Towards an ASR error robust Spoken Language Understanding System

Weitong Ruan, Yaroslav Nechaev, Luoxin Chen, Chengwei Su, Imre Kiss

Amazon Alexa, Cambridge, USA

{weiton,nechaey,luoxchen,chengwes,ikiss}@amazon.com

Abstract

A modern Spoken Language Understanding (SLU) system usually contains two sub-systems, Automatic Speech Recognition (ASR) and Natural Language Understanding (NLU), where ASR transforms voice signal to text form and NLU provides intent classification and slot filling from the text. In practice, such decoupled ASR/NLU design facilitates fast model iteration for both components. However, this makes downstream NLU susceptible to errors from the upstream ASR, causing significant performance degradation. Therefore, dealing with such errors is a major opportunity to improve overall SLU model performance. In this work, we first propose a general evaluation criterion that requires an ASR error robust model to perform well on both transcription and ASR hypothesis. Then robustness training techniques for both classification task and NER task are introduced. Experimental results on two datasets show that our proposed approaches improve model robustness to ASR errors for both tasks.

Index Terms: spoken language understanding, ASR error robustness, Marginalized CRF

1. Introduction

Personal assistants, such as Siri, Google Assistant, and Alexa, fulfill an incoming user requests through the use of a Spoken Language Understanding (SLU) system. A standard modern SLU system contains at least two major components ASR and NLU. During interaction with these assistants, human voice is first transcribed to either text form or lattice by ASR and then interpreted by NLU. During the interpretation stage at NLU, each utterance[1] is assigned an intent from an Intent Classifier (IC) model and slots from each utterance are labelled by an Named Entity Recognition (NER) model, also known as a semantic frame in [1, 2]. As an example, "what is the weather in Boston today" is mapped to a `get_weather` intent and slots of `city:Boston` and `date:today`.

In practice, both ASR and NLU are decoupled for fast model iteration and both are trained in a supervised manner with a dataset of voice signals, human-transcriptions and human-annotations based on transcriptions. In this dataset, voice signals and human-transcriptions are used to improve the ASR performance where voice signals serve as input and transcriptions as output. Similarly, transcriptions and annotations are used to boost the NLU accuracy.

This decoupled design allows ASR and NLU to scale at their own speed, however, the cascade design propagates and even magnifies upstream ASR error into downstream NLU. Moreover, NLU models built in this manner has never seen ASR errors in offline training and evaluation while when deployed in practice, it takes ASR outputs or ASR hypotheses as inputs which inevitably contains ASR errors and hence NLU model performance is suboptimal. A common ASR error is due to homophones, for example,"buy the first item" is often incorrectly recognized as "by the first item"; and "bye WW[2]" as "buy WW". For an NLU trained in the aforementioned manner, a change from `buy` to `by` can easily modify the intent classification result from `Shopping` to `Global`. More details on ASR errors and ASR error induced NLU errors are included in Sec. 2.

To increase the NLU robustness, a straightforward approach is to change the training data: instead of training on the ASR error free transcriptions, NLU models can be trained on annotated ASR hypotheses. However, this approach is not practical. First, the ASR model updates regularly and its error distribution drifts along with those updates, the effort spent on annotating the ASR hypotheses will, therefore, be useless. Second, annotating the ASR hypotheses which contains ASR errors is a more time-consuming effort than annotating the error-free transcriptions. Finally, the original semantics of user's input can change significantly due to an ASR error.

Prior work on solving this problem mostly relies on using ASR n-best hypotheses instead of 1-best and through reranking to recover from ASR errors [3, 4, 5, 6, 7, 8]. After the deep learning based approaches became mainstream [9, 10], a much more variety of approaches have been proposed. One direction is to design an end-to-end SLU system that interprets voice signal directly without the need of the intermediate error-prone step which converts voice signal to text [11, 12, 13, 14], however, this approach has not yet shown similar performance compared with the decoupled structure and the coupled structure makes the model not easy to scale. Other approaches build ASR-robust representations, for example, using a lattice [15] or a confusion matrix [16, 17, 18, 19].

In this paper, we first propose an evaluation guideline criterion to measure the model robustness towards ASR errors, then propose to change both classification and NER model training technique to account for robustness with respect to (w.r.t.) ASR errors. For the classification task, we are motivated by the idea of Virtual Adversarial Training (VAT), and use the ASR hypothesis as adversarial samples. The loss function includes a Kullback-Leibler (KL) divergence term that penalizes the difference between predictive distribution from transcription and from ASR hypothesis. For NER task, our proposed approach involves a two-step approach where the first step is to pseudo-label ASR hypotheses using annotated transcriptions and then use marginalized-CRF to train NER with pseudo-labelled ASR hypotheses.

2. ASR Error Problem

Deepak Kumar et al. [20] conducted skill squatting attacks on the ASR engine used by Amazon Alexa. They observed three common systematic errors: homophones, compound words, and phonetic confusions.

[1] In this paper, we only use ASR 1-best.

[2] Here WW refers to the wakeword.

For homophone errors, similar to the example used in Sec. 1, they noticed that Alexa's ASR often makes error by misrecognizing `sail` as `sale`, `calm` as `com`, `main` as `maine` and etc [20]. As for compound word errors, some compound words are split into constituent words. For example, `outdoors` is split into `out doors`. The most prominent cause of error is phonetic confusion where an error in the phonemes leads to a different word. For example, as noticed in [20], the phonetic spelling of word `coal` is `K OW L`, and the ASR often confuses `OW` with `AO`, leading to an incorrect word `call`.

In this work, we mitigate the above issue by making the NLU models robust to ASR errors. Before we move onto model details and training techniques, we propose a guideline on how to evaluate robustness towards ASR errors by formulating the following evaluation criterion:

> *Model robustness towards ASR errors is only improved given increased performance tested on a set with ASR errors and no performance degradation on a set without ASR errors.*

The idea of this guide is to emphasize model performance on transcriptions, since increased performance on ASR hypotheses and decreased performance on transcriptions leads to an overfitting on ASR hypotheses. In our results reported in Sec. 5, we report results on both ASR hypotheses and transcriptions.

The evaluation on ASR hypotheses poses another level of complexity. In most cases, especially in industry, only annotations on transcriptions are available. For classification task, one can always assume that transcription and ASR hypotheses can share the same label, however, for NER task, which targets token level slot-filling, this is problematic. For example, suppose we have an annotated transcription "what is the weather in amsterdam netherlands today", in which we have two slots `city:amsterdam netherlands` and `date:today` and a corresponding ASR hypothesis "what is the weather in amsterdam nether lands today", where the word "netherlands" is split into two words. An ideal NER model will be able to recognize the the slot `date:today`, however, for each other slot, it can, at its best, output `city:amsterdam nether lands`, which is not the same as ground truth and will be considered as an error in a lot of widely used evaluations metrics, for example, "slot-F1" [21]. Given this *unrecoverable* error issue when evaluating on the ASR hypotheses with ASR errors, in this work, we define NER model robustness as model performance only on matched entities (`date:today` in the previous example).

3. Models

In this section, we provide a detailed description of our NLU models. We employ the state-of-the-art CNN-LSTM-CRF [22] model, since it has proven to be effective across many NLP tasks. All models share the same encoder architecture with an appropriate decoder for classification and NER.

3.1. Encoder Layer

As illustrated in Fig. 1, our encoder layer mainly uses two Bi-LSTM layers of size 300 per direction to encode word level and character level information from both directions. The word level information is represented by a set of 300 dimensional word embeddings of size T, $\{\boldsymbol{e}_l\}_{l=1,\cdots,T}$, pre-trained with the skip-gram model using fastText [23]. Out Of Vocabulary (OOV) words are represented as an embedding formed from an average of 150 least frequent embeddings in the embedding space. To ensure the generalization capability of pre-trained word embeddings, word embeddings are fixed during training process. Also, as indicated by the investigation in [22], character level information captures morphological features, we therefore uses a set of C randomly initialized 32-dimension character embeddings, $\{\boldsymbol{c}_l\}_{l=1,\cdots,C}$ for each character together with a CNN layer of 32 trigram filters. Note character embeddings will be updated during training. Specifically, for the i-th token in an input utterance, its corresponding 300-dim word embedding and 32-dim CNN based character representation are concatenated as input to the Bi-LSTM layer

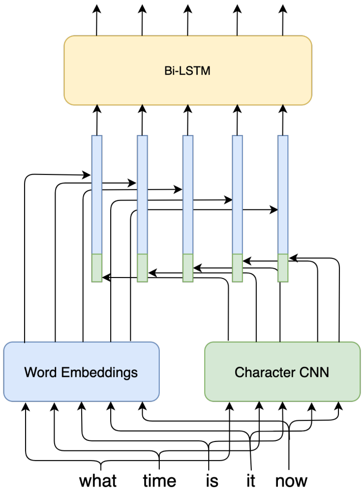

Figure 1: *Encoder architectures*

$$\boldsymbol{x}_i = [\boldsymbol{e}_i; \mathbf{CNN}(\boldsymbol{c}_i)], \tag{1}$$

and its contextual representation through 2 layers of Bi-LSTM is a also a concatenation of LSTM outputs from both directions

$$\boldsymbol{r}_i = \left[\overrightarrow{\boldsymbol{h}}_i; \overleftarrow{\boldsymbol{h}}_i\right],$$

where $\overrightarrow{\boldsymbol{h}}_i$ represents the 300-dim top left to right LSTM output for the i-th input token and $\overleftarrow{\boldsymbol{h}}_i$ from right to left.

3.2. Decoder Layer for Classification

For classification task, the concatenated Bi-LSTM outputs for each input token $\boldsymbol{r}_i, i = 1, \cdots, t$, where t denotes the length of the input, are first sent through a pooling layer which sums up all individual token representations as utterance representation

$$\boldsymbol{u} = \sum_{i=1}^{t} \boldsymbol{r}_i.$$

Note, we choose sum-pooling over average-pooling simply because it performed slightly better during our experiments. A fully connected layer of size 600 × 600 with Exponential Linear Unit (ELU) [24] is connected on top of the utterance representation before it is sent to the softmax layer.

3.3. Decoder Layer for NER

For NER task, instead of using a softmax at each output as in [25], the concatenated Bi-LSTM outputs are fed to a linear-clain Conditional Random Field (CRF) layer as in most state-of-the-art architectures [22, 26] for NER, to better modeling label transition probabilities and avoiding label bias problems [27].

4. Robustness Training

Robustness training techniques for both classification task and NER are discussed in details in this section.

4.1. Classification Task

The classic loss function used in any classification task is the Negative Log-Likelihood (NLL) loss, which is equivalent to the Cross Entropy (CE) loss defined as the cross entropy between the "empirical" distribution $p\left(\boldsymbol{y}_i|\boldsymbol{X}_i\right) = \mathbb{1}\left[y = y_l\right]$ and the "predictive" distribution $\hat{p}\left(\boldsymbol{y}_i|\boldsymbol{X}_i\right)$:

$$CE\left(\boldsymbol{y}_i;\boldsymbol{X}_i,\boldsymbol{\theta}\right) = -\sum_{l=1}^{L} p\left(y_l|\boldsymbol{X}_i\right)\log\hat{p}\left(y_l|\boldsymbol{X}_i\right),$$

where $\boldsymbol{\theta}$ is the set of parameters used in the model, L is the total number of labels and

$$\boldsymbol{X}_i = \left[\boldsymbol{x}_1^i,\cdots,\boldsymbol{x}_j^i,\cdots,\boldsymbol{x}_T^i\right],$$

is a column stack of input representation (in (1)) for each token in the i-th utterance.

Motivated by the VAT proposed in [28, 29], where the model robustness is achieved through the added KL loss penalizing the distance between the predictive distribution from original inputs and its noisy neighbors, we propose to add a KL term that measures the distance between the prediction distribution from transcriptions and ASR hypotheses, resulting in a new loss function:

$$\begin{aligned} Loss\left(\boldsymbol{y}_i;\boldsymbol{X}_i,\boldsymbol{A}_i,\boldsymbol{\theta}\right) = \quad & \epsilon_1 * CE\left(\boldsymbol{y}_i;\boldsymbol{X}_i,\boldsymbol{\theta}\right) \\ & + \epsilon_2 * CE\left(\boldsymbol{y}_i;\boldsymbol{A}_i,\boldsymbol{\theta}\right) \\ & + \epsilon_3 * KL\big(p\left(\boldsymbol{y}_i|\boldsymbol{A}_i\right), p\left(\boldsymbol{y}_i|\boldsymbol{X}_i\right)\big) \end{aligned}$$

where $\boldsymbol{A}_i$ denotes the corresponding ASR hypothesis of the i-th input utterance and $\{\epsilon_i\}_{i=1}^{3}$ are a group of weights that control the relative importance of each loss term.

In the above loss function, the first CE term is designed as the NLL loss for model performance on the transcription, the second term is equivalent as adding pseudo-labelled ASR hypotheses into training data and the third term aims to improve model robustness towards ASR hypotheses by forcing models to predict similar predictions on both transcription and ASR hypotheses. From the experimental results, we noticed the combination of the first and third loss achieves the best model performance.

4.2. NER Task

Our proposed approach to improve NER model robustness towards ASR errors relies on the idea of training data augmentation, which is a widely used technique to improve model performance [30]. Our approaches involves two steps: 1) pesudo-label utterances where ASR hypotheses are different from transcription; 2) add those pesudo-labelled ASR hypotheses in the training set with a modified decoder loss function. In this subsection, we first introduce our proposed Marginalized CRF model[3], then provide an example pseudo-labelling technique we can use.

4.2.1. *Marginalized CRF*

CFR is often used to model the input-output relationship within a sequence $\{(x_i, y_i)_{i=1}^{t}\}$. The traditional CRF requires a complete set of input and output pairs due to the fact that it uses the NLL loss [27], where the likelihood is defined as

$$p\left(\boldsymbol{y}|\boldsymbol{x}\right) = \frac{\exp\left(\boldsymbol{w}^T\boldsymbol{\Phi}\left(\boldsymbol{x},\boldsymbol{y}\right)\right)}{Z\left(\boldsymbol{x},\boldsymbol{Y}\right)}, \tag{2}$$

where

$$Z\left(\boldsymbol{x},\boldsymbol{S}\right) = \sum_{\boldsymbol{y}\in\boldsymbol{S}}\exp\left(\boldsymbol{w}^T\boldsymbol{\Phi}\left(\boldsymbol{x},\boldsymbol{y}\right)\right)$$

is a normalizing term and $\boldsymbol{\Phi}\left(\boldsymbol{x},\boldsymbol{y}\right)$ is a set of feature vectors.

However, in a lot of cases, the complete annotation is not available. Given the input, $\{x_i\}_{i=1}^{t}$, its corresponding output is either unknown or uncertain for some time step t. In our case with pseudo-labelling detailed in Sec. 4.2.2, some pseudo-labelled tokens in ASR hypotheses do not have corresponding labels. To train an NER model with such incomplete dataset, we propose to generalize traditional CRF by marginalizing the previously defined likelihood over unknown/uncertain labels, which avoids penalizing uncertain input output pairs. To be more specific, the revised likelihood is defined as:

$$p\left(\boldsymbol{Y}|\boldsymbol{x}\right) = \sum_{\boldsymbol{y}\in\boldsymbol{Y}} p\left(\boldsymbol{y}|\boldsymbol{x}\right),$$

where $\boldsymbol{Y}$ is a set of all valid output combinations.

4.2.2. *Pseudo-labelling for Marginalized CRF*

In general, the pseudo-labelling problem is the following: Given an annotated sequence $\{(x_i, y_i)_{i=1}^{t}\}$, we want to pseudo-label another non-annotated sequence $(\hat{x}_i)_{i=1}^{t'}$, where in our case, the annotated sequence is annotated transcription while the non-annotated sequence is its corresponding ASR hypotheses. Here we propose two type of exact match based pseudo-labelling rules: Token Exact Match (TEM) and Entity Exact Match (EEM). The TEM rule assigns pseudo-labels per token and for every token level exact match, i.e. $x_i = \hat{x}_l$ for some i and l, then $\hat{y}_l = y_i$. In EEM, the exact match is not on token level but on entity level. For example, given an annotated transcription:

$$\frac{\text{play}}{\text{O}}\,\frac{\text{welcome to new york}}{\text{Song}}\,\frac{\text{by}}{\text{O}}\,\frac{\text{taylor swift}}{\text{Artist}}.$$

The TEM based pseudo-labels on an ASR hypotheses "play welcome to new by taylor swift" is:

$$\frac{\text{play}}{\text{O}}\,\frac{\text{welcome to new}}{\text{Song}}\,\frac{\text{by}}{\text{O}}\,\frac{\text{taylor swift}}{\text{Artist}},$$

and the Entity Exact Match (EEM) gives

$$\frac{\text{play}}{\text{O}}\,\frac{\text{welcome to new}}{\text{NA}}\,\frac{\text{by}}{\text{O}}\,\frac{\text{taylor swift}}{\text{Artist}},$$

where "NA" is short for "Not Applicable", a new label assigned to handle tokens without a certain label. During the training process, utterances with such labels will be properly taken care of using the Marginalized CRF model detailed in Sec. 4.2.1. Also, it is obvious from the above comparison that the TEM focuses on recall and the EEM is more conservative and prioritizes on precision. In this paper, we use the EEM to create pseudo-labels for ASR hypotheses which are different from the transcription and add these pseudo-labelled ASR hypotheses to the training set to improve model robustness in cases where the utterance contains ASR errors.

[3] During the preparation of this manuscript, we noticed similar approach was also used in [31] and further back in [32] to handle similar situations

Table 1: *Performance (F1) comparison on ATIS dataset*

Task	Models	Trans.	ASR-3.5	ASR-4.0
Classification	Baseline	0.9713	0.9642	0.9615
	Train with ASR-3.5	0.9714	**0.9686**	N/A
	Train with ASR-4.0	0.9710	N/A	**0.9653**
NER	Baseline	0.9603	0.9521	0.9483
	Train with ASR-3.5	0.9598	**0.9537**	N/A
	Train with ASR-4.0	0.9597	N/A	**0.9501**

5. Experiments and Results

In this section, we present experimental results to demonstrate that our proposed training techniques are able to provide enhancement to model robustness w.r.t. ASR errors. Evaluation is performed on both widely used Airline Travel Information Systems (ATIS) dataset [33, 34, 35, 19] and a private dataset developed within Alexa. Both the classification task and NER task are evaluated separately in the following subsections.

5.1. Datasets

We evaluate our proposed approaches using both the ATIS dataset and the Alexa dataset collected within Alexa AI. The ATIS dataset is a popular benchmark for spoken language systems which contains audio recordings and the corresponding annotated transcripts. In the last years, textual version of this dataset containing 5871 utterances (4478, 500 and 893 utterances in train, validation and test set respectively) was used to benchmark NLU approaches. We align the original audio recordings to each of those utterances and use Amazon Transcribe,[4] a the publicly available speech-to-text service, to acquire the n-best ASR hypotheses for each audio file. Then, we have constructed two datasets by picking either the ASR with the best word error rate (WER) or the worst. This resulted in two datasets with 3.5% (`ASR-3.5`) and 4% (`ASR-4.0`) WER respectively. ATIS dataset is small and represents limited diversity in terms of the target domain and slot values. To prove that our proposed approach works at scale, we also utilized a much larger set, Alexa dataset, which contains a few millions of developer generated and annotated transcriptions and their corresponding ASR hypotheses. The ASR hypotheses in the Alexa dataset are decoded by the 2019 Alexa ASR model used to serve the live traffic.

5.2. Classification Task

The classification performance (F1) on ATIS dataset is tabulated in the top section in Table 1. The first observation is that baseline model performance degrades as the WER increases on ASR hypotheses, which highlights the importance of NLU model robustness towards ASR errors. The second observation is models trained with ASR hypotheses using our proposed loss performed significantly better (around 0.4% absolute) on their corresponding ASR hypotheses and at the same time maintained similar performance on the transcriptions. Note, some slots are ”N/A” because we expect to learn and evaluate on the set with similar ASR error patterns.

Table 2 reports an ablation study result on the effect of hyper-parameters used in the loss function. Note, all results are relative w.r.t. the baseline model performance on either transcription or ASR. The first row denotes the baseline model performance where the model is trained with transcriptions. The second row corresponds to model trained with both transcription and ASR hypotheses. The result verifies that data augmentation can improve model robustness. The third row represents model trained with only ASR hypotheses. Compared with the baseline result, models trained with only ASR hypotheses performs better on the ASR, while not as well on transcription, suggesting an overfit on ASR. Our proposed model utilizes only the first and third component in the loss and achieves the best performance on both transcription and ASR.

Table 2: *Relative classification performance (F1) w.r.t. baseline model performance on Alexa dataset (negative means performance degradation).*

model	parameters			Trans.(%)	ASR(%)
	ϵ_1	ϵ_2	ϵ_3		
baseline	1.0	0.0	0.0	0	0
data augmentation	1.0	1.0	0.0	0	1.76
train on ASR	0.0	1.0	0.0	-3.76	1.41
proposed model	1.0	0.0	40.0	**0.88**	**4.41**

Table 3: *Relative NER performance (slot-F1) w.r.t. baseline model on Alexa dataset (larger value means better model)*

Domain subset in Alexa dataset	Improvements on Trans. (%)	Improvements on ASR (%)
Music	-0.35	0.9
SmartHome	0.19	5.3
Notifications	0.38	10.72
Shopping	0	2.58

5.3. NER Task

For the NER task, evaluations are performed using the slot-F1 metric [21]. Note, when evaluating on ASRs, we only evaluate on slots where the entire slot exist in the ASR hypothesis, hence in some experiments, the F1 on ASRs appears to be higher than F1 on transcriptions. On the ATIS dataset, where results are tabulated in the bottom section in Table 1, similar improvements as the classification task are observed.

As for the Alexa dataset, since the label space is too large, we first partition the entire dataset into subsets according to their groundtruth domain labels and train an NER model for each domain subset. Domain-wise model performance are presented in Table 3. From all domain subsets, our proposed robust model achieves better performance on the ASRs while maintaining the performance on transcriptions.

6. Conclusions and Future Work

In this paper, we addressed the model robustness issue in a decoupled SLU system where upstream ASR errors become a bottleneck in the overall SLU model performance. We first defined an evaluation criterion for SLU model robustness towards ASR errors and then proposed approaches to improve the model robustness for both classification and NER tasks. We evaluated our proposed approaches on both the public ATIS dataset and our private dataset based on Alexa data. In all cases, we observed significant performance improvements compared to baseline models. In future, we plan to evaluate our proposed training technique on more advanced contextual representation, for example, BERT.

[4] `https://aws.amazon.com/transcribe`

7. References

[1] G. Tur and R. De Mori, *Spoken language understanding: Systems for extracting semantic information from speech*. John Wiley & Sons, 2011.

[2] Y.-Y. Wang, L. Deng, and A. Acero, "Semantic frame-based spoken language understanding," *Spoken Language Understanding: Systems for Extracting Semantic Information from Speech*, pp. 41–91, 2011.

[3] R. Jonson, "Dialogue context-based re-ranking of asr hypotheses," in *2006 IEEE Spoken Language Technology Workshop*, 2006, pp. 174–177.

[4] H. Sak, M. Saraclar, and T. Gungor, "Discriminative reranking of asr hypotheses with morpholexical and n-best-list features," in *2011 IEEE Workshop on Automatic Speech Recognition Understanding*, 2011, pp. 202–207.

[5] F. Peng, S. Roy, B. Shahshahani, and F. Beaufays, "Search results based n-best hypothesis rescoring with maximum entropy classification," in *2013 IEEE Workshop on Automatic Speech Recognition and Understanding*, 2013, pp. 422–427.

[6] F. Morbini, K. Audhkhasi, R. Artstein, M. V. Segbroeck, K. Sagae, P. Georgiou, D. R. Traum, and S. Narayanan, "A reranking approach for recognition and classification of speech input in conversational dialogue systems," in *2012 IEEE Spoken Language Technology Workshop (SLT)*, 2012, pp. 49–54.

[7] A. J. Kumar, C. Morales, M.-E. Vidal, C. Schmidt, and S. Auer, "Use of knowledge graph in rescoring the n-best list in automatic speech recognition." *arXiv preprint arXiv:1705.08018*, 2017.

[8] A. Ogawa, M. Delcroix, S. Karita, and T. Nakatani, "Improved deep duel model for rescoring n-best speech recognition list using backward lstmlm and ensemble encoders," in *Interspeech 2019*, 2019.

[9] K. Yao, B. Peng, Y. Zhang, D. Yu, G. Zweig, and Y. Shi, "Spoken language understanding using long short-term memory neural networks," in *2014 IEEE Spoken Language Technology Workshop (SLT)*, 2014, pp. 189–194.

[10] M. Henderson, B. Thomson, and S. Young, "Deep neural network approach for the dialog state tracking challenge," in *Proceedings of the SIGDIAL 2013 Conference*, 2013, pp. 467–471.

[11] L. Lugosch, M. Ravanelli, P. Ignoto, V. S. Tomar, and Y. Bengio, "Speech model pre-training for end-to-end spoken language understanding," in *Interspeech 2019*, 2019.

[12] D. Serdyuk, Y. Wang, C. Fuegen, A. Kumar, B. Liu, and Y. Bengio, "Towards end-to-end spoken language understanding," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 5754–5758.

[13] P. Haghani, A. Narayanan, M. Bacchiani, G. Chuang, N. Gaur, P. Moreno, R. Prabhavalkar, Z. Qu, and A. Waters, "From audio to semantics: Approaches to end-to-end spoken language understanding," in *2018 IEEE Spoken Language Technology Workshop (SLT)*, 2018, pp. 720–726.

[14] N. Tomashenko, A. Caubrire, and Y. Estve, "Investigating adaptation and transfer learning for end-to-end spoken language understanding from speech," in *Interspeech 2019*, 2019, pp. 824–828.

[15] F. Ladhak, A. Gandhe, M. Dreyer, L. Mathias, A. Rastrow, and B. Hoffmeister, "Latticernn: Recurrent neural networks over lattices." in *Interspeech 2016*, 2016, pp. 695–699.

[16] M. Henderson, B. Thomson, and S. Young, "Word-based dialog state tracking with recurrent neural networks," in *Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, 2014, pp. 292–299.

[17] J. Liu and X. Yang, "Using word confusion networks for slot filling in spoken language understanding," in *INTERSPEECH*, 2015, pp. 1353–1357.

[18] P. G. Shivakumar, M. Yang, and P. G. Georgiou, "Spoken language intent detection using confusion2vec," in *Interspeech 2019*, 2019.

[19] C.-W. Huang and Y.-N. Chen, "Learning asr-robust contextualized embeddings for spoken language understanding," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020.

[20] D. Kumar, R. Paccagnella, P. Murley, E. Hennenfent, J. Mason, A. Bates, and M. Bailey, "Skill squatting attacks on amazon alexa," in *27th {USENIX} Security Symposium ({USENIX} Security 18)*, 2018, pp. 33–47.

[21] T. K. Sang, F. Erik, and S. Buchholz, "Introduction to the conll-2000 shared task: chunking," in *Proceedings of CoNLL-2000, Lisbon, Portugal*, 2000, pp. 127–132.

[22] X. Ma and E. Hovy, "End-to-end sequence labeling via bi-directional lstm-cnns-crf," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2016, pp. 1064–1074.

[23] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," *arXiv preprint arXiv:1607.04606*, 2016.

[24] D.-A. Clevert, T. Unterthiner, and S. Hochreiter, "Fast and accurate deep network learning by exponential linear units (elus)," *arXiv preprint arXiv:1511.07289*, 2015.

[25] K. Clark, M.-T. Luong, C. D. Manning, and Q. Le, "Semi-supervised sequence modeling with cross-view training," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018, pp. 1914–1925.

[26] M. Peters, W. Ammar, C. Bhagavatula, and R. Power, "Semi-supervised sequence tagging with bidirectional language models," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2017, pp. 1756–1765.

[27] J. Lafferty, A. McCallum, and F. C. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," 2001.

[28] T. Miyato, S.-i. Maeda, M. Koyama, and S. Ishii, "Virtual adversarial training: a regularization method for supervised and semi-supervised learning," *IEEE transactions on pattern analysis and machine intelligence*, vol. 41, no. 8, pp. 1979–1993, 2018.

[29] T. Miyato, A. M. Dai, and I. Goodfellow, "Adversarial training methods for semi-supervised text classification," *arXiv preprint arXiv:1605.07725*, 2016.

[30] K. M. Yoo, Y. Shin, and S. goo Lee, "Data augmentation for spoken language understanding via joint variational generation," *AAAI 2019 : Thirty-Third AAAI Conference on Artificial Intelligence*, vol. 33, no. 1, pp. 7402–7409, 2019.

[31] A. Chaudhary, J. Xie, Z. Sheikh, G. Neubig, and J. G. Carbonell, "A little annotation does a lot of good: A study in bootstrapping low-resource named entity recognizers," in *EMNLP-IJCNLP*, 2019, pp. 5167–5177.

[32] Y. Tsuboi, H. Kashima, H. Oda, S. Mori, and Y. Matsumoto, "Training conditional random fields using incomplete annotations," in *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1*. Association for Computational Linguistics, 2008, pp. 897–904.

[33] C. T. Hemphill, J. J. Godfrey, and G. R. Doddington, "The atis spoken language systems pilot corpus," in *HLT '90 Proceedings of the workshop on Speech and Natural Language*, 1990, pp. 96–101.

[34] D. A. Dahl, M. Bates, M. Brown, W. Fisher, K. Hunicke-Smith, D. Pallett, C. Pao, A. Rudnicky, and E. Shriberg, "Expanding the scope of the atis task: the atis-3 corpus," in *HLT '94 Proceedings of the workshop on Human Language Technology*, 1994, pp. 43–48.

[35] G. Tur, D. Hakkani-Tur, and L. Heck, "What is left to be understood in atis," in *2010 IEEE Spoken Language Technology Workshop*, 2010, pp. 19–24.