CAN UNPAIRED TEXTUAL DATA REPLACE SYNTHETIC SPEECH IN ASR MODEL ADAPTATION?

Pasquale D'Alterio, Christian Hensel, Bashar Awwad Shiekh Hasan

Amazon Alexa

ABSTRACT

To boost training and adaptation of end to end (E2E) automatic speech recognition (ASR) models, several approaches to use paired speech-text input together with unpaired text input have emerged. They aim at improving the model performance on rare words, personalisation, and long tail. In this work, we present a systematic study of the impact of such training/adaptation and compare it to training with synthetic utterances generated by text-to-speech engines. We experiment with in-house and CommonVoice datasets and conclude that using text data for adaptation is effective, but is outperformed by adapting with synthetic audio even when the TTS engine is sub-optimal. This challenges recent literature on the difficulties of using TTS data including catastrophic forgetting, feature misalignment, and pronunciation errors, which motivated the use of text-only adaptation.

Index Terms— speech recognition, unpaired text training, synthetic speech

1. INTRODUCTION

The dominance of E2E ASR models is mainly driven by their ease of productionisation and superior performance [1, 2, 3, 4]. However, they are much harder to adapt to out of vocabulary (OOV) words, long tail, and personalisation [5, 6]. Model adaptation/fine tuning [7], Shallow fusion [6], External Language Model (LM) rescoring [8], contextual biasing [9], and adapting with Text-To-Speech (TTS) data [10] are amongst the most common approaches to address said issues.

Recently, several approaches were used to incorporate text data that is not paired with speech utterances during the training/fine-tuning of the E2E models: Textograms were proposed in [11] as a representation for text data which mimics time varying speech utterances. Textograms are used to train the model from scratch and to adapt to new samples without changing the standard RNN-T loss. This approach was demonstrated to be able to adapt the model's internal component with text-only data successfully. JOIST [12] targeted training large volumes of text (in billions of samples) jointly with speech data in streaming models resulting in 4-14% relative improvement on rare words. In order to support personalisation on edge devices, [13] used random audio encoder features to pair with textual input in order to fine tune an E2E hybrid autoregressive transducer (HAT) [4] model. Maestro was proposed in [14] to learn unified representations from both text and speech modalities simultaneously. This requires sequence alignment, duration prediction and matching embeddings in the learned space through an aligned masked-language model loss.

One of the main advantages of directly using text in model training is to eliminate the need for pairing text with synthetic speech. This can be computationally expensive and limited by the availability and quality of the TTS model [11, 12]. However, most prior work did not compare directly text-based approach with TTS based adaptation. The one exception is in [6] where the authors compared their approach to one in which the model is pretrained with TTS data instead of text. The results show that TTS lead to performance degredation compared to the author's text-based approach, with the caveat that the used TTS engine was trained on Librispeech data only.

The focus of this work is to present important data points on the compromise resulting from moving from paired speech and text to unpaired text only (i.e. text with no paired audio) adaptation. We will demonstrate that including a speech component is superior to text adaption even when using a lower quality (concatenative) TTS engine.

Although, at first glance it might seem intuitive that adding paired synthetic speech should improve the performance compared to unpaired text, there has been a consistent literature demonstrating the drawbacks of using TTS: I) catastrophic forgetting [15], II) Significant disparity between TTS and real audio features making it hard for the ASR model to generalize [16] III) TTS models tend to have pronunciation issues especially with entity names and therefore can mislead the ASR model. As unpaired text mitigates some of these shortcomings, it is not trivial to assume that TTS data would outperform text without running an experimental analysis. A gap that is addressed here.

2. METHODS

In this work we use the textogram approach [11] as an example of text-based model training/adaptation to compare against. This is mainly driven by the fact that it works well with low volumes of text data, making the analysis easier to

979-8-3503-0689-7/23/$31.00 ©2023 IEEE

conduct and interpret.

In [11], text is used directly as an input modality to the encoder of the neural network transducer. In contrast to related language model (LM) adaptation techniques [17], this allows to keep the training of the model the same as before without additional steps or changes to the loss function. In this setting, the text is encoded as *textograms* which are constructed by one-hot encoding the sub-words (e.g. graphemes).

During training, textogram features are concatenated with the audio features along the frame-length dimension. A disparity in length between the audio and text signals can be addressed by including a duration modeling strategy that repeats sub-word one-hot vectors for a number of frames. For more details on how to construct textograms please refer to [11].

Here, an example is either presented as audio-only to the model by setting the textogram features to zero or it's presented as a textogram and the audio-features are zeroed out. Put differently, the size of the input to the first layer in the encoder is increased so it is logically divided into two sections, one for audio features and one for textogram features. In order to make the learning task non-trivial, every sub-word can be masked with a certain probability (i.e. label masking).

3. EXPERIMENTS

For all the experiments that follow, we use a recurrent neural network transducer (RNN-T) [18] with an 8-layer long short-term memory unit (LSTM) [19] encoder, a 2-layer LSTM prediction network and a single-layer joint network with Softmax in the output layer. The size of each layer is 1024 units for the encoder and prediction network, 512 for the joint network and 4001 for the output layer, i.e. 4000 word-pieces plus the *blank* symbol. The word-piece model was trained on a large dataset of in-house voice assistant data with a unigram language model. The experiments have been designed to test the impact of unpaired textual and TTS-generated data in (OOV) word recognition after the model adaptation. To train the model on unpaired textual data, we encode the words on a grapheme-level with a repetition of 4 times per grapheme so that the length of textogram features is, on average, similar to the sequence length of audio features. Different duration modeling strategies have shown no significant impact [12]. We employ label masking with probability 25%.

We opted for one architecture for the ASR model to allow ourselves to conduct extensive the experiments. We chose one of the most (if not the most) popular architecture for real-world ASR. We also limited our experimentation to the the textogram representation proposed in [11], as the other variants introduced afterwards didn't seem to make any significant changes to justify further experimentation.

ID	Model	Core Test (WERR)
$B_{0\%}$	Baseline (No Text)	0.0
$B_{10\%}$	With 10% textograms	-0.53
$B_{50\%}$	With 50% textograms	+4.8

Table 1. Relative change of word error rate (WERR) for the training from scratch setup

3.1. Textual data in from-scratch training

When training the model from scratch, we designed three experiments to examine whether the use of textograms at this stage would provide some performance gains on general domains thanks to a regularization effect as observed in [11].

The models were trained using an in-house de-identified dataset made of 120k hours of German voice assistant speech (core train). The dataset contains both human and machine transcribed data. Each of the models was trained for 650k steps on 48 GPUs with a total batch size of 3072 samples/step. The models use Adam optimizer [20] with a starting value of 1e-7, a hold value of 5e-4 (between 150k, 500k steps) and a terminal value after decay of 1e-5.

The baseline model was trained only on core train data, two other variations were also trained with respectively a 10% and 50% probability to use a textogram presentation of the utterances. The models were tested on in-house German voice assistant data coming from general traffic (core test). The results in Table 1 report the relative word error rate (WERR) with respect to the model with no textogram (i.e. $B_{0\%}$).

3.2. Textual data and model adaptation

In these experiments, we explore the use of textual data in model adaptation when in-domain real audio is not available while textual data is easy to obtain (e.g. generated manually, through LM models or by web crawlers). All the models below were fine tuned starting from $B_{10\%}$ reported in Table 1. This model was chosen over $B_{50\%}$ because of the better performance on core test data. All the datasets used in the adaptation come from specific domains and are not included in the core training data used to train the models from scratch.

3.2.1. New skill support

This experiment covers the scenario in which a voice assistant needs to support a new skill and therefore recognises some new OOV words and commands. Before the skill is launched, real speech is rarely available. However, textual data covering the required commands is generally easy to generate. This data is made of 1k German in-house text-only entries designed to support the new skill, where each textual entry represents a plausible request from a customer involving such a skill. An additional 250 real-speech utterances are used for evaluation only.

The adapted models were trained for 25k steps using the same hyperparameters and GPU setup as described in Section 3.1 with the exception of the learning rate, that was fixed to the terminal value after the decay $(1e^{-5})$.

Six different experiments were designed and evaluated on the dedicated real-speech test set of the new skill as well as core test data (Table 2). We added an experiment (S2) in which we fine-tune the $B_{10\%}$ with text, similar to S1, but with the freezing of the encoder params in $B_{10\%}$. Notoriously, fine-tuned models can suffer from catastrophic forgetting. Hence, experiments S3-S6 use data replay, in which a set percentage (r) of each training batch is sampled from the core training data. The remaining 100-r% of the data is sampled from the new domain. Here, we use $r = 90\%$. Although this technique has the disadvantage of requiring the from-scratch training data to be available at adaptation, it was shown to be very effective at countering side effects of model fine-tuning [7].

As one of the main goals of the paper is to investigate how text-based adaptation compares to one that uses synthetic audio, 1k synthetic utterances were generated through an in-house German TTS engine (as in [21]). Those were created by feeding the text-only examples into the TTS engine with a randomly sampled voice profile from a pool of 250 voices.

3.2.2. FIFA world cup adaptation

The second scenario that we examined is one that involves a larger number of rare entities names. This highlights how TTS adaptation compares to text-based adaptation when the data contains entity names that might be hard to pronounce correctly for TTS engines.

This specific dataset is made of 50k German in-house text-only entries containing FIFA world cup 2022 player names. Possible examples of entries in the dataset are "*in what team does Harry Kane play?*" or *"who is Harry Kane?"*. We use a similar setup for model adaptation and audio synthesis to Sec 3.2.1. The models were evaluated using a real speech test set from the same domain with the results in Table 3.

3.2.3. Music misrecognitions

In our third series of experiments, we consider the use case of adapting a model towards known challenging entity names for the trained model. We want to quantify the performance of fine-tuning on text and TTS data compared to adapting on real speech data. To this end, we construct a music misrecognition dataset (MM) by decoding a set of utterances from the music domain using $B_{10\%}$ and taking all utterances that have misrecognised entity names (such as artist or album name). The misrecognitions could be due to rare entity names, difficult pronunciations and/or challenging acoustics. We partition MM into train and evaluation sets containing 10789 and 5075 utterances, respectively. [1]

To account for the larger volume and variance of the adaptation data, we use $r = 60\%$ data replay and run the adaptation for 30k steps with constant learning rate 1e-5 on 24 GPUs with an effective batch size of 1536 (Table 4). To estimate a upper limit of the achievable improvement by adaptation, we use the train partition of MM with real speech in one of the experiments (M1). Finally, we study the performance of combining text and TTS in M4.

3.2.4. Common Voice

The last set of experiments is based on the open-source dataset Common Voice (CV) [2] 9.0. This dataset is made of roughly 1k hours of German speech created by 16k speakers. This is very different from the in-house datasets used so far in many aspects, including acoustics, utterance length and style. The goal is to see how much text-only adaptation can improve the model performance when the domain to adapt to is very different, particularly in some aspects that cannot be covered by text alone (e.g. acoustics).

The data was split randomly into 3 partitions (80/10/10) and an equivalent synthetic version was reproduced for the train partition using, the same TTS settings above. We also use the same adaptation setup, with the exception of the number of steps (75k) and the data replay percentage, $r = 60\%$ to make up for the big volume of fine-tuning data. One model (C2) was adapted using real audio coming from the CV train partition to provide an upper bound to the achievable improvement. The results are reported in Table 5.

3.3. Quality of synthetic audio

We set up a set of experiments to quantify the role of the quality of the synthetic audio on adaptation compared to the text. We use two datasets: 1) the skill-support dataset as in Sec. 3.2.1 and 2) the names of stadiums that hosted the FIFA world cup 2022. These datasets were passed to both the in-house neural TTS used in all the previous experiments, as well as a concatenative TTS model, with "standard" voices), to create two comparable (i.e. based on the same text) TTS datasets.

Concatenative TTS engines are known to be inferior in terms of speech quality and realism when compared with more modern neural solutions [22]. The results for this set of experiments are reported in Table 6 and Table 7 for the stadium and skill case studies, respectively.

4. RESULTS

Adding unpaired text data to the from-scratch training setup, we observed only marginal gains in the model with 10%

[1] Note that, by construction, sentence error rate of $B_{10\%}$ on both splits is 100%.

[2] https://commonvoice.mozilla.org/

ID	Model description	Skill	Core test
S1	$B_{10\%}$ + text	-82.63	+392.14
S2	S1 + frozen encoder	-39.49	+1405.54
S3	S1 + data replay	-86.56	+0.36
S4	$B_{10\%}$ + TTS + data replay	-89.93	-0.36
S5	S4 + text (90% TTS; 10% text)	-90.21	-0.18
S6	S4 + text (50% TTS; 50% text)	-89.93	-0.18

Table 2. Adaptation results for the skill invocation data (WERR to $B_{10\%}$)

ID	Model description	FWC	Core test
F1	$B_{10\%}$ + text + replay	-27.29	-0.36
F2	$B_{10\%}$ + TTS + replay	-40.75	+0.18
F3	F2 + text (90% TTS; 10% text)	-41.03	-0.18
F4	F2 + text (50% TTS; 50% text)	-40.75	-0.54

Table 3. Adaptation results for the FIFA world cup 2022 (player names) data (WERR to $B_{10\%}$)

audio-to-text conversion rate and a significant degradation when the conversion rate was set to 50% (Section 3.1). These results differ from the ones reported in [11], where significant improvements were achieved. This benefit was mainly attributed by the authors to the regularization effect of feeding two different types of data to the encoder. We speculate that this discrepancy is explained by the larger data volumes used to train the model here, which makes the need for regularization less pronounced. They also differ from the results reported in [12], where the main difference with our approach is the use of *additional* textual data (i.e. data containing new information) instead of textual data generated from transcriptions of existing utterances.

The results on the domain adaptation confirm the usefulness of the text-only adaptation for all the domains. From the experiments on the skill-support data, however, it seems clear that fine-tuning only on the data from the new domain leads to catastrophic forgetting of the core (general) data. This is generally not acceptable in real-world production scenarios, where the goal is to improve on the new domain with minimal or no degradation on the core traffic data. Data replay consistently leads to being on-par with the baseline without sacrificing in-domain performance.

Contrary to what was observed in [11], freezing the encoder led to a lower WER reduction on the new domain and a high degradation on the general data. For these reasons, setups without data replay or with frozen encoder were dropped in all other experiments.

Table 2 shows that adaptation with synthesised audio (S4) slightly outperforms text adaptation (S1), while combining both (S6) performs best. The much larger and complex adaptation scenario with the FIFA world cup data (Table 3) presents the same picture but shows a much bigger gap between the text adaptation (F1) and TTS (F2) with respectively

ID	Model description	MM	Core test
M1	$B_{10\%}$ + real speech + replay	-39.56	+5.0
M2	$B_{10\%}$ + text + replay	-14.76	-0.89
M3	$B_{10\%}$ + TTS + replay	-18.39	+0.18
M4	M3 + text (50% TTS; 50% text)	-19.94	0.0

Table 4. Adaptation results for misrecognised music entities (WERR to $B_{10\%}$).

ID	Model description	CV	Core test
C1	$B_{10\%}$ + text + replay	-14.0	-0.36
C2	$B_{10\%}$ + real speech + replay	-55.34	+0.18
C3	$B_{10\%}$ + TTS + replay	-26.87	+0.18
C4	C3 + text (90% TTS; 10% text)	-28.05	-0.18
C5	C3 + text (50% TTS; 50% text)	-26.94	-0.54

Table 5. Adaptation results for the common voice dataset (WERR to $B_{10\%}$)

28% and 40% relative improvement. Also in this case, mixing TTS audio and text provides the best result (F3).

We make a similar observation for the misrecognised music entities dataset (Table 4). Again, text-based adaptation (M2) works well, but is outperformed by TTS (M3). Combining the two (M4) again works marginally better. Unsurprisingly, both text and TTS are significantly less effective than real speech.

Most adaptations sustain the performance on the core test set. However, in Table 4 and despite data replay M1 degrades model performance on the core test set by 5% relative. It is possible that a larger data volume for replay during the adaptation would have mitigated this degradation.

Surprisingly, for the CV dataset (Table 5), text-based adaptation (C1) is still able to provide significant gains over the baseline. We speculate that the main contribution of the textual data lies in the increase of the prior probabilities of the OOV words appearing in CV and in the adaptation of the model to the new style and utterance length. These gains, however, are overshadowed by those obtained by TTS adaptation (C3). In this case as well, the combination of text and audio (C4) provides better performance than either approach separately. Unsurprisingly, the best overall model is the one that used real in-domain audio for the adaption (C2). Such model was only included to provide an upper bound to the achievable improvement.

The comparison between neural TTS, concatenative (lower quality) TTS and unpaired text (Table 6, Table 7) also reveals some interesting information. In both cases, the best result is achieved when neural TTS is used (T2 and S4), as expected. However, the adaptation carried out using the audio generated through the concatenative engine is a very close second, distancing the text-based adaptation by a good margin, especially in the stadium name recognition scenario.

ID	Model description	Stad.	Core test
T1	$B_{10\%}$ + text + replay	-65.88	+0.16
T2	$B_{10\%}$ + neural TTS + replay	-86.68	-0.32
T3	$B_{10\%}$ + conc. TTS + replay	-86.38	+0.32

Table 6. Comparison of adaptation on the stadium dataset with data generated by neural TTS, concatenative TTS and unpaired text data (WERR to $B_{10\%}$)

ID	Model description	Skill	Core test
S3	$B_{10\%}$ + text + replay	-86.56	+0.36
S4	$B_{10\%}$ + neural TTS + replay	-89.93	-0.36
S7	$B_{10\%}$ + conc. TTS + replay	-89.65	-0.18

Table 7. Comparison of adaptation on the skill dataset with data generated by neural TTS, concatenative TTS and unpaired text data (WERR to $B_{10\%}$)

4.1. Utterance-level analysis

We look at differences in n-best lists generated by beam search on selected examples from the music misrecognitions dataset to study the effects of text and TTS adaptations on individual utterances.

Generally, we find that both adaptations are able to enable recognition of words that were OOV before. For example, the baseline model ($B_{10\%}$) does not recognise "*spiel skald lyfjaberg*" ("*play skald lyfjaberg*") as it fails to transcribe the song title correctly in any of the 8 hypotheses produced by beam search. Even though there is only one relevant utterance "*spiele lyfjaberg von wardruna*" ("*play lyfjaberg by wardruna*") in the adaptation set, both adaptations correct the misrecognition.

Next, we consider the following utterance "*spiele never going home von firelite*" ("*play never going home by firelite*"). $B_{10\%}$ does not recognise *firelite* in all of the hypotheses and favors the homophone *firelight*. The text-adapted model has the right transcription in the 2-best, yet still favors *firelight*. The model adapted with TTS produces the correct transcription as the best hypothesis. We find many utterances where text adaptation makes the right hypothesis more likely, but TTS adaptation influences the model more strongly and bubbles the right hypothesis during beam search. This is consistent with the observations in [12]. In fact, when considering oracle WERR, we see that text-based adaptation (-18.71 oracle WERR over $B_{10\%}$) slightly outperforms TTS (-17.08).

We find anecdotal evidence that text adaptation can help in boosting hypotheses for utterances that contain mispronunciations. For example we analyse an utterance "*füge despacito zu der band playlist hinzu*" ("*add despacito to the band playlist*") where the song title is mispronounced as *daspa'sito*. $B_{10\%}$ misrecognises *despacito* as *das pasito* and TTS adaptation (M3) using the right pronunciation does not change this. In contrast, text adaptation seems to boost the right hypothesis independent of acoustics and the adapted model corrects the mispronunciation in the input.

5. DISCUSSION AND CONCLUSION

The four adaptation experiments presented in this paper had the goal to compare synthetic audio and text based model adaptation in a set of scenarios with no real audio available and that cover different aspects of adaptation: simple OOV recognition, large and foreign catalogue, fixing production misrecognitions, adaptation to a domain with large differences that are hard to capture for synthetic audio and text (acoustics, style).

We aim in this work to contribute to knowledge in the literature on understanding the differences and trade offs between text-based adaptation and those based on synthetic speech, as well as providing initial data points on the importance of TTS quality.

The picture presented in our experiments is very consistent across all the domains analysed: text-based adaptation is an effective approach. However, it was overshadowed by the usage of TTS-generated speech in all the analysed scenarios, in most cases by a very large margin.

Furthermore, the FIFA world cup and music misrecognition experiments showed that even in domains containing words that are possibly hard to pronounce correctly by the TTS engine, synthetic audio is a better choice in terms of performance when used with data replay.

The combination of both synthetic audio and text did provide some additional value, achieving the lowest WER in all of the domains analysed. We attribute this beneficial effect to either a regularization or the ability of unpaired text to sometimes correct pronunciation mistakes made by the TTS engine or boost OOV words independently of acoustic evidence.

The use of synthetic audio, however, comes with some downsides, for instance: the synthesis can be costly, storing synthetic audio takes much more memory than storing just text, and building a TTS model for some low resource languages can be challenging.

Our work shows that some trade offs must be considered when deciding between text and synthetic audio usage in model adaptation, particularly between cost, convenience and improvement on the target domain. In cases in which the adaptation to perform is relatively simple, like in Sec. 3.2.1, the experimental results do not seem to justify the extra cost and effort of synthetic audio. Similarly, when the data size is extraordinarily large (e.g. billions of lines of text as in [12]) the cost and time of synthesis plus the storage required would make the use of TTS engines impractical.

For all the other cases, that from our experience are very common in real-world production models, our results support the idea that the usage of synthetic audio should be strongly considered to achieve significantly better performance, especially since commercial TTS solutions are easy to use, rela-

tively affordable and cover almost all languages (e.g. AWS Polly or Google Cloud Text-to-Speech[3]). This recommendation is further strengthened by analysis carried out on the performance difference between neural TTS, concatenative TTS and text-based adaptation, which suggests that the quality of the TTS engine is a secondary factor and concatenative TTS represent a cheaper yet powerful tool.

6. LIMITATIONS

The ability to generalize the results in this paper may be impacted by some limitations. First, the experiments only take into account a single language (German): it is possible that the results would differ when more phonetic languages (e.g. Italian, Turkish) are studied as the added value of having a dedicated TTS engine could decrease.

The work also only focuses on RNN-T models (similarly to [11]) and does not consider other popular architectures such as HAT [4]. Furthermore, other encoding strategies for the text (e.g. by using phonemes as in [12]) have not been explored, although from previous results we do not think this will have a major impact on the outcomes reported here.

[3]https://cloud.google.com/text-to-speech

7. REFERENCES

[1] Chung-Cheng Chiu, Tara N Sainath, Yonghui Wu, Rohit Prabhavalkar, Patrick Nguyen, Zhifeng Chen, Anjuli Kannan, Ron J Weiss, Kanishka Rao, Ekaterina Gonina, et al., "State-of-the-art speech recognition with sequence-to-sequence models," in *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2018, pp. 4774–4778.

[2] Suyoun Kim, Takaaki Hori, and Shinji Watanabe, "Joint ctc-attention based end-to-end speech recognition using multi-task learning," in *2017 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2017, pp. 4835–4839.

[3] Jinyu Li, Rui Zhao, Hu Hu, and Yifan Gong, "Improving rnn transducer modeling for end-to-end speech recognition," in *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2019, pp. 114–121.

[4] Ehsan Variani, David Rybach, Cyril Allauzen, and Michael Riley, "Hybrid autoregressive transducer (hat)," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 6139–6143.

[5] Tara Sainath, Yanzhang He, Arun Narayanan, Rami Botros, Ruoming Pang, David Rybach, Cyril Allauzen, Ehsan Variani, James Qin, Nam Lê Quoc, Shuo-Yiin Chang, Bo Li, Anmol Gulati, Jiahui Yu, Chung-Cheng Chiu, Diamantino Caseiro, Wei Li, Qiao Liang, and Pat Rondon, "An efficient streaming non-recurrent on-device end-to-end model with improvements to rare-word modeling," in *Interspeech*, 08 2021, pp. 1777–1781.

[6] Ding Zhao, Tara N Sainath, David Rybach, Pat Rondon, Deepti Bhatia, Bo Li, and Ruoming Pang, "Shallow-fusion end-to-end contextual biasing.," in *Interspeech*, 2019, pp. 1418–1422.

[7] Deepak Baby, Pasquale D'Alterio, and Valentin Mendelev, "Incremental learning for rnn-transducer based speech recognition models," in *Interspeech 2022*, 2022.

[8] Liyan Xu, Yile Gu, Jari Kolehmainen, Haidar Khan, Ankur Gandhe, Ariya Rastrow, Andreas Stolcke, and Ivan Bulyko, "Rescorebert: Discriminative speech recognition rescoring with bert," in *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2022, Virtual and Singapore, 23-27 May 2022*. 2022, pp. 6117–6121, IEEE.

[9] Mahaveer Jain, Gil Keren, Jay Mahadeokar, Geoffrey Zweig, Florian Metze, and Yatharth Saraf, "Contextual RNN-T For Open Domain ASR," in *Interspeech*, 2020.

[10] Xianrui Zheng, Yulan Liu, Deniz Gunceler, and Daniel Willett, "Using synthetic audio to improve the recognition of out-of-vocabulary words in end-to-end asr systems," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 5674–5678.

[11] Samuel Thomas, Brian Kingsbury, George Saon, and Hong-Kwang J Kuo, "Integrating text inputs for training and adapting rnn transducer asr models," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 8127–8131.

[12] Tara N Sainath, Rohit Prabhavalkar, Ankur Bapna, Yu Zhang, Zhouyuan Huo, Zhehuai Chen, Bo Li, Weiran Wang, and Trevor Strohman, "Joist: A joint speech and text streaming model for asr," in *2022 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2023, pp. 52–59.

[13] Adam Stooke, Khe Chai Sim, Mason Chua, Tsendsuren Munkhdalai, and Trevor Strohman, "Internal language model personalization of e2e automatic speech recognition using random encoder features," in *2022 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2023, pp. 213–220.

[14] Zhehuai Chen, Yu Zhang, Andrew Rosenberg, Bhuvana Ramabhadran, Pedro J. Moreno, Ankur Bapna, and Heiga Zen, "MAESTRO: matched speech text representations through modality matching," in *Interspeech 2022, 23rd Annual Conference of the International Speech Communication Association, Incheon, Korea, 18-22 September 2022*, Hanseok Ko and John H. L. Hansen, Eds. 2022, pp. 4093–4097, ISCA.

[15] Zhehuai Chen, Yu Zhang, Andrew Rosenberg, Bhuvana Ramabhadran, Pedro Moreno, and Gary Wang, "Tts4pretrain 2.0: Advancing the use of Text and Speech in ASR Pretraining with Consistency and Contrastive Losses," 2022.

[16] Ting-Yao Hu, Mohammadreza Armandpour, Ashish Shrivastava, Jen-Hao Rick Chang, Hema Koppula, and Oncel Tuzel, "SYNT++: Utilizing Imperfect Synthetic Data to Improve Speech Recognition," 2022.

[17] Janne Pylkkönen, Antti Ukkonen, Juho Kilpikoski, Samu Tamminen, and Hannes Heikinheimo, "Fast text-only domain adaptation of rnn-transducer prediction network," in *Interspeech 2021, 22nd Annual Conference of the International Speech Communication Association, Brno, Czechia, 30 August - 3 September*

2021, Hynek Hermansky, Honza Cernocký, Lukás Burget, Lori Lamel, Odette Scharenborg, and Petr Motlícek, Eds. 2021, pp. 1882–1886, ISCA.

[18] Alex Graves, "Sequence transduction with recurrent neural networks," 2012.

[19] Sepp Hochreiter and Jürgen Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[20] Diederik P Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[21] Iván Vallés-Pérez, Julian Roth, Grzegorz Beringer, Roberto Barra-Chicote, and Jasha Droppo, "Improving multi-speaker TTS prosody variance with a residual encoder and normalizing flows," in *Interspeech*, 2021.

[22] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu, "Wavenet: A generative model for raw audio," 2016.