 15% (J. K. Chorowski et al., 2015). Second, this approach assumes that this is the only 'correct' way for a sentence to be pronounced, but due to phonetic variability this is not always true. For example, the word 'enough' can be pronounced by native speakers in multiple correct ways: /ih n ah f/ or /ax n ah f/ (short 'i' or 'schwa' phoneme at the beginning). These assumptions do not always hold which can result in a significant amount of false mispronunciation alarms and making students confused when it happens.

We propose a novel approach that results in fewer false mispronunciation alarms, by formalizing the intuition that we will not be able to recognize exactly what a student has pronounced or say precisely how a native speaker would pronounce it. First, the model estimates a belief over the phonemes produced by the student, intuitively representing the uncertainty in the student's pronunciation. Then, the model converts this belief into the probabilities that a native speaker would pronounce it, accounting for phonetic variability. Finally, the model makes a decision on which words were mispronounced in the sentence by processing three pieces of information: a) what the student pronounced, b) how likely a native speaker would pronounce it that way, and c) what the student was expected to pronounce.

In Section 3.3.2, we review the related work. In Section 3.3.3, we describe the proposed model. In Section 3.3.4, we present the experiments, and we conclude in Section 3.3.5.

3.3.2 Related work

In 2000, Witt et al. coined the term Goodness of Pronunciation (GoP) (Witt et al., 2000). GoP starts by aligning the canonical phonemes with the speech signal using

a forced-alignment technique. This technique aims to find the most likely mapping between phonemes and the regions of a corresponding speech signal. In the next step, GoP computes the ratio between the likelihoods of the canonical and the most likely pronounced phonemes. Finally, it detects a mispronunciation if the ratio falls below a given threshold. GoP was further extended with Deep Neural Networks (DNNs), replacing Hidden Markov Model (HMM) and Gaussian Mixture Model (GMM) techniques for acoustic modeling (K. Li, Qian, and Meng, 2016; Sudhakara, Ramanathi, Yarra, and Ghosh, 2019). Cheng et al. (Cheng et al., 2020) improved the performance of GoP with the latent representation of speech extracted in an unsupervised way.

As opposed to GoP, we do not use forced-alignment that requires both speech and phoneme inputs. Following the work of Leung et al. (Leung et al., 2019), we use a phoneme recognizer, which recognizes phonemes from only the speech signal. The phoneme recognizer is based on a Convolutional Neural Network (CNN), a Gated Recurrent Unit (GRU), and Connectionist Temporal Classification (CTC) loss. Leung et al. report that it outperforms other forced-alignment (K. Li, Qian, and Meng, 2016) and forced-alignment-free (Harrison et al., 2009) techniques on the task of detecting phoneme-level mispronunciations in L2 English. Contrary to Leung et al., who rely only on a single recognized sequence of phonemes, we obtain top N decoded sequences of phonemes, along with the phoneme-level posterior probabilities.

It is common in pronunciation assessment to employ the speech signal of a reference speaker. Xiao et al. use a pair of speech signals from a student and a native speaker to classify native and non-native speech (Xiao et al., 2018). Mauro et al. incorporate the speech of a reference speaker to detect mispronunciations at the phoneme level (Nicolao, Beeston, et al., 2015). Wang et al. use siamese networks for modeling discrepancy between normal and distorted children's speech (J. Wang et al., 2019). We take a similar approach but we do not need a database of reference speech. Instead, we train a statistical model to estimate the probability of pronouncing a sentence by a native speaker. Qian et al. propose a statistical pronunciation model as well (Qian et al., 2010). Unlike our work, in which we create a model of 'correct' pronunciation, they build a model that generates hypotheses of mispronounced speech.

3.3.3 Proposed model

The design consists of three subsystems: a Phoneme Recognizer (PR), a Pronunciation Model (PM), and a Pronunciation Error Detector (PED), illustrated in Figure 3.4. The PR recognizes phonemes spoken by a student. The PM estimates the probabilities of having been pronounced by a native speaker. Finally, the PED computes word-level mispronunciation probabilities. In Figure 3.5, we present detailed architectures of the PR, PM, and PED.

For example, considering the text: 'I said alone not gone' with the canonical representation of /ay - s eh d - ax l ow n - n aa t - g aa n/. Polish L2 speakers of

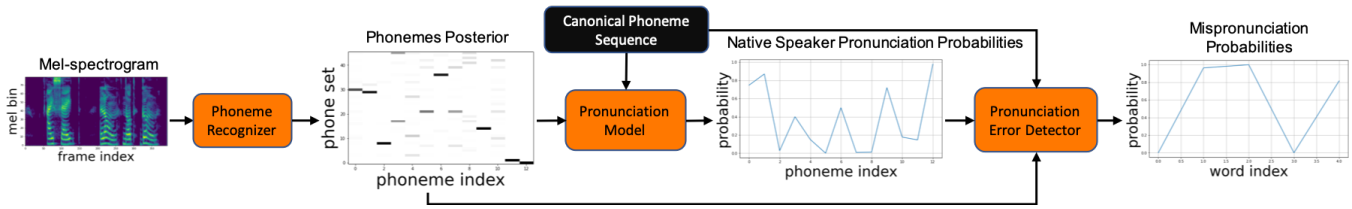


FIGURE 3.4: Architecture of the system for detecting mispronounced words in a spoken sentence.

English often mispronounce the /eh/ phoneme in the second word as /ey/. The PM would identify the /ey/ as having a low probability of being pronounced by a native speaker in the middle of the word 'said', which the PED would translate into a high probability of mispronunciation.

3.3.3.1 Phoneme recognizer

The PR (Figure 3.5a) uses beam decoding (Graves et al., 2013) to estimate N hypotheses of the most likely sequences of phonemes that are recognized in the speech signal $\mathbf{o}$. A single hypothesis is denoted as $\mathbf{r_o} \sim p(\mathbf{r_o}|\mathbf{o})$. The speech signal $\mathbf{o}$ is represented by a mel-spectrogram with f frames and 80 mel-bins. Each sequence of phonemes $\mathbf{r_o}$ is accompanied by the posterior phoneme probabilities of shape: $(l_{r_o}, l_s + 1)$. l_{r_o} is the length of the sequence and l_s is the size of the phoneme set (45 phonemes including 'pause', 'end of sentence (eos)', and a 'blank' label required by the CTC-based model).

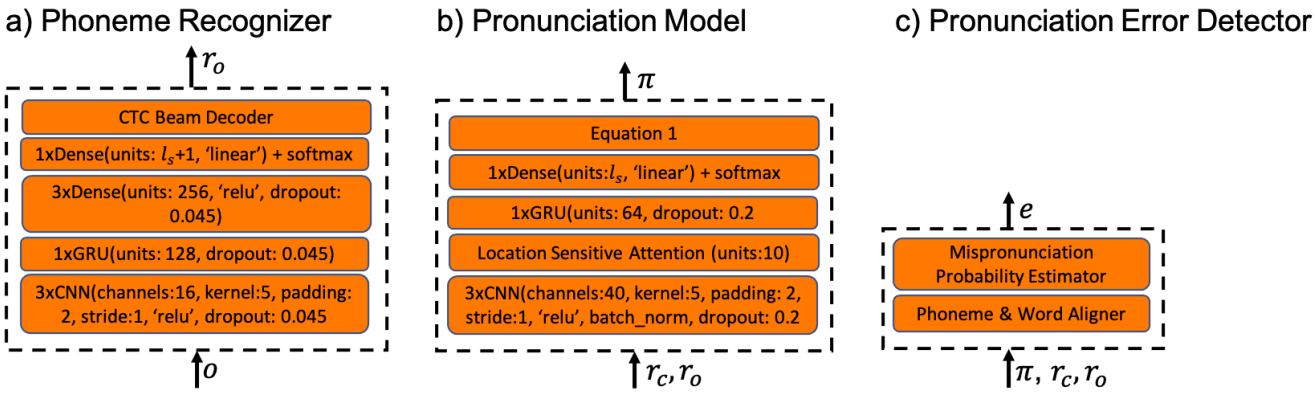


FIGURE 3.5: Architecture of the PR, PM, and PED subsystems. l_s - the size of the phoneme set.

3.3.3.2 Pronunciation model

The PM (Figure 3.5b) is an encoder-decoder neural network following Sutskever et al. (Sutskever et al., 2014). Instead of building a text-to-text translation system between two languages, we use it for phoneme-to-phoneme conversion. The sequence of phonemes $\mathbf{r_c}$ that a native speaker was expected to pronounce is converted into the

sequence of phonemes $\mathbf{r}$ they had pronounced, denoted as $\mathbf{r} \sim p(\mathbf{r}|\mathbf{r_c})$. Once trained, the PM acts as a probability mass function, computing the likelihood sequence $\boldsymbol{\pi}$ of the phonemes $\mathbf{r_o}$ pronounced by a student conditioned on the expected (canonical) phonemes $\mathbf{r_c}$. The PM is denoted in Eq. 3.2, which we implemented in MxNet (T. e. a. Chen, 2015) using 'sum' and 'element-wise multiply' linear-algebra operations.

$$\boldsymbol{\pi} = \sum_{\mathbf{r_o}} p(\mathbf{r_o}|\mathbf{o}) p(\mathbf{r} = \mathbf{r_o}|\mathbf{r_c}) \tag{3.2}$$

The model is trained on phoneme-to-phoneme speech data created automatically by passing the speech of the native speakers through the PR. By annotating the data with the PR, we can make the PM model more resistant to possible phoneme recognition inaccuracies of the PR at testing time.

3.3.3.3 Pronunciation error detector

The PED (Figure 3.5c) computes the probabilities of mispronunciations $\mathbf{e}$ at the word level, denoted as $\mathbf{e} \sim p(\mathbf{e}|\mathbf{r_o}, \boldsymbol{\pi}, \mathbf{r_c})$. The PED is conditioned on three inputs: the phonemes $\mathbf{r_o}$ recognized by the PR, the corresponding pronunciation likelihoods $\boldsymbol{\pi}$ from the PM, and the canonical phonemes $\mathbf{r_c}$. The model starts with aligning the canonical and recognized sequences of phonemes. We adopted a dynamic programming algorithm for aligning biological sequences developed by Needleman-Wunsch (Needleman et al., 1970). Then, the probability of mispronunciation for a given word is computed with Eq. 3.3, k denotes the word index, and j is the phoneme index in the word with the lowest probability of pronunciation.

$$p(\mathbf{e}_k) = \begin{cases} 0 & \text{if aligned phonemes match,} \\ 1 - \pi_{k,j} & \text{otherwise.} \end{cases} \tag{3.3}$$

We compute the probabilities of mispronunciation for N phoneme recognition hypotheses from the PR. Mispronunciation for a given word is detected if the probability of mispronunciation falls below a given threshold for all hypotheses. The hyper-parameter $N = 4$ was manually tuned on a single L2 speaker from the testing set to optimize the PED in the precision metric.

3.3.4 Experiments and discussion

We want to understand the effect of accounting for uncertainty in the PR-PM system presented in Section 3.2.2. To do this, we compare it with two other variants, PR-LIK and PR-NOLIK, and analyze precision and recall metrics. The PR-LIK system helps us understand how important is it to account for the phonetic variability in the PM. To switch the PM off, we modify it so that it considers only a single way for a sentence to be pronounced correctly.

The PR-NOLIK variant corresponds to the CTC-based mispronunciation detection model proposed by Leung et al. (Leung et al., 2019). To reflect this, we make two

modifications compared to the PR-PM system. First, we switch the PM off in the same way we did it in the PR-LIK system. Second, we set the posterior probabilities of recognized phonemes in the PR to 100%, which means that the PR is always certain about the phonemes produced by a speaker. There are some slight implementation differences between Leung's model and PR-NOLIK, for example, regarding the number of units in the neural network layers. We use our configuration to make a consistent comparison with PR-PM and PR-LIK systems. One can hence consider PR-NOLIK as a fair state-of-the-art baseline (Leung et al., 2019).

3.3.4.1 Model details

For extracting mel-spectrograms, we used a time step of 10 ms and a window size of 40 ms. The PR was trained with CTC Loss and Adam Optimizer (batch size: 32, learning rate: 0.001, gradient clipping: 5). We tuned the following hyper-parameters of the PR with Bayesian Optimization: dropout, CNN channels, GRU, and dense units. The PM was trained with the cross-entropy loss and AdaDelta optimizer (batch size: 20, learning rate: 0.01, gradient clipping: 5). The location-sensitive attention in the PM follows the work by Chorowski et al. (J. K. Chorowski et al., 2015). The PR and PM models were implemented in MxNet Deep Learning framework.

3.3.4.2 Speech corpora

For training and testing the PR and PM, we used 125.28 hours of L1 and L2 English speech from 983 speakers segmented into 102812 sentences, sourced from multiple speech corpora: TIMIT (Garofolo et al., 1993), LibriTTS (Zen et al., 2019), Isle (Atwell et al., 2003) and GUT Isle (Weber et al., 2020). We summarize it in Table 3.6. All speech data were downsampled to 16 kHz. Both L1 and L2 speech were phonetically transcribed using Amazon proprietary grapheme-to-phoneme model and used by the PR. Automatic transcriptions of L2 speech do not capture pronunciation errors, but we found it is still worth including automatically transcribed L2 speech in the PR. L2 corpora were also annotated by 5 native speakers of American English for word-level pronunciation errors. There are 3624 mispronounced words out of 13191 in the Isle Corpus and 1046 mispronounced words out of 5064 in the GUT Isle Corpus.

From the collected speech, we held out 28 L2 speakers and used them only to assess the performance of the systems in the mispronunciation detection task. It includes 11 Italian and 11 German speakers from the Isle corpus (Atwell et al., 2003), and 6 Polish speakers from the GUT Isle corpus (Weber et al., 2020).

3.3.4.3 Experimental results

The PR-NOLIK detects mispronounced words based on the difference between the canonical and recognized phonemes. Therefore, this system does not offer any flexibility in optimizing the model for higher precision.

TABLE 3.6: The summary of speech corpora used by the PR.

Native Language	Hours	Speakers
English	90.47	640
Unknown	19.91	285
German and Italian	13.41	46
Polish	1.49	12

The PR-LIK system incorporates posterior probabilities of recognized phonemes. It means that we can tune this system towards higher precision, as illustrated in Figure 3.6. Accounting for uncertainty in the PR helps when there is more than one likely sequence of phonemes that could have been uttered by a user, and the PR model is uncertain which one it is. For example, the PR reports two likely pronunciations for the text 'I said' /ay s eh d/. The first one, /s eh d/ with /ay/ phoneme missing at the beginning and the alternative one /ay s eh d/ with the /ay/ phoneme present. If the PR considered only the mostly likely sequence of phonemes, like PR-NOLIK does, it would incorrectly raise a pronunciation error. In the second example, a student read the text 'six' /s ih k s/ mispronouncing the first phoneme /s/ as /t/. The likelihood of the recognized phoneme is only 34%. It suggests that the PR model is quite uncertain on what phoneme was pronounced. However, sometimes even in such cases, we can be confident that the word was mispronounced. It is because the PM computes the probability of pronunciation based on the posterior probability from the PR model. In this particular case, other phoneme candidates that account for the remaining 66% of uncertainty are also unlikely to be pronounced by a native speaker. The PM can take it into account and correctly detect a mispronunciation.

However, we found that the effect of accounting for uncertainty in the PR is quite limited. Compared to the PR-NOLIK system, the PR-LIK raises precision on the GUT Isle corpus only by 6% (55% divided by 52%), at the cost of dropping recall by about 23%. We can observe a much stronger effect when we account for uncertainty in the PM model. Compared to the PR-LIK system, the PR-PM system further increases precision between 11% and 18%, depending on the decrease in recall between 20% to 40%. One example where the PM helps is illustrated by the word 'enough' that can be pronounced in two similar ways: /ih n ah f/ or /ax n ah f/ (short 'i' or 'schwa' phoneme at the beginning.) The PM can account for phonetic variability and recognize both versions as pronounced correctly. Another example is word linking (Hieke, 1984). Native speakers tend to merge phonemes of neighboring words. For example, in the text 'her arrange' /hh er - er ey n jh/, two neighboring phonemes /er/ can be pronounced as a single phoneme: /hh er ey n jh/. The PM model can correctly recognize multiple variations of such pronunciations.

Complementary to precision-recall curve showed in Figure 3.6, we present in Table 3.7 one configuration of the precision and recall scores for the PR-LIK and PR-PM systems. This configuration is selected in such a way that: a) recall for both

systems is close to the same value, b) to illustrate that the PR-PM model has a much bigger potential of increasing precision than the PR-LIK system. A similar conclusion can be made by inspecting multiple different precision and recall configurations in the precision and recall plots for both Isle and GUT Isle corpora.

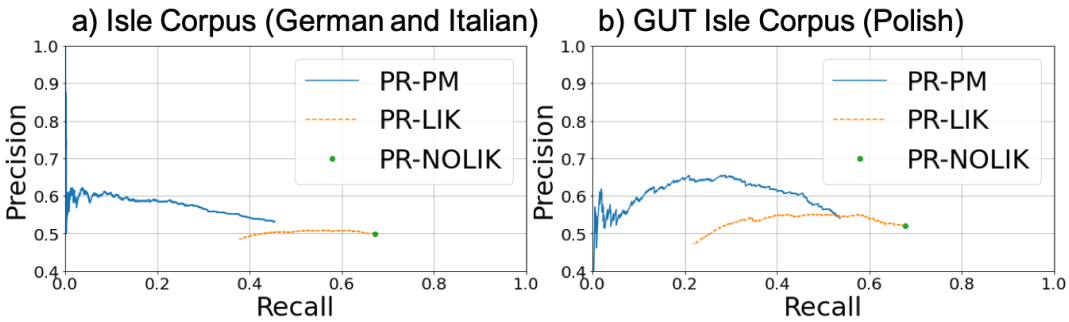


FIGURE 3.6: Precision-recall curves for the evaluated systems.

TABLE 3.7: Precision and recall of detecting word-level mispronunciations. CI - Confidence Interval.

Model	Precision [%,95%CI]	Recall [%,95%CI]
Isle corpus (German and Italian)		
PR-LIK	49.39 (47.59-51.19)	40.20 (38.62-41.81)
PR-PM	54.20 (52.32-56.08)	40.20 (38.62-41.81)
GUT Isle corpus (Polish)		
PR-LIK	54.91 (50.53-59.24)	40.29 (36.66-44.02)
PR-PM	61.21 (56.63-65.65)	40.15 (36.51-43.87)

3.3.5 Conclusion and future work

To report fewer false pronunciation alarms, it is important to move away from the two simplifying assumptions that are usually made by common methods for pronunciation assessment: a) phonemes can be recognized with high accuracy, b) a sentence can be read in a single correct way. We acknowledged that these assumptions do not always hold. Instead, we designed a model that: a) accounts for the uncertainty in phoneme recognition and b) accounts for multiple ways a sentence can be pronounced correctly due to phonetic variability. We found that to optimize precision, it is more important to account for the phonetic variability of speech than accounting for uncertainty in phoneme recognition. We showed that the proposed model can raise the precision of detecting mispronounced words by up to 18% compared to the common methods.

In the future, we plan to adapt the PM model to correctly pronounced L2 speech to account for phonetic variability of non-native speakers. We plan to combine the

PR, PM, and PED modules and train the model jointly to eliminate accumulation of statistical errors coming from disjoint training of the system.

3.4 Detection of Lexical Stress Errors in Non-Native (L2) English with Data Augmentation and Attention

Daniel Korzekwa, Roberto Barra-Chicote, Szymon Zaporowski, Grzegorz Beringer, Jaime Lorenzo-Trueba, Alicja Serafinowicz, Jasha Droppo, Thomas Drugman, Bozena Kostek,
Detection of Lexical Stress Errors in Non-Native (L2) English with Data Augmentation and Attention, Interspeech, 2021

Abstract

This paper describes two novel complementary techniques that improve the detection of lexical stress errors in non-native (L2) English speech: attention-based feature extraction and data augmentation based on Neural Text-To-Speech (TTS). In a classical approach, audio features are usually extracted from fixed regions of speech such as the syllable nucleus. We propose an attention-based deep learning model that automatically derives optimal syllable-level representation from frame-level and phoneme-level audio features. Training this model is challenging because of the limited amount of incorrect stress patterns. To solve this problem, we propose to augment the training set with incorrectly stressed words generated with Neural TTS. Combining both techniques achieves 94.8% precision and 49.2% recall for the detection of incorrectly stressed words in L2 English speech of Slavic and Baltic speakers.

3.4.1 Introduction

Computer Assisted Pronunciation Training (CAPT) usually focuses on practicing pronunciation of phonemes (Witt et al., 2000; Leung et al., 2019; Korzekwa, Lorenzo-Trueba, Zaporowski, et al., 2021), while there is evidence in non-native (L2) English speakers that practicing lexical stress improves speech intelligibility (Field, 2005; Lepage et al., 2014). Lexical stress is a syllable-level phonological feature. It is a part of the phonological rules that define how words should be spoken in a given language. Stressed syllables are usually longer, louder, and expressed with a higher pitch than their unstressed counterparts (Jung et al., 2018). Lexical stress is inter-connected with phonemic representation. For example, placing lexical stress on a different syllable of a word may lead to different phonemic realizations known as 'vowel reduction' (Bergem, 1991).

The focal point of our work is the detection of words with incorrect stress patterns. The training data with human speech is usually highly imbalanced, with few training examples of incorrectly stressed words. It makes training machine learning models for this task challenging. We address this problem by augmenting the training set

with synthetic speech that is generated with Neural Text-To-Speech (TTS) (Latorre, Lachowicz, Lorenzo-Trueba, Merritt, Drugman, Ronanki, and Klimkov, 2019). Neural TTS allows us generating words with both correct and incorrect stress patterns.

Most of the existing approaches for automated lexical stress assessment are based on carefully designed features that are extracted from fixed regions of speech signal such as the syllable nucleus (Ferrer et al., 2015; Shahin et al., 2016; J.-Y. Chen et al., 2010). We introduce attention mechanism (Vaswani, Shazeer, Parmar, Uszkoreit, Jones, Gomez, Ł. Kaiser, et al., 2017) to automatically learn optimal syllable-level representation. Attention-based approach originates from the intuition of how people detect specific patterns in high dimensional and unstructured data such as visual and speech signals (Posner et al., 1990). For example, we might focus our attention on the duration ratio between nuclei of two neighboring syllables, incidentally, an important predictor of lexical stress. The syllable-level representation is derived from frame-level (F0, intensity) and phoneme-level (duration) audio features and the corresponding phonetic representation of a word. We do not indicate precisely the regions of the audio signal that are important for the detection of lexical stress errors. The attention mechanism does it automatically.

To the best of our knowledge, this paper is the first attempt, for the task of lexical stress error detection, to: *i)* augment the training data with Neural TTS, *ii)* use attention mechanisms to automatically extract syllable-level features for lexical stress error detection. Ruan et al. (Ruan et al., 2019) used attention-based architecture of transformers for lexical stress detection. However, their paper concerns recognizing stressed and unstressed phonemes. They do not detect lexical stress errors, which is crucial in CAPT applications.

The paper is structured as follows. In Section 3.4.2, we review the related work. Section 3.4.3 describes the proposed model. Section 3.4.4 reviews human and synthetic speech corpora. In Section 3.4.5, we present our experiments, and Section 3.4.6 concludes the paper.

3.4.2 Related work

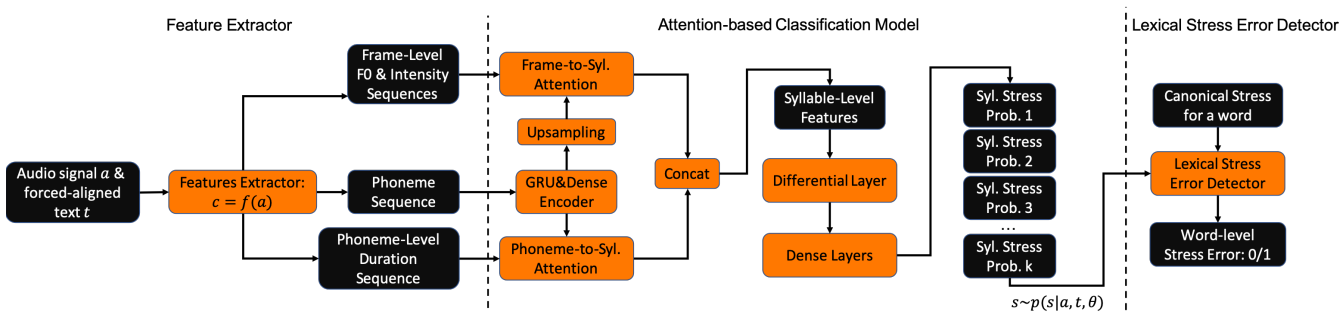


FIGURE 3.7: Attention-based Deep Learning model for the detection of lexical stress errors.

The existing work focuses on the supervised classification of lexical stress using Neural Networks (K. Li, Mao, et al., 2018; Shahin et al., 2016), Support Vector Machines (J.-Y. Chen et al., 2010; J. Zhao et al., 2011) and Fisher's linear discriminant (N. Chen et al., 2007). There are two popular variants: a) discriminating syllables between primary stress/no stress (Ferrer et al., 2015), and b) classifying between primary stress/secondary stress/no stress (K. Li, Qian, Kang, et al., 2013; K. Li, Mao, et al., 2018). Ramanathi et al. (Ramanathi et al., 2019) have followed an alternative unsupervised way of classifying lexical stress, which is based on computing the likelihood of an acoustic signal for a number of possible lexical stress representations of a word.

Accuracy is the most commonly used performance metric, and it indicates the ratio of correctly classified stress patterns on a syllable (K. Li, Qian, Kang, et al., 2013) or word level (J.-Y. Chen et al., 2010). On the contrary, following Ferrer et al. (Ferrer et al., 2015), we analyze precision and recall metrics because we aim to detect lexical stress errors and not just classify them.

Existing approaches for the classification and detection of lexical stress errors are based on carefully designed features. They start with aligning a speech signal with phonetic transcription, performed via forced-alignment (Shahin et al., 2016; J.-Y. Chen et al., 2010). Alternatively, Automatic Speech Recognition (ASR) can provide both phonetic transcription and its alignment with a speech signal (K. Li, Qian, Kang, et al., 2013). Then, prosodic features such as duration, energy and pitch (J.-Y. Chen et al., 2010) and cepstral features such as MFCC and Mel-Spectrogram (Ferrer et al., 2015; Shahin et al., 2016) are extracted. These features can be extracted on the syllable (Shahin et al., 2016) or syllable nucleus (Ferrer et al., 2015; J.-Y. Chen et al., 2010) level.

Shahin et al. (Shahin et al., 2016) computed features of neighboring vowels, and Li et al. (K. Li, Qian, Kang, et al., 2013) included the features for two preceding and two following syllables in the model. The features are often preprocessed and normalized to avoid potential confounding variables (Ferrer et al., 2015), and to achieve better model generalization by normalizing the duration and pitch on a word level (Ferrer et al., 2015; N. Chen et al., 2007). Li et al. (K. Li, Mao, et al., 2018) added canonical lexical stress to input features, which improves the accuracy of the model.

In our approach, we use attention mechanisms to derive automatically regions of the audio signal that are important for the detection of lexical stress errors. We also use data augmentation through the generation of artificial data with Neural TTS.

3.4.3 Proposed model

The proposed model consists of three subsystems: Feature Extractor, Attention-based Classification Model, and Lexical Stress Error Detector. It is illustrated in Figure 3.7.

3.4.3.1 Feature extractor

The Feature Extractor extracts prosodic features and phonemes from speech signal **a** and forced-aligned text **t**. To obtain forced-alignment, we used Montreal toolkit (McAuliffe et al., 2017) along with an acoustic model pretrained on LibriSpeech ASR corpus (Panayotov et al., 2015). The prosodic features $\mathbf{c} = f(\mathbf{a})$ are formed by: F0, intensity [dB SPL] and phoneme-level durations. The F0 and intensity features are computed at the frame level using Praat library (Boersma, 2006) (time step: 10 ms, window size: 40 ms). The F0 contour is linearly interpolated in unvoiced regions. These raw features will be further transformed by the attention-based model to the syllable-level representation.

3.4.3.2 Attention-based classification model

The Attention-based Classification Model maps frame-level and phoneme-level features to the syllable-level representation. Then, it produces a lexical stress pattern **s**, modeled as a sequence of Bernoulli random variables $\mathbf{s} = \{s_1, .., s_k\}$ (stressed/unstressed) over K syllables of a multi-syllable word, conditioned on audio **a** and text **t** representations. Let us define it as a conditional probability distribution $\mathbf{s} \sim p(\mathbf{s}|\mathbf{a}, \mathbf{t}, \boldsymbol{\theta})$, where $\boldsymbol{\theta}$ are the parameters of the model.

To extract syllable-level features, we use two dot-product attentions operating on the frame and phoneme levels. To build better intuition on what these two attention do, in Figure 3.8 we show the frame-level and phoneme-level attention plots for the word 'garage' pronounced by a Polish speaker and incorrectly stressed on the first syllable in reference to American English. This word has a similar pronunciation but different lexical stress in Polish and American English languages ('G AA1 R AA0 ZH' vs 'G ER0 AA1 ZH'). Both attentions find the most relevant regions of the frame-level and phoneme-level features.

The dot-product attention is presented in Eq. 3.4, and it follows the notation proposed by Vaswani et al. (Vaswani, Shazeer, Parmar, Uszkoreit, Jones, Gomez, Ł. Kaiser, et al., 2017). It is based on three inputs: Query (**Q**), Keys (**K**) and Values (**V**), where d_k is the dimensionality of **K**.

$$Attention(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = softmax(\frac{\mathbf{Q}\mathbf{K}^t}{\sqrt{d_k}})\mathbf{V} \tag{3.4}$$

The attention inputs are represented as follows. Query refers to the syllable positional embeddings defined by one-hot syllable index encodings. Keys represents a sequence of sub-phonemes. Each sub-phoneme is represented by a set of features: *phoneme_id*, *syllable_index*, *is_vowel*, *left_or_right_sub_phoneme*. All features are one-hot encoded and processed with a Gated Recurrent Unit (GRU) layer (Cho, Van Merriënboer, et al., 2014) (units:4, dropout: 0.24). In the end, encoded sub-phoneme sequence is passed through linear dense layers. In the case of the frame-level attention, the encoded sub-phoneme sequence is upsampled to the frame level using phoneme

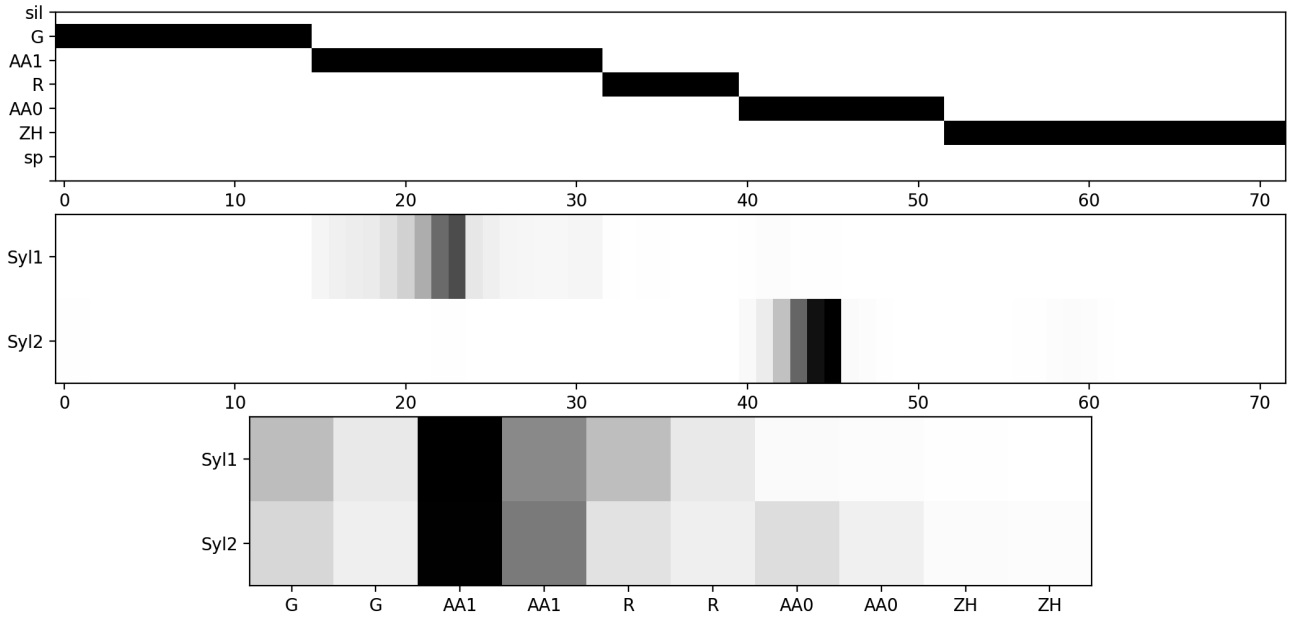


FIGURE 3.8: Top: forced-alignment mapping between phonemes and frames for the word 'garage'. Middle: Frame-to-syllable attention weights matrix. Bottom: (Sub)Phoneme-to-syllable attention weights matrix.

durations from forced-alignment. In upsampling, we simply replicate phonemes across aligned frames of audio signal. Similar phoneme-to-frame upsampling has been recently adopted in Text-To-Speech (Elias et al., 2020). Finally, Values are the $F0/intensity$ and $duration$ features for frame-level and phoneme-level attentions respectively.

To model relative prominence, we introduce a differential bi-directional layer that computes the ratios of syllable-level acoustic features for each syllable and its two neighbors (Figure 3.7). The bi-directional layer is implemented as a simple 'division' math operation and it does not contain any trainable parameters. The output of the differential layer is further processed by three dense layers (units: 4, activation: tanh, dropout: 0.24), followed by a linear dense layer (units: 2, dropout: 0.24) that produces a two-dimensional output for each syllable. It is then squeezed by a softmax function to generate lexical stress probabilities.

3.4.3.3 Training of the classification model

We train the model on a set of N triplets that contains 1) human recorded words and 2) synthetic words generated using Neural TTS. A single triplet is represented by $\{\mathbf{s_n}, \mathbf{a_n}, \mathbf{t_n}\}$, where $n = 1..N$ is the index of a training example.

The concept of data augmentation can be explained using a framework of Bayesian Inference. Consider three random variables, lexical stress $\mathbf{s_n}$, audio signal $\mathbf{a_n}$ and text $\mathbf{t_n}$. All variables are observed for the training examples of human speech. However,

for the synthetic speech, we only observe the lexical stress and text variables. The audio signal is unobserved (hidden) because we have to generate it.

To train this model, we derive a negative log-likelihood loss over a joint probability distribution of lexical stress **s** and audio **a** random variables, as depicted in Eq. 3.5. The loss is further approximated with the variational lower bound (Jordan et al., 1999), as presented in Eq. 3.6 (we omit $\boldsymbol{\theta}$ for brevity). For the training examples of synthetic speech, the conditional probability distribution over the audio signal $\mathbf{a_n} \sim p(\mathbf{a_n}|\mathbf{s_n}, \mathbf{t_n})$ is estimated with Neural TTS, and for human recorded words, it is given explicitly.

$$\mathcal{L}(\subseteq) = -\sum_{n}^{N} log \int p(\mathbf{s_n}, \mathbf{a_n}|\mathbf{t_n}, \boldsymbol{\theta}) d\mathbf{a_n} \tag{3.5}$$

$$log \int p(\mathbf{s_n}, \mathbf{a_n}|\mathbf{t_n}) \mathbf{da_n} \approx E_{\mathbf{a_n} \sim p(\mathbf{a_n}|\mathbf{t_n}, \mathbf{s_n})}[log p(\mathbf{s_n}|\mathbf{a_n}, \mathbf{t_n})] \tag{3.6}$$

The model was implemented in MxNet (T. e. a. Chen, 2015), trained with Stochastic Gradient Descent optimizer (learning rate: 0.1, batch size: 20) and tuned with Bayesian optimization (Paleyes et al., 2019). Training data were split into buckets based on the number of frames in an audio signal, using Gluon-NLP package (Guo et al., 2020). A single bucket contains words with the same number of syllables with zero-padded acoustic and sub-phoneme sequences.

3.4.3.4 Lexical stress error detector

The Lexical Stress Error Detector reports on lexical stress error if the expected (canonical) and estimated lexical stress for a given syllable do not match and the corresponding probability is higher than a given threshold.

3.4.4 Speech corpus

Our speech corpus consists of human and synthetic speech. The data were split into training and testing sets with disjointed speakers ascribed to each set. Human speech contains L1 and L2 speakers of English. Synthetic data were generated with Neural TTS and are included only in the training set. All audio files were downsampled to a 16 kHz sampling rate. The data are summarized in Table 3.8, and we provide more details in the following subsections.

TABLE 3.8: Train and test sets details.

Data set	Speakers (L2)	Words (unique)	Stress Errors
Train set (human)	473 (10)	8223 (1528)	425
Train set (TTS)	1 (0)	3937 (1983)	2005
Test set (human)	176 (21)	2108 (378)	189

3.4.4.1 Human speech

Due to the limited availability of L2 corpora, we recorded our own L2-English corpus of Slavic and Baltic speakers. It also allows us to evaluate the model during interactive English learning sessions with our students. The corpus contains speech from 25 speakers (23 Polish, 1 Ukrainian and 1 Lithuanian): 7 females and 18 males, all between 24 and 40 years old. All speakers read a list of two hundred words. One hundred words were prepared by a professional English teacher, including frequently mispronounced words by Slavic and Baltic students. The second half consists of the most common words that were obtained from Google's Trillion Word Corpus (Michel et al., 2011) based on n-gram frequency analysis. We excluded abbreviations and one-syllable words.

Additionally, L1 and L2 English speech was collected from publicly available speech data sets, including TIMIT (Garofolo et al., 1993), Arctic (Kominek et al., 2004), L2-Arctic (G. Zhao et al., 2018) and Porzuczek (Porzuczek et al., 2017).

3.4.4.2 Synthetic speech

Complementary to human recordings, synthetic speech was generated with Neural TTS by Latorre et al. (Latorre, Lachowicz, Lorenzo-Trueba, Merritt, Drugman, Ronanki, and Klimkov, 2019). The Neural TTS consists of two modules. Context-generation module is an attention-based encoder-decoder neural network that generates a mel-spectrogram from a sequence of phonemes. Then, a Neural Vocoder converts it to the speech signal. The Neural Vocoder is a neural network of architecture similar to the work by (Oord et al., 2018). The Neural TTS was trained using speech of a professional American voice talent. To generate words with different lexical stress patterns, we modify lexical stress markers associated with the vowels in the phonemic transcription of a word. For example, with the input of /r iy1 m ay0 n d/ we can place lexical stress on the first syllable of the word 'remind'. 1980 popular English words were synthesized with correct and incorrect stress patterns.

3.4.4.3 Lexical stress annotations

L1 corpora were segmented into words and annotated automatically using a proprietary Amazon American English Lexicon, taking into account the syntactic context of the word. Neural TTS speech and the speech of L2 speakers were annotated by 5 American English linguists into 'primary' and 'no stress' categories, keeping the words for which a minimum of 4 out of 5 linguists agreed on the stress pattern. Annotators were not able to distinguish between primary and secondary lexical stress. 81.5% of synthesized words matched the intended stress patterns with a minimum of 4 annotators' agreement. It shows that Neural TTS can be used to generate incorrectly stressed speech.

3.4.5 Experiments

The proposed model (Att_TTS) from Section 3.4.3 is compared to three baseline models that are designed to measure the impact of the Neural TTS data augmentation and the attention mechanism. To compare these models, we plotted their precision-recall curves and gave their corresponding area under a curve (AUC) along with our results, see Figure 3.9.

The Att_NoTTS model has the same architecture as the Att_TTS, but the synthetic speech is excluded from the 'training set'. The NoAtt_TTS model uses the same training set as the Att_TTS, but it has no attention mechanism. Instead, as a syllable-level representation, it uses mean values of acoustic features for the corresponding syllable nucleus. The NoAtt_NoTTS model has no attention, and it does not use Neural TTS data augmentation.

As a state-of-the-art baseline, we use the work by Ferrer et al. (Ferrer et al., 2015). However, a direct comparison is not possible. In their test corpus, there were 46.4% (191 out of 411) of incorrectly stressed words, far more than 9.4% (189 out of 2109) words in our experiment. The fewer lexical stress errors are made by users, the more challenging it is to detect it. They also used proprietary L2 English of Japanese speakers. Due to the lack of available benchmark and standard speech corpora for the task of lexical stress assessment, we could not make a fairer comparison with the state-of-the-art.

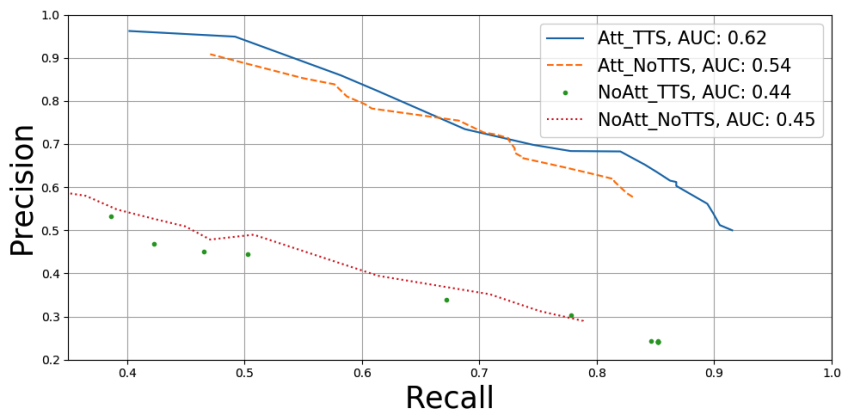


FIGURE 3.9: Precision-recall curves for evaluated systems.

3.4.5.1 Experimental results

First, we compare Att_NoTTS and NoAtt_NoTTS models. Using the attention mechanism for automatic extraction of syllable-level features significantly improves the detection of lexical stress errors. It is illustrated by precision-recall curves and AUC metric in Figure 3.9. To be comparable with the study by Ferrer et al., we fix recall to around 50% and compare the models using precision as shown in Table 3.9.

The Att_NoTTS attention-based can be further improved. Augmenting the training set with incorrectly stressed words (Att_TTS) boosts precision from 87.85% to 94.8%, at a recall level of 50%. Data augmentation helps because it increases the number of words with incorrect stress patterns in the training set. It prevents the model from exploiting a strong correlation between phonemes and lexical stress in correctly stressed words. Using data augmentation in the simpler no-attention-based model (NoAtt_TTS) does not help. It is because NoAtt_TTS uses only prosodic features for fixed regions of speech, so this model cannot overfit to phonetic input.

TABLE 3.9: Precision and recall [%, 95% Confidence Interval] of detecting lexical stress errors, at around 50% recall. * - Ferrer et al. model has been evaluated on the data with 46.4% of lexical stress errors, compared to 9.4% of errors on our data set. This data point indicates that our proposed model AttTTS should outperform Ferrr et al. model if both were evaluated exactly in the same conditions.

Model	**Precision**	**Recall**
AttTTS	94.8 (89.18-98.03)	49.2 (42.13-56.3)
AttNoTTS	87.85 (80.67-93.02)	49.74 (42.66-56.82)
NoAttTTS	44.39 (37.85-51.09)	50.26 (43.18-57.34)
NoAttNoTTS	48.98 (42.04-55.95)	50.79 (43.70-57.86)
Ferrer et al. (Ferrer et al., 2015) *	95.00 (na-na)	48.3 (na-na)

Ferrer et al. (Ferrer et al., 2015) reported on a similar performance to our Att_TTS model with a precision of 95% and a recall of 48.3% on L2 English speech of Japanese speakers. However, in their testing data, the proportion of incorrectly stressed words is much larger, which makes it easier to detect lexical stress errors.

3.4.6 Conclusion and future work

Using an attention-based neural network for the automatic extraction of syllable-level features significantly improves the detection of lexical stress errors in L2 English speech, compared to baseline models. However, this model has a tendency to classify lexical stress based on highly-correlated phonemes. We can counteract this effect by augmenting the training set with incorrectly stressed words generated with Neural TTS. It boosts the performance of the attention-based model by 14.8% in the AUC metric and by 7.9% in precision, while maintaining recall at a level close to 50%. Data Augmentation, however, does not help when applied to a simpler model without an attention mechanism.

We found that the current word-level model is not able to correctly classify lexical stress when two words are linked (Hieke, 1984) and stress shift may occur (Shattuck-Hufnagel et al., 1994). For example, two neighboring phonemes /er/ in the text 'her arrange' /hh er - er ey n jh/ are pronounced as a single phoneme. Therefore, in future, we plan to move away from the assessment of isolated words and extend the current model to detect lexical stress errors at the sentence level. We plan to

replace a single-speaker TTS model to generate synthetic lexical stress errors with a multi-speaker model. We plan to analyze the accuracy of detecting lexical stress errors for speakers with different proficiency levels of English.

3.5 Speech synthesis is almost all you need

Daniel Korzekwa, Jaime Lorenzo-Trueba, Thomas Drugman, Bozena Kostek,
Computer-assisted Pronunciation Training - Speech synthesis is almost all you need, accepted for publication in Speech Communication Journal on June 17 '2022, in print

Abstract

The research community has long studied computer-assisted pronunciation training (CAPT) methods in non-native speech. Researchers focused on studying various model architectures, such as Bayesian networks and deep learning methods, as well as on the analysis of different representations of the speech signal. Despite significant progress in recent years, existing CAPT methods are not able to detect pronunciation errors with high accuracy (only 60% precision at 40%-80% recall). One of the key problems is the low availability of mispronounced speech that is needed for the reliable training of pronunciation error detection models. If we had a generative model that could mimic non-native speech and produce any amount of training data, then the task of detecting pronunciation errors would be much easier. We present three innovative techniques based on phoneme-to-phoneme (P2P), text-to-speech (T2S), and speech-to-speech (S2S) conversion to generate correctly pronounced and mispronounced synthetic speech. We show that these techniques not only improve the accuracy of three machine learning models for detecting pronunciation errors but also help establish a new state-of-the-art in the field. Earlier studies have used simple speech generation techniques such as P2P conversion, but only as an additional mechanism to improve the accuracy of pronunciation error detection. We, on the other hand, consider speech generation to be the first-class method of detecting pronunciation errors. The effectiveness of these techniques is assessed in the tasks of detecting pronunciation and lexical stress errors. Non-native English speech corpora of German, Italian, and Polish speakers are used in the evaluations. The best proposed S2S technique improves the accuracy of detecting pronunciation errors in AUC metric by 41% from 0.528 to 0.749 compared to the state-of-the-art approach.

3.5.1 Introduction

Language plays a key role in online education, giving people access to large amounts of information contained in articles, books, and video lectures. Thanks to spoken language and other forms of communication, such as a sign-language, people can participate in interactive discussions with teachers and take part in lively brainstorming with other people. Unfortunately, education is not available to everybody. According

to the UNESCO report, 40% of the global population do not have access to education in the language they understand (UNESCO, 2016). 'If you don't understand, how can you learn?' the report says. English is the leading language on the Internet, representing 25.9% of the world's population (Statista, 2021). Regrettably, research by EF (Education First) (EF-Education-First, 2020) shows a large disproportion in English proficiency across countries and continents. People from regions of 'very low' language proficiency, such as the Middle East, are unable to navigate through English-based websites or communicate with people from an English-speaking country.

Computer-Assisted Language Learning (CALL) helps to improve the English language proficiency of people in different regions (Levy et al., 2013). CALL relies on computerized self-service tools that are used by students to practice a language, usually a foreign language, also known as a non-native (L2) language. Students can practice multiple aspects of the language, including grammar, vocabulary, writing, reading, and speaking. Computer-based tools can also be used to measure student's language skills and their learning potential by using Computerized Dynamic Assessment (C-DA) test (Mehri Kamrood et al., 2019). CALL can complement traditional language learning provided by teachers. It also has a chance to make second language learning more accessible in scenarios where traditional ways of learning languages are not possible due to the cost of learning or the lack of access to foreign language teachers.

Computer-Assisted Pronunciation Training (CAPT) is a part of CALL responsible for learning pronunciation skills. It has been shown to help people practice and improve their pronunciation skills (Neri et al., 2008; Golonka et al., 2014; Tejedor-Garcıa et al., 2020). CAPT consists of two components: an automated pronunciation evaluation component (Leung et al., 2019; Z. Zhang et al., 2021; Korzekwa, Lorenzo-Trueba, Zaporowski, et al., 2021) and a feedback component (Ai, 2015). The automated pronunciation evaluation component is responsible for detecting pronunciation errors in spoken speech, for example, for detecting words pronounced incorrectly by the speaker. The feedback component informs the speaker about mispronounced words and advises how to pronounce them correctly. This article is devoted to the topic of automated detection of pronunciation errors in non-native speech. This area of CAPT can take advantage of technological advances in machine learning and bring us closer to creating a fully automated assistant based on artificial intelligence for language learning.

The research community has long studied the automated detection of pronunciation errors in non-native speech. Existing work has focused on various tasks such as detecting mispronounced phonemes (Leung et al., 2019) and lexical stress errors (Ferrer et al., 2015). Researcher have given most attention to studying various machine learning models such as Bayesian networks (Witt et al., 2000; H. Li et al., 2011) and deep learning methods (Leung et al., 2019; Z. Zhang et al., 2021), as well as analyzing different representations of the speech signal such as prosodic features

(duration, energy and pitch) (J.-Y. Chen et al., 2010), and cepstral/spectral features (Ferrer et al., 2015; Shahin et al., 2016; Leung et al., 2019). Despite significant progress in recent years, existing CAPT methods detect pronunciation errors with relatively low accuracy of 60% precision at 40%-80% recall (Leung et al., 2019; Korzekwa, Lorenzo-Trueba, Zaporowski, et al., 2021; Z. Zhang et al., 2021). Highlighting correctly pronounced words as pronunciation errors by the CAPT tool can demotivate students and lower the confidence in the tool. Likewise, missing pronunciation errors can slow down the learning process.

One of the main challenges with the existing CAPT methods is poor availability of mispronounced speech, which is required for the reliable training of pronunciation error detection models. We propose a reformulation of the problem of pronunciation error detection as a task of synthetic speech generation. Intuitively, if we had a generative model that could mimic mispronounced speech and produce any amount of training data, then the task of detecting pronunciation errors would be much easier. The probability of pronunciation errors for all the words in a sentence can then be calculated using the Bayes rule (Bishop, 2006). In this new formulation, we move the complexity to learning the speech generation process that is well suited to the problem of limited speech availability (Huybrechts et al., 2021; Shah et al., 2021; Fazel et al., 2021). The proposed method outperforms the state-of-the-art model (Leung et al., 2019) in detecting pronunciation errors in AUC metric by 41% from 0.528 to 0.749 on the GUT Isle Corpus of L2 Polish speakers.

To put the new formulation of the problem into action, we propose three innovative techniques based on phoneme-to-phoneme (P2P), text-to-speech (T2S), and speech-to-speech (S2S) conversion to generate correctly pronounced and mispronounced synthetic speech. We show that these techniques not only improve the accuracy of three machine learning models for detecting pronunciation errors but also help establish a new state-of-the-art in the field. The effectiveness of these techniques is assessed in two tasks: detecting mispronounced words (replacing, adding, removing phonemes, or pronouncing an unknown speech sound) and detecting lexical stress errors. The results presented in this study are the culmination of our recent work on speech generation in pronunciation error detection task (Korzekwa, Lorenzo-Trueba, Zaporowski, et al., 2021; Korzekwa, Lorenzo-Trueba, Drugman, Calamaro, et al., 2021; Korzekwa, Barra-Chicote, Zaporowski, et al., 2021), including a new S2S technique.

In short, the contributions of the paper are as follows:

- A new paradigm for the automated detection of pronunciation errors is proposed, reformulating the problem as a task of generating synthetic speech.
- A unified probabilistic view on P2P, T2S, and S2S techniques is presented in the context of detecting pronunciation errors.
- A new S2S method to generate synthetic speech is proposed, which outperforms the state-of-the-art model (Leung et al., 2019) in detecting pronunciation errors.

- Comprehensive experiments are described to demonstrate the effectiveness of speech generation in the tasks of pronunciation and lexical stress error detection.

The outline of the rest of this paper is: Section 3.5.2 presents related work. Section 3.5.3 describes the proposed methods of generating synthetic speech for automatic detection of pronunciation errors. Section 3.5.4 describes the human speech corpora used to train the pronunciation error detection models in the experiments. Section 3.5.5 presents experiments demonstrating the effectiveness of various synthetic speech generation methods in improving the accuracy of the detection of pronunciation and lexical stress errors. Finally, conclusions and future work are presented in Section 3.5.6.

3.5.2 Related work

3.5.2.1 Pronunciation error detection

Phoneme recognition approaches

Most existing CAPT methods are designed to recognize the phonemes pronounced by the speaker and compare them with the expected (canonical) pronunciation of correctly pronounced speech (Witt et al., 2000; K. Li, Qian, and Meng, 2016; Sudhakara, Ramanathi, Yarra, and Ghosh, 2019; Leung et al., 2019). Any discrepancy between the recognized and canonical phonemes results in a pronunciation error at the phoneme level. Phoneme recognition approaches generally fall into two categories: methods that align a speech signal with phonemes (forced-alignment techniques) and methods that first recognize the phonemes in the speech signal and then align the recognized and canonical phoneme sequences. Aside these two categories, CAPT methods can be split into multiple other categories:

Forced-alignment techniques (H. Li et al., 2011; K. Li, Qian, and Meng, 2016; Sudhakara, Ramanathi, Yarra, and Ghosh, 2019; Cheng et al., 2020) are based on the work of Franco et al. (Franco et al., 1997) and the Goodness of Pronunciation (GoP) method (Witt et al., 2000). In the first step, GoP uses Bayesian inference to find the most likely alignment between canonical phonemes and the corresponding audio signal (forced alignment). In the next step, GoP calculates the ratio between the likelihoods of the canonical and the most likely pronounced phonemes. Finally, it detects mispronunciation if the ratio drops below a certain threshold. GoP has been further extended with Deep Neural Networks (DNNs), replacing the Hidden Markov Model (HMM) and Gaussian Mixture Model (GMM) techniques for acoustic modeling (K. Li, Qian, and Meng, 2016; Sudhakara, Ramanathi, Yarra, and Ghosh, 2019). Cheng et al. (Cheng et al., 2020) improves GoP performance with the hidden representation of speech extracted in an unsupervised way. This model can detect pronunciation errors based on the input speech signal and the reference canonical speech signal, without using any linguistic information such as text and phonemes.

The methods that do not use forced-alignment recognize the phonemes pronounced by the speaker purely from the speech signal and only then align them with the canonical phonemes (Minematsu, 2004; Harrison et al., 2009; A. Lee and Glass, 2013; Plantinga et al., 2019; Sudhakara, Ramanathi, Yarra, Das, et al., 2019; L. Zhang et al., 2020). Leung et al. (Leung et al., 2019) use a phoneme recognizer that recognizes phonemes only from the speech signal. The phoneme recognizer is based on Convolutional Neural Network (CNN), a Gated Recurrent Unit (GRU), and Connectionist Temporal Classification (CTC) loss. Leung et al. report that it outperforms other forced-alignment (K. Li, Qian, and Meng, 2016) and forced-alignment-free (Harrison et al., 2009) techniques in the task of detecting mispronunciations at the phoneme-level in L2 English.

There are two challenges with presented approaches for pronunciation error detection. First, phonemes pronounced by the speaker must be recognized accurately, which has been proved difficult (Z. Zhang et al., 2021; J. Chorowski, Bahdanau, et al., 2014; J. K. Chorowski et al., 2015; Bahdanau et al., 2016). Phoneme recognition is difficult, especially in non-native speech, as different languages have different phoneme spaces. Second, standard approaches assume only one canonical pronunciation of a given text, but this assumption is not always true due to the phonetic variability of speech, e.g., differences between regional accents. For example, the word 'enough' can be pronounced by native speakers in multiple ways: /ih n ah f/ or /ax n ah f/ (short 'i' or 'schwa' phoneme at the beginning). In our previous work, we solve these problems by creating a native speech pronunciation model that returns the probability of the sentence to be spoken by a native speaker (Korzekwa, Lorenzo-Trueba, Zaporowski, et al., 2021).

Techniques based on phoneme recognition can be supplemented by a reference speech signal obtained from the speech database (Xiao et al., 2018; Nicolao, Beeston, et al., 2015; J. Wang et al., 2019) or generated from the phonetic representation (Korzekwa, Lorenzo-Trueba, Zaporowski, et al., 2021; Qian et al., 2010). Xiao et al. (Xiao et al., 2018) use a pair of speech signals from a student and a native speaker to classify native and non-native speech. Mauro et al. (Nicolao, Beeston, et al., 2015) use the speech of the reference speaker to detect mispronunciation errors at the phoneme level. Wang et al. (J. Wang et al., 2019) use Siamese networks to model the discrepancy between normal and distorted children's speech. Qian et al. (Qian et al., 2010) propose a statistical model of pronunciation in which they build a model that generates hypotheses of mispronounced speech.

In this work, we use the end-to-end method to detect pronunciation errors directly, without having to recognize phonemes as an intermediate step. The end-to-end approach is discussed in more detail in the next section.

End-to-end methods

The phoneme recognition approaches presented so far rely on phonetically transcribed speech labeled by human listeners. Phonetic transcriptions are needed to

train a phoneme recognition model. Human-based transcription is a time-consuming task, especially with L2 speech, where listeners need to recognize mispronunciation errors. Sometimes L2 speech transcription may be even impossible because different languages have different phoneme sets, and it is unclear which phonemes were pronounced by the speaker. In our recent work, we have introduced a novel model (known as WEAKLY-S, i.e., weakly supervised) for detecting pronunciation errors at the world level that does not require phonetically transcribed L2 speech (Korzekwa, Lorenzo-Trueba, Drugman, Calamaro, et al., 2021). During training, the model is weakly supervised, in the sense that in L2 speech, only mispronounced words are marked, and the data do not need to be phonetically transcribed. In addition to the primary task of detecting mispronunciation errors at the world level, the second task uses a phoneme recognizer trained on automatically transcribed L1 speech. Zhang et al. (Z. Zhang et al., 2021) employ a multi-task model with two tasks: phoneme-recognition and pronunciation error detection tasks. Unlike our WEAKLY-S model, they use the Needleman-Wunsch algorithm (Needleman et al., 1970) from bioinformatics to align the canonical and recognized phoneme sequences, but this algorithm cannot be tuned to detect pronunciation errors. The WEAKLY-S model automatically learns the alignment, thus eliminating a potential source of inaccuracy. The alignment is learned through an attention mechanism that automatically maps the speech signal to a sequence of pronunciation errors at the word level. Tong et al. [39] propose to use a multi-task framework in which a neural network model is used to learn the joint space between the acoustic characteristics of adults and children. Additionally, Duan et al. (Duan et al., 2019) propose a multi-task model for acoustical modeling with two tasks for native and non-native speech respectively.

The work of Zhang et al. (Z. Zhang et al., 2021) and our recent work (Korzekwa, Lorenzo-Trueba, Drugman, Calamaro, et al., 2021) are end-to-end methods of direct estimation of pronunciation errors, setting up a new trend in the field of automated pronunciation assessment. In this article, we use the end-to-end method as well, but we extend it by the S2S method of generating mispronounced speech.

Other trends

All the works presented so far treat pronunciation errors as discrete categories, at best producing the probability of mispronunciation. In contrast, Bi-Cheng et al. (Yan, M.-C. Wu, et al., 2020) propose a model capable of identifying phoneme distortions, giving the user more detailed feedback on mispronunciation. In our recent work, we provide more fine-grained feedback by indicating the severity level of mispronunciation (Korzekwa, Lorenzo-Trueba, Drugman, Calamaro, et al., 2021).

Active research is conducted not only on modeling techniques but also on speech representation. Xu et al. (Xu et al., 2021) and Peng et al. (Peng et al., 2021) use the Wav2vec 2.0 speech representation that is created in an unsupervised way. They report that it outperforms existing methods and requires three times less speech training data. Lin et al. (B. Lin et al., 2021) use transfer learning by taking advantage

of deep latent features extracted from the Automated Speech Recognition (ASR) acoustic model and report improvements over the classic GOP-based method.

In this work, we use a mel-spectrogram as a speech representation in the pronunciation error detection model. We also use a mel-spectrogram to represent the speech signal in the T2S and S2S methods of generating mispronounced speech.

3.5.2.2 Lexical stress error detection

CAPT usually focuses on practicing the pronunciation of phonemes (Witt et al., 2000; Leung et al., 2019; Korzekwa, Lorenzo-Trueba, Zaporowski, et al., 2021). However, there is evidence that practicing lexical stress improves the intelligibility of non-native English speech (Field, 2005; Lepage et al., 2014). Lexical stress is a phonological feature of a syllable. It is part of the phonological rules that govern how words should be pronounced in a given language. Stressed syllables are usually longer, louder, and expressed with a higher pitch than their unstressed counterparts (Jung et al., 2018). The lexical stress is related to the phonemic representation. For example, placing lexical stress on a different syllable of a word can lead to various phonemic realizations known as 'vowel reduction' (Bergem, 1991). Students should be able to practice both pronunciation and lexical stress in spoken language. We study both topics to better understand the potential of using speech generation methods in CAPT.

The existing works focus on the supervised classification of lexical stress using Neural Networks (K. Li, Mao, et al., 2018; Shahin et al., 2016), Support Vector Machines (J.-Y. Chen et al., 2010; J. Zhao et al., 2011), and Fisher's linear discriminant (N. Chen et al., 2007). There are two popular variants: a) discriminating syllables between primary stress/no stress (Ferrer et al., 2015), and b) classifying between primary stress/secondary stress/no stress (K. Li, Qian, Kang, et al., 2013; K. Li, Mao, et al., 2018). Ramanathi et al. (Ramanathi et al., 2019) have followed an alternative unsupervised way of classifying lexical stress, which is based on computing the likelihood of an acoustic signal for a number of possible lexical stress representations of a word.

Accuracy is the most commonly used performance metric, and it indicates the ratio of correctly classified stress patterns on a syllable (K. Li, Qian, Kang, et al., 2013) or word level (J.-Y. Chen et al., 2010). On the contrary, Ferrer et al. (Ferrer et al., 2015), analyzed the precision and recall metrics to detect lexical stress errors and not just classify them.

Most existing approaches for the classification and detection of lexical stress errors are based on carefully designed features. They start with aligning a speech signal with phonetic transcription, performed via forced-alignment (Shahin et al., 2016; J.-Y. Chen et al., 2010). Alternatively, ASR can provide both phonetic transcription and its alignment with a speech signal (K. Li, Qian, Kang, et al., 2013). Then, prosodic features such as duration, energy and pitch (J.-Y. Chen et al., 2010) and cepstral features such as Mel Frequency Cepstral Coefficients (MFCC) and Mel-Spectrogram (Ferrer et al.,

2015; Shahin et al., 2016) are extracted. These features can be extracted on the syllable (Shahin et al., 2016) or syllable nucleus (Ferrer et al., 2015; J.-Y. Chen et al., 2010) level. Shahin et al. (Shahin et al., 2016) computes features of neighboring vowels, and Li et al. (K. Li, Qian, Kang, et al., 2013) includes the features for two preceding and two following syllables in the model. The features are often preprocessed and normalized to avoid potential confounding variables (Ferrer et al., 2015), and to achieve better model generalization by normalizing the duration and pitch on a word level (Ferrer et al., 2015; N. Chen et al., 2007). Li et al. (K. Li, Mao, et al., 2018) adds canonical lexical stress to input features, which improves the accuracy of the model.

In our recent work, we use attention mechanisms to automatically derive areas of the audio signal that are important for the detection of lexical stress errors (Korzekwa, Barra-Chicote, Zaporowski, et al., 2021). In this work, we use the T2S method to generate synthetic lexical stress errors to improve the accuracy of detecting lexical stress errors.

3.5.2.3 Synthetic speech generation for pronunciation error detection

Existing synthetic speech generation techniques for detecting pronunciation errors can be divided into two categories: data augmentation and data generation.

Data augmentation techniques are designed to generate new training examples for existing mispronunciation labels. Badenhorst et al. (Badenhorst et al., 2017) simulate new speakers by adjusting the speed of raw audio signals. Eklund (Eklund, 2019) generates additional training data by adding background noise and convolving the audio signal with the impulse responses of the microphone of a mobile device and a room.

Data generation techniques are designed to generate new training data with new labels of both correctly pronounced and mispronounced speech. Most existing works are based on the P2P technique to generate mispronounced speech by perturbing the phoneme sequence of the corresponding audio using a variety of strategies (A. Lee et al., 2016; Komatsu et al., 2019; Fu, J. Lin, et al., 2021; Yan, Jiang, et al., 2021; Korzekwa, Lorenzo-Trueba, Zaporowski, et al., 2021). In addition to P2P techniques, in our recent work, we use T2S to generate synthetic lexical stress errors (Korzekwa, Lorenzo-Trueba, Drugman, Calamaro, et al., 2021). Qian et al. (Qian et al., 2010) introduce a generative model to create hypotheses of mispronounced speech and use it as a reference speech signal to detect pronunciation errors. Recently, we proposed a similar technique to create a pronunciation model of native speech to account for many ways of correctly pronouncing a sentence by a native speaker (Korzekwa, Lorenzo-Trueba, Zaporowski, et al., 2021).

Synthetic speech generation techniques have recently gained attention in other related fields. Fazel et al. (Fazel et al., 2021) use synthetic speech generated with T2S to improve accuracy in ASR. Huang et al. (G. Huang et al., 2016) use a machine translation technique to generate text to train an ASR language model in a low-resource language. At the same time, Shah et al. (Shah et al., 2021) and Huybrechts et

al. (Huybrechts et al., 2021) employ S2S voice conversion to improve the quality of speech synthesis in the data reduction scenario.

All the presented works on the detection of pronunciation errors treat synthetic speech generation as a secondary contribution. In this article, we present a unified perspective of synthetic speech generation methods for detecting pronunciation errors. This article extends our previous work (Korzekwa, Lorenzo-Trueba, Zaporowski, et al., 2021; Korzekwa, Barra-Chicote, Zaporowski, et al., 2021; Korzekwa, Lorenzo-Trueba, Drugman, Calamaro, et al., 2021) and introduces a new S2S method to detect pronunciation errors. To the best of our knowledge, there are no papers devoted to generating pronunciation errors with the S2S technique and using it in the detection of pronunciation errors.

3.5.3 Methods of generating pronunciation errors

To detect pronunciation errors, first, the spoken language must be separated from other factors in the signal and then incorrectly pronounced speech sounds have to be identified. Separating speech into multiple factors is difficult, as speech is a complex signal. It consists of prosody (F0, duration, energy), timbre of the voice, and the representation of the spoken language. Spoken language is defined by the sounds (phones) perceived by people. Phones are the realizations of phonemes - a human abstract representation of how to pronounce a word/sentence. Speech may also present variability due to the recording channel and environmental effects such as noise and reverberation. Detecting pronunciation errors is very challenging, also because of the limited amount of recordings with mispronounced speech. To address these challenges, we reformulate the problem of pronunciation error detection as the task of synthetic speech generation.

Let $\mathbf{s}$ be the speech signal, $\mathbf{r}$ be the sequence of phonemes that the user is trying to pronounce (canonical pronunciation), and $\mathbf{e}$ be the sequence of probabilities of mispronunciation at the phoneme or word level. The original task of detecting pronunciation errors is defined by:

$$\mathbf{e} \sim p(\mathbf{e}|\mathbf{s}, \mathbf{r}) \tag{3.7}$$

where the formulation of the problem as the task of synthetic speech generation is defined as follows:

$$\mathbf{s} \sim p(\mathbf{s}|\mathbf{e}, \mathbf{r}) \tag{3.8}$$

The probability of pronunciation errors for all the words in a sentence can then be calculated using the Bayes rule (Bishop, 2006):

$$p(\mathbf{e}|\mathbf{s}, \mathbf{r}) = \frac{p(\mathbf{e}|\mathbf{r})p(\mathbf{s}|\mathbf{e}, \mathbf{r})}{p(\mathbf{s}|\mathbf{r})} \tag{3.9}$$

From Eq. 3.9, one can see that there is no need to directly learn the probability of pronunciation errors $p(\mathbf{e}|\mathbf{s},\mathbf{r})$, since the complexity of the problem has now been transferred to learning the speech generation process $p(\mathbf{s}|\mathbf{e},\mathbf{r})$. Such a formulation of the problem opens the way to the inclusion of additional prior knowledge into the model:

1. Replacing the phoneme in a word while preserving the original speech signal results in a pronunciation error (P2P method).
2. Changing the speech signal while retaining the original pronunciation results in a pronunciation error (T2S method).
3. There are many variations of mispronounced speech that differ in terms of the voice timbre and the prosodic aspects of speech (S2S method).

To solve Eq. 3.9, we use Markov Chain Monte Carlo Sampling (MCMC) (Koller et al., 2009). In this way, the prior knowledge can be incorporated by generating N training examples $\{\mathbf{e_i},\mathbf{s_i},\mathbf{r_i}\}$ for $i = 1..N$ with the use of P2P (prior knowledge 1), T2S (prior knowledge 2), and S2S (prior knowledge 3) methods. Accounting for the prior knowledge, intuitively corresponds to an increase in the amount of training data, which contributes to outperforming state-of-the-art models for detecting pronunciation errors, as presented in Section 3.5.5. Eq. 3.9 can then be optimized with standard gradient-based optimization techniques. In the following subsections, we present the P2P conversion, T2S, and S2S methods of generating correctly and incorrectly pronounced speech in details.

3.5.3.1 P2P method

To generate synthetic mispronounced speech, it is enough to start with correctly pronounced speech and modify the corresponding sequence of phonemes. This simple idea does not even require generating the speech signal itself. It can be observed that the probability of mispronunciations depends on the discrepancy between the speech signal and the corresponding canonical pronunciation. This leads to the P2P conversion model shown in Figure 3.10a.

Let $\{\mathbf{e_{noerr}},\mathbf{s},\mathbf{r}\}$ be a single training example containing: the sequence of $0s$ denoting correctly pronounced phonemes, the speech signal, and the sequence of phonemes representing the canonical pronunciation. Let $\mathbf{r}^{'}$ be the sequence of phonemes with injected mispronunciations such as phoneme replacements, insertions, and deletions:

$$\mathbf{r}^{'} \sim p(\mathbf{r}^{'}|\mathbf{r}) \tag{3.10}$$

then the probability of mispronunciation for the jth phoneme is defined by:

$$e_j^{'} = \begin{cases} 1 & \text{if } r_j^{'}! = r_j \\ 0 & \text{otherwise} \end{cases} \tag{3.11}$$

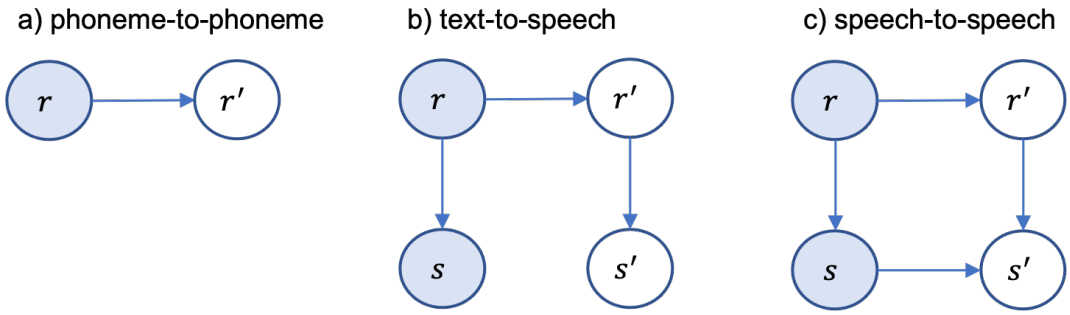


FIGURE 3.10: Probabilistic graphical models for three methods to generate pronunciation errors: P2P, T2S and S2S. Empty circles represent hidden (latent) variables, while filled (blue) circles represent observed variables. **s** - the speech signal, **r** - the sequence of phonemes that the user is trying to pronounce (canonical pronunciation), the superscript $'$ represents a variable with generated mispronunciations.

The probabilities of mispronunciation can be projected from the level of phonemes to the level of words. A word is treated as mispronounced if at least one pair of phonemes in the word $\{r_j^{'}, r_j\}$ does not match. At the end of this process, a new training example is created with artificially introduced pronunciation errors: $\{\mathbf{e_{err}}, \mathbf{s}, \mathbf{r}^{'}\}$. Note that the speech signal **s** in the new training example is unchanged from the original training example and only phoneme transcription is manipulated.

Implementation

To generate synthetic pronunciation errors, we use a simple approach of perturbing phonetic transcription for the corresponding speech audio. First, we sample these utterances with replacement from the input corpora of human speech. Then, for each utterance, we replace the phonemes with random phonemes with a given probability.

3.5.3.2 T2S method

The T2S method expands on P2P by making it possible to create speech signals that match the synthetic mispronunciations. The T2S method for generating mispronounced speech is a generalization of the P2P method, as can be seen by the comparison of the two methods shown in Figures 3.10a and 3.10b.

One problem with the P2P method is that it cannot generate a speech signal for the newly created sequence of phonemes $\mathbf{r}^{'}$. As a result, pronunciation errors will dominate in the training data containing new sequences of phonemes $\mathbf{r}^{'}$. Therefore, it will be possible to detect pronunciation errors only from the canonical representation $\mathbf{r}^{'}$, ignoring information contained in the speech signal. To mitigate this issue, there should be two training examples for the phonemes $\mathbf{r}^{'}$, one representing mispronounced speech: $\{\mathbf{e_{err}}, \mathbf{s}, \mathbf{r}^{'}\}$, and the second one for correct pronunciation: $\{\mathbf{e_{noerr}}, \mathbf{s}^{'}, \mathbf{r}^{'}\}$, where:

$$\mathbf{s}^{'} \sim p(\mathbf{s}^{'} | \mathbf{e_{noerr}}, \mathbf{r}^{'}) \tag{3.12}$$

Because we now have the speech signal $\mathbf{s}^{'}$, another training example can be created as: $\{\mathbf{e}_{\mathbf{err}}, \mathbf{s}^{'}, \mathbf{r}\}$. In summary, T2S method extends a single training example of correctly pronounced speech to four combinations of correctly and incorrect pronunciations:

- $\{\mathbf{e}_{\mathbf{noerr}}, \mathbf{s}, \mathbf{r}\}$ – correctly pronounced input speech
- $\{\mathbf{e}_{\mathbf{err}}, \mathbf{s}, \mathbf{r}^{'}\}$ – mispronounced speech generated by the P2P method
- $\{\mathbf{e}_{\mathbf{noerr}}, \mathbf{s}^{'}, \mathbf{r}^{'}\}$ – correctly pronounced speech generated by the T2S method
- $\{\mathbf{e}_{\mathbf{err}}, \mathbf{s}^{'}, \mathbf{r}\}$ – mispronounced speech generated by the T2S method

Implementation

The synthetic speech is generated with the Neural TTS described by Latorre et al. (Latorre, Lachowicz, Lorenzo-Trueba, Merritt, Drugman, Ronanki, and Klimkov, 2019). The Neural TTS consists of two modules. The context-generation module is an attention-based encoder-decoder neural network that generates a mel-spectrogram from a sequence of phonemes. The Neural Vocoder then converts it into a speech signal. The Neural Vocoder is a neural network of architecture similar to Parallel Wavenet (Oord et al., 2018). The Neural TTS is trained using the speech of a single native speaker. To generate words with different lexical stress patterns, we modify the lexical stress markers associated with the vowels in the phonetic transcription of the word. For example, with the input of `/r iy1 m ay0 n d/` we can place lexical stress on the first syllable of the word 'remind'.

3.5.3.3 S2S method

The S2S method is designed to simulate the diverse nature of speech, as there are many ways to correctly pronounce a sentence. The prosodic aspects of speech, such as pitch, duration, and energy, can vary. Similarly, phonemes can be pronounced differently. To mimic human speech, speech generation techniques should allow a similar level of variability. The T2S method outlined in the previous section always produces the same output for the same phoneme input sequence. The S2S method is designed to overcome this limitation.

S2S converts the input speech signal $\mathbf{s}$ in a way to change the pronounced phonemes (phoneme replacements, insertions, and deletions) from the input phonemes $\mathbf{r}$ to target phonemes $\mathbf{r}^{'}$ while preserving other aspects of speech, including voice timbre and prosody (Eq. 3.13 and Figure 3.10c). In this way, the natural variability of human speech is preserved, resulting in generating many variations of incorrectly pronounced speech. The prosody will differ in various versions of the sentence of the same speaker, while the same sentence spoken by many speakers will differ in the voice timbre.

$$\mathbf{s}^{'} \sim p(\mathbf{s}^{'} | \mathbf{e}_{\mathbf{noerr}}, \mathbf{r}^{'}, \mathbf{s}) \tag{3.13}$$

Similarly to the T2S method, the S2S method outputs four types of speech pronounced correctly and incorrectly: $\{\mathbf{e}_{\mathbf{noerr}}, \mathbf{s}, \mathbf{r}\}$, $\{\mathbf{e}_{\mathbf{err}}, \mathbf{s}, \mathbf{r}^{'}\}$, $\{\mathbf{e}_{\mathbf{noerr}}, \mathbf{s}^{'}, \mathbf{r}^{'}\}$, and $\{\mathbf{e}_{\mathbf{err}}, \mathbf{s}^{'}, \mathbf{r}\}$.

Implementation

Synthetic speech is generated by introducing mispronunciations into the input speech, while preserving the duration of the phonemes and timbre of the voice. The architecture of the S2S model is shown in Figure 3.11. The mel-spectrogram of the input speech signal $\mathbf{s}$ is forced-aligned with the corresponding canonical phonemes $\mathbf{r}$ to get the duration of the phonemes. The speaker id has to be provided together with the input speech to enable the source speaker's voice to be maintained. Mispronunciations are introduced into the canonical phonemes $\mathbf{r}$ according to the P2P method described in Section 3.5.3.1. Mispronounced phonemes $\mathbf{r}^{'}$ along with phonemes duration and speaker id are processed by the encoder-decoder, which generates the mel-spectrogram $\mathbf{s}^{'}$. The encoder-decoder transforms the phoneme-level representation into frame-level features and then generates all mel-spectrogram frames in parallel. The mel-spectrogram is converted to an audio signal with Universal Vocoder (Jiao et al., 2021). Without the Universal Vocoder, it would not be possible to generate the raw audio signal for hundreds of speakers included in the LibriTTS corpus. Details of the S2S method are shown in the works of Shah et al. (Shah et al., 2021) and Jiao et al. (Jiao et al., 2021). The main difference between these two models and our S2S model is the use of the P2P mapping to introduce pronunciation errors.

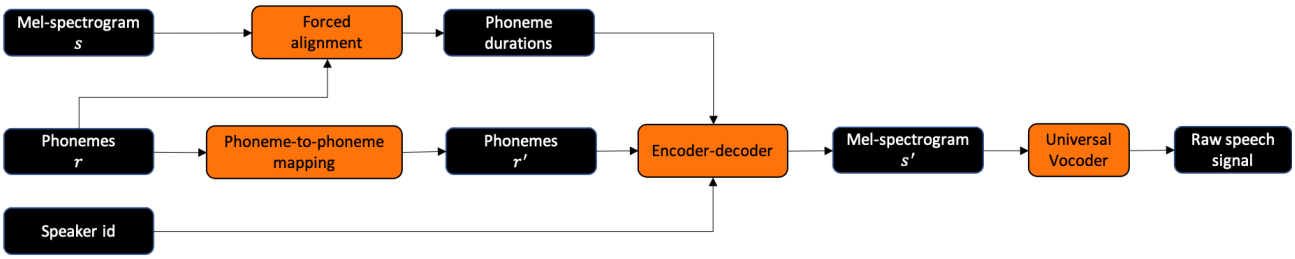


FIGURE 3.11: Architecture of the S2S model to generate mispronounced synthetic speech while maintaining prosody and voice timbre of the input speech. The black rectangles represent the data (tensors) and the orange boxes represent processing blocks. This color notation is used in all machine learning model diagrams throughout the article.

3.5.3.4 Summary of mispronounced speech generation

Generation of synthetic mispronounced speech and detection of pronunciation errors were presented from the probabilistic perspective of the Bayes-rule. With this formulation, we can better understand the relationship between P2P, T2S and S2S methods, and see that the S2S method generalizes two simpler methods. Following this reasoning, we can argue that using the Bayes rule gives us a nice mathematical framework to potentially further generalize the S2S method, e.g. by adding a language variable to the model to support multilingual pronunciation error detection.

There is another advantage of modeling pronunciation error detection from the probabilistic perspective - it paves the way for joint training of mispronounced speech generation and pronunciation error detection models. In the present work, we are training separate machine learning models for both tasks, but it should be possible to train both models jointly using the framework of Variational Inference (Jordan et al., 1999) instead of MCMC to infer the probability of mispronunciation in Eq. 3.9.

3.5.4 Speech corpora

3.5.4.1 Corpora of continuous speech

Speech corpora of recorded sentences is a combination of L1 and L2 English speech. L1 speech is obtained from the TIMIT (Garofolo et al., 1993) and the LibriTTS (Zen et al., 2019) corpora. L2 speech comes from the Isle (Atwell et al., 2003) corpus (German and Italian speakers) and the GUT Isle (Weber et al., 2020) corpus (Polish speakers). In total, we used 125.28 hours of L1 and L2 English speech from 983 speakers segmented into 102812 sentences. A summary of the speech corpora is presented in Table 3.10, whereas the details are presented in our recent work (Korzekwa, Lorenzo-Trueba, Drugman, Calamaro, et al., 2021).

The speech data are used in all the pronunciation error detection experiments presented in Section 3.5.5. From the collected speech, we held out 28 L2 speakers and used them only to assess the performance of the systems in the mispronunciation detection task. It includes 11 Italian and 11 German speakers from the Isle corpus (Atwell et al., 2003), and 6 Polish speakers from the GUT Isle corpus (Weber et al., 2020). The human speech training data is extended with synthetic pronunciation errors generated by the methods presented in Section 3.5.3.

TABLE 3.10: Summary of human speech corpora used in the pronunciation error detection experiments. * - audiobooks read by volunteers from all over the world (Zen et al., 2019)

Native Language	Hours	Speakers
English	90.47	640
Unknown*	19.91	285
German and Italian	13.41	46
Polish	1.49	12

3.5.4.2 Corpora of isolated words

The speech corpora consist of human and synthetic speech. The data were divided into training and testing sets with separate speakers assigned to each set. Human speech includes native (L1) and non-native (L2) English speech. L1 speech corpora are made of TIMIT (Garofolo et al., 1993) and Arctic (Kominek et al., 2004). L2 corpora contain speech from L2-Arctic [32], Porzuczek (Porzuczek et al., 2017), and our own

TABLE 3.11: Details of the training and test sets for the lexical stress error detection model.

Data set	Speakers (L2)	Words (unique)	Stress Errors
Train set (human)	473 (10)	8223 (1528)	425
Train set (TTS)	1 (0)	3937 (1983)	2005
Test set (human)	176 (21)	2108 (378)	189

recordings of 25 speakers (23 Polish, 1 Ukrainian and 1 Lithuanian). The synthetic data were generated using the T2S method and are only included in the training set. The data are summarized in Table 3.11. For a more detailed description of speech corpora, see Section 4 of our recent work (Korzekwa, Barra-Chicote, Zaporowski, et al., 2021). The speech corpora of isolated words are used in the lexical stress error detection experiment presented in Section 3.5.5.3.

3.5.5 Experiments

3.5.5.1 Generation of mispronounced speech

Experimental setup

The effect of using synthetic pronunciation errors based on the P2P, T2S and S2S methods is evaluated in the task of detecting pronunciation errors in spoken sentences at the word level. First, we analyze the P2P method by comparing it with the state-of-the-art techniques and measure the effect of adding synthetic pronunciation errors to the training data. We then compare P2P with T2S and S2S to assess the benefits of using more complex methods of generating pronunciation errors. The accuracy of detecting pronunciation errors is reported in standard Area Under the Curve (AUC), precision and recall metrics.

Overview of our WEAKLY-S model

We use the pronunciation error detection model (WEAKLY-S) recently proposed by us (Korzekwa, Lorenzo-Trueba, Drugman, Calamaro, et al., 2021). To train the model, the human speech training set is extended with 292,242 utterances of L1 speech with synthetically generated pronunciation errors. To generate pronunciation errors, the P2P, T2S, and S2S methods described in Section 3.5.3 are used.

The WEAKLY-S model produces probabilities of mispronunciation for all words, conditioned by the spoken sentence and canonical phonemes. Mispronunciation errors include phoneme replacement, addition, deletion, or an unknown speech sound. During training, the model is weakly supervised, in the sense that only mispronounced words in L2 speech are marked by listeners and the data do not have to be phonetically transcribed. Due to the limited availability of L2 speech and the

fact that it is not phonetically transcribed, the model is more likely to overfit. To solve this problem, the model is trained in a multi-task setup. In addition to the primary task of detecting mispronunciation error at the word level, the second task uses a phoneme recognizer which is trained on automatically transcribed L1 speech. Both tasks share components of the model, which makes the primary task less likely to overfit.

The architecture of the pronunciation error detection model is shown in Figure 3.12. The model consists of two sub-networks. The Mispronunciations Detection Network (MDN) detects word-level pronunciation errors **e** from the audio signal **s** and canonical phonemes **r**, while the Phoneme Recognition Network (PRN) recognizes phonemes $\mathbf{r_o}$ pronounced by a speaker from the audio signal **s**. The detailed model architecture is presented in Section 2 of our recent work (Korzekwa, Lorenzo-Trueba, Drugman, Calamaro, et al., 2021).

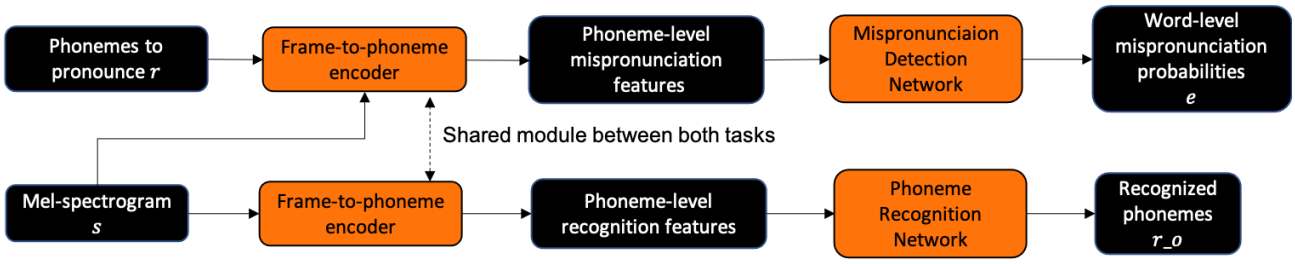


FIGURE 3.12: Architecture of the WEAKLY-S model for word-level pronunciation error detection trained in the multi-task setup. Task 1 - to detect pronunciation errors e. Task 2 - to recognize phonemes r_o.

Results - P2P method

We conducted an ablation study to measure the effect of removing synthetic pronunciation errors from the training data. We trained four variants of the WEAKLY-S model to measure the effect of using synthetic data against other elements of the model. WEAKLY-S is a complete model that also includes synthetic data during training. In the NO-SYNTH-ERR model, we exclude synthetic samples of mispronounced L1 speech, significantly reducing the number of mispronounced words seen during training from 1,129,839 to just 5,273 L2 words. The NO-L2-ADAPT variant does not fine-tune the model on L2 speech, although it is still exposed to L2 speech while being trained on a combined corpus of L1 and L2 speech. The NO-L1L2-TRAIN model is not trained on L1/L2 speech, and fine-tuning on L2 speech starts from scratch. This means that this model will not use a large amount of phonetically transcribed L1 speech data and ultimately no secondary phoneme recognition task will be used.

L2 fine-tuning (NO-L2-ADAPT) is the most important factor influencing the performance of the model (Fig. 3.13 and Table 3.12), with an AUC of 0.517 compared to 0.686 for the full model. Training the model on both L2 and L1 human speech together is not enough. This is because L2 speech accounts for less than 1% of the training data

TABLE 3.12: Ablation study for the GUT Isle corpus to show the effect of using synthetic data and other elements of the WEAKLY-S model. Pr. - Precision, Re. - Recall

Model	Description	AUC	Pr. [%]	Re.[%]
NO-L2-ADAPT	No fine-tuning on L2 speech	0.517	57.89	40.11
NO-L1L2-TRAIN	No pretraining on L1&L2 speech	0.565	59.73	40.20
NO-SYNTH-ERR	No synthetically generated pronunciation errors in the training data	0.615	67.22	40.38
WEAKLY-S	Complete model	**0.686**	75.25	40.38

and the model naturally leans towards L1 speech. The second most important feature is training the model on a combined set of L1 and L2 speech (NO-L1L2-TRAIN), with an AUC of 0.565. L1 speech accounts for over 99% of training data. These data are also phonetically transcribed, and therefore can be used for the phoneme recognition task. The phoneme recognition task acts as a 'backbone' and reduces the effect of overfitting in the main task of detecting errors in the pronunciation of words. Finally, excluding synthetically generated pronunciation errors (NO-SYNTH-ERR) reduces an AUC from 0.686 to 0.615. Although, the synthetic data provides the least improvement to the model, it still increases the accuracy of the model by 11.5% in AUC, contributing to setting up a new state-of-the-art.

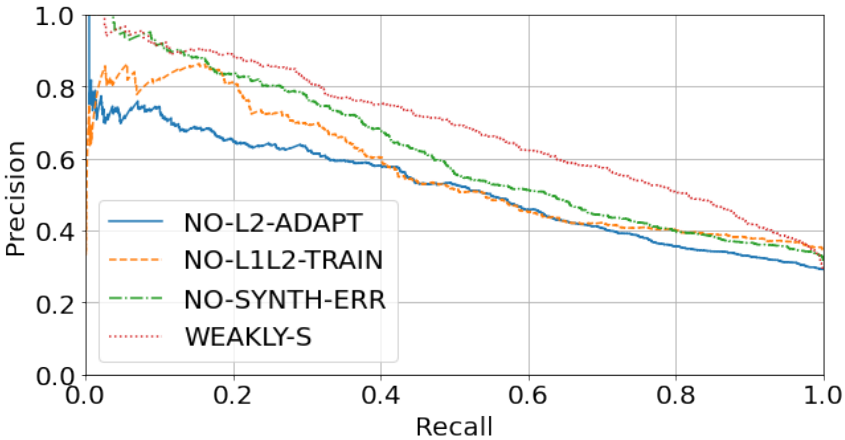


FIGURE 3.13: Precision-recall curve for the ablation study on the GUT Isle corpus, illustrating the effect of using synthetic pronunciation errors generated by the P2P method.

We compare the WEAKLY-S model with two state-of-the-art baselines. The Phoneme Recognizer (PR) model by Leung et al. (Leung et al., 2019) is our first baseline. The PR is based on the CTC loss (Graves, 2012) and outperforms multiple alternative approaches of pronunciation assessment. The original CTC-based model uses a hard likelihood threshold applied to the recognized phonemes. To compare

TABLE 3.13: Accuracy metrics of detecting word-level pronunciation errors. WEAKLY-S vs. baseline models.

Model	AUC	Precision [%,95%CI]	Recall [%,95%CI]
Isle corpus (German and Italian)			
PR	0.555	49.39 (47.59-51.19)	40.20 (38.62-41.81)
PR-PM	0.480	54.20 (52.32-56.08)	40.20 (38.62-41.81)
WEAKLY-S	**0.678**	71.94 (69.96, 73.87)	40.14 (38.56, 41.75)
GUT Isle corpus (Polish)			
PR	0.528	54.91 (50.53-59.24)	40.29 (36.66-44.02)
PR-PM	0.505	61.21 (56.63-65.65)	40.15 (36.51-43.87)
WEAKLY-S	**0.686**	75.25 (71.67-78.59)	40.38 (37.52-43.29)

it with two other models, following our recent work (Korzekwa, Lorenzo-Trueba, Zaporowski, et al., 2021), we have replaced the hard likelihood threshold with a soft threshold. The second baseline is PR extended by the pronunciation model (PR-PM model (Korzekwa, Lorenzo-Trueba, Zaporowski, et al., 2021)). The pronunciation model takes into account the phonetic variability of the speech spoken by native speakers, which results in greater precision in detecting pronunciation errors. The results are shown in Table 3.13. It turns out that the WEAKLY-S model outperforms the second-best model in terms of an AUC by 30% from 0.528 to 0.686 and precision by 23% from 0.612 to 0.752 on the GUT Isle Corpus of Polish speakers. We are seeing similar improvements on the Isle Corpus of German and Italian speakers. The use of synthetic data is an important contribution to the performance of the WEAKLY-S model.

Results - T2S and S2S methods

The main limitation of the P2P method is that it does not generate a new speech signal. The method introduces mispronunciations by operating only on the sequence of phonemes for the corresponding speech. In this experiment, we demonstrate the T2S and S2S methods that can directly generate a speech signal to overcome this limitation. The S2S method introduces mispronunciations into the input native speech while preserving the prosody (phoneme durations) and timbre of the voice. Preserving speech attributes other than pronunciation increases speech variability during training and makes the pronunciation error detection model more reliable during testing. The T2S method can be considered as a simplified variant of the S2S method, in which there is only text as input.

The T2S and S2S methods are compared with the P2P method. Three WEAKLY-S models are trained, differing in the technique of generating mispronounced speech contained in the training data. The S2S method outperforms the P2P method by increasing an AUC score by 9% from 0.686 to 0.749 in the Gut Isle corpus of Polish speakers (Table 3.14). Additionally, an AUC increases from 0.815 to 0.834 for major pronunciation errors (Table 3.15), according to a similar experiment presented in

TABLE 3.14: Comparison of the P2P, T2S and S2S methods in the task of pronunciation error detection assessed on the GUT Isle corpus.

Model	AUC	Precision [%]	Recall [%]
P2P	0.686	75.25 (71.67-78.59)	40.38 (37.52-43.29)
T2S	0.695	76.15 (72.59-79.36)	40.25 (37.44-43.22)
S2S	0.749	80.45 (76.94-83.47)	40.12 (37.12-43.02)

TABLE 3.15: Comparison of the P2P, T2S and S2S methods in the task of pronunciation error detection assessed on the GUT Isle corpus only for major pronunciation errors.

Model	AUC	Precision [%]	Recall [%]
P2P	0.815	91.67 (88.55-94.45)	40.31 (37.43-43.23)
T2S	0.819	92.11 (89.09-94.83)	40.21 (36.81-43.31)
S2S	0.834	93.54 (90.53-96.23)	40.15 (37.26-43.11)

Section 3.4 of (Korzekwa, Lorenzo-Trueba, Drugman, Calamaro, et al., 2021). Interestingly, the T2S method is only slightly better than the P2P method, which suggests that the variability of the generated mispronounced speech provided by the S2S method is really important. The presented experiments show the potential of the S2S method in improving the accuracy of detecting pronunciation errors. The S2S method is able to control voice timbre, phoneme duration, and pronunciation, opening the door to transplanting all three properties from non-native speech and potentially further improving the accuracy of the model.

One downside of the S2S method is its complexity. Compared to the straightforward P2P method, the 9% improvement in an AUC is associated with high costs. The method involves training a complex multi-speaker S2S model to convert between input and output mel-spectrograms and requires training a Universal Vocoder model to convert a mel-spectrogram into a raw speech signal.

To better understand what prevents the model from achieving higher accuracy, we measure the performance of the model on synthetic pronunciation errors. We divide all synthetic pronunciation errors into four categories to reflect the severity of pronunciation errors. The 'low' category includes mispronounced words with only one mismatched phoneme between the canonical and pronounced phonemes of the word. The 'medium' category includes two mispronounced phonemes. The 'high' category gets three, and the 'very high' category includes four mispronounced errors. The AUC across different severity levels varies from 0.928 (low severity) to 1.00 (very high severity) as shown in Table 3.16. These AUC values are significantly higher than the results for non-native human speech, suggesting that making synthetic speech errors more similar to non-native speech may improve the accuracy of detecting pronunciation errors.

TABLE 3.16: Accuracy (AUC) in detecting pronunciation errors assessed in synthetic speech at different severity levels of mispronunciation for the best S2S method.

Severity	AUC
Low (phoneme distance=1)	0.928
Medium (phoneme distance=2)	0.974
High (phoneme distance=3)	0.993
Very High (phoneme distance=4)	1.00

3.5.5.2 Model of native speech pronunciation

Experimental setup

The P2P, T2S, and S2S are generative models that provide the probability of generating a particular output sequence. This probability can be used directly to detect pronunciation errors without generating the mispronounced speech and adding it to the training data. In this experiment, we show how to apply this approach in practice.

One of the challenges in detecting pronunciation errors is that a native speaker can pronounce a sentence correctly in many ways. The classic approach for detecting pronunciation errors is based on identifying the difference between pronounced and canonical phonemes. All pronunciations that do not correspond precisely to the canonical pronunciation will result in false pronunciation errors. One way to solve this problem is to use the P2P technique to create a native speech Pronunciation Model (PM) that determines the probability that a sentence is pronounced by a native speaker. A low likelihood value indicates a high probability of mispronunciation.

To evaluate the performance of the PM model, the pronunciation error detection model has been designed such that the PM model can be turned on and off. To disable the PM, we are modifying it so that it only takes into account one way of correctly pronouncing a sentence. In an ablation study, we measure whether the PM model improves the accuracy in detecting pronunciation errors at the word level. Note that in this experiment, synthetically generated pronunciation errors are not used explicitly. Instead, the native speech pronunciation model is used to implicitly represent the generative speech process.

Overview of the pronunciation error detection model

The design of the pronunciation error detection model consists of three subsystems: a Phoneme Recognizer (PR), a Pronunciation Model (PM), and a Pronunciation Error Detector (PED), shown in Figure 3.14. First, the PR model estimates a belief over the phonemes produced by the student, intuitively representing the uncertainty in the student's pronunciation. The PM model transforms this belief into a probability that a native speaker would pronounce the sentence this way, given the phonetic

variability. Finally, the PED model decides which words were mispronounced in the sentence by processing three pieces of information: a) what the student pronounced, b) how likely it is that the native speaker would pronounce it that way, and c) what the student was supposed to pronounce. Details of the entire model of pronunciation error detection are presented in Section 3 of our recent work (Korzekwa, Lorenzo-Trueba, Zaporowski, et al., 2021). We will now only show the details of the PM model that are relevant to this experiment.

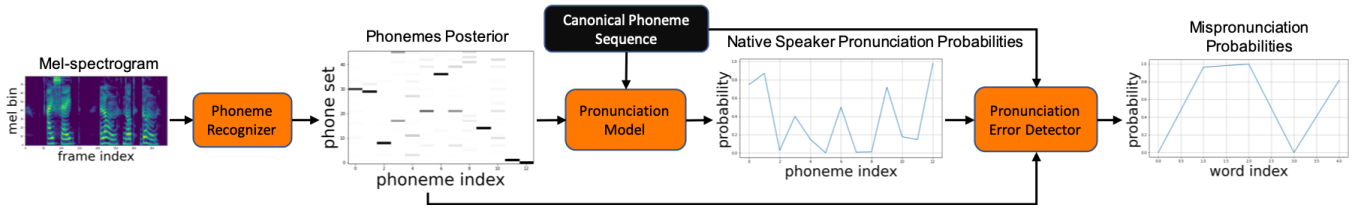


FIGURE 3.14: Architecture of the system for detecting mispronounced words in a spoken sentence based on the native speech pronunciation model.

Overview of the native speech pronunciation model

PM is an encoder-decoder neural network following Sutskever et al. (Sutskever et al., 2014). Instead of building a text-to-text translation system between two languages, we use it for the P2P conversion. The sequence of phonemes $\mathbf{r}$ that the native speaker was supposed to pronounce is converted to the sequence of phonemes $\mathbf{r}'$ they had pronounced, denoted as $\mathbf{r}' \sim p(\mathbf{r}'|\mathbf{r})$. Once trained, PM acts as a probability mass function, computing the probability sequence π of the recognized phonemes $\mathbf{r_o}$ pronounced by the student conditioned by the expected (canonical) phonemes $\mathbf{r}$. PM is denoted as in Eq. 3.14.

$$\pi = \sum_{\mathbf{r_o}} p(\mathbf{r_o}|\mathbf{o}) p(\mathbf{r}' = \mathbf{r_o}|\mathbf{r}) \tag{3.14}$$

The PM model is trained on P2P speech data generated automatically by passing the speech of the native speakers through the PR. By using PR to annotate the data, we can make the PM model more robust against possible phoneme recognition inaccuracies in PR at the time of testing.

Results

The complete model with PM enabled is called PR-PM that stands for a Phoneme Recognizer + Pronunciation Model. The model with PM turned off is called PR-LIK that stands for Phoneme Recognizer outputting the likelihoods of recognized phonemes. PR-LIK is an extension of the PR-NOLIK model – the mispronunciation detection model proposed by Leung et al. (Leung et al., 2019) that only returns the most likely recognized phonemes and does not use phoneme likelihoods to

detect pronunciation errors. PR-NOLIK detects mispronounced words based on the difference between the canonical and recognized phonemes. Therefore, this system does not offer any flexibility in optimizing the model for higher precision by fine-tuning the threshold applied to the phoneme recognition probabilities.

Turning off PM reduces the precision between 11% and 18%, depending on the decrease in recall between 20% to 40%, as shown in Figure 3.15. One example where the PM helps is the word 'enough' that can be pronounced in two similar ways: /ih n ah f/ or /ax n ah f/ (short 'i' or 'schwa' phoneme at the beginning.) The PM can take into account the phonetic variability and recognize both versions as correctly pronounced. Another example is coarticulation (Hieke, 1984). Native speakers tend to merge phonemes of adjacent words. For example, in the text 'her arrange' /hh er - er ey n jh/, two adjacent phonemes /er/ can be pronounced as one phoneme: /hh er ey n jh/. The PM model can correctly recognize multiple variations of such pronunciations.

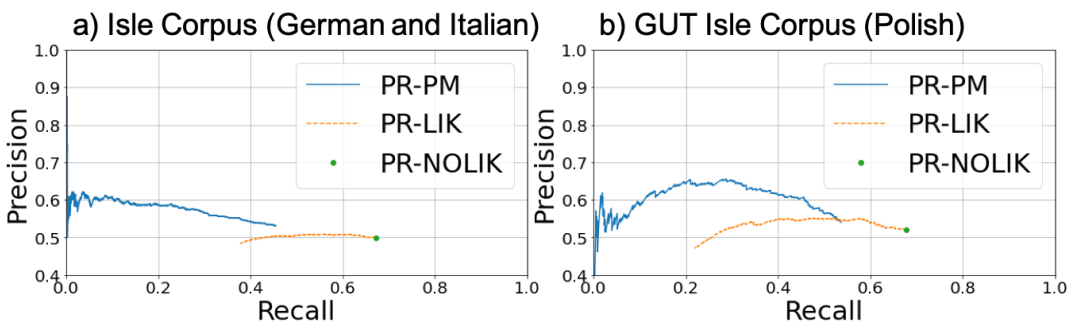


FIGURE 3.15: Precision-recall curves for the evaluated systems to measure the effect of using the PM model in detecting pronunciation errors. PR-PM - full model with the PM enabled. PR-LIK - the PR-PM model with the PM disabled. PR-NOLIK - non-probabilistic variant of the PR-LIK model proposed by Leung et al. (Leung et al., 2019).

Complementary to the precision-recall curve shown in Figure 3.15, we present in Table 3.17 one configuration of the precision and recall scores for the PR-LIK and PR-PM systems. This configuration is chosen in a way to: a) make the recall for both systems close to the same value, and b) to illustrate that the PR-PM model has much greater potential to increase precision than the PR-LIK system. A similar conclusion can be drawn by checking various different precision and recall configurations in the precision and recall plots for both Isle and GUT Isle corpora.

3.5.5.3 Lexical stress error detection

Experimental setup

The full CAPT learning experience includes both the detection of pronunciation and lexical stress errors. To investigate the potential of speech generation in the

TABLE 3.17: Precision and recall of detecting word-level mispronunciations. CI - Confidence Interval. PR-PM - full model with the PM enabled. PR-LIK - the PR-PM model with the PM disabled.

Model	Precision [%,95%CI]	Recall [%,95%CI]
Isle corpus (German and Italian)		
PR-LIK	49.39 (47.59-51.19)	40.20 (38.62-41.81)
PR-PM	54.20 (52.32-56.08)	40.20 (38.62-41.81)
GUT Isle corpus (Polish)		
PR-LIK	54.91 (50.53-59.24)	40.29 (36.66-44.02)
PR-PM	61.21 (56.63-65.65)	40.15 (36.51-43.87)

lexical stress error detection task, we evaluate the T2S method, which is a simpler version of the S2S method evaluated in Section 3.5.5.1.

The lexical stress error detection model is trained to measure the benefits of employing synthetic mispronounced speech. The first model, denoted as Att_TTS is based on an attention mechanism and is trained on both human and synthetic speech with pronunciation errors. In this model, 1980 the most popular English words (Michel et al., 2011) were synthesized with correct and incorrect stress patterns using the method outlined in Section 3.5.3.2, and added to the speech corpora of isolated words presented in Section 3.5.4.2. The Att_NoTTS model is trained only on human speech. Each of the two models presented has its simpler version without the attention mechanism, marked as NoAtt_TTS and NoAtt_NoTTS. Both models will help to understand whether the benefits of using synthetic pronunciation errors depend on the model capacity.

The accuracy of detecting lexical stress errors is measured in terms of an AUC metric. To be comparable to the study by Ferrer et al. (Ferrer et al., 2015), we use precision as an additional metric, while setting recall to 50%.

Overview of the lexical stress detection model

As shown in Figure 3.16, the lexical stress error detection model consists of three subsystems: Feature Extractor, Attention-based Classification Model, and Lexical Stress Error Detector. The Feature Extractor extracts prosodic features and phonemes from the speech signal **s** and the forced-aligned canonical phonemes **r**. Prosodic features include: F0, intensity [dB SPL] and duration of phonemes. The F0 and intensity features are computed at the frame level. The Attention-based Classification Model uses the attention mechanism (Vaswani, Shazeer, Parmar, Uszkoreit, Jones, Gomez, Ł. Kaiser, et al., 2017) to map frame-level and phoneme-level features to a syllable-level representation. It then produces lexical stress error probabilities at the syllable level. The Lexical Stress Error Detector reports a lexical stress error if the expected (canonical) and estimated lexical stress for a given syllable do not match and the corresponding probability is higher than the specified threshold. The detailed

architecture of the model is presented in Section 3 of our recent work (Korzekwa, Barra-Chicote, Zaporowski, et al., 2021).

The NoAtt_TTS and NoAtt_NoTTS models do not have the attention mechanism. Instead, as a representation at the syllable level, they use the average acoustic feature values for the corresponding syllable nucleus. The hypothesis is that synthetic data will not be beneficial to a simpler model due to its limited capacity.

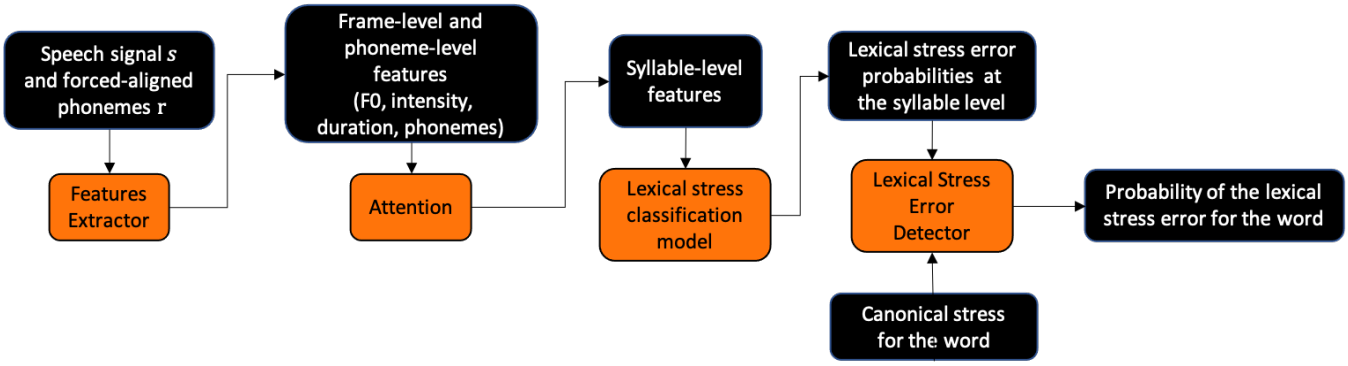


FIGURE 3.16: Attention-based model for the detection of lexical stress errors.

Results

Enriching the training set with the incorrectly stressed words increases an AUC score from 0.54 to 0.62 (Att_TTS vs. Att_NoTTS in Figure 3.17 and Table 3.18). Data augmentation helps because it increases the number of words with incorrect stress patterns in the training set. This prevents the model from using the strong correlation between phonemes and lexical stress in the correctly stressed words. Using data augmentation in the simpler model without the attention mechanism slightly reduced an AUC score from 0.45 to 0.44 (NoAtt_NoTTS vs NoAtt_TTS). The NoAtt_TTS model has limited capacity due to not using the attention mechanism to model prosodic features, and thus is unable to benefit from synthetic speech.

We compare our results with the work of Ferrer et al. (Ferrer et al., 2015). There were 46.4% (191 out of 411) of incorrectly stressed words in their corpus, well over 9.4% (189 out of 2109) words in our experiment. The fewer lexical stress errors that users make, the more difficult it is to detect them. Under these conditions, we can state that our lexical stress detection model based on T2S generated synthetic speech achieves higher scores in precision and recall compared to the work of Ferrer et al. (Ferrer et al., 2015).

3.5.6 Conclusions

We propose a new paradigm for detecting pronunciation errors in non-native speech. Rather than focusing on detecting pronunciation errors directly, we reformulate the detection problem as a speech generation task. This approach is based on the

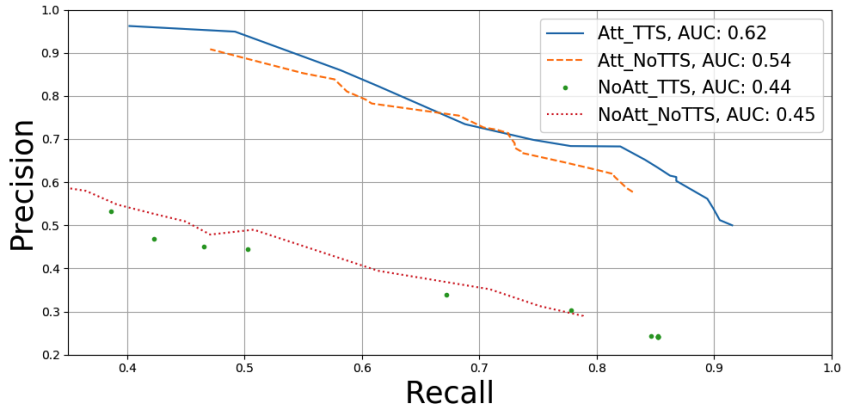


FIGURE 3.17: Precision-recall curves for lexical stress error detection models.

TABLE 3.18: AUC, precision and recall [%, 95% Confidence Interval] metrics for lexical stress error detection models. Att. - Model with attention. Syn. - Synthetic mispronunciations.

Model	**Att.**	**Syn.**	**AUC**	**Precision [%]**	**Recall[%]**
Att_TTS	yes	yes	0.62	94.8 (89.18-98.03)	49.2 (42.13-56.3)
Att_NoTTS	yes	no	0.54	87.85 (80.67-93.02)	49.74 (42.66-56.82)
NoAtt_TTS	no	yes	0.44	44.39 (37.85-51.09)	50.26 (43.18-57.34)
NoAtt_NoTTS	no	no	0.45	48.98 (42.04-55.95)	50.79 (43.70-57.86)
(Ferrer et al., 2015)	na	na	na	95.00 (na-na)	48.3 (na-na)

assumption that it is easier to generate speech with specific characteristics than to detect those characteristics in speech with limited availability. In this way, we address one of the main problems of the existing CAPT methods, which is the low availability of mispronounced speech for reliable training of pronunciation error detection models.

We present a unified look at three different speech generation techniques for detecting pronunciation errors based on P2P, T2S and S2S conversion. The P2P, T2S, and S2S methods improve the accuracy of detecting pronunciation and lexical stress errors. The methods outperform strong baseline models and establish a new state-of-the-art. The best S2S method outperforms the baseline method (Leung et al., 2019) by improving the accuracy of detecting pronunciation errors in AUC metric by 41% from 0.528 to 0.749. The S2S method has the ability to control many properties of speech, such as voice timbre, prosody (duration), and pronunciation. This opens the door to the generation of mispronounced speech that can mimic certain aspects of non-native speech, such as voice timbre. The S2S method can be seen as a generalization of the simpler methods, T2S and P2P, providing a general framework for building a first-class models of pronunciation assessment. For better reproducibility, in addition

to using publicly available speech corpora, we recorded the GUT Isle corpus of non-native English speech (Weber et al., 2020). The corpus is available to other researchers in the field.

In the future, we plan to extend the S2S method in order to generate synthetic speech as close as possible to non-native speech: a) we will extract the voice timbre from the speech of non-native speakers and transfer it to native speech, following the paper of Merritt et al. on text-free voice conversion (Merritt, Ezzerg, et al., 2022), and b) we will mimic the distribution of pronunciation errors in non-native speech. We expect both changes to increase the accuracy of detecting pronunciation errors in non-native speech. In the long run, we hope to demonstrate that "synthetic speech is all you need" by training the model with synthetic speech only and achieving state-of-the-art results in the pronunciation error detection task. This may revolutionize computer-assisted English L2 learning and CAPT. Moreover, such a paradigm may be transferred to the whole domain of computer-assisted foreign language learning.

Chapter 4

Generalization of deep learning methods for pronunciation error detection

In this section, we explore the generalization capabilities of deep learning methods for pronunciation error detection. For this purpose, the following secondary research thesis has been formulated:

Deep learning methods for the detection of pronunciation errors in non-native speech are transferable to the related tasks of detection and reconstruction of dysarthric speech.

The first task related to pronunciation error detection is the detection of dysarthric speech. For this purpose, generalization capabilities of the attention mechanism and the multi-task deep learning techniques are investigated.

The reconstruction of dysarthric speech was selected for the second related task. Reconstructing dysarthric speech and generating synthetic pronunciation errors are examples of speech-to-speech deep learning methods, therefore, similar deep learning techniques may perform well in both scenarios.

The research on both topics, detection and reconstruction of dysarthric speech, resulted in a publication at the Interspeech 2019 conference, which is presented in this chapter.

Daniel Korzekwa, Roberto Barra-Chicote, Bozena Kostek, Thomas Drugman, Mateusz Lajszczak, Interpretable deep learning model for the detection and reconstruction of dysarthric speech, Interspeech, 2019

Abstract

We present a novel deep learning model for the detection and reconstruction of dysarthric speech. We train the model with a multi-task learning technique to jointly solve dysarthria detection and speech reconstruction tasks. The model key feature is a low-dimensional latent space that is meant to encode the properties of dysarthric

speech. It is commonly believed that neural networks are "black boxes" that solve problems but do not provide interpretable outputs. On the contrary, we show that this latent space successfully encodes interpretable characteristics of dysarthria, is effective at detecting dysarthria, and that manipulation of the latent space allows the model to reconstruct healthy speech from dysarthric speech. This work can help patients and speech pathologists to improve their understanding of the condition, lead to more accurate diagnoses and aid in reconstructing healthy speech for afflicted patients.

4.1 Introduction

Dysarthria is a motor speech disorder manifesting itself by a weakness of muscles controlled by the brain and nervous system that are used in the process of speech production, such as lips, jaw and throat (ASHA, 2018). Patients with dysarthria produce harsh and breathy speech with abnormal prosodic patterns, such as very low speech rate or flat intonation, which makes their speech unnatural and difficult to comprehend. Damage to the nervous system is the main cause of dysarthria (ASHA, 2018). It can happen as an effect of multiple possible neurological disorders such as cerebral palsy, brain stroke, dementia or brain cyst (M. L. Cuny et al., 2017; Banovic, L. Zunic, et al., 2018).

Early onset detection of dysarthria may improve the quality of life for people affected by these neurological disorders. According to Alzheimer's Research UK2015 (Alzheimersresearchuk, 2015), 1 out of 3 people in the UK born in 2015 will develop dementia in their life. Manual detection of dysarthria conducted in clinical conditions by speech pathologists is costly, time-consuming and can lead to an incorrect diagnosis (Yamagishi et al., 2012; Carmichael et al., 2008). With an automated analysis of speech, we can detect an early onset of dysarthria and recommend further health checks with a clinician even when a human speech pathologist is not available. Speech reconstruction may help with better identification of the symptoms and enable patients with severe dysarthria to communicate with other people.

Section 2 presents related work. In Section 3 we describe the proposed model for detection and reconstruction of dysarthria. In Section 4 we demonstrate the performance of the model with experiments on detection, interpretability, and reconstruction of healthy speech from dysarthric speech. We conclude with our remarks.

4.2 Related work

4.2.1 Dysarthria detection

Deep neural networks can automatically detect dysarthric patterns without any prior expert knowledge (Krishna, 2018; Vásquez-Correa et al., 2018). Unfortunately, these models are difficult to interpret because they are usually composed of multiple layers

producing multidimensional outputs with an arbitrary meaning and representation. Contrarily, statistical models based on a fixed vector of handcrafted prosodic and spectral features such as jitter, shimmer, Noise to Harmonic Ratio (NHR) or Mel-Frequency Cepstral Coefficients (MFCC) offer good interpretability but require experts to manually design predictor features (Falk et al., 2012; Sarria-Paja et al., 2012; Gillespie et al., 2017; Lansford et al., 2014).

The work of Tu Ming et al. on interpretable objective evaluation of dysarthria (Tu et al., 2017) is the closest we found to our proposal. The main difference is that our model not only provides interpretable characteristics of dysarthria but also reconstructs healthy speech. Their model is based on feed-forward deep neural networks with a latent layer representing four dimensions of dysarthria: nasality, vocal quality, articulatory precision, and prosody. The final output of the network represents general dysarthria severity on a scale from 1 to 7. The input to this model is described by a 1201-dimensional vector of spectral and cepstral features that capture various aspects of dysarthric speech such as rhythm, glottal movement or formants. As opposed to this work, we use only mel-spectrograms to present the input speech to the model. Similarly to our approach, Vasquez-Correa et al. (Vásquez-Correa et al., 2018) uses a mel-spectrogram representation for dysarthria detection. However, they use 160 ms long time windows at the transition points between voiced and unvoiced speech segments, in contrast to using a full mel-spectrogram in our approach.

4.2.2 Speech reconstruction

There are three different approaches to the reconstruction of dysarthric speech: voice banking, voice adaptation and voice reconstruction (Yamagishi et al., 2012). Voice banking is a simple idea of collecting a patient's speech samples before their speech becomes unintelligible and using it to build a personalized Text-To-Speech (TTS) voice. It requires about 1800 utterances for a basic unit-selection TTS technology (Modeltalker, n.d.) and more than 5K utterances for building a Neural TTS voice (Latorre, Lachowicz, Lorenzo-Trueba, Merritt, Drugman, Ronanki, and Viacheslav, 2018). Voice adaptation requires as little as 7 minutes of recordings. In this approach, we start with a TTS model of an average speaker and adapt its acoustic and articulatory parameters to the target speaker (Ahmad Khan et al., 2011).

Both voice banking and voice adaptation techniques rely on the availability of recordings for a healthy speaker. The voice reconstruction technique overcomes this shortcoming. This technique aims at restoring damaged speech by tuning parameters representing the glottal source and the vocal tract filter (Rabiner et al., 1978; Drugman et al., 2014). In our model, we take a similar approach. However, instead of making assumptions on what parameters should be restored, we let the model automatically learn the best dimensions of the latent space that are responsible for dysarthric speech. Reconstruction of healthy speech by manipulating the latent space of a dysarthric speech is a promising direction, however, so far we only managed to successfully apply this technique in a single-speaker setup.

Variational Auto-Encoder (VAE) (Doersch, 2016) is a probabilistic latent space model that has recently become popular for the reconstruction of various signals such as text (Hu et al., 2017; Bowman et al., 2015) and speech (Y.-J. Zhang et al., 2018; Hsu et al., 2017).

4.3 Proposed model

The model consists of two output networks, jointly trained, with a shared encoder as shown in Figure 4.1. The audio and text encoders produce a low-dimensional dysarthric latent space and a sequential encoding of the input text. The audio decoder reconstructs input mel-spectrogram from a dysarthric latent space and encoded text. Logistic classification model predicts the probability of dysarthric speech from the dysarthric latent space. In Table 4.1 we present the details of various neural blocks used in the model.

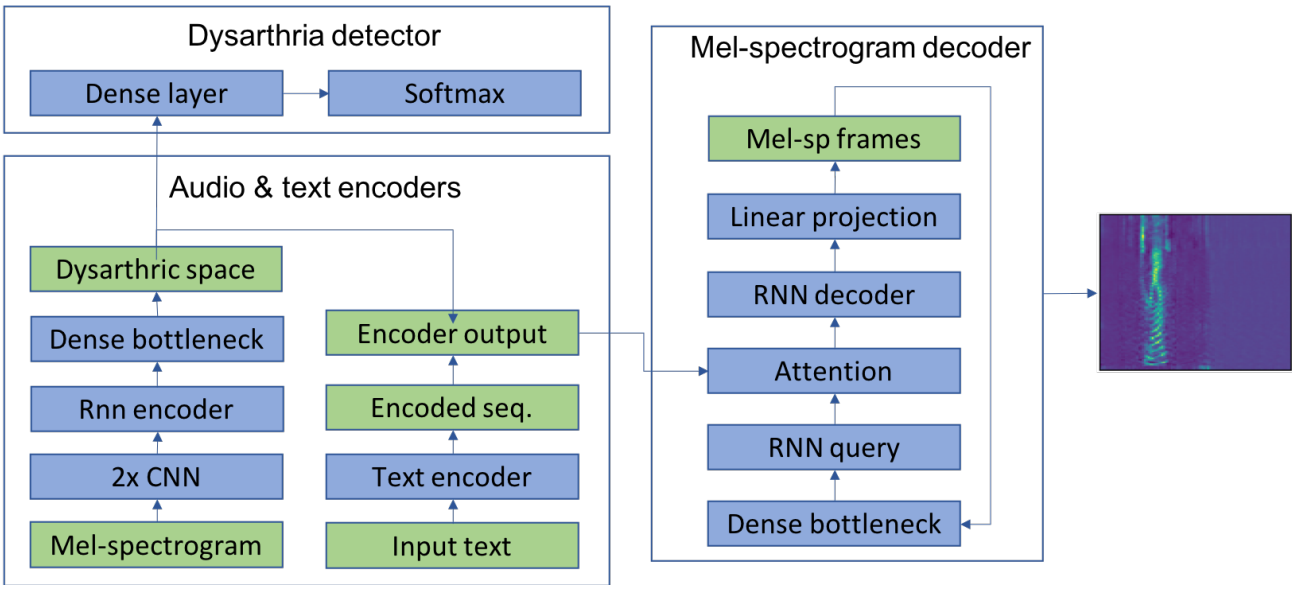


FIGURE 4.1: Architecture of deep learning model for detection and reconstruction of dysarthric speech.

Let us define a matrix $X : [n_{mels}, n_f]$ representing a mel-spectrogram (frame length=50ms and frame shift=12.5ms), where $n_{mels} = 128$ is the number of mel-frequency bands and n_f is the number of frames. Let us define a matrix $T : [n_c, n_t]$ representing a one-hot encoded input text, where n_c is the number of unique characters in the alphabet and n_t is the number of characters in the input text. The mel-spectrogram X is encoded into 2-dimensional dysarthria latent space $\mathbf{l} = \{l_1, l_2\}$ and then used as a conditioning variable for estimating the probability of dysarthria $d \sim p(d|X, \theta)$ and reconstructing the mel-spectrogram $Y \sim p(Y|X, T, \theta)$. Limiting the latent space to 2 dimensions makes the model more resilient to overfitting. The theta is a vector of trainable parameters of the model.

TABLE 4.1: Configuration of the neural network blocks.

Neural block	Config
	Audio encoder
2x CNN	20 channels, 5x5 kernel, RELU, VALID
GRU	20 hidden states, 1 layer
Dense	20 units, tanh
Dysarthric space	2 units, linear
	Text encoder
3x CNN	40 channels, 5x5 kernel, RELU, SAME
GRU	27 hidden states, 1 layer
	Audio decoder
Dense bottleneck	96 units, RELU
GRU query	29 hidden states, 1 layer
GRU decoder	128 hidden states, 1 layer
Linear projection	frames_num x melsp bins units, linear

Let us define a training set of m tuples of $((X, T), y)$, where $y \in \{0, 1\}$ is the label for normal/dysarthric speech and m is the number of speech mel-spectrograms for dysarthric and normal speakers. We optimize a joint cost of the predicted probability of dysarthria and mel-spectrogram reconstruction defined as a weighted function:

$$\sum_{i=1}^{m} \alpha log(p(d_i|X_i, \theta)) + (1 - \alpha) log(p(Y_i|X_i, T_i, \theta)) \tag{4.1}$$

where $log(p(d_i|X_i, \theta))$ is the cross-entropy between the predicted and actual labels of dysarthria, and $log(p(Y_i|X_i, T_i, \theta))$ is the log-likelihood of a Gaussian distribution for the predicted mel-spectrogram with a unit variance, a.k.a L2 loss. We used backpropagation and mini-batch stochastic gradient descent with a learning rate of 0.03 and a batch size of 50. The whole model is initialized with Xavier's method (Glorot et al., 2010) using the magnitude value of 2.24. Hyper-parameters of the model presented in Table 4.1 were tuned with a grid search optimization. We used MxNet framework for implementing the model (T. Chen et al., 2015).

4.3.1 Mel-spectrogram and text encoders

For the spectrogram encoder, we use a Recurrent Convolutional Neural Network model (RCNN) (R. J. Skerry-Ryan et al., 2018). The convolutional layers, each followed by a max-pooling layer, extract local and time-invariant patterns of the glottal source and the vocal tract. The GRU layer models temporal patterns of dysarthric speech (Cho, Merrienboer, et al., 2014). The last state of the GRU layer is processed by two dense layers. Dropout (Srivastava et al., 2014) with probability of 0.5 is applied to the output of the activations for both CNN layers, GRU layer, and the dense layer.

Text encoder encodes the input text using one-hot encoding, followed by three CNN layers and one GRU layer. Outputs of both audio and text encoders are concatenated via matrix broadcasting, producing a matrix $E : [n_c + n_l, n_t]$, where n_l is dimensionality of the dysarthria latent space.

4.3.2 Spectrogram decoder and dysarthria detector

For decoding a mel-spectrogram, similarly to Wang et al. (Y. Wang, R. J. Skerry-Ryan, et al., 2017), we use a Recurrent Neural Network (RNN) model with attention. The dot-product attention mechanism (Vaswani, Shazeer, Parmar, Uszkoreit, Jones, Gomez, L. Kaiser, et al., 2017) plays a crucial role. It informs to which elements of the encoder output the decoder should pay attention at every decoder step. The RNN network that produces a query vector for the attention, takes as input r predicted mel-spectrogram frames from the previous time-step. The output of the RNN decoder is projected via a linear dense layer into r number of mel-spectrogram frames. Similarly to Wang et al. (Y. Wang, R. J. Skerry-Ryan, et al., 2017), we found that it is important to preprocess the mel-spectrogram with a dense layer and dropout regularization to improve the overall generalization of the model.

The dysarthria detector is created from a 2-dimensional dense layer. It uses a tanh activation followed by a softmax function that represents the probability of dysarthric speech.

4.4 Experiments

4.4.1 Dysarthric speech database

There is no well-established benchmark in the literature to compare different models for detecting dysarthria. Aside from the most popular dysarthric corpora, UA-Speech (Kim et al., 2008) and TORGO (Rudzicz et al., 2012), there are multiple speech databases created for the purpose of a specific study, for example, corpora of 57 dysarthric speakers (Lansford et al., 2014) and Enderby Frenchay Assessment dataset (Carmichael et al., 2008). Many corpora, including TORGO and HomeService (Nicolao, Christensen, et al., 2016), are available under non-commercial license.

In our experiments we use the UA-Speech database from the University of Illinois (Kim et al., 2008). It contains 11 male and 4 female dysarthric speakers of different dysarthria severity levels and 13 control speakers. 455 isolated words are recorded for each speaker with 1 to 3 repetitions. Every word is recorded through a 7-channel microphone array, producing a separate wav file of 16 kHz sampling rate for every channel. It contains 9.4 hours of speech for dysarthric speakers and 4.85 hours for control speakers. UA-Speech corpus comes with intelligibility scores that are obtained from a transcription task performed by 5 naive listeners.

To control variabilities in recording conditions, we normalized mel-spectrograms for every recorded word independently with a z-score normalization. We considered

removing the initial period of silence at the beginning of recorded words but we decided against it. We found that for dysarthric speakers of high speech intelligibility, the average length of the initial silence period that lasts 0.569sec +- 0.04674 (99% CI) is comparable with healthy speakers with the length of 0.532sec +- 0.055. Because we can predict unvoiced periods with merely 85% of accuracy (Johnston et al., 2012), removing the periods of silence for dysarthric speakers with poor intelligibility is very inaccurate.

4.4.2 Automatic detection of dysarthria

To define the training and test sets, we use a Leave-One-Subject-Out (LOSO) cross-validation scheme. For each training, we include all speakers but one that is left out to measure the prediction accuracy on unseen examples. The accuracy, precision and recall metrics are computed at a speaker level (the average dysarthria probability of all the words produced by the speaker is compared to a target speaker dysarthria label $\in \{0, 1\}$), and a word level (comparing target dysarthria label with predicted dysarthria probability for all words independently).

As a baseline, we use the Gillespie's et al. model that is based on Support Vector Machine classifier (Gillespie et al., 2017). It uses 1595 low-level predictor features processed with a global z-score normalization. It reports a 75.3 and 92.9 accuracy in the dysarthria detection task at the word and speaker levels respectively, following LOSO cross-validation. However, Gillespie uses 336 words from the UA-Speech corpus with 12 words per speaker, whereas we use all 455 words across all speakers.

In our first model, only dysarthric labels are observed and we achieved an accuracy on the word and speaker levels of 82% and 93% respectively. By training the multi-task model, in which both targets, i.e. mel-spectrogram and dysarthric labels, are observed, the accuracy on the word level increased by 3 percents to the value of 85.3% (Table 4.2). We found that the UA-Speech database includes multiple recorded words for healthy speakers that contain intelligibility errors, different words than asked or background speech of other people. These issues affect the accuracy of detecting dysarthric speech.

Krishna reports a 97.5% accuracy on UA-Corpus (Krishna, 2018). However, after email clarification with the author, we found that they estimated the accuracy taking into account only the speakers with a medium level of dysarthria. Narendra et al. achieved 93.06% utterance level accuracy on the TORGO dysarthric speech database (Narendra et al., 2018). As opposed to the related work, our model does not need any expert knowledge to design hand-crafted features and it can learn automatically using a low-dimensional latent space that encodes characteristics of dysarthria.

4.4.3 Interpretable modeling of dysarthric patterns

We analyze the correlation between the dysarthric latent space and the intelligibility of speakers. We look at 550 audio samples of a single 'Command' word across the 15

TABLE 4.2: Accuracy of dysarthria detection including 95% CI. Classifier task - target mel-spectrogram (ML) is not observed during training. Multitask - both targets ML and dysarthric labels are observed

System	Accuracy	Precision	Recall
Word level			
Multitask	0.853 (0.849 - 0.857)	0.831	0.911
Classifier task	0.820 (0.815 - 0.824)	0.818	0.855
Gillespie et al.(Gillespie et al., 2017)	0.753 (na)	0.823	0.728
Speaker level			
Multitask	0.929 (0.790-0.984)	1.000	0.867
Classifier task	0.929 (0.790-0.984)	0.933	0.933
Gillespie et al.(Gillespie et al., 2017)	0.929 (na)	na	na

dysarthric speakers and 13 healthy speakers.

In an unsupervised training (Figure 4.2), target labels of dysarthric/normal speech are not presented to the model. Dysarthric speakers are well separated from normal speakers and the dimension 2 of the latent space is negatively correlated with the intelligibility scores (Pearson correlation of -0.84, two-sided p-value < 0.001). In a supervised variant (Figure 4.3), we train the model jointly with both reconstructed mel-spectrogram and the target dysarthria labels observed. Both dimensions of the latent space are highly correlated with the intelligibility scores (dimension 1 with correlation of -0.76 and dimension 2 with correlation of 0.70, both with p-value < 0.001).

The sign of the correlation has no particular meaning. Retraining the model multiple times results in both positive and negative correlations between the latent space and the intelligibility of speech. A high correlation between dysarthric latent space and intelligibility scores suggests that by moving along the dimensions of the latent space, we should be able to reconstruct speech of dysarthric speakers and improve its intelligibility. We explore this in the next experiment.

4.4.4 Reconstruction of dysarthric speech

First we trained a supervised multi-speaker model with all dysarthric and control speakers but we achieved poor reconstruction results with almost unintelligible speech. We think this is due to a high variability of dysarthric speech across all speakers, including various articulation, prosody and fluency problems. To better understand the potential for speech reconstruction, we narrowed the experiment down to two speakers, male speaker M05 and a corresponding control speaker. We have chosen M05 subject because their speech varies across different levels of fluency and we wanted to observe this pattern when manipulating the latent space. For example, when pronouncing the word 'backspace', M05 uttered consonants 'b' and 's' multiple times, resulting in 'ba ba cs space'.

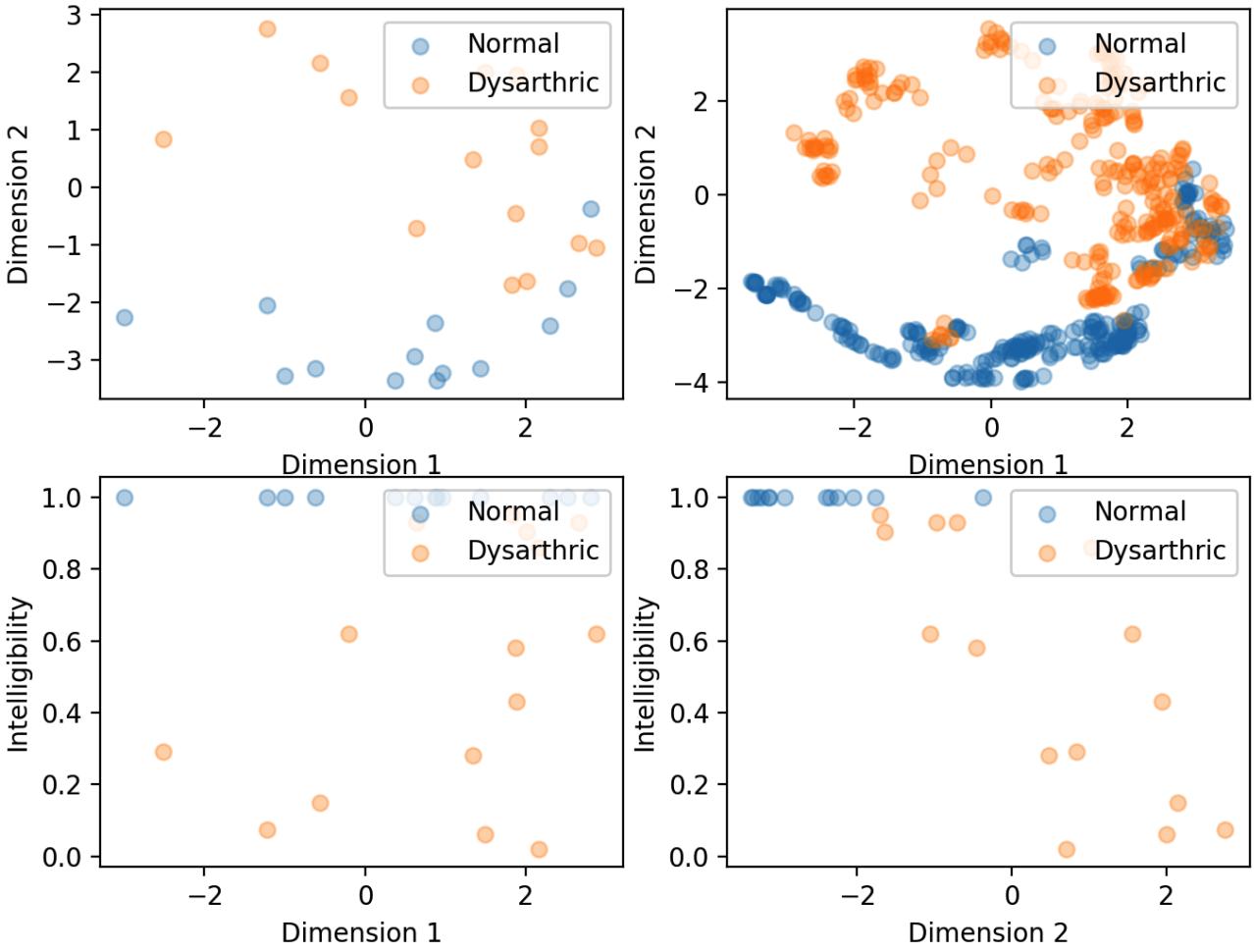


FIGURE 4.2: Unsupervised learning. Top row: Separation between dysarthric and control speakers in the latent space on a speaker (left) and word (right) level. Bottom row: Correlation between both dimensions of the latent space and the intelligibility scores.

We analyzed a single category of 19 computer command words, such as 'command' or 'backspace'. For every word uttered by M05, we generated 5 different versions of speech, fixing dimension 2 of the latent space to the value of -0.1, and using the values of [-0.5, 0, 0.5, 1, 1.5] for dimension 1. Audio samples of reconstructed speech were obtained by converting predicted mel-spectrograms to waveforms using the Griffin-Lim algorithm (Griffin et al., 1984).

We conducted MUSHRA perceptual test (Merritt, Putrycz, et al., 2018). Every listener was presented with 6 versions of a given word at the same time, 5 reconstructions and one version of recorded speech. We asked listeners to evaluate the fluency of speech on a scale from 0 to 100. We used 10 US based listeners from the Amazon mTurk platform, in total providing us with 1140 evaluated speech samples.

As shown in Figure 4.4, by moving along dimension 1 of the latent space, we can improve the fluency of speech, generating speech with levels of fluency not observed in the training data. In the pairwise two-sided Wilcoxon signed-rank, all

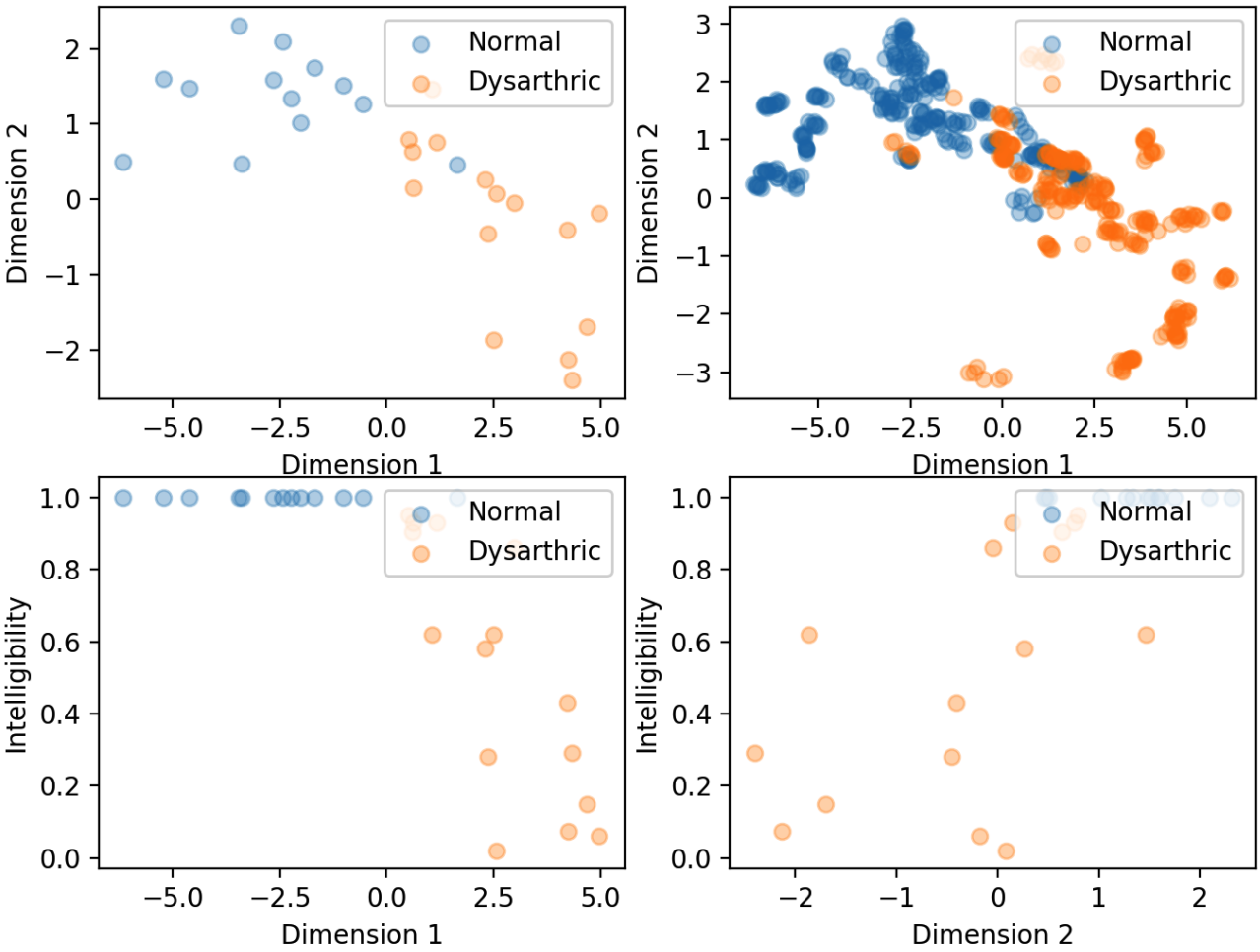


FIGURE 4.3: Supervised learning. As in Figure 4.2.

pairs of ranks are different from each other with p-value < 0.001, except of {orig, d1=1.0}, {d1=-0.5, d1=0.0}, {d1=-0.5, d1=0.5}. Examples of original and reconstructed mel-spectrograms are shown in Figure 4.5.

We found that manipulation of the latent space changes both the fluency of speech and the timbre of voice and it is possible that dysarthria is so tied up with speaker identify making it fruitless to disentangle them. We replaced a deterministic dysarthric latent space with a Gaussian variable and trained the model with an additional Kullback-Leibler loss (Doersch, 2016; Mathieu et al., 2018) but we did not manage to separate the timbre of voice from dysarthria. Training the model with an additional discriminative cost to ensure that every dimension of the latent space is directly associated with a particular speech factor can potentially help with this problem (Hu et al., 2017).

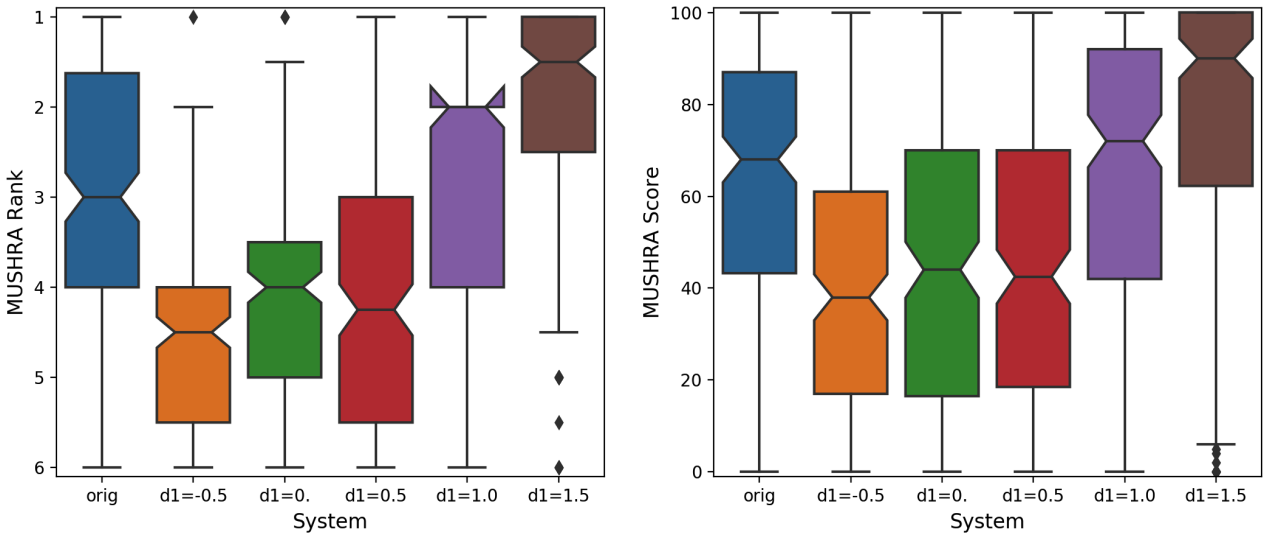


FIGURE 4.4: MUSHRA results for the fluency of speech for 5 reconstructions and one recorded speech. Rank order (left) and the median score on the scale from 0 to 100 (right).

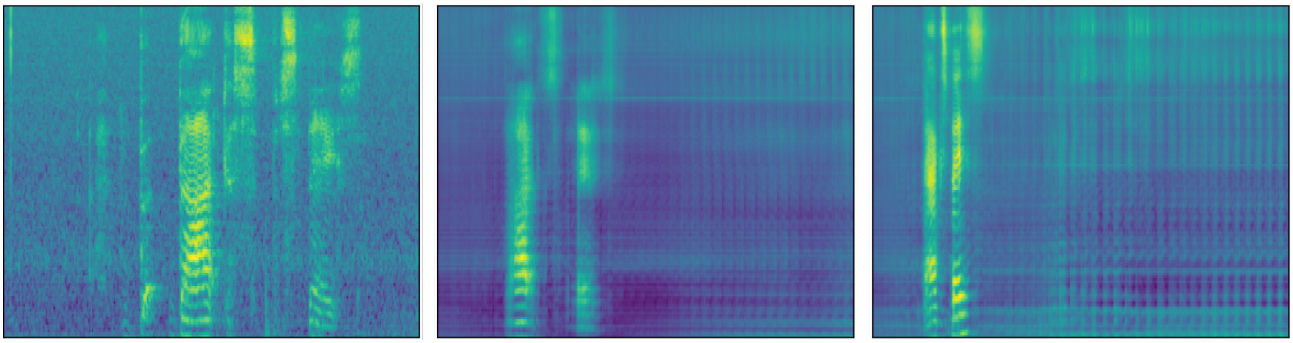


FIGURE 4.5: Reconstruction of dysarthric speech ('command' word). From left to right (MUSHRA scores of 51.8, 61.9 and 89.5): Recorded dysarthric speech. Reconstructed speech with dimension 1 of 0.0 and 1.5 respectively.

4.5 Conclusions

This paper proposed a novel approach for the detection and reconstruction of dysarthric speech. The encoder-decoder model factorizes speech into a low-dimensional latent space and encoding of the input text. We showed that the latent space conveys interpretable characteristics of dysarthria, such as intelligibility and fluency of speech. MUSHRA perceptual test demonstrated that the adaptation of the latent space let the model generate speech of improved fluency. The multi-task supervised approach for predicting both the probability of dysarthric speech and the mel-spectrogram helps improve the detection of dysarthria with higher accuracy. This is thanks to a low-dimensional latent space of the auto-encoder as opposed to directly predicting dysarthria from a highly dimensional mel-spectrogram.

4.6 Acknowledgements

We would like to thank A. Nadolski, J. Droppo, J. Rohnke and V. Klimkov for insightful discussions on this work.

Chapter 5

Conclusions

5.1 Summary

Within the research carried out in the framework of the Ph.D. work, novel deep learning methods were developed to detect pronunciation errors in non-native English speech automatically. Detecting pronunciation errors is part of CAPT that enables people to learn foreign languages without the assistance of a language teacher. As already mentioned, regarding the UNESCO report, 40% of the world's population does not have access to education in a language they understand, so there is a great potential for the new CAPT methods to make education more accessible to people all over the world.

Existing CAPT methods based on deep learning cannot detect pronunciation errors with high accuracy. The best method proposed in this Ph.D. research improves the accuracy of detecting pronunciation errors in the AUC metric by 41%, from 0.528 to 0.749, compared to the state-of-the-art approach (Korzekwa, Lorenzo-Trueba, Drugman, and Kostek, 2022). This improvement corresponds to 80.45% precision and 40.12% recall. Taking into account only severe pronunciation errors, the AUC metric raises from 0.749 to 0.834, corresponding to 93.54% precision and 40.15% recall. These achievements successfully validate the primary research thesis:

It is possible to improve the accuracy of deep learning methods for detecting pronunciation errors in non-native English by employing synthetic speech generation and end-to-end modeling techniques that reduce the need for phonetically transcribed mispronounced speech.

Extensive experiments have been conducted to measure the effectiveness of the proposed methods in CAPT. Deep learning models were developed and assessed to detect both pronunciation and lexical stress errors. Non-native speech of German, Italian and Polish speakers were used in the evaluations. As part of the doctoral research, two speech corpora of non-native Slavic and Baltic speakers have been recorded (Weber et al., 2020).

To investigate generalization capabilities, the developed deep learning techniques for detecting pronunciation errors were successfully applied to the related areas of detection and reconstruction of dysarthric speech (Korzekwa, Barra-Chicote, Kostek,

et al., 2019). The auto-encoder model was proposed to factorize dysarthric speech into a low-level latent representation. By controlling the latent representation, the fluency of the output speech can be improved, as shown in the MUSHRA perceptual speech test. In addition, the latent presentation can be used to detect dysarthric speech at the word level with 83.1% precision and 91.1% recall metrics. The new deep learning techniques applied to the topic of dysarthric speech successfully prove the secondary research thesis:

Deep learning methods for the detection of pronunciation errors in non-native speech are transferable to the related tasks of detection and reconstruction of dysarthric speech.

5.2 Novelty

Many important observations have been made on existing state-of-the-art methods, which led to the development of novel techniques for detecting pronunciation errors.

Performing phonetic transcription of non-native speech is time-consuming, and sometimes, transcription is impossible due to differences between spoken languages. A new method of detecting pronunciation errors called WEAKLY-S (Weakly-supervised) has been proposed, which does not require phonetic transcriptions of non-native speech (Korzekwa, Lorenzo-Trueba, Drugman, Calamaro, et al., 2021).

State-of-the-art methods align the canonical and recognized phoneme sequences to identify mispronounced speech segments such as phonemes and words. Any inaccuracies introduced in the alignment process would lower the accuracy of detecting pronunciation errors. As part of the WEAKLY-S model, a new end-2-end method has been proposed to directly detect pronunciation errors at the word level without having to align with canonical and recognized phoneme sequences (Korzekwa, Lorenzo-Trueba, Drugman, Calamaro, et al., 2021). The WEAKLY-S model increases the accuracy of detecting pronunciation errors in the AUC metric by up to 30% compared to the state-of-the-art approach.

There are two sources of variability and uncertainty that can affect the accuracy of detecting pronunciation errors. First, the same sentence can be pronounced in multiple correct ways, which should not trigger a pronunciation error. Second, it is challenging to recognize phonemes pronounced by the speaker accurately, and this ubiquitous uncertainty has to be accounted for. A new method has been proposed to this end, accounting for: i) multiple correct ways of pronouncing the same sentence and ii) uncertainty in phoneme recognition (Korzekwa, Lorenzo-Trueba, Zaporowski, et al., 2021). The proposed method increases the precision of detecting mispronunciations by up to 18% compared to the state-of-the-art approach.

Existing methods of detecting pronunciation errors often rely on hand-crafted speech features such as f0, energy, and phoneme alignment. A new method based on the attention mechanism has been proposed to automatically extract optimal speech

features from a speech signal (Korzekwa, Barra-Chicote, Zaporowski, et al., 2021). The method introduced plays a vital role in all proposed deep learning models in detecting pronunciation and lexical stress errors.

The attention mechanism helps factorize a black-box deep learning model into multiple dependent components. Factorization leads to better interpretability of the model, e.g., visualizing the attention of the model for detecting lexical stress errors (Korzekwa, Barra-Chicote, Zaporowski, et al., 2021). Multi-task learning is a type of model factorization that can make a deep learning model more robust and less prone to overfitting (Korzekwa, Lorenzo-Trueba, Drugman, Calamaro, et al., 2021). Training the proposed multi-task WEAKLY-S pronunciation error detection model with two tasks, phoneme recognizer and pronunciation error detector, increase the accuracy of detecting pronunciation errors. Factorization can also take the form of an interpretable bottleneck layer that can be used to modify specific characteristics of the signal, e.g., make dysarthric speech more fluent and intelligible (Korzekwa, Barra-Chicote, Kostek, et al., 2019).

There is limited availability of non-native speech that is time-consuming to collect and difficult to annotate with phonetic transcriptions. Resorting to the probability theory and Bayes-rule, the problem of pronunciation error detection is reformulated as a speech generation task. Intuitively, if we had an unlimited amount of synthetic speech that could mimic non-native human speech, deep learning models for detecting pronunciation errors would be less prone to overfitting. The best proposed speech-to-speech generation method for generating mispronounced speech increases the accuracy of detecting pronunciation errors in the AUC metric by 41%, from 0.528 to 0.749, compared to the state-of-the-art approach (Korzekwa, Lorenzo-Trueba, Drugman, and Kostek, 2022).

The experiments carried out to investigate the performance of the proposed approaches supported research theses no. 1 and no. 2 of this doctoral dissertation. In summary, the following major original contributions were introduced in this Ph.D. dissertation:

1. To reduce the need for phonetically transcribed non-native speech, the problem of pronunciation error detection has been reformulated as a speech generation task (Korzekwa, Lorenzo-Trueba, Drugman, and Kostek, 2022), which enables to generate synthetic mispronounced speech.

2. To eliminate the need to align canonical and recognized phoneme sequences and not rely on transcribed non-native speech, a novel end-to-end multi-task technique to directly detect pronunciation errors was proposed, called WEAKLY-S (Weakly-supervised) (Korzekwa, Lorenzo-Trueba, Drugman, Calamaro, et al., 2021).

3. To take into account the variability of pronunciation and the uncertainty in phoneme recognition while recognizing pronunciation errors, a new probabilistic deep learning architecture was proposed (Korzekwa, Lorenzo-Trueba,

Zaporowski, et al., 2021).

4. To automatically extract speech features in the pronunciation (Korzekwa, Lorenzo-Trueba, Drugman, Calamaro, et al., 2021) and lexical stress (Korzekwa, Barra-Chicote, Zaporowski, et al., 2021) error detection tasks, the attention mechanism was proposed.

5. To enable the generation of mispronounced speech (Korzekwa, Lorenzo-Trueba, Drugman, Calamaro, et al., 2021) and improve the fluency of disordered speech (Korzekwa, Barra-Chicote, Kostek, et al., 2019), controllable deep learning models were proposed.

5.3 Applicability

The machine learning models created as part of the doctoral dissertation can be divided into two groups: models for automated pronunciation error detection and models of speech synthesis and voice conversion. Both types of models have been applied to real-world problems at Amazon.

The pronunciation error detection models were applied to automatically detect pronunciation errors in a synthetic speech in two scenarios: during inference and training of speech synthesis models. During inference, the goal is to automatically evaluate the quality of speech generated by speech synthesis models. After the speech synthesis model is trained, a large number of synthetic utterances are synthesized and automatically processed by the pronunciation error detection model. Automatically detecting pronunciation errors enables to evaluate synthetic voices on a large scale and greatly reduces the number of perceptual tests conducted by human listeners. During training, the pronunciation error detection model is used as a perceptual loss to ensure that the speech synthesis model will generate intelligible speech.

Speech synthesis and voice conversion pipelines consist of two steps, a context generation module that generates a mel-spectrogram from the input text and/or the input speech signal and a vocoder component that produces a raw speech signal based on the mel-spectrogram. Both components have been implemented into Alexa devices and serve millions of Amazon customers worldwide. In addition, synthetic speech generated by speech synthesis and voice conversion models improved the accuracy of the pronunciation error detection models in the synthetic speech evaluation task.

5.4 Future work

During the doctoral research, multiple interesting research directions emerged. The most forward-thinking idea is to continue the work from the Ph.D. research on reformulating the problem of pronunciation error detection as a speech generation task (Korzekwa, Lorenzo-Trueba, Drugman, and Kostek, 2022). The proposed Speech-to-Speech (S2S) method can generate synthetic mispronounced speech but is not yet

able to perfectly mimic non-native human speech. To improve the S2S method, a universal speech model should be created in order to generate any type of speech. The model should be able of transforming native speech into non-native speech, reflecting the identity, prosody, speaking style, and pronunciation of the target speaker. This approach could make non-native human speech unnecessary, as the pronunciation error detection model will only be trained on synthetic speech data.

Another interesting research direction is to explore unsupervised speech representations such as Wav2vec (Peng et al., 2021). A more compact speech representation might reduce the need for a large amount of speech data for training pronunciation error detection models. Multi-modal pronunciation error detection to benefit from audio-visual speech corpora is an attractive future direction as well (Czyzewski et al., 2017; Oneata et al., 2022).

So the vision is that future work will also focus on the development of a complete CAPT system with the goal of raising foreign language proficiency in the global population. An AI-based conversational agent will be created. The agent will consist of two elements: a pronunciation error detection model and a feedback component. The pronunciation error detection model will be based on the results of this doctoral research, while the feedback component will require additional research. The CAPT system will only be controlled via the voice interface and the student will have a learning experience similar to the one provided by a human language teacher.

Appendix A

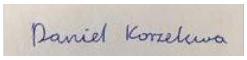
Declaration of authorship

Declaration of Authorship

I, Daniel Korzekwa, declare that this thesis entitled “Automated detection of pronunciation errors in non-native English speech employing deep learning” and the work presented in it are performed by me. I confirm that:

- This work, with respect to the publications with me as the main author, was done mainly within the framework of Implementation Doctoral School (doktorat wdrożeniowy) for a research degree at Gdańsk University of Technology.
- I have made clear what I have contributed myself, as stated in the following author contribution statements.

2 June 2022



Date

Daniel Korzekwa

Author Contribution Statement

I declare that in the publication:

Daniel Korzekwa, Roberto Barra-Chicote, Bozena Kostek, Thomas Drugman, Mateusz Lajszczak (2019). "Interpretable Deep Learning Model for the Detection and Reconstruction of Dysarthric Speech". In: Proc. Interspeech 2019, pp. 3890–3894.

my contribution, in accordance with CRediT (Contributor Roles Taxonomy), was as follows: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Data Curation, Writing - Original Draft, Visualization.

19 May 2022 — Daniel Korzekwa

Date — Daniel Korzekwa

I, the undersigned, hereby certify that the information given by Daniel Korzekwa is correct.

25/5/2022

Date — Roberto Barra-Chicote

16.05.22

Date — Bożena Kostek

28 May 2022

Date — Thomas Drugman

31 May 2022

Date — Mateusz Lajszczak

Author Contribution Statement

I declare that in the publication:

Korzekwa, Daniel and Bozena Kostek (2019). "Deep learning model for automated assessment of lexical stress of non-native English speakers". In: The Journal of the Acoustical Society of America 146.4, pp. 2956–2957.

my contribution, in accordance with CRediT (Contributor Roles Taxonomy), was as follows: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Data Curation, Joint Writing - Original Draft, Visualization.

28 May 2022 Daniel Korzekwa

Date Daniel Korzekwa

I, the undersigned, hereby certify that the information given by Daniel Korzekwa is correct.

30.05.22

Date Bożena Kostek

Author Contribution Statement

I declare that in the publication:

Daniel Korzekwa, Jaime Lorenzo-Trueba, Szymon Zaporowski, Shira Calamaro, Thomas Drugman, Bozena Kostek (2021b). "Mispronunciation Detection in Non-Native (L2) English with Uncertainty Modeling". In: ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, pp. 7738–7742.

my contribution, in accordance with CRediT (Contributor Roles Taxonomy), was as follows: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Data Curation, Writing - Original Draft, Visualization.

19 May 2022	Daniel Korzekwa
Date	Daniel Korzekwa

I, the undersigned, hereby certify that the information given by Daniel Korzekwa is correct.

19/05/2022	
Date	Jaime Lorenzo-Trueba
25/05/2022	
Date	Szymon Zaporowski
24/05/22	
Date	Shira Calamaro
22 May 2022	
Date	Thomas Drugman
19.05.22	
Date	Bożena Kostek

Author Contribution Statement

I declare that in the publication:

Daniel Korzekwa, Roberto Barra-Chicote, Szymon Zaporowski, Grzegorz Beringer, Jaime Lorenzo-Trueba, Alicja Serafinowicz, Jasha Droppo, Thomas Drugman, Bozena Kostek (2021a). "Detection of Lexical Stress Errors in Non Native (L2) English with Data Augmentation and Attention". In: Proc. Inter-speech 2021, pp. 3915–3919. DOI: 10.21437/Interspeech.2021-86.

my contribution, in accordance with CRediT (Contributor Roles Taxonomy), was as follows: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Data Curation, Writing - Original Draft, Visualization.

19 May 2022 — Daniel Korzekwa
Date — Daniel Korzekwa

I, the undersigned, hereby certify that the information given by Daniel Korzekwa is correct.

25/05/2022
Date — Roberto Barra-Chicote

25/05/2022
Date — Szymon Zaporowski

30/05/2022
Date — Grzegorz Beringer

19/05/2022
Date — Jaime Lorenzo-Trueba

2 June 2022
Date — Alicja Serafinowicz

31 MAY 2022
Date — Jasha Droppo

28 May 2022
Date — Thomas Drugman

19.05.22
Date — Bożena Kostek

Author Contribution Statement

I declare that in the publication:

Daniel Korzekwa, Jaime Lorenzo-Trueba, Thomas Drugman, Shira Calamaro, Bozena Kostek (2021c). "Weakly-Supervised Word-Level Pronunciation Error Detection in Non-Native English Speech". In: Proc. Interspeech 2021, pp. 4408–4412. DOI: 10.21437/Interspeech.2021-38.

my contribution, in accordance with CRediT (Contributor Roles Taxonomy), was as follows: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Data Curation, Writing - Original Draft, Visualization.

19 May 2022 — Daniel Korzekwa
Date — Daniel Korzekwa

I, the undersigned, hereby certify that the information given by Daniel Korzekwa is correct.

Date — Jaime Lorenzo-Trueba

22 May 2022
Date — Thomas Drugman

Date — Shira Calamaro

19.05.22
Date — Bożena Kostek

Author Contribution Statement

I declare that in the publication:

Daniel Korzekwa, Jaime Lorenzo-Trueba, Thomas Drugman, Bozena Kostek (2022). "Computer-assisted Pronunciation Training - Speech synthesis is almost all you need". In: Submitted to Speech Communication Journal.

my contribution, in accordance with CRediT (Contributor Roles Taxonomy), was as follows: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Data Curation, Writing - Original Draft, Visualization.

19 May 2022	Daniel Korzekwa
Date	Daniel Korzekwa

I, the undersigned, hereby certify that the information given by Daniel Korzekwa is correct.

Date	Jaime Lorenzo-Trueba
22 May 2022	
Date	Thomas Drugman
19.05.22	
Date	Bozena Kostek

Appendix B

List of publications of the author of the doctoral dissertation

The articles published or accepted for publication with Daniel Korzekwa as the primary author:

1. Korzekwa, D., J. Lorenzo-Trueba, T. Drugman, and B. Kostek (2022). "Computer-assisted Pronunciation Training - Speech synthesis is almost all you need". In: *accepted for publication in Speech Communication Journal on June 17 '2022, in print*.

2. Korzekwa, D., R. Barra-Chicote, S. Zaporowski, G. Beringer, J. Lorenzo-Trueba, A. Serafinowicz, J. Droppo, T. Drugman, and B. Kostek (2021). "Detection of Lexical Stress Errors in Non-Native (L2) English with Data Augmentation and Attention". In: *Proc. Interspeech 2021*, pp. 3915–3919. DOI: `10.21437/Interspeech.2021-86`.

3. Korzekwa, D., J. Lorenzo-Trueba, T. Drugman, S. Calamaro, and B. Kostek (2021). "Weakly-Supervised Word-Level Pronunciation Error Detection in Non-Native English Speech". In: *Proc. Interspeech 2021*, pp. 4408–4412. DOI: `10.21437/Interspeech.2021-38`.

4. Korzekwa, D., J. Lorenzo-Trueba, S. Zaporowski, S. Calamaro, T. Drugman, and B. Kostek (2021). "Mispronunciation Detection in Non-Native (L2) English with Uncertainty Modeling". In: *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, pp. 7738–7742. DOI: `10.1109/ICASSP39728.2021.9413953`.

5. Korzekwa, D., R. Barra-Chicote, B. Kostek, T. Drugman, and M. Lajszczak (2019). "Interpretable Deep Learning Model for the Detection and Reconstruction of Dysarthric Speech". In: *Proc. Interspeech 2019*, pp. 3890–3894. DOI: `10.21437/Interspeech.2019-1206`.

6. Korzekwa, D. and B. Kostek (2019). "Deep learning model for automated assessment of lexical stress of non-native English speakers". In: *The Journal of the Acoustical Society of America* 146.4, pp. 2956–2957. DOI: `10.1121/1.5137270`.

Other articles co-authored by Daniel Korzekwa:

1. Bilinski, P., T. Merritt, A. Ezzerg, K. Pokora, S. Cygert, K. Yanagisawa, R. Barra-Chicote, and D. Korzekwa (2022). "Creating New Voices using Normalizing Flows". In: *accepted to Interspeech 2022*.

2. Merritt, T., A. Ezzerg, P. Biliński, M. Proszewska, K. Pokora, R. Barra-Chicote, and D. Korzekwa (2022). "Text-Free Non-Parallel Many-To-Many Voice Conversion Using Normalising Flow". In: *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6782–6786. DOI: 10.1109/ICASSP43922.2022.9746368.

3. Zhang, D., A. Ganesan, S. Campbell, and D. Korzekwa (2022). "L2-GEN: A Neural Phoneme Paraphrasing Approach to L2 Speech Synthesis for Mispronunciation Diagnosis". In: *accepted to Interspeech 2022*.

4. Ezzerg, A., A. Gabrys, B. Putrycz, D. Korzekwa, D. Saez-Trigueros, D. McHardy, K. Pokora, J. Lachowicz, J. Lorenzo-Trueba, and V. Klimkov (2021). "Enhancing audio quality for expressive Neural Text-to-Speech". In: *Proc. 11th ISCA Speech Synthesis Workshop (SSW 11)*, pp. 78–83. DOI: 10.21437/SSW.2021-14.

5. Gabryś, A., Y. Jiao, V. Klimkov, D. Korzekwa, and R. Barra-Chicote (2021). "Improving the Expressiveness of Neural Vocoding with Non-Affine Normalizing Flows". In: *Proc. Interspeech 2021*, pp. 1679–1683. DOI: 10.21437/Interspeech.2021-1555.

6. Jiao, Y., A. Gabryś, G. Tinchev, B. Putrycz, D. Korzekwa, and V. Klimkov (2021). "Universal neural vocoding with parallel wavenet". In: *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, pp. 6044–6048. DOI: 10.1109/ICASSP39728.2021.9414444.

7. Shah, R., K. Pokora, A. Ezzerg, V. Klimkov, G. Huybrechts, B. Putrycz, D. Korzekwa, and T. Merritt (2021). "Non-Autoregressive TTS with Explicit Duration Modelling for Low-Resource Highly Expressive Speech". In: *Proc. 11th ISCA Speech Synthesis Workshop (SSW 11)*, pp. 96–101. DOI: 10.21437/SSW.2021-17.

8. Beringer, G., D. Korzekwa, A. Sanchez, B. Wang, and J. Lorenzo-Trueba (2020). "Extending Goodness of Pronunciation to generate mispronunciationhypotheses for pronunciation assessment in L2-English". In: *Amazon Machine Learning Conference, Seattle*.

9. Weber, D., S. Zaporowski, and D. Korzekwa (2020). "Constructing a Dataset of Speech Recordings with Lombard Effect". In: *24th IEEE SPA*. DOI: 10.23919/SPA50552.2020.9241266.

10. Merritt, T., B. Putrycz, A. Nadolski, T. Ye, D. Korzekwa, W. Dolecki, T. Drugman, V. Klimkov, A. Moinet, A. Breen, et al. (2018). "Comprehensive evaluation of statistical speech waveform synthesis". In: *2018 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, pp. 325–331.

Appendix C

Primary author publications in the original format

Computer-assisted Pronunciation Training - Speech synthesis is almost all you need

Daniel Korzekwa[1,2], Jaime Lorenzo-Trueba[1], Thomas Drugman[1], Bozena Kostek[2]

[1]Amazon Speech Research
[2]Gdansk University of Technology, Faculty of ETI, Poland

ARTICLE HISTORY
Compiled June 17, 2022

ABSTRACT
The research community has long studied computer-assisted pronunciation training (CAPT) methods in non-native speech. Researchers focused on studying various model architectures, such as Bayesian networks and deep learning methods, as well as on the analysis of different representations of the speech signal. Despite significant progress in recent years, existing CAPT methods are not able to detect pronunciation errors with high accuracy (only 60% precision at 40%-80% recall). One of the key problems is the low availability of mispronounced speech that is needed for the reliable training of pronunciation error detection models. If we had a generative model that could mimic non-native speech and produce any amount of training data, then the task of detecting pronunciation errors would be much easier. We present three innovative techniques based on phoneme-to-phoneme (P2P), text-to-speech (T2S), and speech-to-speech (S2S) conversion to generate correctly pronounced and mispronounced synthetic speech. We show that these techniques not only improve the accuracy of three machine learning models for detecting pronunciation errors but also help establish a new state-of-the-art in the field. Earlier studies have used simple speech generation techniques such as P2P conversion, but only as an additional mechanism to improve the accuracy of pronunciation error detection. We, on the other hand, consider speech generation to be the first-class method of detecting pronunciation errors. The effectiveness of these techniques is assessed in the tasks of detecting pronunciation and lexical stress errors. Non-native English speech corpora of German, Italian, and Polish speakers are used in the evaluations. The best proposed S2S technique improves the accuracy of detecting pronunciation errors in AUC metric by 41% from 0.528 to 0.749 compared to the state-of-the-art approach.

KEYWORDS
computer-assisted pronunciation training; automated pronunciation error detection; automated lexical stress error detection; speech synthesis; voice conversion; deep learning

1. Introduction

Language plays a key role in online education, giving people access to large amounts of information contained in articles, books, and video lectures. Thanks to spoken language and other forms of communication, such as a sign-language, people can participate in interactive discussions with teachers and take part in lively brainstorming with other people. Unfortunately, education is not available to everybody. According to the UNESCO report, 40% of the global population do not have access to education in the language they understand [1]. 'If you don't understand, how can you learn?'

the report says. English is the leading language on the Internet, representing 25.9% of the world's population [2]. Regrettably, research by EF (Education First) [3] shows a large disproportion in English proficiency across countries and continents. People from regions of 'very low' language proficiency, such as the Middle East, are unable to navigate through English-based websites or communicate with people from an English-speaking country.

Computer-Assisted Language Learning (CALL) helps to improve the English language proficiency of people in different regions [4]. CALL relies on computerized self-service tools that are used by students to practice a language, usually a foreign language, also known as a non-native (L2) language. Students can practice multiple aspects of the language, including grammar, vocabulary, writing, reading, and speaking. Computer-based tools can also be used to measure student's language skills and their learning potential by using Computerized Dynamic Assessment (C-DA) test [5]. CALL can complement traditional language learning provided by teachers. It also has a chance to make second language learning more accessible in scenarios where traditional ways of learning languages are not possible due to the cost of learning or the lack of access to foreign language teachers.

Computer-Assisted Pronunciation Training (CAPT) is a part of CALL responsible for learning pronunciation skills. It has been shown to help people practice and improve their pronunciation skills [6–8]. CAPT consists of two components: an automated pronunciation evaluation component [9–11] and a feedback component [12]. The automated pronunciation evaluation component is responsible for detecting pronunciation errors in spoken speech, for example, for detecting words pronounced incorrectly by the speaker. The feedback component informs the speaker about mispronounced words and advises how to pronounce them correctly. This article is devoted to the topic of automated detection of pronunciation errors in non-native speech. This area of CAPT can take advantage of technological advances in machine learning and bring us closer to creating a fully automated assistant based on artificial intelligence for language learning.

The research community has long studied the automated detection of pronunciation errors in non-native speech. Existing work has focused on various tasks such as detecting mispronounced phonemes [9] and lexical stress errors [13]. Researcher have given most attention to studying various machine learning models such as Bayesian networks [14, 15] and deep learning methods [9, 10], as well as analyzing different representations of the speech signal such as prosodic features (duration, energy and pitch) [16], and cepstral/spectral features [9, 13, 17]. Despite significant progress in recent years, existing CAPT methods detect pronunciation errors with relatively low accuracy of 60% precision at 40%-80% recall [9–11]. Highlighting correctly pronounced words as pronunciation errors by the CAPT tool can demotivate students and lower the confidence in the tool. Likewise, missing pronunciation errors can slow down the learning process.

One of the main challenges with the existing CAPT methods is poor availability of mispronounced speech, which is required for the reliable training of pronunciation error detection models. We propose a reformulation of the problem of pronunciation error detection as a task of synthetic speech generation. Intuitively, if we had a generative model that could mimic mispronounced speech and produce any amount of training data, then the task of detecting pronunciation errors would be much easier. The probability of pronunciation errors for all the words in a sentence can then be calculated using the Bayes rule [18]. In this new formulation, we move the complexity to learning the speech generation process that is well suited to the problem of lim-

ited speech availability [19–21]. The proposed method outperforms the state-of-the-art model [9] in detecting pronunciation errors in AUC metric by 41% from 0.528 to 0.749 on the GUT Isle Corpus of L2 Polish speakers.

To put the new formulation of the problem into action, we propose three innovative techniques based on phoneme-to-phoneme (P2P), text-to-speech (T2S), and speech-to-speech (S2S) conversion to generate correctly pronounced and mispronounced synthetic speech. We show that these techniques not only improve the accuracy of three machine learning models for detecting pronunciation errors but also help establish a new state-of-the-art in the field. The effectiveness of these techniques is assessed in two tasks: detecting mispronounced words (replacing, adding, removing phonemes, or pronouncing an unknown speech sound) and detecting lexical stress errors. The results presented in this study are the culmination of our recent work on speech generation in pronunciation error detection task [11, 22, 23], including a new S2S technique.

In short, the contributions of the paper are as follows:

- A new paradigm for the automated detection of pronunciation errors is proposed, reformulating the problem as a task of generating synthetic speech.
- A unified probabilistic view on P2P, T2S, and S2S techniques is presented in the context of detecting pronunciation errors.
- A new S2S method to generate synthetic speech is proposed, which outperforms the state-of-the-art model [9] in detecting pronunciation errors.
- Comprehensive experiments are described to demonstrate the effectiveness of speech generation in the tasks of pronunciation and lexical stress error detection.

The outline of the rest of this paper is: Section 2 presents related work. Section 3 describes the proposed methods of generating synthetic speech for automatic detection of pronunciation errors. Section 4 describes the human speech corpora used to train the pronunciation error detection models in the experiments. Section 5 presents experiments demonstrating the effectiveness of various synthetic speech generation methods in improving the accuracy of the detection of pronunciation and lexical stress errors. Finally, conclusions and future work are presented in Section 6.

2. Related work

2.1. Pronunciation error detection

2.1.1. Phoneme recognition approaches

Most existing CAPT methods are designed to recognize the phonemes pronounced by the speaker and compare them with the expected (canonical) pronunciation of correctly pronounced speech [9, 14, 24, 25]. Any discrepancy between the recognized and canonical phonemes results in a pronunciation error at the phoneme level. Phoneme recognition approaches generally fall into two categories: methods that align a speech signal with phonemes (forced-alignment techniques) and methods that first recognize the phonemes in the speech signal and then align the recognized and canonical phoneme sequences. Aside these two categories, CAPT methods can be split into multiple other categories:

Forced-alignment techniques [15, 24–26] are based on the work of Franco et al. [27] and the Goodness of Pronunciation (GoP) method [14]. In the first step, GoP uses Bayesian inference to find the most likely alignment between canonical phonemes and the corresponding audio signal (forced alignment). In the next step, GoP calcu-

lates the ratio between the likelihoods of the canonical and the most likely pronounced phonemes. Finally, it detects mispronunciation if the ratio drops below a certain threshold. GoP has been further extended with Deep Neural Networks (DNNs), replacing the Hidden Markov Model (HMM) and Gaussian Mixture Model (GMM) techniques for acoustic modeling [24, 25]. Cheng et al. [26] improves GoP performance with the hidden representation of speech extracted in an unsupervised way. This model can detect pronunciation errors based on the input speech signal and the reference canonical speech signal, without using any linguistic information such as text and phonemes.

The methods that do not use forced-alignment recognize the phonemes pronounced by the speaker purely from the speech signal and only then align them with the canonical phonemes [28–33]. Leung et al. [9] use a phoneme recognizer that recognizes phonemes only from the speech signal. The phoneme recognizer is based on Convolutional Neural Network (CNN), a Gated Recurrent Unit (GRU), and Connectionist Temporal Classification (CTC) loss. Leung et al. report that it outperforms other forced-alignment [24] and forced-alignment-free [29] techniques in the task of detecting mispronunciations at the phoneme-level in L2 English.

There are two challenges with presented approaches for pronunciation error detection. First, phonemes pronounced by the speaker must be recognized accurately, which has been proved difficult [10, 34–36]. Phoneme recognition is difficult, especially in non-native speech, as different languages have different phoneme spaces. Second, standard approaches assume only one canonical pronunciation of a given text, but this assumption is not always true due to the phonetic variability of speech, e.g., differences between regional accents. For example, the word 'enough' can be pronounced by native speakers in multiple ways: /ih n ah f/ or /ax n ah f/ (short 'i' or 'schwa' phoneme at the beginning). In our previous work, we solve these problems by creating a native speech pronunciation model that returns the probability of the sentence to be spoken by a native speaker [11].

Techniques based on phoneme recognition can be supplemented by a reference speech signal obtained from the speech database [37–39] or generated from the phonetic representation [11, 40]. Xiao et al. [37] use a pair of speech signals from a student and a native speaker to classify native and non-native speech. Mauro et al. [38] use the speech of the reference speaker to detect mispronunciation errors at the phoneme level. Wang et al. [39] use Siamese networks to model the discrepancy between normal and distorted children's speech. Qian et al. [40] propose a statistical model of pronunciation in which they build a model that generates hypotheses of mispronounced speech.

In this work, we use the end-to-end method to detect pronunciation errors directly, without having to recognize phonemes as an intermediate step. The end-to-end approach is discussed in more detail in the next section.

2.1.2. End-to-end methods

The phoneme recognition approaches presented so far rely on phonetically transcribed speech labeled by human listeners. Phonetic transcriptions are needed to train a phoneme recognition model. Human-based transcription is a time-consuming task, especially with L2 speech, where listeners need to recognize mispronunciation errors. Sometimes L2 speech transcription may be even impossible because different languages have different phoneme sets, and it is unclear which phonemes were pronounced by the speaker. In our recent work, we have introduced a novel model (known as WEAKLY-S, i.e., weakly supervised) for detecting pronunciation errors at the world level that

does not require phonetically transcribed L2 speech [22]. During training, the model is weakly supervised, in the sense that in L2 speech, only mispronounced words are marked, and the data do not need to be phonetically transcribed. In addition to the primary task of detecting mispronunciation errors at the world level, the second task uses a phoneme recognizer trained on automatically transcribed L1 speech.

Zhang et al. [10] employ a multi-task model with two tasks: phoneme-recognition and pronunciation error detection tasks. Unlike our WEAKLY-S model, they use the Needleman-Wunsch algorithm [41] from bioinformatics to align the canonical and recognized phoneme sequences, but this algorithm cannot be tuned to detect pronunciation errors. The WEAKLY-S model automatically learns the alignment, thus eliminating a potential source of inaccuracy. The alignment is learned through an attention mechanism that automatically maps the speech signal to a sequence of pronunciation errors at the word level. Tong et al. [39] propose to use a multi-task framework in which a neural network model is used to learn the joint space between the acoustic characteristics of adults and children. Additionally, Duan et al. [42] propose a multi-task model for acoustical modeling with two tasks for native and non-native speech respectively.

The work of Zhang et al. [10] and our recent work [22] are end-to-end methods of direct estimation of pronunciation errors, setting up a new trend in the field of automated pronunciation assessment. In this article, we use the end-to-end method as well, but we extend it by the S2S method of generating mispronounced speech.

2.1.3. Other trends

All the works presented so far treat pronunciation errors as discrete categories, at best producing the probability of mispronunciation. In contrast, Bi-Cheng et al. [43] propose a model capable of identifying phoneme distortions, giving the user more detailed feedback on mispronunciation. In our recent work, we provide more fine-grained feedback by indicating the severity level of mispronunciation [22].

Active research is conducted not only on modelling techniques but also on speech representation. Xu et al. [44] and Peng et al. [45] use the Wav2vec 2.0 speech representation that is created in an unsupervised way. They report that it outperforms existing methods and requires three times less speech training data. Lin et al. [46] use transfer learning by taking advantage of deep latent features extracted from the Automated Speech Recognition (ASR) acoustic model and report improvements over the classic GOP-based method.

In this work, we use a mel-spectrogram as a speech representation in the pronunciation error detection model. We also use a mel-spectrogram to represent the speech signal in the T2S and S2S methods of generating mispronounced speech.

2.2. Lexical stress error detection

CAPT usually focuses on practicing the pronunciation of phonemes [9, 11, 14]. However, there is evidence that practicing lexical stress improves the intelligibility of non-native English speech [47, 48]. Lexical stress is a phonological feature of a syllable. It is part of the phonological rules that govern how words should be pronounced in a given language. Stressed syllables are usually longer, louder, and expressed with a higher pitch than their unstressed counterparts [49]. The lexical stress is related to the phonemic representation. For example, placing lexical stress on a different syllable of a word can lead to various phonemic realizations known as 'vowel reduction' [50].

Students should be able to practice both pronunciation and lexical stress in spoken language. We study both topics to better understand the potential of using speech generation methods in CAPT.

The existing works focus on the supervised classification of lexical stress using Neural Networks [17, 51], Support Vector Machines [16, 52], and Fisher's linear discriminant [53]. There are two popular variants: a) discriminating syllables between primary stress/no stress [13], and b) classifying between primary stress/secondary stress/no stress [51, 54]. Ramanathi et al. [55] have followed an alternative unsupervised way of classifying lexical stress, which is based on computing the likelihood of an acoustic signal for a number of possible lexical stress representations of a word.

Accuracy is the most commonly used performance metric, and it indicates the ratio of correctly classified stress patterns on a syllable [54] or word level [16]. On the contrary, Ferrer et al. [13], analyzed the precision and recall metrics to detect lexical stress errors and not just classify them.

Most existing approaches for the classification and detection of lexical stress errors are based on carefully designed features. They start with aligning a speech signal with phonetic transcription, performed via forced-alignment [16, 17]. Alternatively, ASR can provide both phonetic transcription and its alignment with a speech signal [54]. Then, prosodic features such as duration, energy and pitch [16] and cepstral features such as Mel Frequency Cepstral Coefficients (MFCC) and Mel-Spectrogram [13, 17] are extracted. These features can be extracted on the syllable [17] or syllable nucleus [13, 16] level. Shahin et al. [17] computes features of neighboring vowels, and Li et al. [54] includes the features for two preceding and two following syllables in the model. The features are often preprocessed and normalized to avoid potential confounding variables [13], and to achieve better model generalization by normalizing the duration and pitch on a word level [13, 53]. Li et al. [51] adds canonical lexical stress to input features, which improves the accuracy of the model.

In our recent work, we use attention mechanisms to automatically derive areas of the audio signal that are important for the detection of lexical stress errors [23]. In this work, we use the T2S method to generate synthetic lexical stress errors to improve the accuracy of detecting lexical stress errors.

2.3. Synthetic speech generation for pronunciation error detection

Existing synthetic speech generation techniques for detecting pronunciation errors can be divided into two categories: data augmentation and data generation.

Data augmentation techniques are designed to generate new training examples for existing mispronunciation labels. Badenhorst et al. [56] simulate new speakers by adjusting the speed of raw audio signals. Eklund [57] generates additional training data by adding background noise and convolving the audio signal with the impulse responses of the microphone of a mobile device and a room.

Data generation techniques are designed to generate new training data with new labels of both correctly pronounced and mispronounced speech. Most existing works are based on the P2P technique to generate mispronounced speech by perturbing the phoneme sequence of the corresponding audio using a variety of strategies [11, 58–61]. In addition to P2P techniques, in our recent work, we use T2S to generate synthetic lexical stress errors [22]. Qian et al. [40] introduce a generative model to create hypotheses of mispronounced speech and use it as a reference speech signal to detect pronunciation errors. Recently, we proposed a similar technique to create a pronun-

ciation model of native speech to account for many ways of correctly pronouncing a sentence by a native speaker [11].

Synthetic speech generation techniques have recently gained attention in other related fields. Fazel et al. [21] use synthetic speech generated with T2S to improve accuracy in ASR. Huang et al. [62] use a machine translation technique to generate text to train an ASR language model in a low-resource language. At the same time, Shah et al. [20] and Huybrechts et al. [19] employ S2S voice conversion to improve the quality of speech synthesis in the data reduction scenario.

All the presented works on the detection of pronunciation errors treat synthetic speech generation as a secondary contribution. In this article, we present a unified perspective of synthetic speech generation methods for detecting pronunciation errors. This article extends our previous work [11, 22, 23] and introduces a new S2S method to detect pronunciation errors. To the best of our knowledge, there are no papers devoted to generating pronunciation errors with the S2S technique and using it in the detection of pronunciation errors.

3. Methods of generating pronunciation errors

To detect pronunciation errors, first, the spoken language must be separated from other factors in the signal and then incorrectly pronounced speech sounds have to be identified. Separating speech into multiple factors is difficult, as speech is a complex signal. It consists of prosody (F0, duration, energy), timbre of the voice, and the representation of the spoken language. Spoken language is defined by the sounds (phones) perceived by people. Phones are the realizations of phonemes - a human abstract representation of how to pronounce a word/sentence. Speech may also present variability due to the recording channel and environmental effects such as noise and reverberation. Detecting pronunciation errors is very challenging, also because of the limited amount of recordings with mispronounced speech. To address these challenges, we reformulate the problem of pronunciation error detection as the task of synthetic speech generation.

Let $\mathbf{s}$ be the speech signal, $\mathbf{r}$ be the sequence of phonemes that the user is trying to pronounce (canonical pronunciation), and $\mathbf{e}$ be the sequence of probabilities of mispronunciation at the phoneme or word level. The original task of detecting pronunciation errors is defined by:

$$\mathbf{e} \sim p(\mathbf{e}|\mathbf{s}, \mathbf{r}) \tag{1}$$

where the formulation of the problem as the task of synthetic speech generation is defined as follows:

$$\mathbf{s} \sim p(\mathbf{s}|\mathbf{e}, \mathbf{r}) \tag{2}$$

The probability of pronunciation errors for all the words in a sentence can then be calculated using the Bayes rule [18]:

$$p(\mathbf{e}|\mathbf{s},\mathbf{r}) = \frac{p(\mathbf{e}|\mathbf{r})p(\mathbf{s}|\mathbf{e},\mathbf{r})}{p(\mathbf{s}|\mathbf{r})} \tag{3}$$

From Equation 3, one can see that there is no need to directly learn the probability of pronunciation errors $p(\mathbf{e}|\mathbf{s},\mathbf{r})$, since the complexity of the problem has now been transferred to learning the speech generation process $p(\mathbf{s}|\mathbf{e},\mathbf{r})$. Such a formulation of the problem opens the way to the inclusion of additional prior knowledge into the model:

(1) Replacing the phoneme in a word while preserving the original speech signal results in a pronunciation error (P2P method).
(2) Changing the speech signal while retaining the original pronunciation results in a pronunciation error (T2S method).
(3) There are many variations of mispronounced speech that differ in terms of the voice timbre and the prosodic aspects of speech (S2S method).

To solve Equation 3, we use Markov Chain Monte Carlo Sampling (MCMC) [63]. In this way, the prior knowledge can be incorporated by generating N training examples $\{\mathbf{e_i}, \mathbf{s_i}, \mathbf{r_i}\}$ for $i = 1..N$ with the use of P2P (prior knowledge 1), T2S (prior knowledge 2), and S2S (prior knowledge 3) methods. Accounting for the prior knowledge, intuitively corresponds to an increase in the amount of training data, which contributes to outperforming state-of-the-art models for detecting pronunciation errors, as presented in Section 5. Equation 3 can then be optimized with standard gradient-based optimization techniques. In the following subsections, we present the P2P conversion, T2S, and S2S methods of generating correctly and incorrectly pronounced speech in details.

3.1. P2P method

To generate synthetic mispronounced speech, it is enough to start with correctly pronounced speech and modify the corresponding sequence of phonemes. This simple idea does not even require generating the speech signal itself. It can be observed that the probability of mispronunciations depends on the discrepancy between the speech signal and the corresponding canonical pronunciation. This leads to the P2P conversion model shown in Figure 1a.

Let $\{\mathbf{e_{noerr}}, \mathbf{s}, \mathbf{r}\}$ be a single training example containing: the sequence of $0s$ denoting correctly pronounced phonemes, the speech signal, and the sequence of phonemes representing the canonical pronunciation. Let $\mathbf{r}^{'}$ be the sequence of phonemes with injected mispronunciations such as phoneme replacements, insertions, and deletions:

$$\mathbf{r}^{'} \sim p(\mathbf{r}^{'}|\mathbf{r}) \tag{4}$$

then the probability of mispronunciation for the jth phoneme is defined by:

$$e_j^{'} = \begin{cases} 1 & \text{if } r_j^{'}! = r_j \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

The probabilities of mispronunciation can be projected from the level of phonemes to

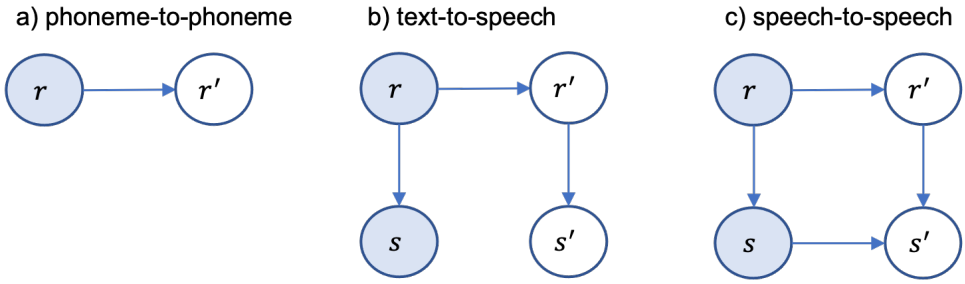


Figure 1. Probabilistic graphical models for three methods to generate pronunciation errors: P2P, T2S and S2S. Empty circles represent hidden (latent) variables, while filled (blue) circles represent observed variables. $\mathbf{s}$ - the speech signal, $\mathbf{r}$ - the sequence of phonemes that the user is trying to pronounce (canonical pronunciation), the superscript $'$ represents a variable with generated mispronunciations.

the level of words. A word is treated as mispronounced if at least one pair of phonemes in the word $\{r_j^{'}, r_j\}$ does not match. At the end of this process, a new training example is created with artificially introduced pronunciation errors: $\{\mathbf{e_{err}}, \mathbf{s}, \mathbf{r}^{'}\}$. Note that the speech signal $\mathbf{s}$ in the new training example is unchanged from the original training example, and only phoneme transcription is manipulated.

Implementation

To generate synthetic pronunciation errors, we use a simple approach of perturbing phonetic transcription for the corresponding speech audio. First, we sample these utterances with replacement from the input corpora of human speech. Then, for each utterance, we replace the phonemes with random phonemes with a given probability.

3.2. T2S method

The T2S method expands on P2P by making it possible to create speech signals that match the synthetic mispronunciations. The T2S method for generating mispronounced speech is a generalization of the P2P method, as can be seen by the comparison of the two methods shown in Figures 1a and 1b.

One problem with the P2P method is that it cannot generate a speech signal for the newly created sequence of phonemes $\mathbf{r}^{'}$. As a result, pronunciation errors will dominate in the training data containing new sequences of phonemes $\mathbf{r}^{'}$. Therefore, it will be possible to detect pronunciation errors only from the canonical representation $\mathbf{r}^{'}$, ignoring information contained in the speech signal. To mitigate this issue, there should be two training examples for the phonemes $\mathbf{r}^{'}$, one representing mispronounced speech: $\{\mathbf{e_{err}}, \mathbf{s}, \mathbf{r}^{'}\}$, and the second one for correct pronunciation: $\{\mathbf{e_{noerr}}, \mathbf{s}^{'}, \mathbf{r}^{'}\}$, where:

$$\mathbf{s}^{'} \sim p(\mathbf{s}^{'} | \mathbf{e_{noerr}}, \mathbf{r}^{'}) \tag{6}$$

Because we now have the speech signal $\mathbf{s}^{'}$, another training example can be created as: $\{\mathbf{e_{err}}, \mathbf{s}^{'}, \mathbf{r}\}$. In summary, T2S method extends a single training example of correctly pronounced speech to four combinations of correctly and incorrect pronunciations:

- $\{\mathbf{e_{noerr}}, \mathbf{s}, \mathbf{r}\}$ – correctly pronounced input speech
- $\{\mathbf{e_{err}}, \mathbf{s}, \mathbf{r}^{'}\}$ – mispronounced speech generated by the P2P method
- $\{\mathbf{e_{noerr}}, \mathbf{s}^{'}, \mathbf{r}^{'}\}$ – correctly pronounced speech generated by the T2S method

- $\{\mathbf{e_{err}}, \mathbf{s}^{'}, \mathbf{r}\}$ – mispronounced speech generated by the T2S method

Implementation

The synthetic speech is generated with the Neural TTS described by Latorre et al. [64]. The Neural TTS consists of two modules. The context-generation module is an attention-based encoder-decoder neural network that generates a mel-spectrogram from a sequence of phonemes. The Neural Vocoder then converts it into a speech signal. The Neural Vocoder is a neural network of architecture similar to Parallel Wavenet [65]. The Neural TTS is trained using the speech of a single native speaker. To generate words with different lexical stress patterns, we modify the lexical stress markers associated with the vowels in the phonetic transcription of the word. For example, with the input of /r iy1 m ay0 n d/ we can place lexical stress on the first syllable of the word 'remind'.

3.3. S2S method

The S2S method is designed to simulate the diverse nature of speech, as there are many ways to correctly pronounce a sentence. The prosodic aspects of speech, such as pitch, duration, and energy, can vary. Similarly, phonemes can be pronounced differently. To mimic human speech, speech generation techniques should allow a similar level of variability. The T2S method outlined in the previous section always produces the same output for the same phoneme input sequence. The S2S method is designed to overcome this limitation.

S2S converts the input speech signal $\mathbf{s}$ in a way to change the pronounced phonemes (phoneme replacements, insertions, and deletions) from the input phonemes $\mathbf{r}$ to target phonemes $\mathbf{r}^{'}$ while preserving other aspects of speech, including voice timbre and prosody (Equation 7 and Figure 1c). In this way, the natural variability of human speech is preserved, resulting in generating many variations of incorrectly pronounced speech. The prosody will differ in various versions of the sentence of the same speaker, while the same sentence spoken by many speakers will differ in the voice timbre.

$$\mathbf{s}^{'} \sim p(\mathbf{s}^{'}|\mathbf{e_{noerr}}, \mathbf{r}^{'}, \mathbf{s}) \tag{7}$$

Similarly to the T2S method, the S2S method outputs four types of speech pronounced correctly and incorrectly: $\{\mathbf{e_{noerr}}, \mathbf{s}, \mathbf{r}\}$, $\{\mathbf{e_{err}}, \mathbf{s}, \mathbf{r}^{'}\}$, $\{\mathbf{e_{noerr}}, \mathbf{s}^{'}, \mathbf{r}^{'}\}$, and $\{\mathbf{e_{err}}, \mathbf{s}^{'}, \mathbf{r}\}$.

Implementation

Synthetic speech is generated by introducing mispronunciations into the input speech, while preserving the duration of the phonemes and timbre of the voice. The architecture of the S2S model is shown in Figure 2. The mel-spectrogram of the input speech signal $\mathbf{s}$ is forced-aligned with the corresponding canonical phonemes $\mathbf{r}$ to get the duration of the phonemes. The speaker id has to be provided together with the input speech to enable the source speaker's voice to be maintained. Mispronunciations are introduced into the canonical phonemes $\mathbf{r}$ according to the P2P method described in Section 3.1. Mispronounced phonemes $\mathbf{r}^{'}$ along with phonemes duration and speaker id are processed by the encoder-decoder, which generates the mel-spectrogram $\mathbf{s}^{'}$. The encoder-decoder transforms the phoneme-level representation into frame-level features

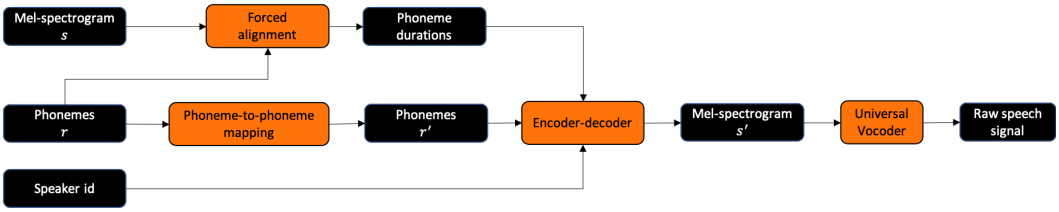


Figure 2. Architecture of the S2S model to generate mispronounced synthetic speech while maintaining prosody and voice timbre of the input speech. The black rectangles represent the data (tensors) and the orange boxes represent processing blocks. This color notation is used in all machine learning model diagrams throughout the article.

and then generates all mel-spectrogram frames in parallel. The mel-spectrogram is converted to an audio signal with Universal Vocoder [66]. Without the Universal Vocoder, it would not be possible to generate the raw audio signal for hundreds of speakers included in the LibriTTS corpus. Details of the S2S method are shown in the works of Shah et al. [20] and Jiao et al. [66]. The main difference between these two models and our S2S model is the use of the P2P mapping to introduce pronunciation errors.

3.4. Summary of mispronounced speech generation

Generation of synthetic mispronounced speech and detection of pronunciation errors were presented from the probabilistic perspective of the Bayes-rule. With this formulation, we can better understand the relationship between P2P, T2S and S2S methods, and see that the S2S method generalizes two simpler methods. Following this reasoning, we can argue that using the Bayes rule gives us a nice mathematical framework to potentially further generalize the S2S method, e.g. by adding a language variable to the model to support multilingual pronunciation error detection. There is another advantage of modelling pronunciation error detection from the probabilistic perspective - it paves the way for joint training of mispronounced speech generation and pronunciation error detection models. In the present work, we are training separate machine learning models for both tasks, but it should be possible to train both models jointly using the framework of Variational Inference [67] instead of MCMC to infer the probability of mispronunciation in Equation 3.

4. Speech corpora

4.1. Corpora of continuous speech

Speech corpora of recorded sentences is a combination of L1 and L2 English speech. L1 speech is obtained from the TIMIT [68] and the LibriTTS [69] corpora. L2 speech comes from the Isle [70] corpus (German and Italian speakers) and the GUT Isle [71] corpus (Polish speakers). In total, we used 125.28 hours of L1 and L2 English speech from 983 speakers segmented into 102812 sentences. A summary of the speech corpora is presented in Table 1, whereas the details are presented in our recent work [22].

The speech data are used in all the pronunciation error detection experiments presented in Section 5. From the collected speech, we held out 28 L2 speakers and used them only to assess the performance of the systems in the mispronunciation detection task. It includes 11 Italian and 11 German speakers from the Isle corpus [70], and 6

Table 1. Summary of human speech corpora used in the pronunciation error detection experiments. * - audiobooks read by volunteers from all over the world [69]

Native Language	Hours	Speakers
English	90.47	640
Unknown*	19.91	285
German and Italian	13.41	46
Polish	1.49	12

Table 2. Details of the training and test sets for the lexical stress error detection model.

Data set	Speakers (L2)	Words (unique)	Stress Errors
Train set (human)	473 (10)	8223 (1528)	425
Train set (TTS)	1 (0)	3937 (1983)	2005
Test set (human)	176 (21)	2108 (378)	189

Polish speakers from the GUT Isle corpus [71]. The human speech training data is extended with synthetic pronunciation errors generated by the methods presented in Section 3.

4.2. Corpora of isolated words

The speech corpora consist of human and synthetic speech. The data were divided into training and testing sets, with separate speakers assigned to each set. Human speech includes native (L1) and non-native (L2) English speech. L1 speech corpora are made of TIMIT [68] and Arctic [72]. L2 corpora contain speech from L2-Arctic [32], Porzuczek [73], and our own recordings of 25 speakers (23 Polish, 1 Ukrainian and 1 Lithuanian). The synthetic data were generated using the T2S method and are only included in the training set. The data are summarized in Table 2. For a more detailed description of speech corpora, see Section 4 of our recent work [23]. The speech corpora of isolated words are used in the lexical stress error detection experiment presented in Section 5.3.

5. Experiments

5.1. Generation of mispronounced speech

5.1.1. Experimental setup

The effect of using synthetic pronunciation errors based on the P2P, T2S and S2S methods is evaluated in the task of detecting pronunciation errors in spoken sentences at the word level. First, we analyze the P2P method by comparing it with the state-of-the-art techniques and measure the effect of adding synthetic pronunciation errors to the training data. We then compare P2P with T2S and S2S to assess the benefits of using more complex methods of generating pronunciation errors. The accuracy of detecting pronunciation errors is reported in standard Area Under the Curve (AUC), precision and recall metrics.

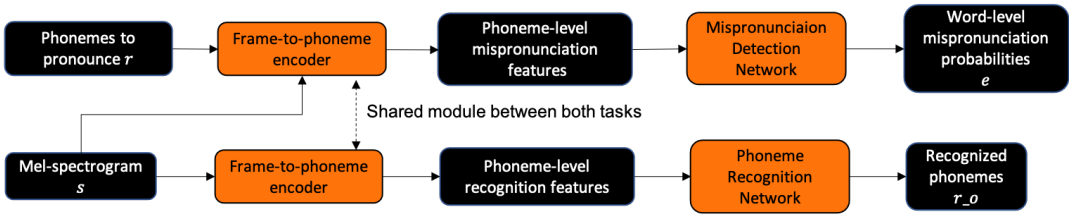


Figure 3. Architecture of the WEAKLY-S model for word-level pronunciation error detection trained in the multi-task setup. Task 1 - to detect pronunciation errors e. Task 2 - to recognize phonemes r_o.

5.1.2. Overview of our WEAKLY-S model

We use the pronunciation error detection model (WEAKLY-S) recently proposed by us [22]. To train the model, the human speech training set is extended with 292,242 utterances of L1 speech with synthetically generated pronunciation errors. To generate pronunciation errors, the P2P, T2S, and S2S methods described in Section 3 are used.

The WEAKLY-S model produces probabilities of mispronunciation for all words, conditioned by the spoken sentence and canonical phonemes. Mispronunciation errors include phoneme replacement, addition, deletion, or an unknown speech sound. During training, the model is weakly supervised, in the sense that only mispronounced words in L2 speech are marked by listeners and the data do not have to be phonetically transcribed. Due to the limited availability of L2 speech and the fact that it is not phonetically transcribed, the model is more likely to overfit. To solve this problem, the model is trained in a multi-task setup. In addition to the primary task of detecting mispronunciation error at the word level, the second task uses a phoneme recognizer which is trained on automatically transcribed L1 speech. Both tasks share components of the model, which makes the primary task less likely to overfit.

The architecture of the pronunciation error detection model is shown in Figure 3. The model consists of two sub-networks. The Mispronunciations Detection Network (MDN) detects word-level pronunciation errors $\mathbf{e}$ from the audio signal $\mathbf{s}$ and canonical phonemes $\mathbf{r}$, while the Phoneme Recognition Network (PRN) recognizes phonemes $\mathbf{r_o}$ pronounced by a speaker from the audio signal $\mathbf{s}$. The detailed model architecture is presented in Section 2 of our recent work [22].

5.1.3. Results - P2P method

We conducted an ablation study to measure the effect of removing synthetic pronunciation errors from the training data. We trained four variants of the WEAKLY-S model to measure the effect of using synthetic data against other elements of the model. WEAKLY-S is a complete model that also includes synthetic data during training. In the NO-SYNTH-ERR model, we exclude synthetic samples of mispronounced L1 speech, significantly reducing the number of mispronounced words seen during training from 1,129,839 to just 5,273 L2 words. The NO-L2-ADAPT variant does not fine-tune the model on L2 speech, although it is still exposed to L2 speech while being trained on a combined corpus of L1 and L2 speech. The NO-L1L2-TRAIN model is not trained on L1/L2 speech, and fine-tuning on L2 speech starts from scratch. This means that this model will not use a large amount of phonetically transcribed L1 speech data and ultimately no secondary phoneme recognition task will be used.

L2 fine-tuning (NO-L2-ADAPT) is the most important factor influencing the performance of the model (Fig. 4 and Table 3), with an AUC of 0.517 compared to 0.686

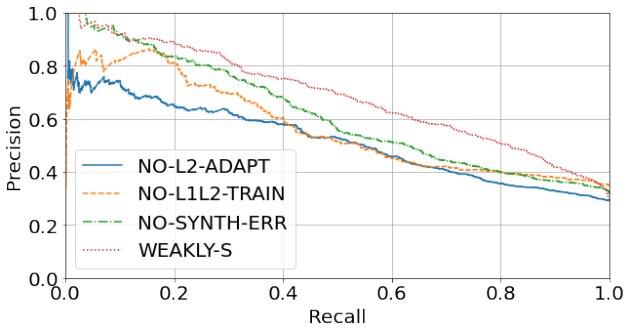


Figure 4. Precision-recall curve for the ablation study on the GUT Isle corpus, illustrating the effect of using synthetic pronunciation errors generated by the P2P method.

Table 3. Ablation study for the GUT Isle corpus to show the effect of using synthetic data and other elements of the WEAKLY-S model.

Model	Description	AUC	Precision [%]	Recall [%]
NO-L2-ADAPT	No fine-tuning on L2 speech	0.517	57.89	40.11
NO-L1L2-TRAIN	No pretraining on L1&L2 speech	0.565	59.73	40.20
NO-SYNTH-ERR	No synthetically generated pronunciation errors in the training data	0.615	67.22	40.38
WEAKLY-S	Complete model	**0.686**	75.25	40.38

for the full model. Training the model on both L2 and L1 human speech together is not enough. This is because L2 speech accounts for less than 1% of the training data and the model naturally leans towards L1 speech. The second most important feature is training the model on a combined set of L1 and L2 speech (NO-L1L2-TRAIN), with an AUC of 0.565. L1 speech accounts for over 99% of training data. These data are also phonetically transcribed, and therefore can be used for the phoneme recognition task. The phoneme recognition task acts as a 'backbone' and reduces the effect of overfitting in the main task of detecting errors in the pronunciation of words. Finally, excluding synthetically generated pronunciation errors (NO-SYNTH-ERR) reduces an AUC from 0.686 to 0.615. Although, the synthetic data provides the least improvement to the model, it still increases the accuracy of the model by 11.5% in AUC, contributing to setting up a new state-of-the-art.

We compare the WEAKLY-S model with two state-of-the-art baselines. The Phoneme Recognizer (PR) model by Leung et al. [9] is our first baseline. The PR is based on the CTC loss [74] and outperforms multiple alternative approaches of pronunciation assessment. The original CTC-based model uses a hard likelihood threshold applied to the recognized phonemes. To compare it with two other models, following our recent work [11], we have replaced the hard likelihood threshold with a soft threshold. The second baseline is PR extended by the pronunciation model (PR-PM model [11]). The pronunciation model takes into account the phonetic variability of the speech spoken by native speakers, which results in greater precision in detecting pronunciation errors. The results are shown in Table 4. It turns out that the WEAKLY-S model outperforms the second-best model in terms of an AUC by 30% from 0.528 to 0.686 and precision by 23% from 0.612 to 0.752 on the GUT Isle Corpus of Polish speakers. We are seeing similar improvements on the Isle Corpus of German and Italian speakers. The use of synthetic data is an important contribution to the performance of the WEAKLY-S model.

Table 4. Accuracy metrics of detecting word-level pronunciation errors. WEAKLY-S vs. baseline models.

Model	AUC	Precision [%,95%CI]	Recall [%,95%CI]
Isle corpus (German and Italian)			
PR	0.555	49.39 (47.59-51.19)	40.20 (38.62-41.81)
PR-PM	0.480	54.20 (52.32-56.08)	40.20 (38.62-41.81)
WEAKLY-S	**0.678**	71.94 (69.96, 73.87)	40.14 (38.56, 41.75)
GUT Isle corpus (Polish)			
PR	0.528	54.91 (50.53-59.24)	40.29 (36.66-44.02)
PR-PM	0.505	61.21 (56.63-65.65)	40.15 (36.51-43.87)
WEAKLY-S	**0.686**	75.25 (71.67-78.59)	40.38 (37.52-43.29)

5.1.4. Results - T2S and S2S methods

The main limitation of the P2P method is that it does not generate a new speech signal. The method introduces mispronunciations by operating only on the sequence of phonemes for the corresponding speech. In this experiment, we demonstrate the T2S and S2S methods that can directly generate a speech signal to overcome this limitation. The S2S method introduces mispronunciations into the input native speech while preserving the prosody (phoneme durations) and timbre of the voice. Preserving speech attributes other than pronunciation increases speech variability during training and makes the pronunciation error detection model more reliable during testing. The T2S method can be considered as a simplified variant of the S2S method, in which there is only text as input.

The T2S and S2S methods are compared with the P2P method. Three WEAKLY-S models are trained, differing in the technique of generating mispronounced speech contained in the training data. The S2S method outperforms the P2P method by increasing an AUC score by 9% from 0.686 to 0.749 in the Gut Isle corpus of Polish speakers (Table 5). Additionally, an AUC increases from 0.815 to 0.834 for major pronunciation errors (Table 6), according to a similar experiment presented in Section 3.4 of [22]. Interestingly, the T2S method is only slightly better than the P2P method, which suggests that the variability of the generated mispronounced speech provided by the S2S method is really important. The presented experiments show the potential of the S2S method in improving the accuracy of detecting pronunciation errors. The S2S method is able to control voice timbre, phoneme duration, and pronunciation, opening the door to transplanting all three properties from non-native speech and potentially further improving the accuracy of the model.

One downside of the S2S method is its complexity. Compared to the straightforward P2P method, the 9% improvement in an AUC is associated with high costs. The method involves training a complex multi-speaker S2S model to convert between input and output mel-spectrograms and requires training a Universal Vocoder model to convert a mel-spectrogram into a raw speech signal.

To better understand what prevents the model from achieving higher accuracy, we measure the performance of the model on synthetic pronunciation errors. We divide all synthetic pronunciation errors into four categories to reflect the severity of pronunciation errors. The 'low' category includes mispronounced words with only one mismatched phoneme between the canonical and pronounced phonemes of the word. The 'medium' category includes two mispronounced phonemes. The 'high' category gets three, and the 'very high' category includes four mispronounced errors. The AUC across different severity levels varies from 0.928 (low severity) to 1.00 (very high severity) as shown in Table 7. These AUC values are significantly higher than the results for non-native human speech, suggesting that making synthetic speech errors more similar

Table 5. Comparison of the P2P, T2S and S2S methods in the task of pronunciation error detection assessed on the GUT Isle corpus.

Model	AUC	Precision [%]	Recall [%]
P2P	0.686	75.25 (71.67-78.59)	40.38 (37.52-43.29)
T2S	0.695	76.15 (72.59-79.36)	40.25 (37.44-43.22)
S2S	0.749	80.45 (76.94-83.47)	40.12 (37.12-43.02)

Table 6. Comparison of the P2P, T2S and S2S methods in the task of pronunciation error detection assessed on the GUT Isle corpus only for major pronunciation errors.

Model	AUC	Precision [%]	Recall [%]
P2P	0.815	91.67 (88.55-94.45)	40.31 (37.43-43.23)
T2S	0.819	92.11 (89.09-94.83)	40.21 (36.81-43.31)
S2S	0.834	93.54 (90.53-96.23)	40.15 (37.26-43.11)

to non-native speech may improve the accuracy of detecting pronunciation errors.

5.2. Model of native speech pronunciation

5.2.1. Experimental setup

The P2P, T2S, and S2S are generative models that provide the probability of generating a particular output sequence. This probability can be used directly to detect pronunciation errors without generating the mispronounced speech and adding it to the training data. In this experiment, we show how to apply this approach in practice.

One of the challenges in detecting pronunciation errors is that a native speaker can pronounce a sentence correctly in many ways. The classic approach for detecting pronunciation errors is based on identifying the difference between pronounced and canonical phonemes. All pronunciations that do not correspond precisely to the canonical pronunciation will result in false pronunciation errors. One way to solve this problem is to use the P2P technique to create a native speech Pronunciation Model (PM) that determines the probability that a sentence is pronounced by a native speaker. A low likelihood value indicates a high probability of mispronunciation.

To evaluate the performance of the PM model, the pronunciation error detection model has been designed such that the PM model can be turned on and off. To disable the PM, we are modifying it so that it only takes into account one way of correctly pronouncing a sentence. In an ablation study, we measure whether the PM model improves the accuracy in detecting pronunciation errors at the word level. Note that in this experiment, synthetically generated pronunciation errors are not used explicitly. Instead, the native speech pronunciation model is used to implicitly represent the

Table 7. Accuracy (AUC) in detecting pronunciation errors assessed in synthetic speech at different severity levels of mispronunciation for the best S2S method.

Severity	AUC
Low (phoneme distance=1)	0.928
Medium (phoneme distance=2)	0.974
High (phoneme distance=3)	0.993
Very High (phoneme distance=4)	1.00

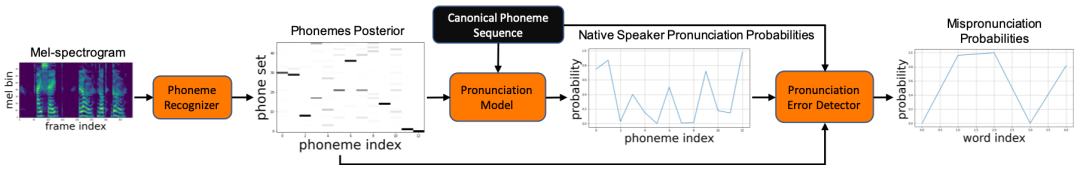


Figure 5. Architecture of the system for detecting mispronounced words in a spoken sentence based on the native speech pronunciation model.

generative speech process.

5.2.2. Overview of the pronunciation error detection model

The design of the pronunciation error detection model consists of three subsystems: a Phoneme Recognizer (PR), a Pronunciation Model (PM), and a Pronunciation Error Detector (PED), shown in Figure 5. First, the PR model estimates a belief over the phonemes produced by the student, intuitively representing the uncertainty in the student's pronunciation. The PM model transforms this belief into a probability that a native speaker would pronounce the sentence this way, given the phonetic variability. Finally, the PED model decides which words were mispronounced in the sentence by processing three pieces of information: a) what the student pronounced, b) how likely it is that the native speaker would pronounce it that way, and c) what the student was supposed to pronounce. Details of the entire model of pronunciation error detection are presented in Section 3 of our recent work [11]. We will now only show the details of the PM model that are relevant to this experiment.

5.2.3. Overview of the native speech pronunciation model

PM is an encoder-decoder neural network, following Sutskever et al. [75]. Instead of building a text-to-text translation system between two languages, we use it for the P2P conversion. The sequence of phonemes $\mathbf{r}$ that the native speaker was supposed to pronounce is converted to the sequence of phonemes $\mathbf{r}^{'}$ they had pronounced, denoted as $\mathbf{r}^{'} \sim p(\mathbf{r}^{'}|\mathbf{r})$. Once trained, PM acts as a probability mass function, computing the probability sequence $\boldsymbol{\pi}$ of the recognized phonemes $\mathbf{r_o}$ pronounced by the student conditioned by the expected (canonical) phonemes $\mathbf{r}$. PM is denoted as in Eq. 8.

$$\boldsymbol{\pi} = \sum_{\mathbf{r_o}} p(\mathbf{r_o}|\mathbf{o}) p(\mathbf{r}^{'} = \mathbf{r_o}|\mathbf{r}) \tag{8}$$

The PM model is trained on P2P speech data generated automatically by passing the speech of the native speakers through the PR. By using PR to annotate the data, we can make the PM model more robust against possible phoneme recognition inaccuracies in PR at the time of testing.

5.2.4. Results

The complete model with PM enabled is called PR-PM that stands for a Phoneme Recognizer + Pronunciation Model. The model with PM turned off is called PR-LIK that stands for Phoneme Recognizer outputting the likelihoods of recognized phonemes. PR-LIK is an extension of the PR-NOLIK model – the mispronunciation

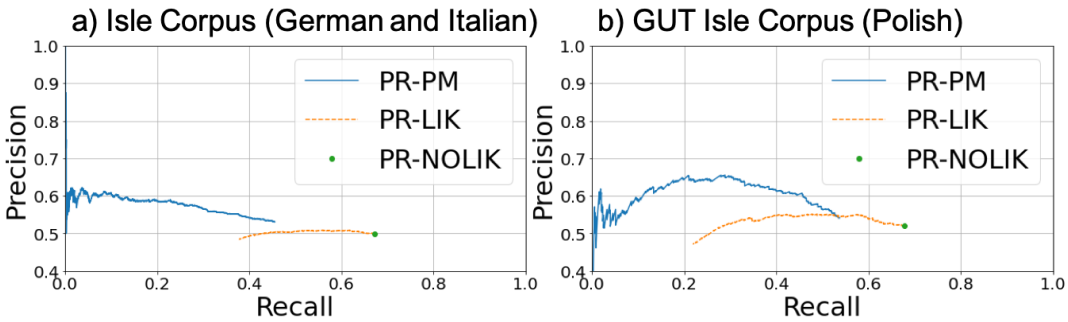


Figure 6. Precision-recall curves for the evaluated systems to measure the effect of using the PM model in detecting pronunciation errors. PR-PM - full model with the PM enabled. PR-LIK - the PR-PM model with the PM disabled. PR-NOLIK - non-probabilistic variant of the PR-LIK model proposed by Leung et al. [9].

Table 8. Precision and recall of detecting word-level mispronunciations. CI - Confidence Interval. PR-PM - full model with the PM enabled. PR-LIK - the PR-PM model with the PM disabled.

Model	Precision [%,95%CI]	Recall [%,95%CI]
Isle corpus (German and Italian)		
PR-LIK	49.39 (47.59-51.19)	40.20 (38.62-41.81)
PR-PM	54.20 (52.32-56.08)	40.20 (38.62-41.81)
GUT Isle corpus (Polish)		
PR-LIK	54.91 (50.53-59.24)	40.29 (36.66-44.02)
PR-PM	61.21 (56.63-65.65)	40.15 (36.51-43.87)

detection model proposed by Leung et al. [9] that only returns the most likely recognized phonemes and does not use phoneme likelihoods to detect pronunciation errors. PR-NOLIK detects mispronounced words based on the difference between the canonical and recognized phonemes. Therefore, this system does not offer any flexibility in optimizing the model for higher precision by fine-tuning the threshold applied to the phoneme recognition probabilities.

Turning off PM reduces the precision between 11% and 18%, depending on the decrease in recall between 20% to 40%, as shown in Figure 6. One example where the PM helps is the word 'enough' that can be pronounced in two similar ways: /ih n ah f/ or /ax n ah f/ (short 'i' or 'schwa' phoneme at the beginning.) The PM can take into account the phonetic variability and recognize both versions as correctly pronounced. Another example is coarticulation [76]. Native speakers tend to merge phonemes of adjacent words. For example, in the text 'her arrange' /hh er - er ey n jh/, two adjacent phonemes /er/ can be pronounced as one phoneme: /hh er ey n jh/. The PM model can correctly recognize multiple variations of such pronunciations.

Complementary to the precision-recall curve shown in Figure 6, we present in Table 8 one configuration of the precision and recall scores for the PR-LIK and PR-PM systems. This configuration is chosen in a way to: a) make the recall for both systems close to the same value, and b) to illustrate that the PR-PM model has much greater potential to increase precision than the PR-LIK system. A similar conclusion can be drawn by checking various different precision and recall configurations in the precision and recall plots for both Isle and GUT Isle corpora.

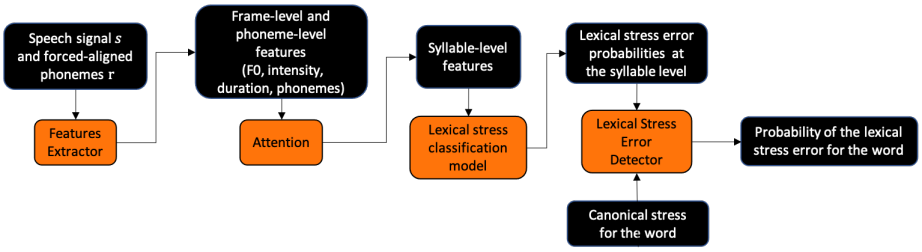


Figure 7. Attention-based model for the detection of lexical stress errors.

5.3. Lexical stress error detection

5.3.1. Experimental setup

The full CAPT learning experience includes both the detection of pronunciation and lexical stress errors. To investigate the potential of speech generation in the lexical stress error detection task, we evaluate the T2S method, which is a simpler version of the S2S method evaluated in Section 5.1.4.

The lexical stress error detection model is trained to measure the benefits of employing synthetic mispronounced speech. The first model, denoted as Att_TTS is based on an attention mechanism and is trained on both human and synthetic speech with pronunciation errors. In this model, 1980 the most popular English words [77] were synthesized with correct and incorrect stress patterns using the method outlined in Section 3.2, and added to the speech corpora of isolated words presented in Section 4.2. The Att_NoTTS model is trained only on human speech. Each of the two models presented has its simpler version without the attention mechanism, marked as NoAtt_TTS and NoAtt_NoTTS. Both models will help to understand whether the benefits of using synthetic pronunciation errors depend on the model capacity.

The accuracy of detecting lexical stress errors is measured in terms of an AUC metric. To be comparable to the study by Ferrer et al. [13], we use precision as an additional metric, while setting recall to 50%.

5.3.2. Overview of the lexical stress detection model

As shown in Figure 7, the lexical stress error detection model consists of three subsystems: Feature Extractor, Attention-based Classification Model, and Lexical Stress Error Detector. The Feature Extractor extracts prosodic features and phonemes from the speech signal **s** and the forced-aligned canonical phonemes **r**. Prosodic features include: F0, intensity [dB SPL] and duration of phonemes. The F0 and intensity features are computed at the frame level. The Attention-based Classification Model uses the attention mechanism [78] to map frame-level and phoneme-level features to a syllable-level representation. It then produces lexical stress error probabilities at the syllable level. The Lexical Stress Error Detector reports a lexical stress error if the expected (canonical) and estimated lexical stress for a given syllable do not match and the corresponding probability is higher than the specified threshold. The detailed architecture of the model is presented in Section 3 of our recent work [23].

The NoAtt_TTS and NoAtt_NoTTS models do not have the attention mechanism. Instead, as a representation at the syllable level, they use the average acoustic feature values for the corresponding syllable nucleus. The hypothesis is that synthetic data will not be beneficial to a simpler model due to its limited capacity.

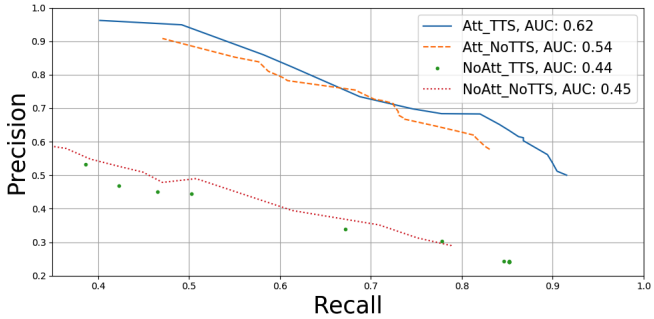


Figure 8. Precision-recall curves for lexical stress error detection models.

Table 9. AUC, precision and recall [%, 95% Confidence Interval] metrics for lexical stress error detection models.

Model	Model with attention	Synthetic mispronunciations	AUC	Precision [%]	Recall[%]
Att_TTS	yes	yes	0.62	94.8 (89.18-98.03)	49.2 (42.13-56.3)
Att_NoTTS	yes	no	0.54	87.85 (80.67-93.02)	49.74 (42.66-56.82)
NoAtt_TTS	no	yes	0.44	44.39 (37.85-51.09)	50.26 (43.18-57.34)
NoAtt_NoTTS	no	no	0.45	48.98 (42.04-55.95)	50.79 (43.70-57.86)
Ferrer et al. [13]	na	na	na	95.00 (na-na)	48.3 (na-na)

5.3.3. Results

Enriching the training set with the incorrectly stressed words increases an AUC score from 0.54 to 0.62 (Att_TTS vs. Att_NoTTS in Figure 8 and Table 9). Data augmentation helps because it increases the number of words with incorrect stress patterns in the training set. This prevents the model from using the strong correlation between phonemes and lexical stress in the correctly stressed words. Using data augmentation in the simpler model without the attention mechanism slightly reduced an AUC score from 0.45 to 0.44 (NoAtt_NoTTS vs NoAtt_TTS). The NoAtt_TTS model has limited capacity due to not using the attention mechanism to model prosodic features, and thus is unable to benefit from synthetic speech.

We compare our results with the work of Ferrer et al. [13]. There were 46.4% (191 out of 411) of incorrectly stressed words in their corpus, well over 9.4% (189 out of 2109) words in our experiment. The fewer lexical stress errors that users make, the more difficult it is to detect them. Under these conditions, we can state that our lexical stress detection model based on T2S generated synthetic speech achieves higher scores in precision and recall compared to the work of Ferrer et al. [13].

6. Conclusions

We propose a new paradigm for detecting pronunciation errors in non-native speech. Rather than focusing on detecting pronunciation errors directly, we reformulate the detection problem as a speech generation task. This approach is based on the assumption that it is easier to generate speech with specific characteristics than to detect those characteristics in speech with limited availability. In this way, we address one of the main problems of the existing CAPT methods, which is the low availability of mispronounced speech for reliable training of pronunciation error detection models.

We present a unified look at three different speech generation techniques for detecting pronunciation errors based on P2P, T2S and S2S conversion. The P2P, T2S, and S2S methods improve the accuracy of detecting pronunciation and lexical stress errors. The methods outperform strong baseline models and establish a new state-of-the-art. The best S2S method outperforms the baseline method [9] by improving the accuracy of detecting pronunciation errors in AUC metric by 41% from 0.528 to 0.749. The S2S method has the ability to control many properties of speech, such as voice timbre, prosody (duration), and pronunciation. This opens the door to the generation of mispronounced speech that can mimic certain aspects of non-native speech, such as voice timbre. The S2S method can be seen as a generalization of the simpler methods, T2S and P2P, providing a general framework for building a first-class models of pronunciation assessment. For better reproducibility, in addition to using publicly available speech corpora, we recorded the GUT Isle corpus of non-native English speech [71]. The corpus is available to other researchers in the field.

In the future, we plan to extend the S2S method in order to generate synthetic speech as close as possible to non-native speech: a) we will extract the voice timbre from the speech of non-native speakers and transfer it to native speech, following the paper of Merritt et al. on text-free voice conversion [79], and b) we will mimic the distribution of pronunciation errors in non-native speech. We expect both changes to increase the accuracy of detecting pronunciation errors in non-native speech. In the long run, we hope to demonstrate that "synthetic speech is all you need" by training the model with synthetic speech only and achieving state-of-the-art results in the pronunciation error detection task. This may revolutionize computer-assisted English L2 learning and CAPT. Moreover, such a paradigm may be transferred to the whole domain of computer-assisted foreign language learning.

References

[1] UNESCO, *If you don't understand, how can you learn?* (2016). Available at https://en.unesco.org/news/40-don-t-access-education-language-they-understand.

[2] Statista, *Most common languages used on the internet as of january 2020, by share of internet users* (2021). Available at https://www.statista.com/statistics/262946/share-of-the-most-common-languages-on-the-internet/.

[3] EF-Education-First, *Ef english proficiency index* (2020). Available at https://www.ef.pl/epi/.

[4] M. Levy and G. Stockwell, *CALL dimensions: Options and issues in computer-assisted language learning*, Routledge, 2013.

[5] A. Mehri Kamrood, M. Davoudi, S. Ghaniabadi, and S.M.R. Amirian, *Diagnosing l2 learners' development through online computerized dynamic assessment*, Computer Assisted Language Learning (2019), pp. 1–30.

[6] A. Neri, O. Mich, M. Gerosa, and D. Giuliani, *The effectiveness of computer assisted pronunciation training for foreign language learning by children*, Computer Assisted Language Learning 21 (2008), pp. 393–408.

[7] E.M. Golonka, A.R. Bowles, V.M. Frank, D.L. Richardson, and S. Freynik, *Technologies for foreign language learning: A review of technology types and their effectiveness*, Computer assisted language learning 27 (2014), pp. 70–105.

[8] C. Tejedor-García, D. Escudero, E. Cámara-Arenas, C. González-Ferreras, and V. Cardeñoso-Payo, *Assessing pronunciation improvement in students of english using a controlled computer-assisted pronunciation tool*, IEEE Transactions on Learning Technologies (2020).

[9] W.K. Leung, X. Liu, and H. Meng, *CNN-RNN-CTC based end-to-end mispronunciation*

detection and diagnosis, in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 8132–8136.

[10] Z. Zhang, Y. Wang, and J. Yang, *Text-conditioned transformer for automatic pronunciation error detection*, Speech Communication 130 (2021), pp. 55–63.

[11] D. Korzekwa, J. Lorenzo-Trueba, S. Zaporowski, S. Calamaro, T. Drugman, and B. Kostek, *Mispronunciation Detection in Non-Native (L2) English with Uncertainty Modeling*, in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 7738–7742.

[12] R. Ai, *Automatic pronunciation error detection and feedback generation for call applications*, in *International Conference on Learning and Collaboration Technologies*. Springer, 2015, pp. 175–186.

[13] L. Ferrer, H. Bratt, C. Richey, H. Franco, V. Abrash, and K. Precoda, *Classification of lexical stress using spectral and prosodic features for computer-assisted language learning systems*, Speech Communication 69 (2015), pp. 31–45.

[14] S.M. Witt and S.J. Young, *Phone-level pronunciation scoring and assessment for interactive language learning*, Speech communication 30 (2000), pp. 95–108.

[15] H. Li, S. Huang, S. Wang, and B. Xu, *Context-Dependent Duration Modeling with Backoff Strategy and Look-Up Tables for Pronunciation Assessment and Mispronunciation Detection*, in *INTERSPEECH 2011, 12th Annual Conference of the International Speech Communication Association, Florence, Italy, August 27-31, 2011*. ISCA, 2011, pp. 1133–1136.

[16] J.Y. Chen and L. Wang, *Automatic lexical stress detection for Chinese learners' of English*, in *2010 7th Intl. Symposium on Chinese Spoken Language Processing*. IEEE, 2010, pp. 407–411.

[17] M.A. Shahin, J. Epps, and B. Ahmed, *Automatic Classification of Lexical Stress in English and Arabic Languages Using Deep Learning.*, in *INTERSPEECH*. 2016, pp. 175–179.

[18] C.M. Bishop, *Pattern recognition*, Machine learning 128 (2006).

[19] G. Huybrechts, T. Merritt, G. Comini, B. Perz, R. Shah, and J. Lorenzo-Trueba, *Low-resource expressive text-to-speech using data augmentation*, in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 6593–6597.

[20] R. Shah, K. Pokora, A. Ezzerg, V. Klimkov, G. Huybrechts, B. Putrycz, D. Korzekwa, and T. Merritt, *Non-Autoregressive TTS with Explicit Duration Modelling for Low-Resource Highly Expressive Speech*, in *Proc. 11th ISCA Speech Synthesis Workshop (SSW 11)*. 2021, pp. 96–101.

[21] A. Fazel, W. Yang, Y. Liu, R. Barra-Chicote, Y. Meng, R. Maas, and J. Droppo, *SynthASR: Unlocking Synthetic Data for Speech Recognition*, in *Proc. Interspeech 2021*. 2021, pp. 896–900.

[22] D. Korzekwa, J. Lorenzo-Trueba, T. Drugman, S. Calamaro, and B. Kostek, *Weakly-Supervised Word-Level Pronunciation Error Detection in Non-Native English Speech*, in *Proc. Interspeech 2021*. 2021, pp. 4408–4412.

[23] D. Korzekwa, R. Barra-Chicote, S. Zaporowski, G. Beringer, J. Lorenzo-Trueba, A. Serafinowicz, J. Droppo, T. Drugman, and B. Kostek, *Detection of Lexical Stress Errors in Non-Native (L2) English with Data Augmentation and Attention*, in *Proc. Interspeech 2021*. 2021, pp. 3915–3919.

[24] K. Li, X. Qian, and H. Meng, *Mispronunciation detection and diagnosis in l2 english speech using multidistribution deep neural networks*, IEEE/ACM Transactions on Audio, Speech, and Language Processing 25 (2016), pp. 193–207.

[25] S. Sudhakara, M.K. Ramanathi, C. Yarra, and P.K. Ghosh, *An Improved Goodness of Pronunciation (GoP) Measure for Pronunciation Evaluation with DNN-HMM System Considering HMM Transition Probabilities.*, in *INTERSPEECH*. 2019, pp. 954–958.

[26] S. Cheng, Z. Liu, L. Li, Z. Tang, D. Wang, and T.F. Zheng, *Asr-free pronunciation assessment*, arXiv preprint arXiv:2005.11902 (2020).

[27] H. Franco, L. Neumeyer, Y. Kim, and O. Ronen, *Automatic pronunciation scoring for*

language instruction, in *1997 IEEE international conference on acoustics, speech, and signal processing*, Vol. 2. IEEE, 1997, pp. 1471–1474.

[28] N. Minematsu, *Pronunciation assessment based upon the phonological distortions observed in language learners' utterances*, in *INTERSPEECH*. 2004.

[29] A.M. Harrison, W.K. Lo, X.j. Qian, and H. Meng, *Implementation of an extended recognition network for mispronunciation detection and diagnosis in computer-assisted pronunciation training*, in *Intl. Workshop on Speech and Language Technology in Education*. 2009.

[30] A. Lee and J.R. Glass, *Pronunciation assessment via a comparison-based system*, in *SLaTE*. 2013.

[31] P. Plantinga and E. Fosler-Lussier, *Towards Real-Time Mispronunciation Detection in Kids' Speech*, in *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2019, pp. 690–696.

[32] S. Sudhakara, M.K. Ramanathi, C. Yarra, A. Das, and P. Ghosh, *Noise robust goodness of pronunciation measures using teacher's utterance*, in *SLaTE*. 2019.

[33] L. Zhang, Z. Zhao, C. Ma, L. Shan, H. Sun, L. Jiang, S. Deng, and C. Gao, *End-to-end automatic pronunciation error detection based on improved hybrid ctc/attention architecture*, Sensors 20 (2020), p. 1809.

[34] J. Chorowski, D. Bahdanau, K. Cho, and Y. Bengio, *End-to-end continuous speech recognition using attention-based recurrent nn: First results*, arXiv preprint arXiv:1412.1602 (2014).

[35] J.K. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, *Attention-based models for speech recognition*, in *Advances in neural information processing systems*. 2015, pp. 577–585.

[36] D. Bahdanau, J. Chorowski, D. Serdyuk, P. Brakel, and Y. Bengio, *End-to-end attention-based large vocabulary speech recognition*, in *2016 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2016, pp. 4945–4949.

[37] Y. Xiao, F. Soong, and W. Hu, *Paired Phone-Posteriors Approach to ESL Pronunciation Quality Assessment*, in *Proc. Interspeech 2018*. 2018, pp. 1631–1635.

[38] M. Nicolao, A.V. Beeston, and T. Hain, *Automatic assessment of English learner pronunciation using discriminative classifiers*, in *2015 IEEE Intl. Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2015, pp. 5351–5355.

[39] J. Wang, Y. Qin, Z. Peng, and T. Lee, *Child Speech Disorder Detection with Siamese Recurrent Network Using Speech Attribute Features.*, in *INTERSPEECH*. 2019, pp. 3885–3889.

[40] X. Qian, H. Meng, and F. Soong, *Capturing L2 segmental mispronunciations with joint-sequence models in computer-aided pronunciation training (CAPT)*, in *2010 7th Intl. Symposium on Chinese Spoken Language Processing*. IEEE, 2010, pp. 84–88.

[41] S.B. Needleman and C.D. Wunsch, *A general method applicable to the search for similarities in the amino acid sequence of two proteins*, Journal of molecular biology 48 (1970), pp. 443–453.

[42] R. Duan, T. Kawahara, M. Dantsuji, and H. Nanjo, *Cross-lingual transfer learning of non-native acoustic modeling for pronunciation error detection and diagnosis*, IEEE/ACM Transactions on Audio, Speech, and Language Processing 28 (2019), pp. 391–401.

[43] B.C. Yan, M.C. Wu, H.T. Hung, and B. Chen, *An End-to-End Mispronunciation Detection System for L2 English Speech Leveraging Novel Anti-Phone Modeling*, in *Proc. Interspeech 2020*. 2020, pp. 3032–3036.

[44] X. Xu, Y. Kang, S. Cao, B. Lin, and L. Ma, *Explore wav2vec 2.0 for Mispronunciation Detection*, in *Proc. Interspeech 2021*. 2021, pp. 4428–4432.

[45] L. Peng, K. Fu, B. Lin, D. Ke, and J. Zhan, *A Study on Fine-Tuning wav2vec2.0 Model for the Task of Mispronunciation Detection and Diagnosis*, in *Proc. Interspeech 2021*. 2021, pp. 4448–4452.

[46] B. Lin and L. Wang, *Deep Feature Transfer Learning for Automatic Pronunciation Assessment*, in *Proc. Interspeech 2021*. 2021, pp. 4438–4442.

[47] J. Field, *Intelligibility and the listener: The role of lexical stress*, TESOL quarterly 39 (2005), pp. 399–423.

[48] A. Lepage and M.G. Busà, *Intelligibility of English L2: The effects of incorrect word stress placement and incorrect vowel reduction in the speech of French and Italian learners of English*, in *Proc. of the Intl. Symposium on the Acquisition of Second Language Speech.* 2014, 2014, pp. 387–400.

[49] Y.J. Jung, S.C. Rhee, *et al.*, *Acoustic analysis of english lexical stress produced by korean, japanese and taiwanese-chinese speakers*, Phonetics and Speech Sciences 10 (2018), pp. 15–22.

[50] D.R.v. Bergem, *Acoustic and lexical vowel reduction*, in *Phonetics and Phonology of Speaking Styles.* 1991.

[51] K. Li, S. Mao, X. Li, Z. Wu, and H. Meng, *Automatic lexical stress and pitch accent detection for l2 english speech using multi-distribution deep neural networks*, Speech Communication 96 (2018), pp. 28–36.

[52] J. Zhao, H. Yuan, J. Liu, and S. Xia, *Automatic lexical stress detection using acoustic features for computer assisted language learning*, Proc. APSIPA ASC (2011), pp. 247–251.

[53] N. Chen and Q. He, *Using nonlinear features in automatic English lexical stress detection*, in *2007 Intl. Conference on Computational Intelligence and Security Workshops (CISW 2007).* IEEE, 2007, pp. 328–332.

[54] K. Li, X. Qian, S. Kang, and H. Meng, *Lexical stress detection for L2 English speech using deep belief networks.*, in *Interspeech.* 2013, pp. 1811–1815.

[55] M.K. Ramanathi, C. Yarra, and P.K. Ghosh, *ASR Inspired Syllable Stress Detection for Pronunciation Evaluation Without Using a Supervised Classifier and Syllable Level Features.*, in *INTERSPEECH.* 2019, pp. 924–928.

[56] J. Badenhorst and F. De Wet, *The limitations of data perturbation for ASR of learner data in under-resourced languages*, in *2017 Pattern Recognition Association of South Africa and Robotics and Mechatronics (PRASA-RobMech).* IEEE, 2017, pp. 44–49.

[57] V.V. Eklund, *Data augmentation techniques for robust audio analysis*, Master's thesis, Tampere University, 2019.

[58] A. Lee, *et al.*, *Language-independent methods for computer-assisted pronunciation training*, Ph.D. diss., Massachusetts Institute of Technology, 2016.

[59] S. Komatsu and M. Sasayama, *Speech Error Detection depending on Linguistic Units*, in *Proceedings of the 2019 3rd International Conference on Natural Language Processing and Information Retrieval.* 2019, pp. 75–79.

[60] K. Fu, J. Lin, D. Ke, Y. Xie, J. Zhang, and B. Lin, *A full text-dependent end to end mispronunciation detection and diagnosis with easy data augmentation techniques*, arXiv preprint arXiv:2104.08428 (2021).

[61] B.C. Yan, S.W.F. Jiang, F.A. Chao, and B. Chen, *Maximum f1-score training for end-to-end mispronunciation detection and diagnosis of l2 english speech*, arXiv preprint arXiv:2108.13816 (2021).

[62] G. Huang, A. Gorin, J.L. Gauvain, and L. Lamel, *Machine translation based data augmentation for cantonese keyword spotting*, in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).* IEEE, 2016, pp. 6020–6024.

[63] D. Koller and N. Friedman, *Probabilistic graphical models: principles and techniques*, MIT press, 2009.

[64] J. Latorre, J. Lachowicz, J. Lorenzo-Trueba, T. Merritt, T. Drugman, S. Ronanki, and V. Klimkov, *Effect of data reduction on sequence-to-sequence neural tts*, in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).* IEEE, 2019, pp. 7075–7079.

[65] A. Oord, Y. Li, I. Babuschkin, K. Simonyan, O. Vinyals, K. Kavukcuoglu, G. Driessche, E. Lockhart, L. Cobo, F. Stimberg, *et al.*, *Parallel wavenet: Fast high-fidelity speech synthesis*, in *International conference on machine learning.* PMLR, 2018, pp. 3918–3926.

[66] Y. Jiao, A. Gabryś, G. Tinchev, B. Putrycz, D. Korzekwa, and V. Klimkov, *Universal neural vocoding with parallel wavenet*, in *ICASSP 2021-2021 IEEE International Conference*

on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2021, pp. 6044–6048.

[67] M.I. Jordan, Z. Ghahramani, T.S. Jaakkola, and L.K. Saul, *An introduction to variational methods for graphical models*, Machine learning 37 (1999), pp. 183–233.

[68] J.S. Garofolo, L.F. Lamel, W.M. Fisher, J.G. Fiscus, and D.S. Pallett, *Darpa timit acoustic-phonetic continous speech corpus cd-rom. nist speech disc 1-1.1*, STIN 93 (1993), p. 27403.

[69] H. Zen, V. Dang, R. Clark, Y. Zhang, R.J. Weiss, Y. Jia, Z. Chen, and Y. Wu, *LibriTTS: A Corpus Derived from LibriSpeech for Text-to-Speech*, in *Proc. Interspeech 2019*. 2019, pp. 1526–1530.

[70] E. Atwell, P. Howarth, and D. Souter, *The isle corpus: Italian and german spoken learner's english*, ICAME Journal: Intl. Computer Archive of Modern and Medieval English Journal 27 (2003), pp. 5–18.

[71] D. Weber, S. Zaporowski, and D. Korzekwa, *Constructing a Dataset of Speech Recordings with Lombard Effect*, in *24th IEEE SPA*. 2020.

[72] J. Kominek and A.W. Black, *The CMU Arctic speech databases*, in *Fifth ISCA workshop on speech synthesis*. 2004.

[73] A. Porzuczek and A. Rojczyk, *English word stress in polish learners speech production and metacompetence*, Research in Language 15 (2017), pp. 313–323.

[74] A. Graves, *Connectionist temporal classification*, in *Supervised Sequence Labelling with Recurrent Neural Networks*, Springer, 2012, pp. 61–93.

[75] I. Sutskever, O. Vinyals, and Q.V. Le, *Sequence to sequence learning with neural networks*, in *Advances in neural information processing systems*. 2014, pp. 3104–3112.

[76] A.E. Hieke, *Linking as a marker of fluent speech*, Language and Speech 27 (1984), pp. 343–354.

[77] J.B. Michel, Y.K. Shen, A.P. Aiden, A. Veres, M.K. Gray, J.P. Pickett, D. Hoiberg, D. Clancy, P. Norvig, J. Orwant, *et al.*, *Quantitative analysis of culture using millions of digitized books*, science 331 (2011), pp. 176–182.

[78] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, L. Kaiser, and I. Polosukhin, *Attention is all you need*, in *Advances in neural information processing systems*. 2017, pp. 5998–6008.

[79] T. Merritt, A. Ezzerg, P. Biliński, M. Proszewska, K. Pokora, R. Barra-Chicote, and D. Korzekwa, *Text-free non-parallel many-to-many voice conversion using normalising flows*, Accepted to Acoustics, Speech and Signal Processing (ICASSP) (2022).

INTERSPEECH 2021
30 August – 3 September, 2021, Brno, Czechia

Weakly-supervised word-level pronunciation error detection in non-native English speech

Daniel Korzekwa[1,2], *Jaime Lorenzo-Trueba*[3], *Thomas Drugman*[3], *Shira Calamaro*[3], *Bozena Kostek*[2]

[1]Amazon, Poland
[2]Gdansk University of Technology, Faculty of ETI, Poland
[3]Amazon, UK
korzekwa@amazon.com

Abstract

We propose a weakly-supervised model for word-level mispronunciation detection in non-native (L2) English speech. To train this model, phonetically transcribed L2 speech is not required and we only need to mark mispronounced words. The lack of phonetic transcriptions for L2 speech means that the model has to learn only from a weak signal of word-level mispronunciations. Because of that and due to the limited amount of mispronounced L2 speech, the model is more likely to overfit. To limit this risk, we train it in a multi-task setup. In the first task, we estimate the probabilities of word-level mispronunciation. For the second task, we use a phoneme recognizer trained on phonetically transcribed L1 speech that is easily accessible and can be automatically annotated. Compared to state-of-the-art approaches, we improve the accuracy of detecting word-level pronunciation errors in AUC metric by 30% on the GUT Isle Corpus of L2 Polish speakers, and by 21.5% on the Isle Corpus of L2 German and Italian speakers.

Index Terms: automated pronunciation assessment, speech processing, second-language learning, deep learning

1. Introduction

It has been shown that Computer-Assisted Pronunciation Training (CAPT) helps people practice and improve pronunciation skills [1, 2]. Despite significant progress over the last two decades, standard methods are still unable to detect mispronunciations with high accuracy. These methods can detect phoneme-level mispronunciations at about 60% precision and 40%-80% recall [3, 4, 5]. By further raising precision we can lower the risk of providing incorrect feedback, whereas with higher recall, we can detect more mispronunciation errors.

Standard methods aim at recognizing the phonemes pronounced by a speaker and compare them with expected (canonical) pronunciation of correctly pronounced speech. Any mismatch between recognized and canonical phonemes yields a pronunciation error at the phoneme level. Phoneme recognition-based approaches rely on phonetically transcribed speech labeled by human listeners. Human-based transcription is a laborious task, especially, in the case of L2 speech where listeners have to identify mispronunciations. Sometimes, it might be even impossible to transcribe L2 speech because different languages have different phoneme sets and it is unclear which phonemes were pronounced by the speaker.

Phoneme recognition-based approaches generally fall into two categories. The first category uses forced-alignment techniques [6, 7, 8, 9] based on the work by Franco et al. [10] and the Goodness of Pronunciation (GOP) method [11]. The GOP uses Bayesian inference to find the most likely alignment between canonical phonemes and the corresponding audio signal (forced alignment). Then, the GOP uses the likelihoods of the aligned audio signal as an indicator for mispronounced phonemes. In the second category there are methods that recognize phonemes pronounced by a speaker purely from a speech signal, and only then align them with canonical phonemes [12, 13, 14, 15, 16]. Techniques falling into both categories can be complemented with the use of a reference signal obtained either from a database of speech [17, 18, 19] or generated from phonetic representation [4, 20].

There are two challenges for the phoneme recognition approaches. First, phonemes pronounced by a speaker have to be recognized accurately, which has been shown to be difficult [5, 21, 22, 23]. Second, standard approaches expect only a single canonical pronunciation of a given text, but this assumption does not always hold true due to phonetic variability of speech. In [4], we addressed these problems by incorporating a pronunciation model of L1 speech, but this approach still relies on phonetically transcribed L2 speech.

In this paper, we introduce a novel model (noted as WEAKLY-S) for the detection of word-level pronunciation errors that does not require phonetically transcribed L2 speech. The model produces the probabilities of mispronunciation for all words, conditioned on a spoken sentence and canonical phonemes. Mispronunciation error types include any of phoneme replacement, addition, deletion or unknown speech sound. During training, the model is weakly supervised, in the sense that we only mark mispronounced words in L2 speech and the data do not have to be phonetically transcribed. Due to the limited availability of L2 speech and the fact it is not phonetically transcribed, the model is more likely to overfit. To solve this problem, we train the model in a multi-task setup. In addition to a primary task of word-level mispronunciation detection, we use a phoneme recognizer trained on automatically transcribed L1 speech for the secondary task. Both tasks share common parts of the model, which makes the primary task less likely to overfit. Additionally, we address the overfitting problem with synthetically generated pronunciation errors that are derived from L1 speech.

Leung et al. [3] used a phoneme recognizer based on Connectionist Temporal Classification (CTC) for pronunciation error detection. Instead, we use an attention-based phoneme recognizer following Chorowski et al. [22] so that we can regularize the model by both tasks sharing a common component (attention). With a CTC-based phoneme recognizer it would not be possible because this technique does not use attention that could be shared between both tasks. Zhang et al. [5] employed a multi-task model for pronunciation assessment, but with two important differences. First, they use a Needleman-Wunsch al-

Copyright © 2021 ISCA

http://dx.doi.org/10.21437/Interspeech.2021-38

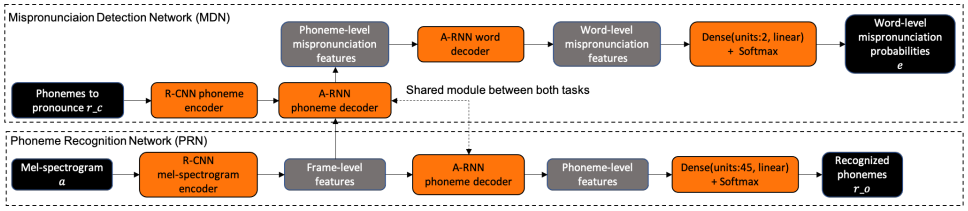


Figure 1: *Neural network architecture of the WEAKLY-S model for word-level pronunciation error detection.*

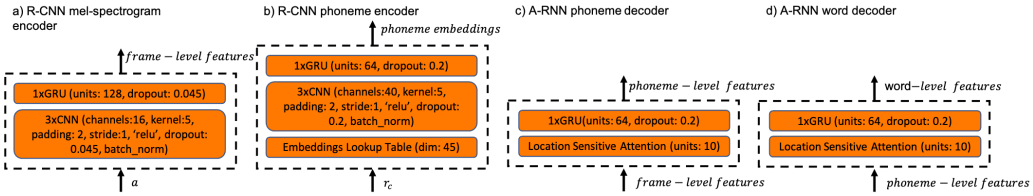


Figure 2: *Details of the neural network architecture of the WEAKLY-S model for word-level pronunciation error detection.*

gorithm [24] for aligning canonical and recognized sequences of phonemes, but this algorithm cannot be tuned towards sequences of phonemes. We use an attention mechanism that automatically maps the speech signal to the sequence of word-level pronunciation errors. Second, Zhang et al. detect pronunciation errors at the phoneme level and they expect L2 speech to be phonetically transcribed. This differs from our method of recognizing pronunciation errors at the word level with no need for phonetic transcriptions of L2 speech. To the best of our knowledge, this is the first approach to train word-level pronunciation error detection model that does not require phonetically transcribed L2 speech and can be optimized directly towards word-level mispronunciation detection.

2. Proposed Model

2.1. Model Definition

The model is made of two sub-networks: *i)* a word-level Mispronunciations Detection Network (MDN) detects word-level pronunciation errors $\mathbf{e}$ from the audio signal $\mathbf{a}$ and canonical phonemes $\mathbf{r_c}$, *ii)* a Phoneme Recognition Network (PRN) recognizes phonemes $\mathbf{r_o}$ pronounced by a speaker from the audio signal $\mathbf{a}$ (Fig. 1).

More formally, let us define the following variables: $\mathbf{a}$ - speech signal represented by a mel-spectrogram, $\mathbf{r_c}$ - canonical phonemes that the speaker was expected to pronounce, $\mathbf{r_o}$ - phonemes pronounced, and $\mathbf{e}$ - the probabilities of mispronouncing words in the spoken sentence. The model outputs the probabilities of word-level mispronunciation, denoted as $\mathbf{e} \sim p(\mathbf{e}|\mathbf{a}, \mathbf{r_c}, \boldsymbol{\theta})$, where $\boldsymbol{\theta}$ represent parameters of the model.

We train the WEAKLY-S model in a multi-task setup. In addition to the primary task $\mathbf{e}$, we use a phoneme recognizer denoted as $\mathbf{r_o} \sim p(\mathbf{r_o}|\mathbf{a}, \boldsymbol{\theta})$ for the secondary task. The parameters $\boldsymbol{\theta}$ are shared between both tasks, which makes the MDN less likely to overfit. We define the loss function as the sum of two losses: a word-level mispronunciation loss and a phoneme recognition loss. Its formulation for the *ith* training example is presented in Eq. 1. We train the model using two types of training data: phonetically transcribed L1 speech (both losses are used) and untranscribed L2 speech (only the mispronunciation loss is used). Having a separate loss for word-level mispronunciation lets us train the model from speech data that are not phonetically transcribed.

$$\mathcal{L}(\boldsymbol{\theta}) = log(p(\mathbf{e}|\mathbf{a}, \mathbf{r_c}, \boldsymbol{\theta})) + log(p(\mathbf{r_o}|\mathbf{a}, \boldsymbol{\theta})) \quad (1)$$

2.2. Neural Network Details

Following Sutskever et al. [25], the MDN network encodes the mel-spectrogram $\mathbf{a}$ and the canonical phonemes $\mathbf{r_c}$ with Recurrent Convolutional Neural Network (RCNN) encoders (Fig. 2a and Fig. 2b). These encoded representations are passed into an attention-based [26] Recurrent Neural Network (A-RNN) decoder (Fig. 2c) that generates phoneme-level mispronunciation features. Phoneme-level features are transformed into word-level features (Fig. 2d) based on an attention mechanism and these finally are used for computing word-level mispronunciation probabilities $\mathbf{e}$.

The PRN recognizes phonemes $\mathbf{r_o}$ pronounced by the speaker. It is similar to the attention-based phoneme recognizer by Chorowski et al. [22]. To generate phoneme-level features, it uses the same RCNN mel-spectrogram encoder and A-RNN decoder as the MDN. The only difference is that the A-RNN decoder is not conditioned on canonical phonemes. Phoneme-level features are transformed to the probabilities of pronounced phonemes. We added a phoneme recognition task due to the limited amount of L2 speech annotated with word-level mispronunciations. Without it, the MDN would be prone to overfitting if it was trained only on its own. By sharing common parts between both models, the PRN acts as a backbone for the MDN and makes it more robust.

The model was implemented in MxNet framework [27] and tuned for hyper-parameters with AutoGluon Bayesian optimization framework [28]. The model was first pretrained on L1 and L2 speech corpora and then the MDN part was fine-tuned only on L2 speech data. We used the Adam optimizer with learning rate 0.001 and gradient clipping 5. Training data were segmented into buckets with batch size 32, using GluonCV [29]. The A-RNN phoneme and word decoders are based on Location Sensitive Attention by Chorowski et al. [22].

3. Experiments

We present three experiments. We start with comparing our model against state-of-the-art approaches in the task of word-

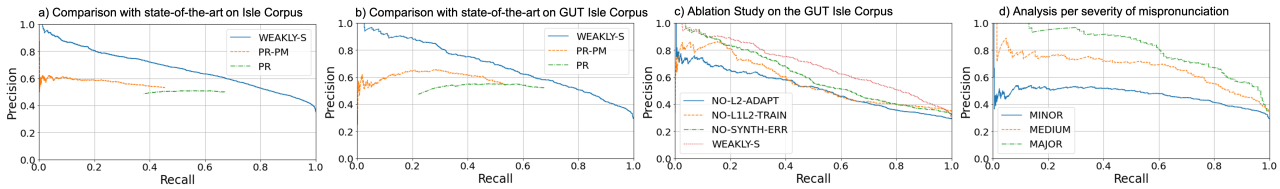


Figure 3: *Precision-recall curves for the WEAKLY-S and baseline models, PR-PM and PR, (a) tested on German and Italian speakers and (b) Polish speakers. (c) Ablation study on the GUT Isle corpus. (d) Analysis of mispronunciation severity levels.*

level mispronunciation detection. In an ablation study we analyze which elements of the model contribute the most to its performance. Finally, we analyze how the severity of pronunciation error affects the accuracy of the model.

3.1. Speech Corpora and Metrics

In our experiments, we use a combination of L1 and L2 English speech. L1 speech is obtained from TIMIT [30] and LibriTTS [31] corpora. L2 data come from the Isle [32] corpus (German and Italian speakers) and the GUT Isle [33] corpus (Polish speakers). In total, we collected 102,812 utterances, summarized in Table 1. We split the data into training and test sets, holding out 28 L2 speakers (11 German, 11 Italian, and 6 Polish) only for testing the performance of the model.

The L2 corpus of Polish speakers was annotated for word-level pronunciation errors by 5 native English speakers. Annotators marked mispronounced words and indicated their severity levels using one of the three possible values: 1 - MINOR, 2 - MEDIUM, 3 - MAJOR. The Isle corpus of German and Italian speakers comes with phoneme level mispronunciations. Words with at least one mispronounced phoneme were automatically marked as mispronounced. The Isle corpus is not mapped to severity levels of mispronunciations. In total, there are 35,555 L2 words, including 8035 mispronounced words. All data were re-sampled to 16 kHz.

We extended the train set with 292,242 utterances of L1 speech with synthetically generated pronunciation errors. We use a simple approach of perturbing phonetic transcription for the corresponding speech audio. First, we sample these utterances with replacement from L1 corpora of human speech. Then, for each utterance, we replace phonemes with random phonemes with a probability of 0.2. In [34] we found that generating incorrectly stressed speech using Text-To-Speech (TTS) improves the accuracy of detecting lexical stress errors in L2 speech. Although, as opposed to using TTS, we create pronunciation errors by perturbing the text, we expect this simpler approach should still help recognizing word-level pronunciation errors.

Table 1: *Summary of speech corpora used in experiments. * - audiobooks read by volunteers from all over the world [31]*

Native Language	Hours	Speakers
English	90.47	640
Unknown*	19.91	285
German and Italian	13.41	46
Polish	1.49	12

To evaluate our model, we use three standard metrics: Area Under Curve (AUC), precision and recall. The AUC metric provides an overall performance of the model accounting for all possible trade offs between precision and recall. Precision-recall plots illustrate relations between both metrics. Complementary, to analyze precision, in all our experiments we consistently fix recall at the value of 0.4 to be comparable with two baseline models that do not cover the whole range of recall values (see Section 3.2).

3.2. Comparison with State-of-the-Art

We compare our proposed WEAKLY-S model against two state-of-the-art baselines. The phoneme recognizer (PR) model by Leung et al. [3] is our first baseline. The PR is based on CTC loss [35] and it outperforms multiple alternative approaches for pronunciation assessment. The original CTC-based model uses a hard likelihood threshold applied to recognized phonemes. To compare it with two other models, following our work in [4], we replaced hard likelihood threshold with a soft threshold. The second baseline is the PR extended by a pronunciation model (PR-PM model [4]). The pronunciation model accounts for phonetic variability of speech produced by native speakers, which results in higher precision of detecting pronunciation errors.

The results are presented in Fig. 3a, Fig. 3b and Table 2. The WEAKLY-S model turns out to outperform the second best model in AUC by 30% from 52.8 to 68.63 and in precision by 23% from 61.21 to 75.25 on the GUT Isle Corpus of Polish speakers. We observe similar improvements on the Isle Corpus of German and Italian speakers.

Table 2: *Accuracy metrics of detecting word-level pronunciation errors. WEAKLY-S vs baseline models.*

Model	AUC [%]	Precision [%,95%CI]	Recall [%,95%CI]
Isle corpus (German and Italian)			
PR	55.52	49.39 (47.59-51.19)	40.20 (38.62-41.81)
PR-PM	48.00	54.20 (52.32-56.08)	40.20 (38.62-41.81)
WEAKLY-S	**67.47**	71.94 (69.96, 73.87)	40.14 (38.56, 41.75)
GUT Isle corpus (Polish)			
PR	52.8	54.91 (50.53-59.24)	40.29 (36.66-44.02)
PR-PM	50.50	61.21 (56.63-65.65)	40.15 (36.51-43.87)
WEAKLY-S	**68.63**	75.25 (71.67-78.59)	40.38 (37.52-43.29)

One difference between our model and the two baselines is that they both use the Needleman-Wunsch algorithm [24] for aligning canonical and recognized sequences of phonemes. This is a dynamic programming-based algorithm for comparing biological sequences and cannot be optimized for mispronunciation errors. Our model automatically finds the mapping between regions in the speech signal and the corresponding canonical phonemes, and then identifies word-level mispronunciation errors. In this way, we eliminate the Needleman-Wunsch algorithm as a possible source of error.

The second difference is the use of phonetic transcriptions for L2 speech. Both baselines use automatic transcriptions provided by an Amazon-proprietary grapheme-to-phoneme model.

In [4] we found that for the PR and PR-PM models it is better to use automatically transcribed L2 speech for training a phoneme recognizer than not use L2 speech at all. Note that these automatic transcriptions will include phoneme mistakes for mispronounced speech. Our model does not use transcriptions of L2 speech, and instead it is guided by the word-level pronunciation errors of L2 speech in a weakly-supervised fashion.

3.3. Ablation Study

We now investigate which elements of our new model contribute the most to its performance. Along with the WEAKLY-S model, we trained three additional variants, each with a certain feature removed. The NO-L2-ADAPT variant does not fine-tune the model on L2 speech, though it is still exposed to L2 speech while it is trained on a combined corpus of L1 and L2 speech. The NO-L1L2-TRAIN model is not trained on L1/L2 speech, and fine-tuning on L2 speech starts from scratch. It means that the model will not use a large amount of phonetically transcribed L1 speech data and ultimately the secondary task of the phoneme recognizer will not be used. In the NO-SYNTH-ERR model, we exclude synthetic samples of mispronounced L1 speech. It significantly reduces the amount of incorrectly pronounced words used during training from 1,129,839 to only 5,273 L2 words.

L2 Fine-tuning (NO-L2-ADAPT) is the most important factor that contributes to the performance of the model (Fig. 3c and Table 3), with an AUC of 51.72% compared to 68.63% for the full model. Training the model on both L2 and L1 speech together is not sufficient. We think it is because L2 speech accounts for less than 1% of the training data and the model naturally leans towards L1 speech. The second most important feature is training the model on a combined set of L1 and L2 speech (NO-L1L2-TRAIN), with AUC of 56.46%. L1 speech accounts for more than 99% of the training data. These data are also phonetically transcribed, and therefore can be used for the phoneme recognition task. The phoneme recognition task acts as a 'backbone' and reduces the effect of overfitting in the main task of detecting word pronunciation errors. Finally, excluding synthetically generated pronunciation errors (NO-SYNTH-ERR) reduces the AUC from 68.63% to 61.54%.

Table 3: *Ablation study for the GUT Isle corpus.*

Model	AUC [%]	Precision [%]	Recall [%]
NO-L2-ADAPT	51.72	57.89	40.11
NO-L1L2-TRAIN	56.46	59.73	40.20
NO-SYNTH-ERR	61.54	67.22	40.38
WEAKLY-S	**68.63**	75.25	40.38

3.4. Severity of Mispronunciation

When providing feedback to the L2 speaker about mispronounced words, we want to reflect the severity of mispronunciation, in order to focus on more severe errors and not report them all at once. We segment pronunciation errors into three categories: LOW, MEDIUM and HIGH, based on an inter-tester agreement of annotating sentences for word-level mispronunciations. Mispronounced words with less than 40% inter-tester agreement belong to the LOW category, between 40% and 80% to MIDDLE, and over 80% to HIGH. We validated that the proposed inter-tester agreement bands are well correlated with explicit listener opinions on the severity of mispronunciation, as shown in Table 4. This result shows that data on mispronunciation severity can be derived automatically, without the need to collect it.

Table 4: *Severity of mispronunciation by inter-tester agreement for the GUT Isle Corpus. 1 - MINOR, 2 - MEDIUM, 3 - MAJOR.*

Inter-tester agreement	Severity [mean and 95% CI]
LOW (Less than 40%)	1.32 (1.28-1.35)
MEDIUM (Between 40% and 80%)	1.58 (1.54-1.62)
HIGH(Higher than 80%)	2.08 (2.03-2.13)

We aim at detecting the words of HIGH inter-tester agreement with higher precision to provide more relevant feedback to L2 speakers. To make AUC, precision, and recall metrics comparable between different levels of inter-tester agreement, we enforce the ratio of mispronounced words across all categories to the same level of 29.2% by randomly down-sampling correctly pronounced words. This value is the proportion of mispronounced words across all inter-tester agreement levels in the GUT Isle Corpus. We observe that we can detect pronunciation errors of HIGH inter-tester agreement with 91.67% precision at 40.38% recall (Fig. 3d and Table 5). By segmenting pronunciation errors into three difference bands, we can report to a language learner only the errors of HIGH inter-tester agreement, and improve their learning experience.

Table 5: *Accuracy metrics for different severity levels of mispronunciation for the GUT Isle Corpus.*

Inter-test agreement	AUC [%]	Precision [%]	Recall [%]
LOW	46.99	51.84	40.48
MEDIUM	66.90	71.89	40.80
HIGH	81.48	91.67	40.31

4. Conclusions and Future Work

We proposed a model for detecting pronunciation errors in English that can be trained from L2 speech labeled only for word-level mispronunciations. The data do not have to be phonetically transcribed. The model outperforms state-of-the-art models in AUC metric on the GUT Isle Corpus of Polish speakers and the Isle Corpus of German and Italian speakers. The limited amount of L2 speech and the lack of phonetically transcribed speech makes this model prone to overfitting. We overcame this issue by proposing a multi-task training with two tasks: a word-level pronunciation error detector trained on L1 and L2 speech, and a phoneme recognizer trained on L1 speech. The most important factors that contribute to the model accuracy are: *i)* fine-tuning on L2 speech, *ii)* pre-training on a joined corpus of L1 and L2 speech, and *iii)* use of synthetically generated pronunciation errors.

The level of inter-tester agreement in annotating pronunciation errors correlates with explicit human opinions about the severity of mispronunciation. By detecting pronunciation errors only for high inter-tester agreement, we may significantly lower the number of false positives reported to a language learner.

In the future, we want to experiment with discrete phoneme representations such as Vector-Quantized Variational-Auto-Encoder (VQ-VAE) [36, 37], which should fit better to discrete nature of phonemes. Second, we plan to generate synthetic mispronounced speech, which is motivated by our recent work on using speech synthesis for generating lexical stress speech errors [34].

5. References

[1] A. Neri, O. Mich, M. Gerosa, and D. Giuliani, "The effectiveness of computer assisted pronunciation training for foreign language learning by children," *Computer Assisted Language Learning*, vol. 21, no. 5, pp. 393–408, 2008.

[2] C. Tejedor-García, D. Escudero, E. Cámara-Arenas, C. González-Ferreras, and V. Cardeñoso-Payo, "Assessing pronunciation improvement in students of english using a controlled computer-assisted pronunciation tool," *IEEE Transactions on Learning Technologies*, 2020.

[3] W.-K. Leung, X. Liu, and H. Meng, "Cnn-rnn-ctc based end-to-end mispronunciation detection and diagnosis," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 8132–8136.

[4] D. Korzekwa, J. Lorenzo-Trueba, S. Zaporowski, S. Calamaro, T. Drugman, and B. Kostek, "Mispronunciation detection in non-native (l2) english with uncertainty modeling," *arXiv preprint arXiv:2101.06396, accepted to ICASSP 2021*, 2021.

[5] Z. Zhang, Y. Wang, and J. Yang, "Text-conditioned transformer for automatic pronunciation error detection," *arXiv preprint arXiv:2008.12424*, 2020.

[6] H. Li, S. Huang, S. Wang, and B. Xu, "Context-dependent duration modeling with backoff strategy and look-up tables for pronunciation assessment and mispronunciation detection," in *INTERSPEECH 2011, 12th Annual Conference of the International Speech Communication Association, Florence, Italy, August 27-31, 2011*. ISCA, 2011, pp. 1133–1136.

[7] K. Li, X. Qian, and H. Meng, "Mispronunciation detection and diagnosis in l2 english speech using multidistribution deep neural networks," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 1, pp. 193–207, 2016.

[8] S. Sudhakara, M. K. Ramanathi, C. Yarra, and P. K. Ghosh, "An improved goodness of pronunciation (gop) measure for pronunciation evaluation with dnn-hmm system considering hmm transition probabilities." in *INTERSPEECH*, 2019, pp. 954–958.

[9] S. Cheng, Z. Liu, L. Li, Z. Tang, D. Wang, and T. F. Zheng, "Asr-free pronunciation assessment," *arXiv preprint arXiv:2005.11902*, 2020.

[10] H. Franco, L. Neumeyer, Y. Kim, and O. Ronen, "Automatic pronunciation scoring for language instruction," in *1997 IEEE international conference on acoustics, speech, and signal processing*, vol. 2. IEEE, 1997, pp. 1471–1474.

[11] S. M. Witt and S. J. Young, "Phone-level pronunciation scoring and assessment for interactive language learning," *Speech communication*, vol. 30, no. 2-3, pp. 95–108, 2000.

[12] N. Minematsu, "Pronunciation assessment based upon the phonological distortions observed in language learners' utterances," in *INTERSPEECH*, 2004.

[13] A. M. Harrison, W.-K. Lo, X.-j. Qian, and H. Meng, "Implementation of an extended recognition network for mispronunciation detection and diagnosis in computer-assisted pronunciation training," in *Intl. Workshop on Speech and Language Technology in Education*, 2009.

[14] A. Lee and J. R. Glass, "Pronunciation assessment via a comparison-based system," in *SLaTE*, 2013.

[15] P. Plantinga and E. Fosler-Lussier, "Towards real-time mispronunciation detection in kids' speech," in *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2019, pp. 690–696.

[16] S. Sudhakara, M. K. Ramanathi, C. Yarra, A. Das, and P. Ghosh, "Noise robust goodness of pronunciation measures using teacher's utterance," in *SLaTE*, 2019.

[17] Y. Xiao, F. K. Soong, and W. Hu, "Paired phone-posteriors approach to esl pronunciation quality assessment," in *bdl*, 2018, vol. 1, no. 782d, p. 3.

[18] M. Nicolao, A. V. Beeston, and T. Hain, "Automatic assessment of english learner pronunciation using discriminative classifiers," in *2015 IEEE Intl. Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2015, pp. 5351–5355.

[19] J. Wang, Y. Qin, Z. Peng, and T. Lee, "Child speech disorder detection with siamese recurrent network using speech attribute features." in *INTERSPEECH*, 2019, pp. 3885–3889.

[20] X. Qian, H. Meng, and F. Soong, "Capturing l2 segmental mispronunciations with joint-sequence models in computer-aided pronunciation training (capt)," in *2010 7th Intl. Symposium on Chinese Spoken Language Processing*. IEEE, 2010, pp. 84–88.

[21] J. Chorowski, D. Bahdanau, K. Cho, and Y. Bengio, "End-to-end continuous speech recognition using attention-based recurrent nn: First results," *arXiv preprint arXiv:1412.1602*, 2014.

[22] J. K. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, "Attention-based models for speech recognition," in *Advances in neural information processing systems*, 2015, pp. 577–585.

[23] D. Bahdanau, J. Chorowski, D. Serdyuk, P. Brakel, and Y. Bengio, "End-to-end attention-based large vocabulary speech recognition," in *2016 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2016, pp. 4945–4949.

[24] S. B. Needleman and C. D. Wunsch, "A general method applicable to the search for similarities in the amino acid sequence of two proteins," *Journal of molecular biology*, vol. 48, no. 3, pp. 443–453, 1970.

[25] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Advances in neural information processing systems*, 2014, pp. 3104–3112.

[26] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[27] T. e. a. Chen, "Mxnet: A flexible and efficient machine learning library for heterogeneous distributed systems," *arXiv preprint arXiv:1512.01274*, 2015.

[28] N. Erickson, J. Mueller, A. Shirkov, H. Zhang, P. Larroy, M. Li, and A. Smola, "Autogluon-tabular: Robust and accurate automl for structured data," *arXiv preprint arXiv:2003.06505*, 2020.

[29] J. Guo *et al.*, "Gluoncv and gluonnlp: Deep learning in computer vision and natural language processing." *Journal of Machine Learning Research*, vol. 21, no. 23, pp. 1–7, 2020.

[30] J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, and D. S. Pallett, "Darpa timit acoustic-phonetic continous speech corpus cd-rom. nist speech disc 1-1.1," *STIN*, vol. 93, p. 27403, 1993.

[31] H. Zen, V. Dang, R. Clark, Y. Zhang, R. J. Weiss, Y. Jia, Z. Chen, and Y. Wu, "LibriTTS: A Corpus Derived from LibriSpeech for Text-to-Speech," in *Proc. Interspeech 2019*, 2019, pp. 1526–1530.

[32] E. Atwell, P. Howarth, and D. Souter, "The isle corpus: Italian and german spoken learner's english," *ICAME Journal: Intl. Computer Archive of Modern and Medieval English Journal*, vol. 27, pp. 5–18, 2003.

[33] D. Weber, S. Zaporowski, and D. Korzekwa, "Constructing a dataset of speech recordings with lombard effect," in *24th IEEE SPA*, 2020.

[34] D. Korzekwa, B. Kostek *et al.*, "Detection of lexical stress errors in non-native (l2) english with data augmentation and attention," *arXiv preprint arXiv:2012.14788*, 2020.

[35] A. Graves, "Connectionist temporal classification," in *Supervised Sequence Labelling with Recurrent Neural Networks*. Springer, 2012, pp. 61–93.

[36] J. Chorowski, R. J. Weiss, S. Bengio, and A. van den Oord, "Unsupervised speech representation learning using wavenet autoencoders," *IEEE/ACM transactions on audio, speech, and language processing*, vol. 27, no. 12, pp. 2041–2053, 2019.

[37] A. Van Den Oord, O. Vinyals *et al.*, "Neural discrete representation learning," *Advances in Neural Information Processing Systems*, vol. 30, pp. 6306–6315, 2017.

ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) | 978-1-7281-7605-5/20/$31.00 ©2021 IEEE | DOI: 10.1109/ICASSP39728.2021.9413953

MISPRONUNCIATION DETECTION IN NON-NATIVE (L2) ENGLISH WITH UNCERTAINTY MODELING

Daniel Korzekwa[⋆†], *Jaime Lorenzo-Trueba*[⋆], *Szymon Zaporowski*[†], *Shira Calamaro*[⋆], *Thomas Drugman*[⋆], *Bozena Kostek*[†]

[⋆] Amazon Speech [†] Gdansk University of Technology, Faculty of ETI, Poland

ABSTRACT

A common approach to the automatic detection of mispronunciation in language learning is to recognize the phonemes produced by a student and compare it to the expected pronunciation of a native speaker. This approach makes two simplifying assumptions: a) phonemes can be recognized from speech with high accuracy, b) there is a single correct way for a sentence to be pronounced. These assumptions do not always hold, which can result in a significant amount of false mispronunciation alarms. We propose a novel approach to overcome this problem based on two principles: a) taking into account uncertainty in the automatic phoneme recognition step, b) accounting for the fact that there may be multiple valid pronunciations. We evaluate the model on non-native (L2) English speech of German, Italian and Polish speakers, where it is shown to increase the precision of detecting mispronunciations by up to 18% (relative) compared to the common approach.

Index Terms— Pronunciation Assessment, Second Language Learning, Uncertainty Modeling, Deep Learning

1. INTRODUCTION

In Computer Assisted Pronunciation Training (CAPT), students are presented with a text and asked to read it aloud. A computer informs students on mispronunciations in their speech, so that they can repeat it and improve. CAPT has been found to be an effective tool that helps non-native (L2) speakers of English to improve their pronunciation skills [1, 2].

A common approach to CAPT is based on recognizing the phonemes produced by a student and comparing them with the expected (canonical) phonemes that a native speaker would pronounce [3, 4, 5, 6]. It makes two simplifying assumptions. First, it assumes that phonemes can be automatically recognized from speech with high accuracy. However, even in native (L1) speech, it is difficult to get the Phoneme Error Rate (PER) below 15% [7]. Second, this approach assumes that this is the only 'correct' way for a sentence to be pronounced, but due to phonetic variability this is not always true. For example, the word 'enough' can be pronounced by native speakers in multiple correct ways: /ih n ah f/ or /ax n ah f/ (short 'i' or 'schwa' phoneme at the beginning). These assumptions do not always hold which can result in a significant amount of false mispronunciation alarms and making students confused when it happens.

We propose a novel approach that results in fewer false mispronunciation alarms, by formalizing the intuition that we will not be able to recognize exactly what a student has pronounced or say precisely how a native speaker would pronounce it. First, the model estimates a belief over the phonemes produced by the student, intuitively representing the uncertainty in the student's pronunciation. Then, the model converts this belief into the probabilities that a native speaker would pronounce it, accounting for phonetic variability. Finally, the model makes a decision on which words were mispronounced in the sentence by processing three pieces of information: a) what the student pronounced, b) how likely a native speaker would pronounce it that way, and c) what the student was expected to pronounce.

In Section 2, we review the related work. In Section 3, we describe the proposed model. In Section 4, we present the experiments, and we conclude in Section 5.

2. RELATED WORK

In 2000, Witt et al. coined the term Goodness of Pronunciation (GoP) [3]. GoP starts by aligning the canonical phonemes with the speech signal using a forced-alignment technique. This technique aims to find the most likely mapping between phonemes and the regions of a corresponding speech signal. In the next step, GoP computes the ratio between the likelihoods of the canonical and the most likely pronounced phonemes. Finally, it detects a mispronunciation if the ratio falls below a given threshold. GoP was further extended with Deep Neural Networks (DNNs), replacing Hidden Markov Model (HMM) and Gaussian Mixture Model (GMM) techniques for acoustic modeling [4, 5]. Cheng et al. [8] improved the performance of GoP with the latent representation of speech extracted in an unsupervised way.

As opposed to GoP, we do not use forced-alignment that requires both speech and phoneme inputs. Following the work of Leung et al. [6], we use a phoneme recognizer,

Authorized licensed use limited to: Amazon. Downloaded on March 22,2022 at 15:47:13 UTC from IEEE Xplore. Restrictions apply.

which recognizes phonemes from only the speech signal. The phoneme recognizer is based on a Convolutional Neural Network (CNN), a Gated Recurrent Unit (GRU), and Connectionist Temporal Classification (CTC) loss. Leung et al. report that it outperforms other forced-alignment [4] and forced-alignment-free [9] techniques on the task of detecting phoneme-level mispronunciations in L2 English. Contrary to Leung et al., who rely only on a single recognized sequence of phonemes, we obtain top N decoded sequences of phonemes, along with the phoneme-level posterior probabilities.

It is common in pronunciation assessment to employ the speech signal of a reference speaker. Xiao et al. use a pair of speech signals from a student and a native speaker to classify native and non-native speech [10]. Mauro et al. incorporate the speech of a reference speaker to detect mispronunciations at the phoneme level [11]. Wang et al. use siamese networks for modeling discrepancy between normal and distorted children's speech [12]. We take a similar approach but we do not need a database of reference speech. Instead, we train a statistical model to estimate the probability of pronouncing a sentence by a native speaker. Qian et al. propose a statistical pronunciation model as well [13]. Unlike our work, in which we create a model of 'correct' pronunciation, they build a model that generates hypotheses of mispronounced speech.

3. PROPOSED MODEL

The design consists of three subsystems: a Phoneme Recognizer (PR), a Pronunciation Model (PM), and a Pronunciation Error Detector (PED), illustrated in Figure 1. The PR recognizes phonemes spoken by a student. The PM estimates the probabilities of having been pronounced by a native speaker. Finally, the PED computes word-level mispronunciation probabilities. In Figure 2, we present detailed architectures of the PR, PM, and PED.

For example, considering the text: 'I said alone not gone' with the canonical representation of /ay - s eh d - ax l ow n - n aa t - g aa n/. Polish L2 speakers of English often mispronounce the /eh/ phoneme in the second word as /ey/. The PM would identify the /ey/ as having a low probability of being pronounced by a native speaker in the middle of the word 'said, which the PED would translate into a high probability of mispronunciation.

3.1. Phoneme Recognizer

The PR (Figure 2a) uses beam decoding [14] to estimate N hypotheses of the most likely sequences of phonemes that are recognized in the speech signal $\mathbf{o}$. A single hypothesis is denoted as $\mathbf{r_o} \sim p(\mathbf{r_o}|\mathbf{o})$. The speech signal $\mathbf{o}$ is represented by a mel-spectrogram with f frames and 80 mel-bins. Each sequence of phonemes $\mathbf{r_o}$ is accompanied by the posterior phoneme probabilities of shape: $(l_{r_o}, l_s + 1)$. l_{r_o} is the length of the sequence and l_s is the size of the phoneme set (45 phonemes including 'pause', 'end of sentence (eos)', and a 'blank' label required by the CTC-based model).

3.2. Pronunciation Model

The PM (Figure 2b) is an encoder-decoder neural network following Sutskever et al. [15]. Instead of building a text-to-text translation system between two languages, we use it for phoneme-to-phoneme conversion. The sequence of phonemes $\mathbf{r_c}$ that a native speaker was expected to pronounce is converted into the sequence of phonemes $\mathbf{r}$ they had pronounced, denoted as $\mathbf{r} \sim p(\mathbf{r}|\mathbf{r_c})$. Once trained, the PM acts as a probability mass function, computing the likelihood sequence $\boldsymbol{\pi}$ of the phonemes $\mathbf{r_o}$ pronounced by a student conditioned on the expected (canonical) phonemes $\mathbf{r_c}$. The PM is denoted in Eq. 1, which we implemented in MxNet [16] using 'sum' and 'element-wise multiply' linear-algebra operations.

$$\boldsymbol{\pi} = \sum_{\mathbf{r_o}} p(\mathbf{r_o}|\mathbf{o}) p(\mathbf{r} = \mathbf{r_o}|\mathbf{r_c}) \tag{1}$$

The model is trained on phoneme-to-phoneme speech data created automatically by passing the speech of the native speakers through the PR. By annotating the data with the PR, we can make the PM model more resistant to possible phoneme recognition inaccuracies of the PR at testing time.

3.3. Pronunciation Error Detector

The PED (Figure 2c) computes the probabilities of mispronunciations $\mathbf{e}$ at the word level, denoted as $\mathbf{e} \sim p(\mathbf{e}|\mathbf{r_o}, \boldsymbol{\pi}, \mathbf{r_c})$. The PED is conditioned on three inputs: the phonemes $\mathbf{r_o}$ recognized by the PR, the corresponding pronunciation likelihoods $\boldsymbol{\pi}$ from the PM, and the canonical phonemes $\mathbf{r_c}$. The model starts with aligning the canonical and recognized sequences of phonemes. We adopted a dynamic programming algorithm for aligning biological sequences developed by Needleman-Wunsch [17]. Then, the probability of mispronunciation for a given word is computed with Equation 2, k denotes the word index, and j is the phoneme index in the word with the lowest probability of pronunciation.

$$p(\mathbf{e}_k) = \begin{cases} 0 & \text{if aligned phonemes match,} \\ 1 - \boldsymbol{\pi}_{k,j} & \text{otherwise.} \end{cases} \tag{2}$$

We compute the probabilities of mispronunciation for N phoneme recognition hypotheses from the PR. Mispronunciation for a given word is detected if the probability of mispronunciation falls below a given threshold for all hypotheses. The hyper-parameter $N = 4$ was manually tuned on a single L2 speaker from the testing set to optimize the PED in the precision metric.

Authorized licensed use limited to: Amazon. Downloaded on March 22,2022 at 15:47:13 UTC from IEEE Xplore. Restrictions apply.

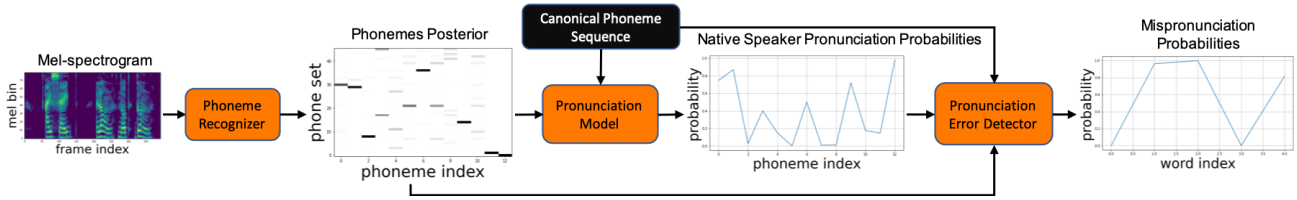


Fig. 1: Architecture of the system for detecting mispronounced words in a spoken sentence.

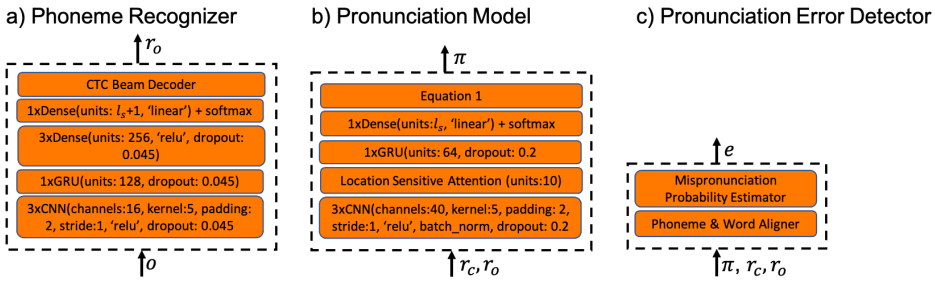


Fig. 2: Architecture of the PR, PM, and PED subsystems. l_s - the size of the phoneme set.

4. EXPERIMENTS AND DISCUSSION

We want to understand the effect of accounting for uncertainty in the PR-PM system presented in Section 3. To do this, we compare it with two other variants, PR-LIK and PR-NOLIK, and analyze precision and recall metrics. The PR-LIK system helps us understand how important is it to account for the phonetic variability in the PM. To switch the PM off, we modify it so that it considers only a single way for a sentence to be pronounced correctly.

The PR-NOLIK variant corresponds to the CTC-based mispronunciation detection model proposed by Leung et al. [6]. To reflect this, we make two modifications compared to the PR-PM system. First, we switch the PM off in the same way we did it in the PR-LIK system. Second, we set the posterior probabilities of recognized phonemes in the PR to 100%, which means that the PR is always certain about the phonemes produced by a speaker. There are some slight implementation differences between Leung's model and PR-NOLIK, for example, regarding the number of units in the neural network layers. We use our configuration to make a consistent comparison with PR-PM and PR-LIK systems. One can hence consider PR-NOLIK as a fair state-of-the-art baseline [6].

4.1. Model Details

For extracting mel-spectrograms, we used a time step of 10 ms and a window size of 40 ms. The PR was trained with CTC Loss and Adam Optimizer (batch size: 32, learning rate: 0.001, gradient clipping: 5). We tuned the following hyper-parameters of the PR with Bayesian Optimization: dropout, CNN channels, GRU, and dense units. The PM was trained with the cross-entropy loss and AdaDelta optimizer (batch size: 20, learning rate: 0.01, gradient clipping: 5). The location-sensitive attention in the PM follows the work by Chorowski et al. [7]. The PR and PM models were implemented in MxNet Deep Learning framework.

4.2. Speech Corpora

For training and testing the PR and PM, we used 125.28 hours of L1 and L2 English speech from 983 speakers segmented into 102812 sentences, sourced from multiple speech corpora: TIMIT [18], LibriTTS [19], Isle [20] and GUT Isle [21]. We summarize it in Table 1. All speech data were downsampled to 16 kHz. Both L1 and L2 speech were phonetically transcribed using Amazon proprietary grapheme-to-phoneme model and used by the PR. Automatic transcriptions of L2 speech do not capture pronunciation errors, but we found it is still worth including automatically transcribed L2 speech in the PR. L2 corpora were also annotated by 5 native speakers of American English for word-level pronunciation errors. There are 3624 mispronounced words out of 13191 in the Isle Corpus and 1046 mispronounced words out of 5064 in the GUT Isle Corpus.

From the collected speech, we held out 28 L2 speakers and used them only to assess the performance of the systems in the mispronunciation detection task. It includes 11 Italian and 11 German speakers from the Isle corpus [20], and 6 Polish speakers from the GUT Isle corpus [21].

4.3. Experimental Results

The PR-NOLIK detects mispronounced words based on the difference between the canonical and recognized phonemes.

Authorized licensed use limited to: Amazon. Downloaded on March 22,2022 at 15:47:13 UTC from IEEE Xplore. Restrictions apply.

Table 1: The summary of speech corpora used by the PR.

Native Language	Hours	Speakers
English	90.47	640
Unknown	19.91	285
German and Italian	13.41	46
Polish	1.49	12

Therefore, this system does not offer any flexibility in optimizing the model for higher precision.

The PR-LIK system incorporates posterior probabilities of recognized phonemes. It means that we can tune this system towards higher precision, as illustrated in Figure 3. Accounting for uncertainty in the PR helps when there is more than one likely sequence of phonemes that could have been uttered by a user, and the PR model is uncertain which one it is. For example, the PR reports two likely pronunciations for the text 'I said' /ay s eh d/. The first one, /s eh d/ with /ay/ phoneme missing at the beginning and the alternative one /ay s eh d/ with the /ay/ phoneme present. If the PR considered only the mostly likely sequence of phonemes, like PR-NOLIK does, it would incorrectly raise a pronunciation error. In the second example, a student read the text 'six' /s ih k s/ mispronouncing the first phoneme /s/ as /t/. The likelihood of the recognized phoneme is only 34%. It suggests that the PR model is quite uncertain on what phoneme was pronounced. However, sometimes even in such cases, we can be confident that the word was mispronounced. It is because the PM computes the probability of pronunciation based on the posterior probability from the PR model. In this particular case, other phoneme candidates that account for the remaining 66% of uncertainty are also unlikely to be pronounced by a native speaker. The PM can take it into account and correctly detect a mispronunciation.

However, we found that the effect of accounting for uncertainty in the PR is quite limited. Compared to the PR-NOLIK system, the PR-LIK raises precision on the GUT Isle corpus only by 6% (55% divided by 52%), at the cost of dropping recall by about 23%. We can observe a much stronger effect when we account for uncertainty in the PM model. Compared to the PR-LIK system, the PR-PM system further increases precision between 11% and 18%, depending on the decrease in recall between 20% to 40%. One example where the PM helps is illustrated by the word 'enough' that can be pronounced in two similar ways: /ih n ah f/ or /ax n ah f/ (short 'i' or 'schwa' phoneme at the beginning.) The PM can account for phonetic variability and recognize both versions as pronounced correctly. Another example is word linking [22]. Native speakers tend to merge phonemes of neighboring words. For example, in the text 'her arrange' /hh er - er ey n jh/, two neighboring phonemes /er/ can be pronounced as a single phoneme: /hh er ey n jh/. The PM model can correctly recognize multiple variations of such pronunciations.

Complementary to precision-recall curve showed in Figure 3, we present in Table 2 one configuration of the precision and recall scores for the PR-LIK and PR-PM systems. This configuration is selected in such a way that: a) recall for both systems is close to the same value, b) to illustrate that the PR-PM model has a much bigger potential of increasing precision than the PR-LIK system. A similar conclusion can be made by inspecting multiple different precision and recall configurations in the precision and recall plots for both Isle and GUT Isle corpora.

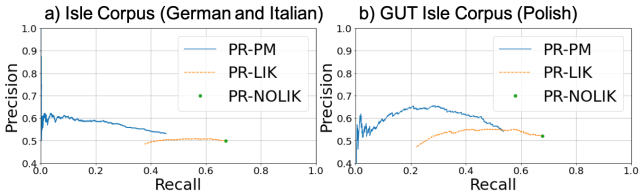


Fig. 3: Precision-recall curves for the evaluated systems.

Table 2: Precision and recall of detecting word-level mispronunciations. CI - Confidence Interval.

Model	Precision [%,95%CI]	Recall [%,95%CI]
Isle corpus (German and Italian)		
PR-LIK	49.39 (47.59-51.19)	40.20 (38.62-41.81)
PR-PM	54.20 (52.32-56.08)	40.20 (38.62-41.81)
GUT Isle corpus (Polish)		
PR-LIK	54.91 (50.53-59.24)	40.29 (36.66-44.02)
PR-PM	61.21 (56.63-65.65)	40.15 (36.51-43.87)

5. CONCLUSION AND FUTURE WORK

To report fewer false pronunciation alarms, it is important to move away from the two simplifying assumptions that are usually made by common methods for pronunciation assessment: a) phonemes can be recognized with high accuracy, b) a sentence can be read in a single correct way. We acknowledged that these assumptions do not always hold. Instead, we designed a model that: a) accounts for the uncertainty in phoneme recognition and b) accounts for multiple ways a sentence can be pronounced correctly due to phonetic variability. We found that to optimize precision, it is more important to account for the phonetic variability of speech than accounting for uncertainty in phoneme recognition. We showed that the proposed model can raise the precision of detecting mispronounced words by up to 18% compared to the common methods.

In the future, we plan to adapt the PM model to correctly pronounced L2 speech to account for phonetic variability of non-native speakers. We plan to combine the PR, PM, and PED modules and train the model jointly to eliminate accumulation of statistical errors coming from disjoint training of the system.

Authorized licensed use limited to: Amazon. Downloaded on March 22,2022 at 15:47:13 UTC from IEEE Xplore. Restrictions apply.

6. REFERENCES

[1] A. Neri, O. Mich, M. Gerosa, and D. Giuliani, "The effectiveness of computer assisted pronunciation training for foreign language learning by children," *Computer Assisted Language Learning*, vol. 21, no. 5, pp. 393–408, 2008.

[2] C. Tejedor-García, D. Escudero, E. Cámara-Arenas, C. González-Ferreras, and V. Cardeñoso-Payo, "Assessing pronunciation improvement in students of english using a controlled computer-assisted pronunciation tool," *IEEE Transactions on Learning Technologies*, 2020.

[3] S. M. Witt and S. J. Young, "Phone-level pronunciation scoring and assessment for interactive language learning," *Speech communication*, vol. 30, no. 2-3, pp. 95–108, 2000.

[4] K. Li, X. Qian, and H. Meng, "Mispronunciation detection and diagnosis in l2 english speech using multidistribution deep neural networks," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 1, pp. 193–207, 2016.

[5] S. Sudhakara, M. K. Ramanathi, C. Yarra, and P. K. Ghosh, "An improved goodness of pronunciation (gop) measure for pronunciation evaluation with dnn-hmm system considering hmm transition probabilities.," in *INTERSPEECH*, 2019, pp. 954–958.

[6] W. Leung, X. Liu, and H. Meng, "Cnn-rnn-ctc based end-to-end mispronunciation detection and diagnosis," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 8132–8136.

[7] J. K. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, "Attention-based models for speech recognition," in *Advances in neural information processing systems*, 2015, pp. 577–585.

[8] S. Cheng et al., "Asr-free pronunciation assessment," *arXiv preprint arXiv:2005.11902*, 2020.

[9] A. M. Harrison, W. Lo, X. Qian, and H. Meng, "Implementation of an extended recognition network for mispronunciation detection and diagnosis in computer-assisted pronunciation training," in *Intl. Workshop on Speech and Language Technology in Education*, 2009.

[10] Y. Xiao and W. Soong, F. K .and Hu, "Paired phone-posteriors approach to esl pronunciation quality assessment," in *bdl*, vol. 1, p. 3. 2018.

[11] M. Nicolao, A. V. Beeston, and T. Hain, "Automatic assessment of english learner pronunciation using discriminative classifiers," in *2015 IEEE Intl. Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2015, pp. 5351–5355.

[12] J. Wang, Y. Qin, Z. Peng, and T. Lee, "Child speech disorder detection with siamese recurrent network using speech attribute features.," in *INTERSPEECH*, 2019, pp. 3885–3889.

[13] X. Qian, H. Meng, and F. Soong, "Capturing l2 segmental mispronunciations with joint-sequence models in computer-aided pronunciation training (capt)," in *2010 7th Intl. Symposium on Chinese Spoken Language Processing*. IEEE, 2010, pp. 84–88.

[14] A. Graves, A. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *2013 IEEE Intl. conference on acoustics, speech and signal processing*. IEEE, 2013, pp. 6645–6649.

[15] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Advances in neural information processing systems*, 2014, pp. 3104–3112.

[16] T. et al. Chen, "Mxnet: A flexible and efficient machine learning library for heterogeneous distributed systems," *arXiv preprint arXiv:1512.01274*, 2015.

[17] S. B. Needleman and C. D. Wunsch, "A general method applicable to the search for similarities in the amino acid sequence of two proteins," *Journal of molecular biology*, vol. 48, no. 3, pp. 443–453, 1970.

[18] J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, and David S. Pallett, "Darpa timit acoustic-phonetic continous speech corpus cd-rom. nist speech disc 1-1.1," *STIN*, vol. 93, pp. 27403, 1993.

[19] H. Zen et al., "LibriTTS: A Corpus Derived from LibriSpeech for Text-to-Speech," in *Proc. Interspeech 2019*, 2019, pp. 1526–1530.

[20] E. S. Atwell, P. A. Howarth, and D. C. Souter, "The isle corpus: Italian and german spoken learner's english," *ICAME Journal: Intl. Computer Archive of Modern and Medieval English Journal*, vol. 27, pp. 5–18, 2003.

[21] D. Weber, S. Zaporowski, and D. Korzekwa, "Constructing a dataset of speech recordings with lombard effect," in *24th IEEE SPA*, 2020.

[22] A. E. Hieke, "Linking as a marker of fluent speech," *Language and Speech*, vol. 27, no. 4, pp. 343–354, 1984.

Authorized licensed use limited to: Amazon. Downloaded on March 22,2022 at 15:47:13 UTC from IEEE Xplore. Restrictions apply.

INTERSPEECH 2021
30 August – 3 September, 2021, Brno, Czechia

Detection of Lexical Stress Errors in Non-Native (L2) English with Data Augmentation and Attention

Daniel Korzekwa[1,2], *Roberto Barra-Chicote*[3], *Szymon Zaporowski*[2], *Grzegorz Beringer*[1], *Jaime Lorenzo-Trueba*[3], *Alicja Serafinowicz*[1], *Jasha Droppo*[4], *Thomas Drugman*[3], *Bozena Kostek*[2]

[1]Amazon, Poland
[2]Gdansk University of Technology, Faculty of ETI, Poland
[3]Amazon, UK
[4]Amazon, USA

korzekwa@amazon.com

Abstract

This paper describes two novel complementary techniques that improve the detection of lexical stress errors in non-native (L2) English speech: attention-based feature extraction and data augmentation based on Neural Text-To-Speech (TTS). In a classical approach, audio features are usually extracted from fixed regions of speech such as the syllable nucleus. We propose an attention-based deep learning model that automatically derives optimal syllable-level representation from frame-level and phoneme-level audio features. Training this model is challenging because of the limited amount of incorrect stress patterns. To solve this problem, we propose to augment the training set with incorrectly stressed words generated with Neural TTS. Combining both techniques achieves 94.8% precision and 49.2% recall for the detection of incorrectly stressed words in L2 English speech of Slavic and Baltic speakers.

Index Terms: lexical stress, language learning, data augmentation, text-to-speech, attention, automated speech assessment

1. Introduction

Computer Assisted Pronunciation Training (CAPT) usually focuses on practicing pronunciation of phonemes [1, 2, 3], while there is evidence in non-native (L2) English speakers that practicing lexical stress improves speech intelligibility [4, 5]. Lexical stress is a syllable-level phonological feature. It is a part of the phonological rules that define how words should be spoken in a given language. Stressed syllables are usually longer, louder, and expressed with a higher pitch than their unstressed counterparts [6]. Lexical stress is inter-connected with phonemic representation. For example, placing lexical stress on a different syllable of a word may lead to different phonemic realizations known as 'vowel reduction' [7].

The focal point of our work is the detection of words with incorrect stress patterns. The training data with human speech is usually highly imbalanced, with few training examples of incorrectly stressed words. It makes training machine learning models for this task challenging. We address this problem by augmenting the training set with synthetic speech that is generated with Neural Text-To-Speech (TTS) [8]. Neural TTS allows us generating words with both correct and incorrect stress patterns.

Most of the existing approaches for automated lexical stress assessment are based on carefully designed features that are extracted from fixed regions of speech signal such as the syllable nucleus [9, 10, 11]. We introduce attention mechanism [12] to automatically learn optimal syllable-level representation. Attention-based approach originates from the intuition of how people detect specific patterns in high dimensional and unstructured data such as visual and speech signals [13]. For example, we might focus our attention on the duration ratio between nuclei of two neighboring syllables, incidentally, an important predictor of lexical stress. The syllable-level representation is derived from frame-level (F0, intensity) and phoneme-level (duration) audio features and the corresponding phonetic representation of a word. We do not indicate precisely the regions of the audio signal that are important for the detection of lexical stress errors. The attention mechanism does it automatically.

To the best of our knowledge, this paper is the first attempt, for the task of lexical stress error detection, to: *i)* augment the training data with Neural TTS, *ii)* use attention mechanisms to automatically extract syllable-level features for lexical stress error detection. Ruan et al. [14] used attention-based architecture of transformers for lexical stress detection. However, their paper concerns recognizing stressed and unstressed phonemes. They do not detect lexical stress errors, which is crucial in CAPT applications.

The paper is structured as follows. In Section 2, we review the related work. Section 3 describes the proposed model. Section 4 reviews human and synthetic speech corpora. In Section 5, we present our experiments, and Section 6 concludes the paper.

2. Related Work

The existing work focuses on the supervised classification of lexical stress using Neural Networks [15, 10], Support Vector Machines [11, 16] and Fisher's linear discriminant [17]. There are two popular variants: a) discriminating syllables between primary stress/no stress [9], and b) classifying between primary stress/secondary stress/no stress [18, 15]. Ramanathi et al. [19] have followed an alternative unsupervised way of classifying lexical stress, which is based on computing the likelihood of an acoustic signal for a number of possible lexical stress representations of a word.

Accuracy is the most commonly used performance metric, and it indicates the ratio of correctly classified stress patterns on a syllable [18] or word level [11]. On the contrary, following Ferrer et al. [9], we analyze precision and recall metrics because we aim to detect lexical stress errors and not just classify them.

Existing approaches for the classification and detection of lexical stress errors are based on carefully designed features. They start with aligning a speech signal with phonetic tran-

Copyright © 2021 ISCA

http://dx.doi.org/10.21437/Interspeech.2021-86

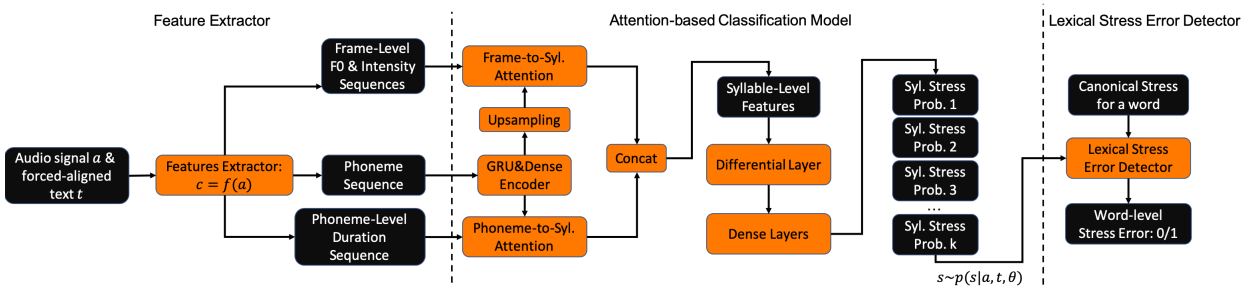


Figure 1: *Attention-based Deep Learning model for the detection of lexical stress errors.*

scription, performed via forced-alignment [10, 11]. Alternatively, Automatic Speech Recognition (ASR) can provide both phonetic transcription and its alignment with a speech signal [18]. Then, prosodic features such as duration, energy and pitch [11] and cepstral features such as MFCC and Mel-Spectrogram [9, 10] are extracted. These features can be extracted on the syllable [10] or syllable nucleus [9, 11] level.

Shahin et al. [10] computed features of neighboring vowels, and Li et al. [18] included the features for two preceding and two following syllables in the model. The features are often preprocessed and normalized to avoid potential confounding variables [9], and to achieve better model generalization by normalizing the duration and pitch on a word level [9, 17]. Li et al. [15] added canonical lexical stress to input features, which improves the accuracy of the model.

In our approach, we use attention mechanisms to derive automatically regions of the audio signal that are important for the detection of lexical stress errors. We also use data augmentation through the generation of artificial data with Neural TTS.

3. Proposed Model

The proposed model consists of three subsystems: Feature Extractor, Attention-based Classification Model, and Lexical Stress Error Detector. It is illustrated in Figure 1.

3.1. Feature Extractor

The Feature Extractor extracts prosodic features and phonemes from speech signal $\mathbf{a}$ and forced-aligned text $\mathbf{t}$. To obtain forced-alignment, we used Montreal toolkit [20] along with an acoustic model pretrained on LibriSpeech ASR corpus [21]. The prosodic features $\mathbf{c} = f(\mathbf{a})$ are formed by: F0, intensity [dB SPL] and phoneme-level durations. The F0 and intensity features are computed at the frame level using Praat library [22] (time step: 10 ms, window size: 40 ms). The F0 contour is linearly interpolated in unvoiced regions. These raw features will be further transformed by the attention-based model to the syllable-level representation.

3.2. Attention-based Classification Model

The Attention-based Classification Model maps frame-level and phoneme-level features to the syllable-level representation. Then, it produces a lexical stress pattern $\mathbf{s}$, modeled as a sequence of Bernoulli random variables $\mathbf{s} = \{s_1, .., s_k\}$ (stressed/unstressed) over K syllables of a multi-syllable word, conditioned on audio $\mathbf{a}$ and text $\mathbf{t}$ representations. Let us define it as a conditional probability distribution $\mathbf{s} \sim p(\mathbf{s}|\mathbf{a}, \mathbf{t}, \boldsymbol{\theta})$, where $\boldsymbol{\theta}$ are the parameters of the model.

To extract syllable-level features, we use two dot-product attentions operating on the frame and phoneme levels. To build better intuition on what these two attention do, in Figure 2 we show the frame-level and phoneme-level attention plots for the word 'garage' pronounced by a Polish speaker and incorrectly stressed on the first syllable in reference to American English. This word has a similar pronunciation but different lexical stress in Polish and American English languages ('G AA1 R AA0 ZH' vs 'G ER0 AA1 ZH'). Both attentions find the most relevant regions of the frame-level and phoneme-level features.

The dot-product attention is presented in Equation 1, and it follows the notation proposed by Vaswani et al. [12]. It is based on three inputs: Query ($\mathbf{Q}$), Keys ($\mathbf{K}$) and Values ($\mathbf{V}$), where d_k is the dimensionality of $\mathbf{K}$.

$$Attention(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = softmax(\frac{\mathbf{Q}\mathbf{K}^t}{\sqrt{d_k}})\mathbf{V} \quad (1)$$

The attention inputs are represented as follows. Query refers to the syllable positional embeddings defined by one-hot syllable index encodings. Keys represents a sequence of sub-phonemes. Each sub-phoneme is represented by a set of features: $phoneme_id$, $syllable_index$, is_vowel, $left_or_right_sub_phoneme$. All features are one-hot encoded and processed with a Gated Recurrent Unit (GRU) layer [23] (units:4, dropout: 0.24). In the end, encoded sub-phoneme sequence is passed through linear dense layers. In the case of the frame-level attention, the encoded sub-phoneme sequence is upsampled to the frame level using phoneme durations from forced-alignment. In upsampling, we simply replicate phonemes across aligned frames of audio signal. Similar phoneme-to-frame upsampling has been recently adopted in Text-To-Speech [24]. Finally, Values are the $F0/intensity$ and $duration$ features for frame-level and phoneme-level attentions respectively.

To model relative prominence, we introduce a differential bi-directional layer that computes the ratios of syllable-level acoustic features for each syllable and its two neighbors (Figure 1). The bi-directional layer is implemented as a simple 'division' math operation and it does not contain any trainable parameters. The output of the differential layer is further processed by three dense layers (units: 4, activation: tanh, dropout: 0.24), followed by a linear dense layer (units: 2, dropout: 0.24) that produces a two-dimensional output for each syllable. It is then squeezed by a softmax function to generate lexical stress probabilities.

3.3. Training of the Classification Model

We train the model on a set of N triplets that contains 1) human recorded words and 2) synthetic words generated using Neural

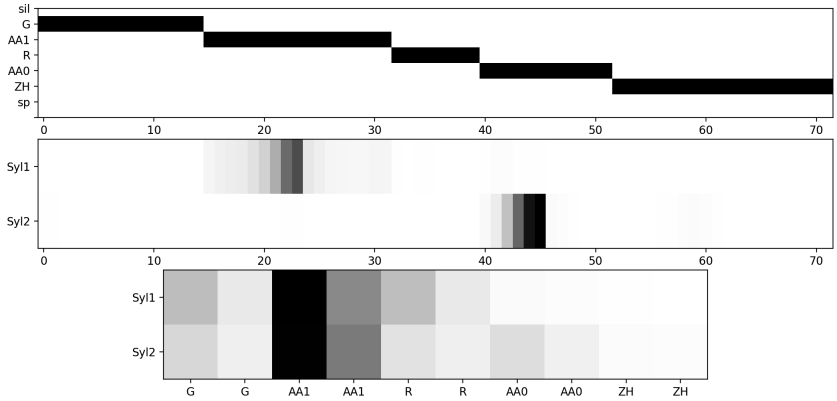


Figure 2: *Top: forced-alignment mapping between phonemes and frames for the word 'garage'. Middle: Frame-to-syllable attention weights matrix. Bottom: (Sub)Phoneme-to-syllable attention weights matrix.*

TTS. A single triplet is represented by $\{\mathbf{s_n}, \mathbf{a_n}, \mathbf{t_n}\}$, where $n = 1..N$ is the index of a training example.

The concept of data augmentation can be explained using a framework of Bayesian Inference. Consider three random variables, lexical stress $\mathbf{s_n}$, audio signal $\mathbf{a_n}$ and text $\mathbf{t_n}$. All variables are observed for the training examples of human speech. However, for the synthetic speech, we only observe the lexical stress and text variables. The audio signal is unobserved (hidden) because we have to generate it.

To train this model, we derive a negative log-likelihood loss over a joint probability distribution of lexical stress $\mathbf{s}$ and audio $\mathbf{a}$ random variables, as depicted in Equation 2. The loss is further approximated with the variational lower bound [25], as presented in Equation 3 (we omit $\boldsymbol{\theta}$ for brevity). For the training examples of synthetic speech, the conditional probability distribution over the audio signal $\mathbf{a_n} \sim p(\mathbf{a_n}|\mathbf{s_n}, \mathbf{t_n})$ is estimated with Neural TTS, and for human recorded words, it is given explicitly.

$$\mathcal{L}(\boldsymbol{\theta}) = -\sum_{n}^{N} log \int p(\mathbf{s_n}, \mathbf{a_n}|\mathbf{t_n}, \boldsymbol{\theta}) d\mathbf{a_n} \tag{2}$$

$$log \int p(\mathbf{s_n}, \mathbf{a_n}|\mathbf{t_n}) \mathbf{da_n} \approx E_{\mathbf{a_n} \sim p(\mathbf{a_n}|\mathbf{t_n}, \mathbf{s_n})}[logp(\mathbf{s_n}|\mathbf{a_n}, \mathbf{t_n})] \tag{3}$$

The model was implemented in MxNet [26], trained with Stochastic Gradient Descent optimizer (learning rate: 0.1, batch size: 20) and tuned with Bayesian optimization [27]. Training data were split into buckets based on the number of frames in an audio signal, using Gluon-NLP package [28]. A single bucket contains words with the same number of syllables with zero-padded acoustic and sub-phoneme sequences.

3.4. Lexical Stress Error Detector

The Lexical Stress Error Detector reports on lexical stress error if the expected (canonical) and estimated lexical stress for a given syllable do not match and the corresponding probability is higher than a given threshold.

4. Speech Corpus

Our speech corpus consists of human and synthetic speech. The data were split into training and testing sets with disjointed speakers ascribed to each set. Human speech contains L1 and L2 speakers of English. Synthetic data were generated with Neural TTS and are included only in the training set. All audio files were downsampled to a 16 kHz sampling rate. The data are summarized in Table 1, and we provide more details in the following subsections.

Table 1: *Train and test sets details.*

Data set	**Speakers (L2)**	**Words (unique)**	**Stress Errors**
Train set (human)	473 (10)	8223 (1528)	425
Train set (TTS)	1 (0)	3937 (1983)	2005
Test set (human)	176 (21)	2108 (378)	189

4.1. Human Speech

Due to the limited availability of L2 corpora, we recorded our own L2-English corpus of Slavic and Baltic speakers. It also allows us to evaluate the model during interactive English learning sessions with our students. The corpus contains speech from 25 speakers (23 Polish, 1 Ukrainian and 1 Lithuanian): 7 females and 18 males, all between 24 and 40 years old. All speakers read a list of two hundred words. One hundred words were prepared by a professional English teacher, including frequently mispronounced words by Slavic and Baltic students. The second half consists of the most common words that were obtained from Google's Trillion Word Corpus [29] based on n-gram frequency analysis. We excluded abbreviations and one-syllable words.

Additionally, L1 and L2 English speech was collected from publicly available speech data sets, including TIMIT [30], Arctic [31], L2-Arctic [32] and Porzuczek [33].

4.2. Synthetic Speech

Complementary to human recordings, synthetic speech was generated with Neural TTS by Latorre et al. [8]. The Neural TTS consists of two modules. Context-generation module is an attention-based encoder-decoder neural network that generates a mel-spectrogram from a sequence of phonemes. Then, a Neural Vocoder converts it to the speech signal. The Neural Vocoder is a neural network of architecture similar to the work by [34]. The Neural TTS was trained using speech of a professional American voice talent. To generate words with different

lexical stress patterns, we modify lexical stress markers associated with the vowels in the phonemic transcription of a word. For example, with the input of /r iy1 m ay0 n d/ we can place lexical stress on the first syllable of the word 'remind'. 1980 popular English words were synthesized with correct and incorrect stress patterns.

4.3. Lexical Stress Annotations

L1 corpora were segmented into words and annotated automatically using a proprietary Amazon American English Lexicon, taking into account the syntactic context of the word. Neural TTS speech and the speech of L2 speakers were annotated by 5 American English linguists into 'primary' and 'no stress' categories, keeping the words for which a minimum of 4 out of 5 linguists agreed on the stress pattern. Annotators were not able to distinguish between primary and secondary lexical stress. 81.5% of synthesized words matched the intended stress patterns with a minimum of 4 annotators' agreement. It shows that Neural TTS can be used to generate incorrectly stressed speech.

5. Experiments

The proposed model (Att_TTS) from Section 3 is compared to three baseline models that are designed to measure the impact of the Neural TTS data augmentation and the attention mechanism. To compare these models, we plotted their precision-recall curves and gave their corresponding area under a curve (AUC) along with our results, see Figure 3.

The Att_NoTTS model has the same architecture as the Att_TTS, but the synthetic speech is excluded from the 'training set'. The NoAtt_TTS model uses the same training set as the Att_TTS, but it has no attention mechanism. Instead, as a syllable-level representation, it uses mean values of acoustic features for the corresponding syllable nucleus. The NoAtt_NoTTS model has no attention, and it does not use Neural TTS data augmentation.

As a state-of-the-art baseline, we use the work by Ferrer et al. [9]. However, a direct comparison is not possible. In their test corpus, there were 46.4% (191 out of 411) of incorrectly stressed words, far more than 9.4% (189 out of 2109) words in our experiment. The fewer lexical stress errors are made by users, the more challenging it is to detect it. They also used proprietary L2 English of Japanese speakers. Due to the lack of available benchmark and standard speech corpora for the task of lexical stress assessment, we could not make a fairer comparison with the state-of-the-art.

5.1. Experimental Results

First, we compare Att_NoTTS and NoAtt_NoTTS models. Using the attention mechanism for automatic extraction of syllable-level features significantly improves the detection of lexical stress errors. It is illustrated by precision-recall curves and AUC metric in Figure 3. To be comparable with the study by Ferrer et al., we fix recall to around 50% and compare the models using precision as shown in Table 2.

The Att_NoTTS attention-based can be further improved. Augmenting the training set with incorrectly stressed words (Att_TTS) boosts precision from 87.85% to 94.8%, at a recall level of 50%. Data augmentation helps because it increases the number of words with incorrect stress patterns in the training set. It prevents the model from exploiting a strong correlation between phonemes and lexical stress in correctly stressed words. Using data augmentation in the simpler no-

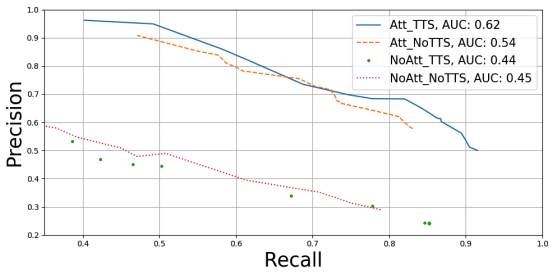


Figure 3: *Precision-recall curves for evaluated systems.*

attention-based model (NoAtt_TTS) does not help. It is because NoAtt_TTS uses only prosodic features for fixed regions of speech, so this model cannot overfit to phonetic input.

Table 2: *Precision and recall [%, 95% Confidence Interval] of detecting lexical stress errors, at around 50% recall. * - Ferrer et al. model has been evaluated on the data with 46.4% of lexical stress errors, compared to 9.4% of errors on our data set. This data point indicates that our proposed model AttTTS should outperform Ferrr et al. model if both were evaluated exactly in the same conditions.*

Model	Precision	Recall
AttTTS	94.8 (89.18-98.03)	49.2 (42.13-56.3)
AttNoTTS	87.85 (80.67-93.02)	49.74 (42.66-56.82)
NoAttTTS	44.39 (37.85-51.09)	50.26 (43.18-57.34)
NoAttNoTTS	48.98 (42.04-55.95)	50.79 (43.70-57.86)
Ferrer et al. [9] *	95.00 (na-na)	48.3 (na-na)

Ferrer et al. [9] reported on a similar performance to our Att_TTS model with a precision of 95% and a recall of 48.3% on L2 English speech of Japanese speakers. However, in their testing data, the proportion of incorrectly stressed words is much larger, which makes it easier to detect lexical stress errors.

6. Conclusion and Future Work

Using an attention-based neural network for the automatic extraction of syllable-level features significantly improves the detection of lexical stress errors in L2 English speech, compared to baseline models. However, this model has a tendency to classify lexical stress based on highly-correlated phonemes. We can counteract this effect by augmenting the training set with incorrectly stressed words generated with Neural TTS. It boosts the performance of the attention-based model by 14.8% in the AUC metric and by 7.9% in precision, while maintaining recall at a level close to 50%. Data Augmentation, however, does not help when applied to a simpler model without an attention mechanism.

We found that the current word-level model is not able to correctly classify lexical stress when two words are linked [35] and stress shift may occur [36]. For example, two neighboring phonemes /er/ in the text 'her arrange' /hh er - er ey n jh/ are pronounced as a single phoneme. Therefore, in future, we plan to move away from the assessment of isolated words and extend the current model to detect lexical stress errors at the sentence level. We plan to replace a single-speaker TTS model to generate synthetic lexical stress errors with a multi-speaker model. We plan to analyze the accuracy of detecting lexical stress errors for speakers with different proficiency levels of English.

7. References

[1] S. M. Witt and S. J. Young, "Phone-level pronunciation scoring and assessment for interactive language learning," *Speech communication*, vol. 30, no. 2-3, pp. 95–108, 2000.

[2] W.-K. Leung, X. Liu, and H. Meng, "Cnn-rnn-ctc based end-to-end mispronunciation detection and diagnosis," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 8132–8136.

[3] D. Korzekwa, J. Lorenzo-Trueba, S. Zaporowski, S. Calamaro, T. Drugman, and B. Kostek, "Mispronunciation detection in non-native (l2) english with uncertainty modeling," in *Accepted to ICASSP 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021.

[4] J. Field, "Intelligibility and the listener: The role of lexical stress," *TESOL quarterly*, vol. 39, no. 3, pp. 399–423, 2005.

[5] A. Lepage and M. G. Busà, "Intelligibility of english l2: The effects of incorrect word stress placement and incorrect vowel reduction in the speech of french and italian learners of english," in *Proc. of the Intl. Symposium on the Acquisition of Second Language Speech*, vol. 5, no. 2014, 2014, pp. 387–400.

[6] Y.-J. Jung, S.-C. Rhee *et al.*, "Acoustic analysis of english lexical stress produced by korean, japanese and taiwanese-chinese speakers," *Phonetics and Speech Sciences*, vol. 10, no. 1, pp. 15–22, 2018.

[7] D. R. v. Bergem, "Acoustic and lexical vowel reduction," in *Phonetics and Phonology of Speaking Styles*, 1991.

[8] J. Latorre, J. Lachowicz, J. Lorenzo-Trueba, T. Merritt, T. Drugman, S. Ronanki, and V. Klimkov, "Effect of data reduction on sequence-to-sequence neural tts," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 7075–7079.

[9] L. Ferrer, H. Bratt, C. Richey, H. Franco, V. Abrash, and K. Precoda, "Classification of lexical stress using spectral and prosodic features for computer-assisted language learning systems," *Speech Communication*, vol. 69, pp. 31–45, 2015.

[10] M. A. Shahin, J. Epps, and B. Ahmed, "Automatic classification of lexical stress in english and arabic languages using deep learning." in *INTERSPEECH*, 2016, pp. 175–179.

[11] J.-Y. Chen and L. Wang, "Automatic lexical stress detection for chinese learners' of english," in *2010 7th Intl. Symposium on Chinese Spoken Language Processing*. IEEE, 2010, pp. 407–411.

[12] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[13] M. I. Posner and S. E. Petersen, "The attention system of the human brain," *Annual review of neuroscience*, vol. 13, no. 1, pp. 25–42, 1990.

[14] Y. Ruan, X. Wang, H. Liu, Z. Ou, Y. Gao, J. Cheng, and Y. Qian, "An end-to-end approach for lexical stress detection based on transformer," *arXiv preprint arXiv:1911.04862*, 2019.

[15] K. Li, S. Mao, X. Li, Z. Wu, and H. Meng, "Automatic lexical stress and pitch accent detection for l2 english speech using multi-distribution deep neural networks," *Speech Communication*, vol. 96, pp. 28–36, 2018.

[16] J. Zhao, H. Yuan, J. Liu, and S. Xia, "Automatic lexical stress detection using acoustic features for computer assisted language learning," *Proc. APSIPA ASC*, pp. 247–251, 2011.

[17] N. Chen and Q. He, "Using nonlinear features in automatic english lexical stress detection," in *2007 Intl. Conference on Computational Intelligence and Security Workshops (CISW 2007)*. IEEE, 2007, pp. 328–332.

[18] K. Li, X. Qian, S. Kang, and H. Meng, "Lexical stress detection for l2 english speech using deep belief networks." in *Interspeech*, 2013, pp. 1811–1815.

[19] M. K. Ramanathi, C. Yarra, and P. K. Ghosh, "Asr inspired syllable stress detection for pronunciation evaluation without using a supervised classifier and syllable level features." in *INTERSPEECH*, 2019, pp. 924–928.

[20] M. McAuliffe, M. Socolof, S. Mihuc, M. Wagner, and M. Sonderegger, "Montreal forced aligner: Trainable text-speech alignment using kaldi." in *Interspeech*, vol. 2017, 2017, pp. 498–502.

[21] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an asr corpus based on public domain audio books," in *2015 IEEE Intl. Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2015, pp. 5206–5210.

[22] P. Boersma, "Praat: doing phonetics by computer," *http://www.praat.org/*, 2006.

[23] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," *arXiv preprint arXiv:1406.1078*, 2014.

[24] I. Elias, H. Zen, J. Shen, Y. Zhang, Y. Jia, R. Weiss, and Y. Wu, "Parallel tacotron: Non-autoregressive and controllable tts," *arXiv preprint arXiv:2010.11439*, 2020.

[25] M. I. Jordan, Z. Ghahramani, T. S. Jaakkola, and L. K. Saul, "An introduction to variational methods for graphical models," *Machine learning*, vol. 37, no. 2, pp. 183–233, 1999.

[26] T. e. a. Chen, "Mxnet: A flexible and efficient machine learning library for heterogeneous distributed systems," *arXiv preprint arXiv:1512.01274*, 2015.

[27] A. Paleyes, M. Pullin, M. Mahsereci, N. Lawrence, and J. Gonzalez, "Emulation of physical processes with emukit," in *Second Workshop on Machine Learning and the Physical Sciences, NeurIPS*, 2019.

[28] J. Guo, H. He, T. He, L. Lausen, M. Li, H. Lin, X. Shi, C. Wang, J. Xie, S. Zha *et al.*, "Gluoncv and gluonnlp: Deep learning in computer vision and natural language processing." *Journal of Machine Learning Research*, vol. 21, no. 23, pp. 1–7, 2020.

[29] J.-B. Michel, Y. K. Shen, A. P. Aiden, A. Veres, M. K. Gray, J. P. Pickett, D. Hoiberg, D. Clancy, P. Norvig, J. Orwant *et al.*, "Quantitative analysis of culture using millions of digitized books," *science*, vol. 331, no. 6014, pp. 176–182, 2011.

[30] J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, and D. S. Pallett, "Darpa timit acoustic-phonetic continous speech corpus cd-rom. nist speech disc 1-1.1," *STIN*, vol. 93, p. 27403, 1993.

[31] J. Kominek and A. W. Black, "The cmu arctic speech databases," in *Fifth ISCA workshop on speech synthesis*, 2004.

[32] G. Zhao, S. Sonsaat, A. O. Silpachai, I. Lucic, E. Chukharev-Hudilainen, J. Levis, and R. Gutierrez-Osuna, "L2-arctic: A non-native english speech corpus," *Perception Sensing Instrumentation Lab*, 2018.

[33] A. Porzuczek and A. Rojczyk, "English word stress in polish learners speech production and metacompetence," *Research in Language*, vol. 15, no. 4, pp. 313–323, 2017.

[34] A. Oord, Y. Li, I. Babuschkin, K. Simonyan, O. Vinyals, K. Kavukcuoglu, G. Driessche, E. Lockhart, L. Cobo, F. Stimberg *et al.*, "Parallel wavenet: Fast high-fidelity speech synthesis," in *International conference on machine learning*. PMLR, 2018, pp. 3918–3926.

[35] A. E. Hieke, "Linking as a marker of fluent speech," *Language and Speech*, vol. 27, no. 4, pp. 343–354, 1984.

[36] S. Shattuck-Hufnagel, M. Ostendorf, and K. Ross, "Stress shift and early pitch accent placement in lexical items in american english," *Journal of Phonetics*, vol. 22, no. 4, pp. 357–388, 1994.

INTERSPEECH 2019
September 15–19, 2019, Graz, Austria

Interpretable Deep Learning Model for the Detection and Reconstruction of Dysarthric Speech

Daniel Korzekwa[1], Roberto Barra-Chicote[1], Bozena Kostek[2], Thomas Drugman[1], Mateusz Lajszczak[1]

[1]Amazon TTS-Research
[2] Gdansk University of Technology, Faculty of ETI, Poland
korzekwa@amazon.com, rchicote@amazon.com, bokostek@multimed.org, drugman@amazon.com, mateuszl@amazon.com

Abstract

We present a novel deep learning model for the detection and reconstruction of dysarthric speech. We train the model with a multi-task learning technique to jointly solve dysarthria detection and speech reconstruction tasks. The model key feature is a low-dimensional latent space that is meant to encode the properties of dysarthric speech. It is commonly believed that neural networks are black boxes that solve problems but do not provide interpretable outputs. On the contrary, we show that this latent space successfully encodes interpretable characteristics of dysarthria, is effective at detecting dysarthria, and that manipulation of the latent space allows the model to reconstruct healthy speech from dysarthric speech. This work can help patients and speech pathologists to improve their understanding of the condition, lead to more accurate diagnoses and aid in reconstructing healthy speech for afflicted patients.

Index Terms: dysarthria detection, speech recognition, speech synthesis, interpretable deep learning models

1. Introduction

Dysarthria is a motor speech disorder manifesting itself by a weakness of muscles controlled by the brain and nervous system that are used in the process of speech production, such as lips, jaw and throat [1]. Patients with dysarthria produce harsh and breathy speech with abnormal prosodic patterns, such as very low speech rate or flat intonation, which makes their speech unnatural and difficult to comprehend. Damage to the nervous system is the main cause of dysarthria [1]. It can happen as an effect of multiple possible neurological disorders such as cerebral palsy, brain stroke, dementia or brain cyst [2, 3].

Early onset detection of dysarthria may improve the quality of life for people affected by these neurological disorders. According to Alzheimer's Research UK2015 [4], 1 out of 3 people in the UK born in 2015 will develop dementia in their life. Manual detection of dysarthria conducted in clinical conditions by speech pathologists is costly, time-consuming and can lead to an incorrect diagnosis [5, 6]. With an automated analysis of speech, we can detect an early onset of dysarthria and recommend further health checks with a clinician even when a human speech pathologist is not available. Speech reconstruction may help with better identification of the symptoms and enable patients with severe dysarthria to communicate with other people.

Section 2 presents related work. In Section 3 we describe the proposed model for detection and reconstruction of dysarthria. In Section 4 we demonstrate the performance of the model with experiments on detection, interpretability, and reconstruction of healthy speech from dysarthric speech. We conclude with our remarks.

2. Related work

2.1. Dysarthria detection

Deep neural networks can automatically detect dysarthric patterns without any prior expert knowledge [7, 8]. Unfortunately, these models are difficult to interpret because they are usually composed of multiple layers producing multidimensional outputs with an arbitrary meaning and representation. Contrarily, statistical models based on a fixed vector of handcrafted prosodic and spectral features such as jitter, shimmer, Noise to Harmonic Ratio (NHR) or Mel-Frequency Cepstral Coefficients (MFCC) offer good interpretability but require experts to manually design predictor features [9, 10, 11, 12].

The work of Tu Ming et al. on interpretable objective evaluation of dysarthria [13] is the closest we found to our proposal. The main difference is that our model not only provides interpretable characteristics of dysarthria but also reconstructs healthy speech. Their model is based on feed-forward deep neural networks with a latent layer representing four dimensions of dysarthria: nasality, vocal quality, articulatory precision, and prosody. The final output of the network represents general dysarthria severity on a scale from 1 to 7. The input to this model is described by a 1201-dimensional vector of spectral and cepstral features that capture various aspects of dysarthric speech such as rhythm, glottal movement or formants. As opposed to this work, we use only mel-spectrograms to present the input speech to the model. Similarly to our approach, Vasquez-Correa et al. [8] uses a mel-spectrogram representation for dysarthria detection. However, they use 160 ms long time windows at the transition points between voiced and unvoiced speech segments, in contrast to using a full mel-spectrogram in our approach.

2.2. Speech reconstruction

There are three different approaches to the reconstruction of dysarthric speech: voice banking, voice adaptation and voice reconstruction [5]. Voice banking is a simple idea of collecting a patient's speech samples before their speech becomes unintelligible and using it to build a personalized Text-To-Speech (TTS) voice. It requires about 1800 utterances for a basic unit-selection TTS technology [14] and more than 5K utterances for building a Neural TTS voice [15]. Voice adaptation requires as little as 7 minutes of recordings. In this approach, we start with a TTS model of an average speaker and adapt its acoustic and articulatory parameters to the target speaker [16].

Both voice banking and voice adaptation techniques rely on the availability of recordings for a healthy speaker. The voice reconstruction technique overcomes this shortcoming. This

Copyright © 2019 ISCA

http://dx.doi.org/10.21437/Interspeech.2019-1206

technique aims at restoring damaged speech by tuning parameters representing the glottal source and the vocal tract filter [17, 18]. In our model, we take a similar approach. However, instead of making assumptions on what parameters should be restored, we let the model automatically learn the best dimensions of the latent space that are responsible for dysarthric speech. Reconstruction of healthy speech by manipulating the latent space of a dysarthric speech is a promising direction, however, so far we only managed to successfully apply this technique in a single-speaker setup.

Variational Auto-Encoder (VAE) [19] is a probabilistic latent space model that has recently become popular for the reconstruction of various signals such as text [20, 21] and speech [22, 23].

3. Proposed model

The model consists of two output networks, jointly trained, with a shared encoder as shown in Figure 1. The audio and text encoders produce a low-dimensional dysarthric latent space and a sequential encoding of the input text. The audio decoder reconstructs input mel-spectrogram from a dysarthric latent space and encoded text. Logistic classification model predicts the probability of dysarthric speech from the dysarthric latent space. In Table 1 we present the details of various neural blocks used in the model.

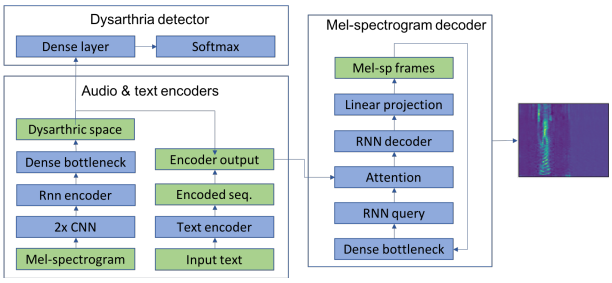


Figure 1: *Architecture of deep learning model for detection and reconstruction of dysarthric speech.*

Let us define a matrix $X : [n_{mels}, n_f]$ representing a mel-spectrogram (frame length=50ms and frame shift=12.5ms), where $n_{mels} = 128$ is the number of mel-frequency bands and n_f is the number of frames. Let us define a matrix $T : [n_c, n_t]$ representing a one-hot encoded input text, where n_c is the number of unique characters in the alphabet and n_t is the number of characters in the input text. The mel-spectrogram X is encoded into 2-dimensional dysarthria latent space $\mathbf{l} = \{l_1, l_2\}$ and then used as a conditioning variable for estimating the probability of dysarthria $d \backsim p(d|X, \theta)$ and reconstructing the mel-spectrogram $Y \backsim p(Y|X, T, \theta)$. Limiting the latent space to 2 dimensions makes the model more resilient to overfitting. The theta is a vector of trainable parameters of the model.

Let us define a training set of m tuples of $((X, T), y)$, where $y \in \{0, 1\}$ is the label for normal/dysarthric speech and m is the number of speech mel-spectrograms for dysarthric and normal speakers. We optimize a joint cost of the predicted probability of dysarthria and mel-spectrogram reconstruction defined as a weighted function:

$$\sum_{i=1}^{m} \alpha log(p(d_i|X_i, \theta)) + (1 - \alpha) log(p(Y_i|X_i, T_i, \theta)) \quad (1)$$

Table 1: *Configuration of the neural network blocks.*

Neural block	Config
	Audio encoder
2x CNN	20 channels, 5x5 kernel, RELU, VALID
GRU	20 hidden states, 1 layer
Dense	20 units, tanh
Dysarthric space	2 units, linear
	Text encoder
3x CNN	40 channels, 5x5 kernel, RELU, SAME
GRU	27 hidden states, 1 layer
	Audio decoder
Dense bottleneck	96 units, RELU
GRU query	29 hidden states, 1 layer
GRU decoder	128 hidden states, 1 layer
Linear projection	frames_num x melsp bins units, linear

where $log(p(d_i|X_i, \theta))$ is the cross-entropy between the predicted and actual labels of dysarthria, and $log(p(Y_i|X_i, T_i, \theta))$ is the log-likelihood of a Gaussian distribution for the predicted mel-spectrogram with a unit variance, a.k.a L2 loss. We used backpropagation and mini-batch stochastic gradient descent with a learning rate of 0.03 and a batch size of 50. The whole model is initialized with Xaviers method [24] using the magnitude value of 2.24. Hyper-parameters of the model presented in Table 1 were tuned with a grid search optimization. We used MxNet framework for implementing the model [25].

3.1. Mel-spectrogram and text encoders

For the spectrogram encoder, we use a Recurrent Convolutional Neural Network model (RCNN) [26]. The convolutional layers, each followed by a max-pooling layer, extract local and time-invariant patterns of the glottal source and the vocal tract. The GRU layer models temporal patterns of dysarthric speech [27]. The last state of the GRU layer is processed by two dense layers. Dropout [28] with probability of 0.5 is applied to the output of the activations for both CNN layers, GRU layer, and the dense layer.

Text encoder encodes the input text using one-hot encoding, followed by three CNN layers and one GRU layer. Outputs of both audio and text encoders are concatenated via matrix broadcasting, producing a matrix $E : [n_c + n_l, n_t]$, where n_l is dimensionality of the dysarthria latent space.

3.2. Spectrogram decoder and dysarthria detector

For decoding a mel-spectrogram, similarly to Wang et al. [29], we use a Recurrent Neural Network (RNN) model with attention. The dot-product attention mechanism [30] plays a crucial role. It informs to which elements of the encoder output the decoder should pay attention at every decoder step. The RNN network that produces a query vector for the attention, takes as input r predicted mel-spectrogram frames from the previous time-step. The output of the RNN decoder is projected via a linear dense layer into r number of mel-spectrogram frames. Similarly to Wang et al. [29], we found that it is important to preprocess the mel-spectrogram with a dense layer and dropout regularization to improve the overall generalization of the model.

The dysarthria detector is created from a 2-dimensional dense layer. It uses a tanh activation followed by a softmax function that represents the probability of dysarthric speech.

4. Experiments

4.1. Dysarthric speech database

There is no well-established benchmark in the literature to compare different models for detecting dysarthria. Aside from the most popular dysarthric corpora, UA-Speech [31] and TORGO [32], there are multiple speech databases created for the purpose of a specific study, for example, corpora of 57 dysarthric speakers [12] and Enderby Frenchay Assessment dataset [6]. Many corpora, including TORGO and HomeService [33], are available under non-commercial license.

In our experiments we use the UA-Speech database from the University of Illinois [31]. It contains 11 male and 4 female dysarthric speakers of different dysarthria severity levels and 13 control speakers. 455 isolated words are recorded for each speaker with 1 to 3 repetitions. Every word is recorded through a 7-channel microphone array, producing a separate wav file of 16 kHz sampling rate for every channel. It contains 9.4 hours of speech for dysarthric speakers and 4.85 hours for control speakers. UA-Speech corpus comes with intelligibility scores that are obtained from a transcription task performed by 5 naive listeners.

To control variabilities in recording conditions, we normalized mel-spectrograms for every recorded word independently with a z-score normalization. We considered removing the initial period of silence at the beginning of recorded words but we decided against it. We found that for dysarthric speakers of high speech intelligibility, the average length of the initial silence period that lasts 0.569sec +- 0.04674 (99% CI) is comparable with healthy speakers with the length of 0.532sec +- 0.055. Because we can predict unvoiced periods with merely 85% of accuracy [34], removing the periods of silence for dysarthric speakers with poor intelligibility is very inaccurate.

4.2. Automatic detection of dysarthria

To define the training and test sets, we use a Leave-One-Subject-Out (LOSO) cross-validation scheme. For each training, we include all speakers but one that is left out to measure the prediction accuracy on unseen examples. The accuracy, precision and recall metrics are computed at a speaker level (the average dysarthria probability of all the words produced by the speaker is compared to a target speaker dysarthria label $\in \{0, 1\}$), and a word level (comparing target dysarthria label with predicted dysarthria probability for all words independently).

As a baseline, we use the Gillespie's et al. model that is based on Support Vector Machine classifier [11]. It uses 1595 low-level predictor features processed with a global z-score normalization. It reports a 75.3 and 92.9 accuracy in the dysarthria detection task at the word and speaker levels respectively, following LOSO cross-validation. However, Gillespie uses 336 words from the UA-Speech corpus with 12 words per speaker, whereas we use all 455 words across all speakers.

In our first model, only dysarthric labels are observed and we achieved an accuracy on the word and speaker levels of 82% and 93% respectively. By training the multi-task model, in which both targets, i.e. mel-spectrogram and dysarthric labels, are observed, the accuracy on the word level increased by 3 percents to the value of 85.3% (Table 2). We found that the UA-Speech database includes multiple recorded words for healthy speakers that contain intelligibility errors, different words than asked or background speech of other people. These issues affect the accuracy of detecting dysarthric speech.

Table 2: *Accuracy of dysarthria detection including 95% CI. Classifier task - target mel-spectrogram (ML) is not observed during training. Multitask - both targets ML and dysarthric labels are observed*

System	Accuracy	Precision	Recall
	Word level		
Multitask	0.853 (0.849 - 0.857)	0.831	0.911
Classifier task	0.820 (0.815 - 0.824)	0.818	0.855
Gillespie et al.[11]	0.753 (na)	0.823	0.728
	Speaker level		
Multitask	0.929 (0.790-0.984)	1.000	0.867
Classifier task	0.929 (0.790-0.984)	0.933	0.933
Gillespie et al.[11]	0.929 (na)	na	na

Krishna reports a 97.5% accuracy on UA-Corpus [7]. However, after email clarification with the author, we found that they estimated the accuracy taking into account only the speakers with a medium level of dysarthria. Narendra et al. achieved 93.06% utterance level accuracy on the TORGO dysarthric speech database [35]. As opposed to the related work, our model does not need any expert knowledge to design hand-crafted features and it can learn automatically using a low-dimensional latent space that encodes characteristics of dysarthria.

4.3. Interpretable modeling of dysarthric patterns

We analyze the correlation between the dysarthric latent space and the intelligibility of speakers. We look at 550 audio samples of a single 'Command' word across the 15 dysarthric speakers and 13 healthy speakers.

In an unsupervised training (Figure 2), target labels of dysarthric/normal speech are not presented to the model. Dysarthric speakers are well separated from normal speakers and the dimension 2 of the latent space is negatively correlated with the intelligibility scores (Pearson correlation of -0.84, two-sided p-value < 0.001). In a supervised variant (Figure 3), we train the model jointly with both reconstructed mel-spectrogram and the target dysarthria labels observed. Both dimensions of the latent space are highly correlated with the intelligibility scores (dimension 1 with correlation of -0.76 and dimension 2 with correlation of 0.70, both with p-value < 0.001).

The sign of the correlation has no particular meaning. Retraining the model multiple times results in both positive and negative correlations between the latent space and the intelligibility of speech. A high correlation between dysarthric latent space and intelligibility scores suggests that by moving along the dimensions of the latent space, we should be able to reconstruct speech of dysarthric speakers and improve its intelligibility. We explore this in the next experiment.

4.4. Reconstruction of dysarthric speech

First we trained a supervised multi-speaker model with all dysarthric and control speakers but we achieved poor reconstruction results with almost unintelligible speech. We think this is due to a high variability of dysarthric speech across all speakers, including various articulation, prosody and fluency problems. To better understand the potential for speech reconstruction, we narrowed the experiment down to two speakers, male speaker M05 and a corresponding control speaker. We have chosen M05 subject because their speech varies across different levels of fluency and we wanted to observe this pattern when manipulating the latent space. For example, when pronounc-

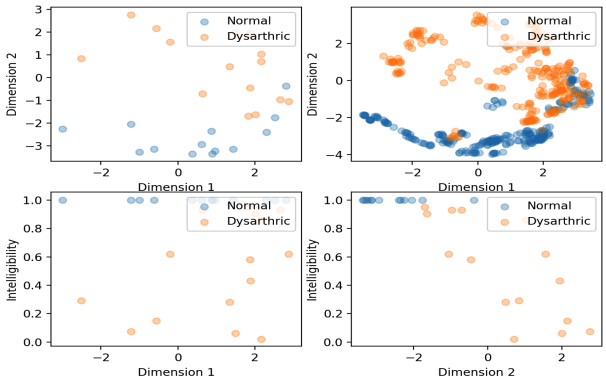


Figure 2: *Unsupervised learning. Top row: Separation between dysarthric and control speakers in the latent space on a speaker (left) and word (right) level. Bottom row: Correlation between both dimensions of the latent space and the intelligibility scores.*

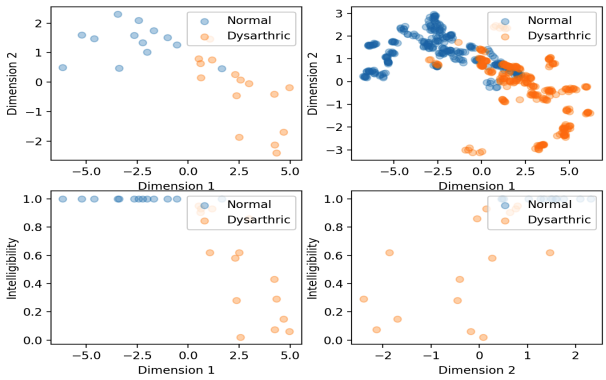


Figure 3: *Supervised learning. As in Figure 2.*

ing the word 'backspace', M05 uttered consonants 'b' and 's' multiple times, resulting in 'ba ba cs space'.

We analyzed a single category of 19 computer command words, such as 'command' or 'backspace'. For every word uttered by M05, we generated 5 different versions of speech, fixing dimension 2 of the latent space to the value of -0.1, and using the values of [-0.5, 0, 0.5, 1, 1.5] for dimension 1. Audio samples of reconstructed speech were obtained by converting predicted mel-spectrograms to waveforms using the Griffin-Lim algorithm [36].

We conducted MUSHRA perceptual test [37]. Every listener was presented with 6 versions of a given word at the same time, 5 reconstructions and one version of recorded speech. We asked listeners to evaluate the fluency of speech on a scale from 0 to 100. We used 10 US based listeners from the Amazon mTurk platform, in total providing us with 1140 evaluated speech samples.

As shown in Figure 4, by moving along dimension 1 of the latent space, we can improve the fluency of speech, generating speech with levels of fluency not observed in the training data. In the pairwise two-sided Wilcoxon signed-rank, all pairs of ranks are different from each other with p-value < 0.001, except of {orig, d1=1.0}, {d1=-0.5, d1=0.0}, {d1=-0.5, d1=0.5}. Examples of original and reconstructed mel-spectrograms are shown in Figure 5.

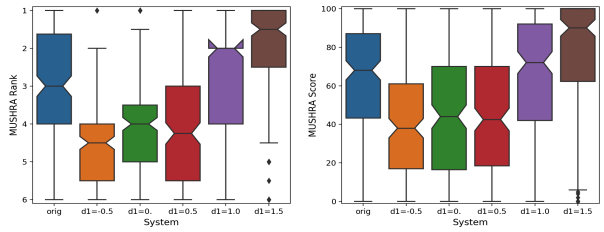


Figure 4: *MUSHRA results for the fluency of speech for 5 reconstructions and one recorded speech. Rank order (left) and the median score on the scale from 0 to 100 (right).*

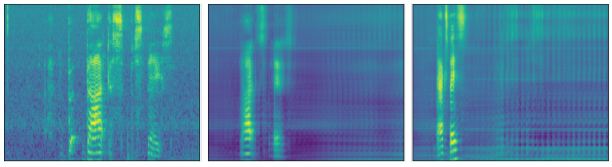


Figure 5: *Reconstruction of dysarthric speech ('command' word). From left to right (MUSHRA scores of 51.8, 61.9 and 89.5): Recorded dysarthric speech. Reconstructed speech with dimension 1 of 0.0 and 1.5 respectively.*

We found that manipulation of the latent space changes both the fluency of speech and the timbre of voice and it is possible that dysarthria is so tied up with speaker identify making it fruitless to disentangle them. We replaced a deterministic dysarthric latent space with a Gaussian variable and trained the model with an additional Kullback–Leibler loss [19, 38] but we did not manage to separate the timbre of voice from dysarthria. Training the model with an additional discriminative cost to ensure that every dimension of the latent space is directly associated with a particular speech factor can potentially help with this problem [20].

5. Conclusions

This paper proposed a novel approach for the detection and reconstruction of dysarthric speech. The encoder-decoder model factorizes speech into a low-dimensional latent space and encoding of the input text. We showed that the latent space conveys interpretable characteristics of dysarthria, such as intelligibility and fluency of speech. MUSHRA perceptual test demonstrated that the adaptation of the latent space let the model generate speech of improved fluency. The multi-task supervised approach for predicting both the probability of dysarthric speech and the mel-spectrogram helps improve the detection of dysarthria with higher accuracy. This is thanks to a low-dimensional latent space of the auto-encoder as opposed to directly predicting dysarthria from a highly dimensional mel-spectrogram.

6. Acknowledgements

We would like to thank A. Nadolski, J. Droppo, J. Rohnke and V. Klimkov for insightful discussions on this work.

7. References

[1] ASHA, "The American Speech-Language-Hearing Association (ASHA) - Dysarthria," 2018.

[2] M. L. Cuny, M. Pallone, H. Piana, N. Boddaert, C. Sainte-Rose, L. Vaivre-Douret, P. Piolino, and S. Puget, "Neuropsychological improvement after posterior fossa arachnoid cyst drainage," *Child's Nervous System*, 2017.

[3] S. Banovic, L. Zunic, and O. Sinanovic, "Communication Difficulties as a Result of Dementia," *Materia Socio Medica*, vol. 30, no. 2, p. 221, 2018. [Online]. Available: https://www.ejmanager.com/fulltextpdf.php?mno=302643414

[4] Alzheimersresearchuk, "One in three people born in 2015 will develop dementia, new analysis shows," 2015.

[5] J. Yamagishi, C. Veaux, S. King, and S. Renals, "Speech synthesis technologies for individuals with vocal disabilities: Voice banking and reconstruction," *Acoustical Science and Technology*, vol. 33, no. 1, pp. 1–5, 2012.

[6] J. Carmichael, V. Wan, and P. Green, "Combining neural network and rule-based systems for dysarthria diagnosis," in *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, 2008.

[7] G. Krishna, "Excitation Source Analysis of Dysarthric Speech for Early Stage Detection of Dysarthria," *WSPD*, 2018.

[8] J. C. Vásquez-Correa, T. Arias-Vergara, J. R. Orozco-Arroyave, and E. Nöth, "A Multitask Learning Approach to Assess the Dysarthria Severity in Patients with Parkinson's Disease," in *Interspeech 2018, 19th Annual Conference of the International Speech Communication Association, Hyderabad, India, 2-6 September 2018.*, B. Yegnanarayana, Ed. ISCA, 2018, pp. 456–460.

[9] T. H. Falk, W. Y. Chan, and F. Shein, "Characterization of atypical vocal source excitation, temporal dynamics and prosody for objective measurement of dysarthric word intelligibility," *Speech Communication*, 2012.

[10] M. Sarria-Paja and T. Falk, "Automated Dysarthria Severity Classification for Improved Objective Intelligibility Assessment of Spastic Dysarthric Speech." in *Interspeech*, 2012.

[11] S. Gillespie, Y. Y. Logan, E. Moore, J. Laures-Gore, S. Russell, and R. Patel, "Cross-database models for the classification of dysarthria presence," in *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, 2017.

[12] K. L. Lansford and J. M. Liss, "Vowel Acoustics in Dysarthria: Speech Disorder Diagnosis and Classification," *Journal of Speech Language and Hearing Research*, 2014.

[13] M. Tu, V. Berisha, and J. Liss, "Interpretable Objective Assessment of Dysarthric Speech Based on Deep Neural Networks," in *Interspeech 2017, 18th Annual Conference of the International Speech Communication Association, Stockholm, Sweden, August 20-24, 2017*, F. Lacerda, Ed. ISCA, 2017, pp. 1849–1853.

[14] Modeltalker, "www.modeltalker.com."

[15] J. Latorre, J. Lachowicz, J. Lorenzo-Trueba, T. Merritt, T. Drugman, S. Ronanki, and K. Viacheslav, "Effect of data reduction on sequence-to-sequence neural {TTS}," *CoRR*, vol. abs/1811.0, 2018.

[16] Z. Ahmad Khan, P. Green, S. Creer, and S. Cunningham, "Reconstructing the voice of an individual following laryngectomy," 2011.

[17] L. Rabiner and R. Schafer, *Digital Processing of Speech Signals*. Englewood Cliffs: Prentice Hall, 1978.

[18] T. Drugman, P. Alku, A. Alwan, and B. Yegnanarayana, "Glottal source processing: from analysis to applications," *Computer Speech and Language*, vol. 28, 09 2014.

[19] C. Doersch, "Tutorial on Variational Autoencoders," 2016.

[20] Z. Hu, Z. Yang, X. Liang, R. Salakhutdinov, and E. P. Xing, "Controllable Text Generation," *CoRR*, vol. abs/1703.0, 2017.

[21] S. R. Bowman, L. Vilnis, O. Vinyals, A. M. Dai, R. Józefowicz, and S. Bengio, "Generating Sentences from a Continuous Space," *CoRR*, vol. abs/1511.0, 2015.

[22] Y.-J. Zhang, S. Pan, L. He, and Z.-H. Ling, "Learning latent representations for style control and transfer in end-to-end speech synthesis," *CoRR*, vol. abs/1812.0, 2018.

[23] W.-N. Hsu, Y. Zhang, and J. R. Glass, "Unsupervised Learning of Disentangled and Interpretable Representations from Sequential Data," *CoRR*, vol. abs/1709.0, 2017.

[24] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks." in *AISTATS*, ser. JMLR Proceedings, Y. W. Teh and D. M. Titterington, Eds., vol. 9. JMLR.org, 2010, pp. 249–256.

[25] T. Chen, M. Li, Y. Li, M. Lin, N. Wang, M. Wang, T. Xiao, B. Xu, C. Zhang, and Z. Zhang, "Mxnet: A flexible and efficient machine learning library for heterogeneous distributed systems," *CoRR*, vol. abs/1512.01274, 2015. [Online]. Available: http://arxiv.org/abs/1512.01274

[26] R. J. Skerry-Ryan, E. Battenberg, Y. Xiao, Y. Wang, D. Stanton, J. Shor, R. J. Weiss, R. Clark, and R. A. Saurous, "Towards End-to-End Prosody Transfer for Expressive Speech Synthesis with Tacotron," *CoRR*, vol. abs/1803.0, 2018.

[27] K. Cho, B. van Merrienboer, Ç. Gülçehre, F. Bougares, H. Schwenk, and Y. Bengio, "Learning Phrase Representations using {RNN} Encoder-Decoder for Statistical Machine Translation," *CoRR*, vol. abs/1406.1, 2014.

[28] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting," *Journal of Machine Learning Research*, vol. 15, pp. 1929–1958, 2014.

[29] Y. Wang, R. J. Skerry-Ryan, D. Stanton, Y. Wu, R. J. Weiss, N. Jaitly, Z. Yang, Y. Xiao, Z. Chen, S. Bengio, Q. V. Le, Y. Agiomyrgiannakis, R. Clark, and R. A. Saurous, "Tacotron: {A} Fully End-to-End Text-To-Speech Synthesis Model," *CoRR*, vol. abs/1703.1, 2017.

[30] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention Is All You Need," *CoRR*, vol. abs/1706.0, 2017.

[31] H. Kim, M. Hasegawa-Johnson, A. Perlman, J. Gunderson, T. Huang, K. Watkin, and S. Frame, "Dysarthric Speech Database for Universal Access Research," *INTERSPEECH*, 2008.

[32] F. Rudzicz, A. K. Namasivayam, and T. Wolff, "The TORGO database of acoustic and articulatory speech from speakers with dysarthria," *Language Resources and Evaluation*, 2012.

[33] M. Nicolao, H. Christensen, S. Cunningham, P. Green, and T. Hain, "A framework for collecting realistic recordings of dysarthric speech - The homeService corpus," in *Proceedings of the 10th International Conference on Language Resources and Evaluation, LREC 2016*, 2016.

[34] A. B. Johnston and D. C. Burnett, *WebRTC: APIs and RTCWEB Protocols of the HTML5 Real-Time Web*. USA: Digital Codex LLC, 2012.

[35] N. P. Narendra and P. Alku, "Dysarthric Speech Classification Using Glottal Features Computed from Non-words, Words and Sentences," in *Interspeech 2018, 19th Annual Conference of the International Speech Communication Association, Hyderabad, India, 2-6 September 2018.*, B. Yegnanarayana, Ed. ISCA, 2018, pp. 3403–3407.

[36] D. W. Griffin and J. S. Lim, "Signal Estimation from Modified Short-Time Fourier Transform," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 1984.

[37] T. Merritt, B. Putrycz, A. Nadolski, T. Ye, D. Korzekwa, W. Dolecki, T. Drugman, V. Klimkov, A. Moinet, A. Breen, R. Kuklinski, N. Strom, and R. Barra-Chicote, "Comprehensive evaluation of statistical speech waveform synthesis," nov 2018.

[38] E. Mathieu, T. Rainforth, N. Siddharth, and Y. W. Teh, "Disentangling Disentanglement in Variational Auto-Encoders," dec 2018.

Appendix D

Co-authored publication on pronunciation error detection prior to Ph.D. research

The first work on detecting pronunciation errors conducted by Daniel Korzekwa, preceding the doctorate, resulted in the co-authorship of the publication by Grzegorz Beringer. Grzegorz conducted a science internship on pronunciation assessment at Amazon, and Daniel Korzekwa was his mentor. The publication was presented at internal Amazon Machine Learning Conference (AMLC) in 2020, Seattle, United States.

Extending Goodness of Pronunciation to generate mispronunciation hypotheses for pronunciation assessment in L2-English

Grzegorz Beringer[1], *Daniel Korzekwa*[1], *Ariadna Sanchez*[2], *Bofei Wang*[3], *Jaime Lorenzo-Trueba*[2]

[1] Amazon.com, Alexa, Poland
[2] Amazon.com, Alexa, United Kingdom
[3] Amazon.com, Alexa, United States

beringg@amazon.com, korzekwa@amazon.com, cariadn@amazon.com, bofei@amazon.com, truebaj@amazon.com

Abstract

We propose a method to extend Goodness of Pronunciation (GOP), a commonly used pronunciation scoring metric, to generate mispronunciation hypotheses, which are then used to find what the speaker has actually uttered. We show that this allows to alleviate GOP's problem of being over-dependant on phone boundaries computed by force-alignment, leading to an improvement in mispronunciation detection and diagnosis. We also argue that introducing hypothesis prior could be used to improve the model in the context of pronunciation teaching, where high precision is required. We demonstrate that a method of increasing the prior of canonical hypothesis by a factor can enable us to have control over precision-recall trade-off. For our experiments, we use a dataset of isolated words, which contain recordings of 23 Polish-based speakers.

Index Terms: speech assessment, pronunciation error detection, L2 English, speech recognition

1. Introduction

A correct pronunciation is one of the key components to being understood. One's pronunciation level can be improved in sessions with language specialists, e.g. English lessons for non-native speakers. Additionally, it is important for the learner to frequently practice this skill outside of the sessions. This is where Computer-Assisted Pronunciation Training (CAPT) software can greatly help, enabling the learner to conduct pronunciation exercises at home and obtain feedback on their performance via pronunciation assessment (PA) models. PA is usually comprised of two areas: error detection (i.e. was a mistake made?) and diagnosis (i.e. what mistake was made?).

Usually, pronunciation error detection is evaluated in terms of accuracy, precision and recall. From the perspective of CAPT user (i.e. learner), it is important to strive for high precision before focusing on recall [1]. This way, we lower the chance of misinforming the user that they have made a mistake, which can be damaging to their morale and overall experience from using the software.

Various approaches to PA were proposed over the years, including pronunciation scoring metrics, such as Goodness of Pronunciation [2] (GOP). They use a pre-trained ASR model to segment the utterance into individual phones (force-alignment), and compute phone likelihoods in each speech frame (recognition). Error detection is then determined based on a threshold, which is applied to these scores. Such approaches show the ability to match expert pronunciation scores [2], making them a popular choice for CAPT software [1].

In this paper we make the following contributions. 1) We demonstrate that GOP achieves poor recall in a high precision setting, which is a prerequisite in pronunciation teaching software. We identify the main problem to be an over-dependance on phone boundaries computed with force-alignment, which can be suboptimal due to the non-native nature of the learner's speech. 2) We propose a multiple hypotheses extension to GOP, which allows to improve detection and diagnosis results in a high precision setting. Mispronunciation hypotheses are generated with GOP diagnosis results, and then scored with alignment likelihood to choose the most likely one. Since alignment likelihood is not as dependant as GOP on having precise phone boundaries, we are able to reduce false-positive rate for higher thresholds. 3) We demonstrate the advantages of adding hypothesis prior to enable better control of precision-recall trade-off.

In Section 2, we review related pronunciation error detection work. In Section 3, we define the problem of pronunciation error detection and describe our baseline, i.e. GOP. In Section 4, we introduce the idea of using GOP to generate mispronunciation hypotheses, which are then assesed with ASR, to choose the most likely recognition. In Section 5, we report experiments on a dataset of isolated words recorded by 23 Polish-English speakers. In Section 6, we draw conclusions and discuss future work.

2. Related Work

A popular way to tackle pronunciation error detection (PED) is by using scoring metrics, which aim to asign a continuous score to each phone p based on how good the pronunciation is. The detection of errors is determined by applying a threshold to these scores. Scores are commonly computed with likelihood-based measures, obtained from the acoustic model. It was shown that likelihood-based scores correlate relatively well with human-expert pronunciation scores, with posterior scores ($p(p|o)$) achieving best results [3]. It has become a defacto standard [1] to use the score named *Goodness of Pronunciation* (GOP) [2]. GOP approximates the posterior score as a ratio of the likelihood of the canonical phone (obtained from the force-alignment) to the likelihood of competing phones (free-phone loop). There have been a few improvements to scoring methods over the years, revolving mostly around changing acoustic models (e.g. move from GMM-HMM to DNN-HMM systems [4]), or incorporating more fine-grained information in the score calculation (e.g. HMM transition probabilities [5]). While above changes result in incremental improvements in terms of lowering the overall detetection error rate, they do not focus on the specific setting of high-precision pronunciation teaching, where GOP performs poorly due to many false positives.

The simplest approach to error detection given pronunci-

ation scores is to set a threshold, which can be global (single threshold for all phones) or phone-specific [2]. To improve results, it is common to build a classifier on top of likelihood-based features (scores for target and competing phones from the phoneset). This improves discerning between correct and incorrect renditions of p, e.g. using SVM [6] or neural network [7] based classifiers. If available, classification can be additionally conditioned on the reference audio observation o_r, which contains correct pronunciation of given text (usually spoken by native), and has shown to greatly improve results [7].

Relying on ASR for phone segmentation of o and likelihood estimation can be problematic, mostly due to the non-native nature of utterances in pronunciation assessment. If segmentation underperforms, then likelihood-based scores are poorly estimated and, therefore, the above methods are likely to fail. It is especially problematic for diagnosis, but can tamper error detection as well. One solution is to introduce multiple hypotheses H at input, which contain possible mispronunciations besides the canonical pronunciation. Acoustic model is then used to run recognition on an Extended Recognition Network (ERN), that is comprised of pronunciation hypotheses H, choosing such $h \in H$, that has the highest likelihood $p(o|h)$. Mispronunciation hypotheses are usually obtained by applying mispronunciation patterns to given text, using either hand-crafted rules [8], or rules extracted from a large-enough phonetically-labeled corpus of non-native speech [9]. Downside to both of these approaches is the L1-dependence (rules are different depending on speaker's native language) and strong bias towards recurrent mistakes. Instead, we suggest the generation of mispronunciation hypotheses by using diagnosis results from GOP, which is L1-independent, unbiased and easy to obtain (Section 4.1).

We can also set hypothesis for H to contain every possible phonetic hypothesis, therefore doing a lexicon-free recognition and relying solely on acoustic model capabilites. In that case, error detection is simply based on comparing recognized phonemes to canonical ones [10].

3. Pronunciation Error Detection (PED)

The goal of PED is to validate if phones uttered by the speaker match the expected phoneme string, i.e. a canonical pronunciation. It could therefore be framed as a binary classification problem of computing $p(e|o,t)$, where e is a Bernoulli variable meaning if an error occurred, o is the acoustic observation, and t is the text that is supposedly uttered. To conduct assessment at phone level, o has to be aligned with a corresponding phoneme sequence, i.e. canonical pronunciation of t. Therefore, a common practice is to use automatic speech recognition (ASR) models to segment audio into phones given the canonical pronunciation [2] [3] [4].

3.1. Goodness of Pronunciation (GOP)

Goodness of Pronunciation (GOP) is an algorithm that provides a score on how likely was each phone correctly pronounced in an utterance [2]. The phoneme sequence is known beforehand, and segmented using force-alignment with an acoustic model. GOP is calculated as shown in Equation 1. The GOP score of a phone p given the observation of p, i.e. o_p (provided by force-alignment technique) is its logarithmic posterior probability, normalised by the duration of the audio segment in frames $|o_p|$. As can be seen in Equation 1, the posterior probability is approximated as the likelihood of phone p over the maximum likelihood of any phone q from the phoneset Q.

$$GOP(p|o_p) = \frac{log(\frac{p(o_p|p)}{max_{q \in Q}\ p(o_p|q)})}{|o_p|} \tag{1}$$

We assume that only one phone can be contained in a given audio segment o_p from force-alignment, which means that a phone q with maximum likelihood can be considered the substitution of expected phone p. This supplies the learner with a diagnosis of the mispronunciation [4].

A threshold is later defined to determine which GOP scores are considered mispronunciations and which belong to correct pronunciations. In this paper, a single threshold is used for all phones. In other literature there has been implementations of phone-specific and speaker-specific thresholds [2] [11], slightly improving performance but not solving the main issues in GOP algorithm.

GOP suffers from a high number of phone-level false-positives, as for example in Table 1. GOP scores phoneme *iy* below the threshold, even though it is correct. It also detects *ih* $\rightarrow$ *iy* substitution, favoured by GOP: the score for the erroneous hypothesis achieves better mean and min GOP scores. This is caused by mismatches in phone boundaries from force-alignment, which leads to noisy score estimations. On the other hand, the likelihood of the whole hypothesis is higher for the actual pronunciation. This brings the intuition that using alignment likelihood to score multiple hypotheses should improve both diagnosis and detection precision performance.

Non-native substitutions are difficult to score reliably, given that Q is the list of canonical phones of a certain language. GOP can handle substitutions reliably, where the amount of phonemes in the utterance equals to the amount of phonemes in the observation. However, it does not handle insertions and deletions well, given that it calculates scores for all the phonemes expected in the utterance, which are incorrectly force-aligned. Handling insertions and deletions is, however, out of scope for this paper.

Hypothesis	GOP (mean)	GOP (min)	Alignment Likelihood
eh - r - **iy** - a	76%	49%	13.12
eh - r - **y** - a (*)	90%	76%	10.16

Table 1: *Example of GOP scoring for different force-alignment hypotheses. GOP (mean) shows the average of GOP scores at the word level; GOP (min) shows the GOP score for phonemes iy and y respectively. (*)Wrong hypothesis.*

4. Proposed approach

We propose to extend GOP to generate multiple hypotheses, which are then scored with alignment likelihood, therefore reducing the need for precise phone boundaries.

4.1. Multi-pass Force-Alignment (MPFA)

As shown in Table 1, alignment likelihood is a better metric than GOP in terms of choosing between mispronunciation hypotheses. On the other hand, GOP is L1-independant, easy to obtain and relatively proficient at detecting and diagnosing substitution errors. Therefore, we propose to combine both approaches: generate the set of hypotheses H with GOP, and then score each hypothesis h with alignment likelihood $p(o|h)$ to find the recognition h_{rec}.

$$h_{\text{rec}} = \arg\max_{h \in H} p(o|h) \tag{2}$$

Recognized hypothesis is then compared to the canonical pronunciation h_{can} to find mistakes. For convience, we call this method MPFA (*multi-pass force-alignment*), since force-alignment is run more than once in the process.

Mispronunciation hypotheses are usually generated based on canonical pronunciation and mispronunciation patterns, which can be extracted from human expertise [8], or automatically from a non-native speech corpus [9]. In our research, we generate such hypotheses using only results from running GOP on canonical pronunciation hypothesis h_{can}. Specifically, we use phones that had the highest GOP in appropriate segments, to generate likely alternative hypotheses. An example of the approach of generating multiple hypotheses with GOP can be seen on Figure 1, where we generate 3 alternative mispronunciation hypotheses for the word *dinosaur*. Although we are still constrained to detect and diagnose substitution errors only, same as for baseline GOP (Section 3.1), this approach has the upside of not being as L1-specific and not as biased towards popular error patterns as ERN [8] [9].

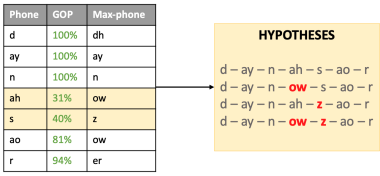


Figure 1: *Hypotheses generation process using GOP results. Alternative hypotheses with phone substitutions are created, using a threshold value to find phones that were likely mispronounced (50% in this case).*

4.2. Hypotheses prior

Equation 2 assumes that all hypotheses are equally likely to occur. In other words, we rely solely on acoustic information to generate and recognize what the speaker actually said.

Word	Hypothesis	Likelihood
assess	ah - s - eh - s	23.27
	f - s - eh - s	**24.67**

Table 2: *Example of scoring GOP-generated hypotheses. Due to non-native nature of evaluated recordings, acoustic model incorrectly recognised f→ ah substitution.*

There are instances where, due to the non-nativeness of a speaker's phone rendition, the acoustic model makes a mistake and recognizes an error, but from a linguistic and phonetic perspective, such mispronunciation is unlikely to occur (Table 2). Therefore, it makes sense to add a prior $p(h)$ to the scoring equation, which decreases the likelihood of unlikely hypotheses, and promotes the ones that are more likely to occur.

Updated equation looks as follows:

$$h_{\text{rec}} = \arg\max_{h \in H} p(o|h)p(h) \tag{3}$$

For the purpose of this paper, we evaluate the idea of positively rescoring the canonical hypothesis, therefore assuming that the canonical pronunciation is the most likely to occur. Priors of mispronunciation hypotheses are all equal in this scenario, increasing only $p(h_{\text{can}})$. This rescoring method is described with a **canonical rescoring factor** f, that demonstrates the scale of canonical prior $p(h_{\text{can}})$ over the prior of any mispronunciation hypothesis $p(h_{\text{mispron}})$:

$$f = \frac{p(h_{\text{can}})}{p(h_{\text{mispron}})} \tag{4}$$

Increasing the prior for canonical hypothesis should lead to higher precision, since the acoustic model needs to be more certain a mispronunciation occurred than with vanilla MPFA. We also hypothesize that f could be used to control precision-recall trade-off in circumstances like limiting strictness of the model for beginner learners.

5. Evaluation

5.1. Dataset

For the purpose of this paper, we recorded 23 L2-English speakers, 21 of which are Polish, 1 Ukrainian and 1 Belarusian. Recordings were done at Amazon Development Center and Gdańsk University of Technology, both located in Poland. Each speaker recorded 100 words, chosen by a professional English teacher as commonly mispronounced words by her students. Unlike other common L2-English datasets [12], we solely focus on isolated words, with the use-case of a word-repeating exercise in mind.

Phone-level annotations of what the speaker actually uttered were conducted by 2 professional linguists. Only words where both of them agree are used in the final evaluation, resulting in 767 recordings.

5.2. Evaluation metrics

Pronunciation error detection is usually evaluated with Equal Error Rate (EER) [4] [7], which evaluates in terms of pure accuracy of the system. However, for our case, false acceptance and false rejection mistakes should not be treated equally [1]. In order to not impede the learner's progress by discouraging them, the precision of the system should have higher priority (i.e. minimizing instances where mistakes are wrongly detected), before focusing on recall (i.e. catching most mispronunciations).

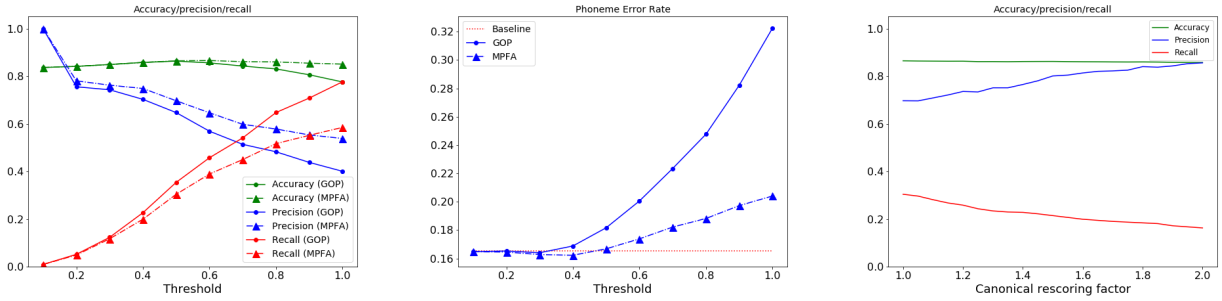
Since the canonical pronunciation, the actual pronunciation (according to annotation), and the recognized pronunciation might have different lengths, we align them by using Needleman-Wunsch sequence alignment algorithm [13] before evaluating with metrics. For detection, we evaluate phone-level precision and recall, based on whether mispronunciation was detected in the right place. For diagnosis, we check if recognized phones agree with the annotation, calculating Phoneme Error Rate (PER).

5.3. Acoustic model

We use a pre-trained Kaldi ASR model that is publicly available[1]. The acoustic model uses a time-delay neural network (TDNN) architecture [14], and was trained on reverberation-augmented Fisher database of mostly American-English telephone speech.

The input of the model are 40-dimensional MFCC features [15] and 100-dimensional iVector features [16] for each frame. Technically, we can improve recognition results on the dataset by estimating iVectors given each speaker's recordings. However, we choose not to include speaker information, treating each utterance separately. This allows us to keep consistent results regardless of the amount of data used per speaker.

[1] https://kaldi-asr.org/models/m1



(a) (b) (c)

Figure 2: *a) Phone-level detection results. b) Phone-level diagnosis results. Baseline is achieved by calculating PER between canonical hypothesis (input) and annotations. c) Influence of rescoring the canonical hypothesis (Eq. 4) for 0.5 threshold.*

Method	Prec.	Recall	Acc. (95% CI)
GOP	70%	22.6%	85.8% ± 1.0%
MPFA		30.4%	86.5% ± 1.0%
MPFA+f		**31.8%**	86.6% ± 1.0%
GOP	80%	3.5%	84.0% ± 1.2%
MPFA		3.3%	84.0% ± 1.2%
MPFA+f		**21.4%**	86.2% ± 1.1%

Table 3: *Performance of GOP, MPFA and MPFA+f (with rescoring) given 70% and 80% precision constraints. t means threshold and f means canonical rescoring factor. Accuracy was computed with 95% confidence interval.*

5.4. Results

Figure 2a shows accuracy, precision and recall of GOP and MPFA on the L2 dataset described in 5.1. Extending GOP to multiple hypotheses allows us to improve precision at the cost of recall, with the same or slightly higher accuracy. The change in performance is most visible in higher thresholds, where GOP normally experiences a lot of false positives. Using alignment likelihood to score mispronunciation hypothesis helps reducing these errors, since we fix instances where GOP score dropped locally due to the mismatches in detected phone boundaries.

Figure 2b shows the diagnosis performance in form of Phoneme Error Rate (PER). MPFA greatly improves recognition compared to GOP, but one has to keep in mind that the baseline of simply outputting the canonical form (which is the input to both GOP and MPFA) gives a relatively low PER. Only for some lower thresholds the baseline is beaten by the multiple hypotheses approach, which suggest that we are able to correctly diagnose some mispronounced phonemes without introducing too many false positives.

Figure 2c shows the influence of manipulating the prior of canonical hypothesis for a chosen threshold of 0.5. Increasing the canonical rescoring factor from default, i.e. $f = 1$ (equal priors), results in further improvements to precision, though the drop in recall is more severe, and overall accuracy slightly drops. The higher the factor, the more likely it is to miss some true phone errors.

As stated in Section 5.2, high precision of the model is very important from the perspective of the learner, and can be treated as a prerequisite. Table 3 demonstrates performance of GOP, MPFA and MPFA with rescoring and with 70% and 80% precision constraints, where we searched for highest recall we can obtain with each method. MPFA allows us to substantially improve recall for the 70% mark (compared to GOP), but in its vanilla form it fails to improve results for the 80% one. A very high precision constraint is where the benefit of canonical hypothesis rescoring is most visible, achieving 21% recall over others' 4%. This is because GOP and MPFA can only achieve such high precision for very low thresholds, where catching any errors is difficult. Hypothesis prior allows us to overcome this constraint and improve precision for higher thresholds. Increasing the prior of the canonical pronunciation allows us to catch mistakes only when the acoustic model is confident on its recognitions.

6. Conclusions and future work

In this paper, we demonstrated the shortcomings of GOP in a high precision setting, which is necessary for pronunciation teaching software. We identified the main issue to be GOP's over-dependance on quality of phone boundaries computed by force-alignment. To alleviate this problem, we proposed to use GOP to generate mispronunciation hypotheses, which are then scored with alignment likelihood to find the recognition. Consequently, it lead to an improvement in detection precision, accuracy and diagnosis.

We further demonstrated the need for adding a prior on the hypotheses. We also presented how positively rescoring the canonical pronunciation in relation to mispronunciation hypotheses can be used to further improve precision, going beyond levels achievable with GOP and MPFA.

Future work includes introducing other sources of hypotheses, e.g. automatically-generated mispronunciation rules [9], and other rescoring methods, e.g. a phone-level language model or prior probabilities of certain mispronunciation patterns. However, we recognize that the quality of the acoustic models used is a limitation for this approach. There are many instances where the correct hypothesis cannot be generated with GOP, and even if we included it in the hypotheses set, it would still score lower than some false hypotheses. Therefore, we believe that the biggest improvement could come from improving the acoustic model specifically for the task of pronunciation assessment of non-native speech, for example, by building a CTC-based [10] or an attention-based phoneme recognizer.

7. Acknowledgements

We would like to thank Amazon employees and Gdańsk Univeristy of Technology students who volunteered to be recorded for the purpose of this research; Amazon ADS team for annotating the dataset on phone level, and Alexa for Everyone team that built the prototype with us.

8. References

[1] N. F. Chen and H. Li, "Computer-assisted pronunciation training: From pronunciation scoring towards spoken language learning," *2016 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference, APSIPA 2016*, 2017.

[2] S. M. Witt and S. J. Young, "Phone-level pronunciation scoring and assessment for interactive language learning," *Speech Communication*, vol. 30, pp. 95–108, 2000.

[3] H. Franco, L. Neumeyer, Y. Kim, and O. Ronen, "Automatic pronunciation scoring for language instruction," *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, vol. 2, no. May 1997, pp. 1471–1474, 1997.

[4] W. Hu, Y. Qian, and F. K. Soong, "An Improved DNN-based Approach to Mispronunciation Detection and Diagnosis of L2 Learners ' Speech," *SLaTE*, pp. 71–76, 2015.

[5] S. Sudhakara, M. K. Ramanathi, C. Yarra, and P. K. Ghosh, "An Improved Goodness of Pronunciation (GoP) Measure for Pronunciation Evaluation with DNN-HMM System Considering HMM Transition Probabilities," in *Proc. Interspeech 2019*, 2019, pp. 954–958.

[6] S. Wei, G. Hu, Y. Hu, and R. H. Wang, "A new method for mispronunciation detection using Support Vector Machine based on Pronunciation Space Models," *Speech Communication*, vol. 51, no. 10, pp. 896–905, 2009.

[7] Y. Xiao, F. Soong, and W. Hu, "Paired phone-posteriors approach to esl pronunciation quality assessment," in *Proc. Interspeech 2018*, 2018, pp. 1631–1635.

[8] G. Kawai and K. Hirose, "A method for measuring the intelligibility and nonnativeness of phone quality in foreign language pronunciation training," *Conference on Spoken Language*, vol. 2, no. JANUARY, pp. 1–5, 1998. [Online]. Available: http://www.isca-speech.org/archive/icslp_1998/i98_0782.html

[9] D. Luo, X. Yang, and L. Wang, "Improvement of segmental mispronunciation detection with prior knowledge extracted from large L2 speech corpus," *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, no. August, pp. 1593–1596, 2011.

[10] W.-K. Leung, X. Liu, and H. Meng, "CNN-RNN-CTC Based End-to-End Mispronunciation Detection and Diagnosis," pp. 8132–8136, 2019.

[11] S. Kanters, C. Cucchiarini, and H. Strik, "The goodness of pronunciation algorithm: a detailed performance study," 2009.

[12] G. Zhao, S. Sonsaat, A. Silpachai, I. Lucic, E. Chukharev-Hudilainen, J. Levis, and R. Gutierrez-Osuna, "L2-arctic: A non-native english speech corpus," in *Proc. Interspeech*, 2018, p. 2783–2787.

[13] S. B. Needleman and C. D. Wunsch, "A general method applicable to the search for similarities in the amino acid sequence of two proteins," *Journal of Molecular Biology*, vol. 48, no. 3, pp. 443 – 453, 1970.

[14] V. Peddinti, G. Chen, V. Manohar, T. Ko, D. Povey, and S. Khudanpur, "JHU ASpIRE system: Robust LVCSR with TDNNS, iVector adaptation and RNN-LMS," *2015 IEEE Workshop on Automatic Speech Recognition and Understanding, ASRU 2015 - Proceedings*, pp. 539–546, 2016.

[15] S. B. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 28, no. 4, pp. 357–366, 1980.

[16] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Trans. Audio, Speech and Language Processing*, vol. 19, no. 4, pp. 788–798, 2011.

References

1. Abdi, H. (2007). "Binomial distribution: Binomial and sign tests". In: *Encyclopedia of measurement and statistics* 1.

2. Ahmad Khan, Z., P. Green, S. Creer, and S. Cunningham (2011). *Reconstructing the voice of an individual following laryngectomy*. DOI: 10.3109/07434618.2010.545078.

3. Ai, R. (2015). "Automatic pronunciation error detection and feedback generation for call applications". In: *International Conference on Learning and Collaboration Technologies*. Springer, pp. 175–186.

4. Ali, J., R. Khan, N. Ahmad, and I. Maqsood (2012). "Random forests and decision trees". In: *International Journal of Computer Science Issues (IJCSI)* 9.5, p. 272.

5. Alzheimersresearchuk (2015). "One in three people born in 2015 will develop dementia, new analysis shows". In:

6. Arpabet, W. (2022). *Arpabebet, accessed on June 2022*. URL: https://en.wikipedia.org/wiki/ARPABET.

7. ASHA (2018). "The American Speech-Language-Hearing Association (ASHA) - Dysarthria". In:

8. — (2022). "American Speech-Language-Hearing Association (ASHA), accessed on June 2022". In: URL: https://www.asha.org.

9. Asrifan, A., C. T. Zita, K. Vargheese, T. Syamsu, and M. Amir (2020). "THE EFFECTS OF CALL (COMPUTER ASSISTED LANGUAGE LEARNING) TOWARD THE STUDENTS'ENGLISH ACHIEVEMENT AND ATTITUDE". In: *Journal of advanced English studies* 3.2, pp. 94–106.

10. Atwell, E., P. Howarth, and D. Souter (2003). "The ISLE corpus: Italian and German spoken learner's English". In: *ICAME Journal: Intl. Computer Archive of Modern and Medieval English Journal* 27, pp. 5–18.

11. Badenhorst, J. and F. De Wet (2017). "The limitations of data perturbation for ASR of learner data in under-resourced languages". In: *2017 Pattern Recognition Association of South Africa and Robotics and Mechatronics (PRASA-RobMech)*. IEEE, pp. 44–49.

12. Bahdanau, D., J. Chorowski, D. Serdyuk, P. Brakel, and Y. Bengio (2016). "End-to-end attention-based large vocabulary speech recognition". In: *2016 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, pp. 4945–4949.

13. Banovic, S., L. Zunic, and O. Sinanovic (2018). "Communication Difficulties as a Result of Dementia". In: *Materia Socio Medica* 30.2, p. 221. ISSN: 1512-7680. DOI: 10.5455/msm.2018.30.221-224. URL: https://www.ejmanager.com/fulltextpdf.php?mno=302643414.

14. Banovic, S., L. J. Zunic, and O. Sinanovic (2018). "Communication difficulties as a result of dementia". In: *Materia socio-medica* 30.3, p. 221.

15. Bergem, D. R. v. (1991). "Acoustic and lexical vowel reduction". In: *Phonetics and Phonology of Speaking Styles*.

16. Beringer, G., D. Korzekwa, A. Sanchez, B. Wang, and J. Lorenzo-Trueba (2020). "Extending Goodness of Pronunciation to generate mispronunciationhypotheses for pronunciation assessment in L2-English". In: *Amazon Machine Learning Conference, Seattle*.

17. Bilinski, P., T. Merritt, A. Ezzerg, K. Pokora, S. Cygert, K. Yanagisawa, R. Barra-Chicote, and D. Korzekwa (2022). "Creating New Voices using Normalizing Flows". In: *accepted to Interspeech 2022*.

18. Bishop, C. M. (2006). "Pattern recognition". In: *Machine learning* 128.9.

19. Boersma, P. (2006). "Praat: doing phonetics by computer". In: *http://www.praat.org/*.

20. Botchkarev, A. (2018). "Performance metrics (error measures) in machine learning regression, forecasting and prognostics: Properties and typology". In: *arXiv preprint arXiv:1809.03006*.

21. Bowman, S. R., L. Vilnis, O. Vinyals, A. M. Dai, R. Józefowicz, and S. Bengio (2015). "Generating Sentences from a Continuous Space". In: *CoRR* abs/1511.0. arXiv: 1511.06349.

22. Brady, M. C., H. Kelly, J. Godwin, P. Enderby, and P. Campbell (2016). "Speech and language therapy for aphasia following stroke". In: *Cochrane database of systematic reviews* 6.

23. Carmichael, J., V. Wan, and P. Green (2008). "Combining neural network and rule-based systems for dysarthria diagnosis". In: *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*.

24. Chaudhari, S., V. Mithal, G. Polatkan, and R. Ramanath (2021). "An attentive survey of attention models". In: *ACM Transactions on Intelligent Systems and Technology (TIST)* 12.5, pp. 1–32.

25. Chen, J.-Y. and L. Wang (2010). "Automatic lexical stress detection for Chinese learners' of English". In: *2010 7th Intl. Symposium on Chinese Spoken Language Processing*. IEEE, pp. 407–411.

26. Chen, N. and Q. He (2007). "Using nonlinear features in automatic English lexical stress detection". In: *2007 Intl. Conference on Computational Intelligence and Security Workshops (CISW 2007)*. IEEE, pp. 328–332.

27. Chen, T., M. Li, Y. Li, M. Lin, N. Wang, M. Wang, T. Xiao, B. Xu, C. Zhang, and Z. Zhang (2015). "MXNet: A Flexible and Efficient Machine Learning Library for Heterogeneous Distributed Systems". In: *CoRR* abs/1512.01274. arXiv: 1512.01274. URL: http://arxiv.org/abs/1512.01274.

28. Chen, T. e. a. (2015). "Mxnet: A flexible and efficient machine learning library for heterogeneous distributed systems". In: *arXiv preprint arXiv:1512.01274*.

29. Cheng, S., Z. Liu, L. Li, Z. Tang, D. Wang, and T. F. Zheng (2020). "ASR-Free Pronunciation Assessment". In: *arXiv preprint arXiv:2005.11902*.

30. Cho, K., B. van Merrienboer, Ç. Gülçehre, F. Bougares, H. Schwenk, and Y. Bengio (2014). "Learning Phrase Representations using {RNN} Encoder-Decoder for Statistical Machine Translation". In: *CoRR* abs/1406.1. arXiv: 1406.1078.

31. Cho, K., B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio (2014). "Learning phrase representations using RNN encoder-decoder for statistical machine translation". In: *arXiv preprint arXiv:1406.1078*.

32. Chorowski, J., D. Bahdanau, K. Cho, and Y. Bengio (2014). "End-to-end continuous speech recognition using attention-based recurrent NN: First results". In: *arXiv preprint arXiv:1412.1602*.

33. Chorowski, J., R. J. Weiss, S. Bengio, and A. van den Oord (2019). "Unsupervised speech representation learning using wavenet autoencoders". In: *IEEE/ACM transactions on audio, speech, and language processing* 27.12, pp. 2041–2053.

34. Chorowski, J. K., D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio (2015). "Attention-based models for speech recognition". In: *Advances in neural information processing systems*, pp. 577–585.

35. Cuny, M. L., M. Pallone, H. Piana, N. Boddaert, C. Sainte-Rose, L. Vaivre-Douret, P. Piolino, and S. Puget (2017). "Neuropsychological improvement after posterior fossa arachnoid cyst drainage". In: *Child's Nervous System*. ISSN: 14330350. DOI: 10.1007/s00381-016-3285-x.

36. — (2017). "Neuropsychological improvement after posterior fossa arachnoid cyst drainage". In: *Child's Nervous System* 33.1, pp. 135–141.

37. Czyzewski, A., B. Kostek, P. Bratoszewski, J. Kotus, and M. Szykulski (2017). "An audio-visual corpus for multimodal automatic speech recognition". In: *Journal of Intelligent Information Systems* 49.2, pp. 167–192.

38. Damianou, A. and N. D. Lawrence (2013). "Deep gaussian processes". In: *Artificial intelligence and statistics*. PMLR, pp. 207–215.

39. Darwiche, A. (2009). *Modeling and reasoning with Bayesian networks*. Cambridge university press.

40. Denham, K. and A. Lobeck (2012). *Linguistics for everyone: An introduction*. Cengage Learning.

41. Doersch, C. (2016). *Tutorial on Variational Autoencoders*.

42. Drugman, T., P. Alku, A. Alwan, and B. Yegnanarayana (Sept. 2014). "Glottal Source Processing: from Analysis to Applications". In: *Computer Speech and Language* 28. DOI: `10.1016/j.csl.2014.03.003`.

43. Duan, R., T. Kawahara, M. Dantsuji, and H. Nanjo (2019). "Cross-lingual transfer learning of non-native acoustic modeling for pronunciation error detection and diagnosis". In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 28, pp. 391–401.

44. Duvenaud, D. (2014). "The Kernel cookbook: Advice on covariance functions, accessed on June 2022". In: URL: `https://www.cs.toronto.edu/~duvenaud/cookbook/`.

45. Eberhard, D. M., G. F. Simons, and C. D. Fennig (2021). "Ethnologue: Languages of the World. Twenty-fourth edition. Dallas". In: URL: `https://www.ethnologue.com/ethnoblog/gary-simons/welcome-24th-edition`.

46. EF-Education-First (2020). *EF English Proficiency Index*. URL: `https://www.ef.pl/assetscdn/WIBIwq6RdJvcD9bc8RMd/legacy/__/~/media/centralefcom/epi/downloads/full-reports/v10/ef-epi-2020-english.pdf`.

47. Eklund, V.-V. (2019). "Data Augmentation Techniques for Robust Audio Analysis". MA thesis. Tampere University.

48. Elias, I., H. Zen, J. Shen, Y. Zhang, Y. Jia, R. Weiss, and Y. Wu (2020). "Parallel Tacotron: Non-Autoregressive and Controllable TTS". In: *arXiv preprint arXiv:2010.11439*.

49. Erickson, N., J. Mueller, A. Shirkov, H. Zhang, P. Larroy, M. Li, and A. Smola (2020). "AutoGluon-Tabular: Robust and Accurate AutoML for Structured Data". In: *arXiv preprint arXiv:2003.06505*.

50. Ezzerg, A., A. Gabrys, B. Putrycz, D. Korzekwa, D. Saez-Trigueros, D. McHardy, K. Pokora, J. Lachowicz, J. Lorenzo-Trueba, and V. Klimkov (2021). "Enhancing audio quality for expressive Neural Text-to-Speech". In: *Proc. 11th ISCA Speech Synthesis Workshop (SSW 11)*, pp. 78–83. DOI: 10.21437/SSW.2021-14.

51. Falk, T. H., W. Y. Chan, and F. Shein (2012). "Characterization of atypical vocal source excitation, temporal dynamics and prosody for objective measurement of dysarthric word intelligibility". In: *Speech Communication*. ISSN: 01676393. DOI: 10.1016/j.specom.2011.03.007.

52. Farrajota, L., C. Maruta, J. Maroco, I. P. Martins, M. Guerreiro, and A. De Mendonca (2012). "Speech therapy in primary progressive aphasia: a pilot study". In: *Dementia and geriatric cognitive disorders extra* 2.1, pp. 321–331.

53. Fazel, A., W. Yang, Y. Liu, R. Barra-Chicote, Y. Meng, R. Maas, and J. Droppo (2021). "SynthASR: Unlocking Synthetic Data for Speech Recognition". In: *Proc. Interspeech 2021*, pp. 896–900. DOI: 10.21437/Interspeech.2021-1882.

54. Ferrer, L., H. Bratt, C. Richey, H. Franco, V. Abrash, and K. Precoda (2015). "Classification of lexical stress using spectral and prosodic features for computer-assisted language learning systems". In: *Speech Communication* 69, pp. 31–45.

55. Field, J. (2005). "Intelligibility and the listener: The role of lexical stress". In: *TESOL quarterly* 39.3, pp. 399–423. DOI: 10.2307/3588487.

56. Fouz-González, J. (2015). "Trends and directions in computer-assisted pronunciation training". In: *Investigating English Pronunciation*, pp. 314–342.

57. Franco, H., L. Neumeyer, Y. Kim, and O. Ronen (1997). "Automatic pronunciation scoring for language instruction". In: *1997 IEEE international conference on acoustics, speech, and signal processing*. Vol. 2. IEEE, pp. 1471–1474.

58. Fu, K., S. Gao, K. Wang, W. Li, X. Tian, and Z. Ma (2022). "Improving Non-native Word-level Pronunciation Scoring with Phone-level Mixup Data Augmentation and Multi-source Information". In: *arXiv preprint arXiv:2203.01826, submitted to INTERSPEECH 2022*. DOI: 10.48550/ARXIV.2203.01826.

59. Fu, K., J. Lin, D. Ke, Y. Xie, J. Zhang, and B. Lin (2021). "A Full Text-Dependent End to End Mispronunciation Detection and Diagnosis with Easy Data Augmentation Techniques". In: *arXiv preprint arXiv:2104.08428*.

60. Gabryś, A., Y. Jiao, V. Klimkov, D. Korzekwa, and R. Barra-Chicote (2021). "Improving the Expressiveness of Neural Vocoding with Non-Affine Normalizing Flows". In: *Proc. Interspeech 2021*, pp. 1679–1683. DOI: 10.21437/Interspeech.2021-1555.

61. Garofolo, J. S., L. F. Lamel, W. M. Fisher, J. G. Fiscus, and D. S. Pallett (1993). "DARPA TIMIT acoustic-phonetic continous speech corpus CD-ROM. NIST speech disc 1-1.1". In: *STIN* 93, p. 27403.

62. Gillespie, S., Y. Y. Logan, E. Moore, J. Laures-Gore, S. Russell, and R. Patel (2017). "Cross-database models for the classification of dysarthria presence". In: *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*. DOI: 10.21437/Interspeech.2017-216.

63. Glorot, X. and Y. Bengio (2010). "Understanding the difficulty of training deep feedforward neural networks." In: *AISTATS*. Ed. by Y. W. Teh and D. M. Titterington. Vol. 9. JMLR Proceedings. JMLR.org, pp. 249–256.

64. Golonka, E. M., A. R. Bowles, V. M. Frank, D. L. Richardson, and S. Freynik (2014). "Technologies for foreign language learning: A review of technology types and their effectiveness". In: *Computer assisted language learning* 27.1, pp. 70–105.

65. Gong, Y., Z. Chen, I.-H. Chu, P. Chang, and J. Glass (2022). "Transformer-Based Multi-Aspect Multi-Granularity Non-Native English Speaker Pronunciation Assessment". In: *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 7262–7266. DOI: 10.1109/ICASSP43922.2022.9746743.

66. Goodfellow, I., Y. Bengio, and A. Courville (2016). *Deep learning*. MIT press.

67. Graves, A., A. Mohamed, and G. Hinton (2013). "Speech recognition with deep recurrent neural networks". In: *2013 IEEE Intl. conference on acoustics, speech and signal processing*. IEEE, pp. 6645–6649.

68. Graves, A. (2012). "Connectionist temporal classification". In: *Supervised Sequence Labelling with Recurrent Neural Networks*. Springer, pp. 61–93.

69. Griffin, D. W. and J. S. Lim (1984). "Signal Estimation from Modified Short-Time Fourier Transform". In: *IEEE Transactions on Acoustics, Speech, and Signal Processing*. ISSN: 00963518. DOI: 10.1109/TASSP.1984.1164317.

70. Gu, J., Z. Wang, J. Kuen, L. Ma, A. Shahroudy, B. Shuai, T. Liu, X. Wang, G. Wang, J. Cai, et al. (2018). "Recent advances in convolutional neural networks". In: *Pattern Recognition* 77, pp. 354–377.

71. Guo, J. et al. (2020). "GluonCV and GluonNLP: Deep Learning in Computer Vision and Natural Language Processing." In: *Journal of Machine Learning Research* 21.23, pp. 1–7.

72. Harrison, A. M., W.-K. Lo, X.-j. Qian, and H. Meng (2009). "Implementation of an extended recognition network for mispronunciation detection and diagnosis in computer-assisted pronunciation training". In: *Intl. Workshop on Speech and Language Technology in Education*.

73. Heck, L. P., Y. Konig, M. K. Sönmez, and M. Weintraub (2000). "Robustness to telephone handset distortion in speaker recognition by discriminative feature design". In: *Speech Communication* 31.2-3, pp. 181–192. DOI: 10.1016/S0167-6393(99)00077-1.

74. Hieke, A. E. (1984). "Linking as a marker of fluent speech". In: *Language and Speech* 27.4, pp. 343–354.

75. Hines, A., E. Gillen, D. Kelly, J. Skoglund, A. Kokaram, and N. Harte (2015). "ViSQOLAudio: An objective audio quality metric for low bitrate codecs". In: *The Journal of the Acoustical Society of America* 137.6, EL449–EL455.

76. Hossin, M. and M. N. Sulaiman (2015). "A review on evaluation metrics for data classification evaluations". In: *International journal of data mining & knowledge management process* 5.2, p. 1.

77. Hsu, W.-N., Y. Zhang, and J. R. Glass (2017). "Unsupervised Learning of Disentangled and Interpretable Representations from Sequential Data". In: *CoRR* abs/1709.0. arXiv: 1709.07902.

78. Hu, Z., Z. Yang, X. Liang, R. Salakhutdinov, and E. P. Xing (2017). "Controllable Text Generation". In: *CoRR* abs/1703.0. arXiv: 1703.00955.

79. Huang, G., A. Gorin, J.-L. Gauvain, and L. Lamel (2016). "Machine translation based data augmentation for cantonese keyword spotting". In: *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, pp. 6020–6024.

80. Huang, W.-C., K. Kobayashi, Y.-H. Peng, C.-F. Liu, Y. Tsao, H.-M. Wang, and T. Toda (2021). "A Preliminary Study of a Two-Stage Paradigm for Preserving Speaker Identity in Dysarthric Voice Conversion". In: *arXiv preprint arXiv:2106.01415*.

81. Huybrechts, G., T. Merritt, G. Comini, B. Perz, R. Shah, and J. Lorenzo-Trueba (2021). "Low-resource expressive text-to-speech using data augmentation". In: *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, pp. 6593–6597.

82. Jia, Y., R. J. Weiss, F. Biadsy, W. Macherey, M. Johnson, Z. Chen, and Y. Wu (2019). "Direct speech-to-speech translation with a sequence-to-sequence model". In: *arXiv preprint arXiv:1904.06037*.

83. Jiao, Y., A. Gabryś, G. Tinchev, B. Putrycz, D. Korzekwa, and V. Klimkov (2021). "Universal neural vocoding with parallel wavenet". In: *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, pp. 6044–6048. DOI: 10.1109/ICASSP39728.2021.9414444.

84. Johnston, A. B. and D. C. Burnett (2012). *WebRTC: APIs and RTCWEB Protocols of the HTML5 Real-Time Web*. USA: Digital Codex LLC. ISBN: 0985978805, 9780985978808.

85. Jordan, M. I., Z. Ghahramani, T. S. Jaakkola, and L. K. Saul (1999). "An introduction to variational methods for graphical models". In: *Machine learning* 37.2, pp. 183–233.

86. Jung, Y.-J., S.-C. Rhee, et al. (2018). "Acoustic analysis of English lexical stress produced by Korean, Japanese and Taiwanese-Chinese speakers". In: *Phonetics and Speech Sciences* 10.1, pp. 15–22.

87. Jurafsky, D. and J. H. Martin (2009). *Speech and Language Processing (2Nd Edition)*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc. ISBN: 0131873210.

88. Kim, H., M. Hasegawa-Johnson, A. Perlman, J. Gunderson, T. Huang, K. Watkin, and S. Frame (2008). "Dysarthric Speech Database for Universal Access Research". In: *INTERSPEECH*. ISSN: 19909772.

89. Kobyzev, I., S. Prince, and M. Brubaker (2020). "Normalizing flows: An introduction and review of current methods". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

90. Koller, D. and N. Friedman (2009). *Probabilistic graphical models: principles and techniques*. MIT press.

91. Komatsu, S. and M. Sasayama (2019). "Speech Error Detection depending on Linguistic Units". In: *Proceedings of the 2019 3rd International Conference on Natural Language Processing and Information Retrieval*, pp. 75–79.

92. Kominek, J. and A. W. Black (2004). "The CMU Arctic speech databases". In: *Fifth ISCA workshop on speech synthesis*.

93. Korzekwa, D., R. Barra-Chicote, B. Kostek, T. Drugman, and M. Lajszczak (2019). "Interpretable Deep Learning Model for the Detection and Reconstruction of Dysarthric Speech". In: *Proc. Interspeech 2019*, pp. 3890–3894. DOI: 10.21437/Interspeech.2019-1206.

94. Korzekwa, D., R. Barra-Chicote, S. Zaporowski, G. Beringer, J. Lorenzo-Trueba, A. Serafinowicz, J. Droppo, T. Drugman, and B. Kostek (2021). "Detection of Lexical Stress Errors in Non-Native (L2) English with Data Augmentation and Attention". In: *Proc. Interspeech 2021*, pp. 3915–3919. DOI: 10.21437/Interspeech.2021-86.

95. Korzekwa, D. and B. Kostek (2019). "Deep learning model for automated assessment of lexical stress of non-native English speakers". In: *The Journal of the Acoustical Society of America* 146.4, pp. 2956–2957. DOI: 10.1121/1.5137270.

96. Korzekwa, D., J. Lorenzo-Trueba, T. Drugman, S. Calamaro, and B. Kostek (2021). "Weakly-Supervised Word-Level Pronunciation Error Detection in Non-Native English Speech". In: *Proc. Interspeech 2021*, pp. 4408–4412. DOI: 10.21437/Interspeech.2021-38.

97. Korzekwa, D., J. Lorenzo-Trueba, T. Drugman, and B. Kostek (2022). "Computer-assisted Pronunciation Training - Speech synthesis is almost all you need". In: *accepted for publication in Speech Communication Journal on June 17 '2022, in print*.

98. Korzekwa, D., J. Lorenzo-Trueba, S. Zaporowski, S. Calamaro, T. Drugman, and B. Kostek (2021). "Mispronunciation Detection in Non-Native (L2) English with Uncertainty Modeling". In: *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, pp. 7738–7742. DOI: 10.1109/ICASSP39728.2021.9413953.

99. Koyuncu, E., P. Çam, N. Altınok, D. E. Çallı, T. Y. Duman, and N. Özgirgin (2016). "Speech and language therapy for aphasia following subacute stroke". In: *Neural Regeneration Research* 11.10, p. 1591.

100. Krishna, G. (2018). "Excitation Source Analysis of Dysarthric Speech for Early Stage Detection of Dysarthria". In: *WSPD*.

101. Kroll, J. F. and P. E. Dussias (2017). "The benefits of multilingualism to the personal and professional development of residents of the US". In: *Foreign Language Annals* 50.2, pp. 248–259.

102. Lake, B. M., R. Salakhutdinov, and J. B. Tenenbaum (2015). "Human-level concept learning through probabilistic program induction". In: *Science* 350.6266, pp. 1332–1338.

103. Lansford, K. L. and J. M. Liss (2014). "Vowel Acoustics in Dysarthria: Speech Disorder Diagnosis and Classification". In: *Journal of Speech Language and Hearing Research*. ISSN: 1092-4388. DOI: 10.1044/1092-4388(2013/12-0262). arXiv: NIHMS150003.

104. Latorre, J., J. Lachowicz, J. Lorenzo-Trueba, T. Merritt, T. Drugman, S. Ronanki, and V. Klimkov (2019). "Effect of data reduction on sequence-to-sequence neural tts". In: *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, pp. 7075–7079.

105. Latorre, J., J. Lachowicz, J. Lorenzo-Trueba, T. Merritt, T. Drugman, S. Ronanki, and K. Viacheslav (2018). "Effect of data reduction on sequence-to-sequence neural {TTS}". In: *CoRR* abs/1811.0. arXiv: 1811.06315.

106. LeCun, Y., Y. Bengio, and G. Hinton (2015). "Deep learning". In: *nature* 521.7553, pp. 436–444.

107. Lee, A. et al. (2016). "Language-independent methods for computer-assisted pronunciation training". PhD thesis. Massachusetts Institute of Technology.

108. Lee, A. and J. R. Glass (2013). "Pronunciation assessment via a comparison-based system". In: *SLaTE*.

109. Lee, Y.-G. and S.-Y. Kim (2008). "Introduction to statistics". In: *Yulgokbooks, Korea*, pp. 342–351.

110. Lepage, A. and M. G. Busà (2014). "Intelligibility of English L2: The effects of incorrect word stress placement and incorrect vowel reduction in the speech of French and Italian learners of English". In: *Proceedings of the International Symposium on the Acquisition of Second Language Speech Concordia Working Papers in Applied Linguistics*. Vol. 5. 2014, pp. 387–400.

111. Leung, W.-K., X. Liu, and H. Meng (2019). "CNN-RNN-CTC based end-to-end mispronunciation detection and diagnosis". In: *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, pp. 8132–8136.

112. Levy, M. and G. Stockwell (2013). *CALL dimensions: Options and issues in computer-assisted language learning*. Routledge.

113. Li, H., S. Huang, S. Wang, and B. Xu (2011). "Context-Dependent Duration Modeling with Backoff Strategy and Look-Up Tables for Pronunciation Assessment and Mispronunciation Detection". In: *INTERSPEECH 2011, 12th Annual Conference of the International Speech Communication Association, Florence, Italy, August 27-31, 2011*. ISCA, pp. 1133–1136.

114. Li, K., S. Mao, X. Li, Z. Wu, and H. Meng (2018). "Automatic lexical stress and pitch accent detection for L2 English speech using multi-distribution deep neural networks". In: *Speech Communication* 96, pp. 28–36.

115. Li, K., X. Qian, S. Kang, and H. Meng (2013). "Lexical stress detection for L2 English speech using deep belief networks." In: *Interspeech*, pp. 1811–1815.

116. Li, K., X. Qian, and H. Meng (2016). "Mispronunciation detection and diagnosis in l2 English speech using multidistribution deep neural networks". In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 25.1, pp. 193–207.

117. Lin, B. and L. Wang (2021). "Deep Feature Transfer Learning for Automatic Pronunciation Assessment". In: *Proc. Interspeech 2021*, pp. 4438–4442. DOI: 10.21437/Interspeech.2021-931.

118. Lorenzo-Trueba, J., T. Drugman, J. Latorre, T. Merritt, B. Putrycz, R. Barra-Chicote, A. Moinet, and V. Aggarwal (2018). "Towards achieving robust universal neural vocoding". In: *arXiv preprint arXiv:1811.06292*.

119. Marcus, G. (2018). "Deep learning: A critical appraisal". In: *arXiv preprint arXiv:1801.00631*.

120. Mathieu, E., T. Rainforth, N. Siddharth, and Y. W. Teh (2018). "Disentangling Disentanglement in Variational Auto-Encoders". In: arXiv: 1812.02833.

121. McAuliffe, M., M. Socolof, S. Mihuc, M. Wagner, and M. Sonderegger (2017). "Montreal Forced Aligner: Trainable Text-Speech Alignment Using Kaldi." In: *Interspeech*. Vol. 2017, pp. 498–502.

122. Mehri Kamrood, A., M. Davoudi, S. Ghaniabadi, and S. M. R. Amirian (2019). "Diagnosing L2 learners' development through online computerized dynamic assessment". In: *Computer Assisted Language Learning*, pp. 1–30.

123. Merritt, T., A. Ezzerg, P. Biliński, M. Proszewska, K. Pokora, R. Barra-Chicote, and D. Korzekwa (2022). "Text-Free Non-Parallel Many-To-Many Voice Conversion Using Normalising Flow". In: *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6782–6786. DOI: 10.1109/ICASSP43922.2022.9746368.

124. Merritt, T., B. Putrycz, A. Nadolski, T. Ye, D. Korzekwa, W. Dolecki, T. Drugman, V. Klimkov, A. Moinet, A. Breen, et al. (2018). "Comprehensive evaluation of statistical speech waveform synthesis". In: *2018 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, pp. 325–331.

125. Meyes, R., M. Lu, C. W. de Puiseau, and T. Meisen (2019). "Ablation studies in artificial neural networks". In: *arXiv preprint arXiv:1901.08644*.

126. Michel, J.-B., Y. K. Shen, A. P. Aiden, A. Veres, M. K. Gray, J. P. Pickett, D. Hoiberg, D. Clancy, P. Norvig, J. Orwant, et al. (2011). "Quantitative analysis of culture using millions of digitized books". In: *science* 331.6014, pp. 176–182.

127. Minematsu, N. (2004). "Pronunciation assessment based upon the phonological distortions observed in language learners' utterances". In: *INTERSPEECH*.

128. Minka, T. P. (2013). "Expectation propagation for approximate Bayesian inference". In: *arXiv preprint arXiv:1301.2294*.

129. Modeltalker (n.d.). *www.modeltalker.com*.

130. Moon, T. K. (1996). "The expectation-maximization algorithm". In: *IEEE Signal processing magazine* 13.6, pp. 47–60. DOI: 10.1109/79.543975.

131. Mu, Z., X. Yang, and Y. Dong (2021). "Review of end-to-end speech synthesis technology based on deep learning". In: *arXiv preprint arXiv:2104.09995*. DOI: 10.48550/ARXIV.2104.09995.

132. Murphy, K. P. (2012). *Machine learning: a probabilistic perspective*. MIT press.

133. Narendra, N. P. and P. Alku (2018). "Dysarthric Speech Classification Using Glottal Features Computed from Non-words, Words and Sentences". In: *Interspeech 2018, 19th Annual Conference of the International Speech Communication Association, Hyderabad, India, 2-6 September 2018*. Ed. by B. Yegnanarayana. ISCA, pp. 3403–3407. DOI: 10.21437/Interspeech.2018-1059.

134. Needleman, S. B. and C. D. Wunsch (1970). "A general method applicable to the search for similarities in the amino acid sequence of two proteins". In: *Journal of molecular biology* 48.3, pp. 443–453.

135. Neri, A., O. Mich, M. Gerosa, and D. Giuliani (2008). "The effectiveness of computer assisted pronunciation training for foreign language learning by children". In: *Computer Assisted Language Learning* 21.5, pp. 393–408.

136. Nicolao, M., A. V. Beeston, and T. Hain (2015). "Automatic assessment of English learner pronunciation using discriminative classifiers". In: *2015 IEEE Intl. Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, pp. 5351–5355.

137. Nicolao, M., H. Christensen, S. Cunningham, P. Green, and T. Hain (2016). "A framework for collecting realistic recordings of dysarthric speech - The home-Service corpus". In: *Proceedings of the 10th International Conference on Language Resources and Evaluation, LREC 2016*. ISBN: 9782951740891.

138. Oneata, D. and H. Cucu (2022). "Improving Multimodal Speech Recognition by Data Augmentation and Speech Representations". In: *arXiv preprint arXiv:2204.13206*.

139. Oord, A., Y. Li, I. Babuschkin, K. Simonyan, O. Vinyals, K. Kavukcuoglu, G. Driessche, E. Lockhart, L. Cobo, F. Stimberg, et al. (2018). "Parallel wavenet: Fast high-fidelity speech synthesis". In: *International conference on machine learning*. PMLR, pp. 3918–3926.

140. Ore, Ø. (2017). *Cardano: The gambling scholar*. Vol. 5063. Princeton University Press.

141. Paleyes, A., M. Pullin, M. Mahsereci, N. Lawrence, and J. Gonzalez (2019). "Emulation of physical processes with emukit". In: *Second Workshop on Machine Learning and the Physical Sciences, NeurIPS*.

142. Panayotov, V., G. Chen, D. Povey, and S. Khudanpur (2015). "Librispeech: an asr corpus based on public domain audio books". In: *2015 IEEE Intl. Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, pp. 5206–5210.

143. Patton, B., Y. Agiomyrgiannakis, M. Terry, K. Wilson, R. A. Saurous, and D. Sculley (2016). "AutoMOS: Learning a non-intrusive assessor of naturalness-of-speech". In: *arXiv preprint arXiv:1611.09207*.

144. Peng, L., K. Fu, B. Lin, D. Ke, and J. Zhan (2021). "A Study on Fine-Tuning wav2vec2.0 Model for the Task of Mispronunciation Detection and Diagnosis". In: *Proc. Interspeech 2021*, pp. 4448–4452. DOI: `10.21437/Interspeech.2021-1344`.

145. Piotrowska, M., A. Czyżewski, T. Ciszewski, G. Korvel, A. Kurowski, and B. Kostek (2021). "Evaluation of aspiration problems in L2 English pronunciation employing machine learning". In: *The Journal of the Acoustical Society of America* 150.1, pp. 120–132.

146. Plantinga, P. and E. Fosler-Lussier (2019). "Towards Real-Time Mispronunciation Detection in Kids' Speech". In: *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, pp. 690–696.

147. Porzuczek, A. and A. Rojczyk (2017). "English word stress in Polish learners speech production and metacompetence". In: *Research in Language* 15.4, pp. 313–323.

148. Posner, M. I. and S. E. Petersen (1990). "The attention system of the human brain". In: *Annual review of neuroscience* 13.1, pp. 25–42.

149. Qian, X., H. Meng, and F. Soong (2010). "Capturing L2 segmental mispronunciations with joint-sequence models in computer-aided pronunciation training (CAPT)". In: *2010 7th Intl. Symposium on Chinese Spoken Language Processing*. IEEE, pp. 84–88.

150. Rabiner, L. and R. Schafer (1978). *Digital Processing of Speech Signals*. Englewood Cliffs: Prentice Hall.

151. Radzikowski, K., L. Wang, and O. Yoshie (2016). "Non-native English speakers' speech correction, based on domain focused document". In: *Proceedings of the 18th International Conference on Information Integration and Web-based Applications and Services*, pp. 276–281.

152. Ramanathi, M. K., C. Yarra, and P. K. Ghosh (2019). "ASR Inspired Syllable Stress Detection for Pronunciation Evaluation Without Using a Supervised Classifier and Syllable Level Features." In: *INTERSPEECH*, pp. 924–928.

153. Ren, Y., Y. Ruan, X. Tan, T. Qin, S. Zhao, Z. Zhao, and T.-Y. Liu (2019). "Fastspeech: Fast, robust and controllable text to speech". In: *arXiv preprint arXiv:1905.09263*.

154. Romana, A., J. Bandon, M. Perez, S. Gutierrez, R. Richter, A. Roberts, and E. M. Provost (2021). "Automatically Detecting Errors and Disfluencies in Read Speech to Predict Cognitive Impairment in People with Parkinson's Disease". In: *22nd Annual Conference of the International Speech Communication Association, INTERSPEECH 2021*. International Speech Communication Association, pp. 156–160.

155. Rosenberg, A. and B. Ramabhadran (2017). "Bias and Statistical Significance in Evaluating Speech Synthesis with Mean Opinion Scores." In: *Interspeech*, pp. 3976–3980.

156. Rosenblatt, F. (1960). "Perceptron simulation experiments". In: *Proceedings of the IRE* 48.3, pp. 301–309.

157. Ruan, Y., X. Wang, H. Liu, Z. Ou, Y. Gao, J. Cheng, and Y. Qian (2019). "An End-to-end Approach for Lexical Stress Detection based on Transformer". In: *arXiv preprint arXiv:1911.04862*.

158. Rudzicz, F., A. K. Namasivayam, and T. Wolff (2012). "The TORGO database of acoustic and articulatory speech from speakers with dysarthria". In: *Language Resources and Evaluation*. ISSN: 1574020X. DOI: 10.1007/s10579-011-9145-0. arXiv: 0507464v2 [arXiv:astro-ph].

159. Särkkä, S. (2013). *Bayesian filtering and smoothing*. 3. Cambridge University Press.

160. Sarria-Paja, M. and T. Falk (2012). "Automated Dysarthria Severity Classification for Improved Objective Intelligibility Assessment of Spastic Dysarthric Speech." In: *Interspeech*. ISBN: 9781622767595.

161. Series, B. (2014). "Method for the subjective assessment of intermediate quality level of audio systems". In: *International Telecommunication Union Radiocommunication Assembly*.

162. Shah, R., K. Pokora, A. Ezzerg, V. Klimkov, G. Huybrechts, B. Putrycz, D. Korzekwa, and T. Merritt (2021). "Non-Autoregressive TTS with Explicit Duration Modelling for Low-Resource Highly Expressive Speech". In: *Proc. 11th ISCA Speech Synthesis Workshop (SSW 11)*, pp. 96–101. DOI: 10.21437/SSW.2021-17.

163. Shahin, M. A., J. Epps, and B. Ahmed (2016). "Automatic Classification of Lexical Stress in English and Arabic Languages Using Deep Learning." In: *INTERSPEECH*, pp. 175–179.

164. Shattuck-Hufnagel, S., M. Ostendorf, and K. Ross (1994). "Stress shift and early pitch accent placement in lexical items in American English". In: *Journal of Phonetics* 22.4, pp. 357–388.

165. Skerry-Ryan, R. J., E. Battenberg, Y. Xiao, Y. Wang, D. Stanton, J. Shor, R. J. Weiss, R. Clark, and R. A. Saurous (2018). "Towards End-to-End Prosody Transfer for Expressive Speech Synthesis with Tacotron". In: *CoRR* abs/1803.0. arXiv: `1803.09047`.

166. Skerry-Ryan, R., E. Battenberg, Y. Xiao, Y. Wang, D. Stanton, J. Shor, R. Weiss, R. Clark, and R. A. Saurous (2018). "Towards end-to-end prosody transfer for expressive speech synthesis with tacotron". In: *international conference on machine learning*. PMLR, pp. 4693–4702.

167. Sofaer, H. R., J. A. Hoeting, and C. S. Jarnevich (2019). "The area under the precision-recall curve as a performance metric for rare binary events". In: *Methods in Ecology and Evolution* 10.4, pp. 565–577.

168. Srivastava, N., G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov (2014). "Dropout: A Simple Way to Prevent Neural Networks from Overfitting". In: *Journal of Machine Learning Research* 15, pp. 1929–1958.

169. Statista (2021). *Most common languages used on the internet as of January 2020, by share of internet users*. URL: `https://www.statista.com/statistics/262946/share-of-the-most-common-languages-on-the-internet/`.

170. Sudhakara, S., M. K. Ramanathi, C. Yarra, A. Das, and P. Ghosh (2019). "Noise robust goodness of pronunciation measures using teacher's utterance". In: *SLaTE*.

171. Sudhakara, S., M. K. Ramanathi, C. Yarra, and P. K. Ghosh (2019). "An Improved Goodness of Pronunciation (GoP) Measure for Pronunciation Evaluation with DNN-HMM System Considering HMM Transition Probabilities." In: *INTERSPEECH*, pp. 954–958.

172. Sutskever, I., O. Vinyals, and Q. V. Le (2014). "Sequence to sequence learning with neural networks". In: *Advances in neural information processing systems*, pp. 3104–3112.

173. Tejedor-Garcıa, C., D. Escudero, E. Cámara-Arenas, C. González-Ferreras, and V. Cardeñoso-Payo (2020). "Assessing pronunciation improvement in students of English using a controlled computer-assisted pronunciation tool". In: *IEEE Transactions on Learning Technologies*.

174. Todhunter, I. (2014). *A history of the mathematical theory of probability*. Cambridge University Press.

175. Trujillo, F. (2006). "The production of speech sounds". In: *English Phonetics and Phonology*. URL: `https://www.ugr.es/~ftsaez/fonetica/production_speech.pdf`.

176. Tu, M., V. Berisha, and J. Liss (2017). "Interpretable Objective Assessment of Dysarthric Speech Based on Deep Neural Networks". In: *Interspeech 2017, 18th Annual Conference of the International Speech Communication Association, Stockholm, Sweden, August 20-24, 2017*. Ed. by F. Lacerda. ISCA, pp. 1849–1853. DOI: 10.21437/Interspeech.2017.

177. UNESCO (2016). *If you don't understand, how can you learn?* URL: https://en.unesco.org/news/40-don-t-access-education-language-they-understand.

178. *University physics Volume 1* (2016). eng. ISBN: 1-938168-27-5.

179. Valizada, A., S. Jafarova, E. Sultanov, and S. Rustamov (2021). "Development and Evaluation of Speech Synthesis System Based on Deep Learning Models". In: *Symmetry* 13.5, p. 819.

180. Van Den Oord, A., O. Vinyals, et al. (2017). "Neural discrete representation learning". In: *Advances in Neural Information Processing Systems* 30, pp. 6306–6315.

181. Vásquez-Correa, J. C., T. Arias-Vergara, J. R. Orozco-Arroyave, and E. Nöth (2018). "A Multitask Learning Approach to Assess the Dysarthria Severity in Patients with Parkinson's Disease". In: *Interspeech 2018, 19th Annual Conference of the International Speech Communication Association, Hyderabad, India, 2-6 September 2018*. Ed. by B. Yegnanarayana. ISCA, pp. 456–460. DOI: 10.21437/Interspeech.2018-1988.

182. Vaswani, A., N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin (2017). "Attention Is All You Need". In: *CoRR* abs/1706.0. arXiv: 1706.03762.

183. Vaswani, A., N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin (2017). "Attention is all you need". In: *Advances in neural information processing systems*, pp. 5998–6008.

184. Wagner, P., J. Beskow, S. Betz, J. Edlund, J. Gustafson, G. Eje Henter, S. Le Maguer, Z. Malisz, É. Székely, C. Tånnander, et al. (2019). "Speech synthesis evaluation—state-of-the-art assessment and suggestion for a novel research program". In: *Proceedings of the 10th Speech Synthesis Workshop (SSW10)*.

185. Wang, J., Y. Qin, Z. Peng, and T. Lee (2019). "Child Speech Disorder Detection with Siamese Recurrent Network Using Speech Attribute Features." In: *INTERSPEECH*, pp. 3885–3889.

186. Wang, Y., R. J. Skerry-Ryan, D. Stanton, Y. Wu, R. J. Weiss, N. Jaitly, Z. Yang, Y. Xiao, Z. Chen, S. Bengio, Q. V. Le, Y. Agiomyrgiannakis, R. Clark, and R. A. Saurous (2017). "Tacotron: {A} Fully End-to-End Text-To-Speech Synthesis Model". In: *CoRR* abs/1703.1. arXiv: 1703.10135.

187. Wang, Y., R. Skerry-Ryan, D. Stanton, Y. Wu, R. J. Weiss, N. Jaitly, Z. Yang, Y. Xiao, Z. Chen, S. Bengio, et al. (2017). "Tacotron: Towards end-to-end speech synthesis". In: *arXiv preprint arXiv:1703.10135*.

188. Weber, D., S. Zaporowski, and D. Korzekwa (2020). "Constructing a Dataset of Speech Recordings with Lombard Effect". In: *24th IEEE SPA*. DOI: 10.23919/SPA50552.2020.9241266.

189. Welch, L. R. (2003). "Hidden Markov models and the Baum-Welch algorithm". In: *IEEE Information Theory Society Newsletter* 53.4, pp. 10–13. URL: http://yanfenglu.net/documents/Baum-Welch_Algorithm.pdf.

190. Williams, C. K. and C. E. Rasmussen (2006). *Gaussian processes for machine learning*. Vol. 2. 3. MIT press Cambridge, MA. DOI: 10.7551/mitpress/3206.001.0001.

191. Witt, S. M. and S. J. Young (2000). "Phone-level pronunciation scoring and assessment for interactive language learning". In: *Speech communication* 30.2-3, pp. 95–108.

192. Wong, S. C., A. Gatt, V. Stamatescu, and M. D. McDonnell (2016). "Understanding data augmentation for classification: when to warp?" In: *2016 international conference on digital image computing: techniques and applications (DICTA)*. IEEE, pp. 1–6.

193. Woolson, R. (2007). "Wilcoxon signed-rank test". In: *Wiley encyclopedia of clinical trials*, pp. 1–3.

194. WorldEconomicForum (2018). *Speaking more than one language can boost economic growth*. URL: https://www.weforum.org/agenda/2018/02/speaking-more-languages-boost-economic-growth.

195. Xiao, Y., F. K. Soong, and W. Hu (2018). "Paired phone-posteriors approach to esl pronunciation quality assessment". In: *bdl*. Vol. 1. 782d, p. 3.

196. Xu, X., Y. Kang, S. Cao, B. Lin, and L. Ma (2021). "Explore wav2vec 2.0 for Mispronunciation Detection". In: *Proc. Interspeech 2021*, pp. 4428–4432. DOI: 10.21437/Interspeech.2021-777.

197. Yamagishi, J., C. Veaux, S. King, and S. Renals (2012). "Speech synthesis technologies for individuals with vocal disabilities: Voice banking and reconstruction". In: *Acoustical Science and Technology* 33.1, pp. 1–5. DOI: 10.1250/ast.33.1.

198. Yan, B.-C. and B. Chen (2021). "End-to-End Mispronunciation Detection and Diagnosis From Raw Waveforms". In: *arXiv preprint arXiv:2103.03023*.

199. Yan, B.-C., S.-W. F. Jiang, F.-A. Chao, and B. Chen (2021). "Maximum F1-score training for end-to-end mispronunciation detection and diagnosis of L2 English speech". In: *arXiv preprint arXiv:2108.13816*.

200. Yan, B.-C., M.-C. Wu, H.-T. Hung, and B. Chen (2020). "An End-to-End Mispronunciation Detection System for L2 English Speech Leveraging Novel Anti-Phone Modeling". In: *Proc. Interspeech 2020*, pp. 3032–3036. DOI: 10.21437/Interspeech.2020-1616.

201. Zen, H., V. Dang, R. Clark, Y. Zhang, R. J. Weiss, Y. Jia, Z. Chen, and Y. Wu (2019). "LibriTTS: A Corpus Derived from LibriSpeech for Text-to-Speech". In: *Proc. Interspeech 2019*, pp. 1526–1530. DOI: 10.21437/Interspeech.2019-2441.

202. Zhang, D., A. Ganesan, S. Campbell, and D. Korzekwa (2022). "L2-GEN: A Neural Phoneme Paraphrasing Approach to L2 Speech Synthesis for Mispronunciation Diagnosis". In: *accepted to Interspeech 2022*.

203. Zhang, Y.-J., S. Pan, L. He, and Z.-H. Ling (2018). "Learning latent representations for style control and transfer in end-to-end speech synthesis". In: *CoRR* abs/1812.0. arXiv: 1812.04342.

204. Zhang, L., Z. Zhao, C. Ma, L. Shan, H. Sun, L. Jiang, S. Deng, and C. Gao (2020). "End-to-end automatic pronunciation error detection based on improved hybrid ctc/attention architecture". In: *Sensors* 20.7, p. 1809.

205. Zhang, Z., Y. Wang, and J. Yang (2021). "Text-conditioned Transformer for automatic pronunciation error detection". In: *Speech Communication* 130, pp. 55–63.

206. Zhao, G., S. Sonsaat, A. O. Silpachai, I. Lucic, E. Chukharev-Hudilainen, J. Levis, and R. Gutierrez-Osuna (2018). "L2-ARCTIC: A non-native English speech corpus". In: *Perception Sensing Instrumentation Lab*.

207. Zhao, J., H. Yuan, J. Liu, and S. Xia (2011). "Automatic lexical stress detection using acoustic features for computer assisted language learning". In: *Proc. APSIPA ASC*, pp. 247–251.