End-to-End Spoken Language Understanding Without Full Transcripts

Hong-Kwang J. Kuo, Zoltán Tüske, Samuel Thomas, Yinghui Huang[*]*, Kartik Audhkhasi*[*]*,*
Brian Kingsbury, Gakuto Kurata, Zvi Kons, Ron Hoory, and Luis Lastras

IBM Research AI

Abstract

An essential component of spoken language understanding (SLU) is slot filling: representing the meaning of a spoken utterance using semantic entity labels. In this paper, we develop end-to-end (E2E) spoken language understanding systems that directly convert speech input to semantic entities and investigate if these E2E SLU models can be trained solely on semantic entity annotations without word-for-word transcripts. Training such models is very useful as they can drastically reduce the cost of data collection. We created two types of such speech-to-entities models, a CTC model and an attention-based encoder-decoder model, by adapting models trained originally for speech recognition. Given that our experiments involve speech input, these systems need to recognize both the entity label and words representing the entity value correctly. For our speech-to-entities experiments on the ATIS corpus, both the CTC and attention models showed impressive ability to skip non-entity words: there was little degradation when trained on just entities versus full transcripts. We also explored the scenario where the entities are in an order not necessarily related to spoken order in the utterance. With its ability to do re-ordering, the attention model did remarkably well, achieving only about 2% degradation in speech-to-bag-of-entities F1 score.

Index Terms: speech recognition, SLU

1. Introduction

Spoken language understanding is essential for a variety of applications including interactive spoken conversational systems and call center analytics that understands agent-customer dialogues. Slot filling is the process where we identify the entities (entity labels (e.g. fromloc.cityname) and values (e.g. Boston)). This type of information is obviously important for completing transactions or information seeking requests.

The ATIS (Air Travel Information Systems) [1–3] corpus, a publicly available corpus from the Linguistic Data Consortium, has been widely used for SLU research. Initially, the best models for slot filling used Conditional Random Fields [4], but more recently the best models use deep learning [5–13]. Surprisingly, most ATIS studies used text transcripts as inputs. They were considered SLU simply because the text transcripts were from actual spoken utterances, and therefore were in a spoken style. Only a few papers [7, 11, 14] use the speech signal as input. In this paper, we use speech inputs in an end-to-end spoken language understanding framework, taking speech as input and returning entity labels and values.

The goal of SLU is to understand the meaning of what was spoken, simplified in ATIS to an overall intent and a set of entities (slots). In contrast with automatic speech recognition (ASR), where word for word accuracy is desired, SLU may not care about every word or even about how it was spoken (order of entities, word choices, etc.) as long as the meaning is preserved. As a result, an SLU system may not need training data in the form of word-for-word transcripts, which are expensive to obtain for a new domain, assuming we are able to apply transfer learning using off-the-shelf general-domain ASR models previously trained on verbatim transcripts.

SLU systems have traditionally been a cascade of an automatic speech recognition (ASR) system converting speech into text followed by a natural language understanding (NLU) system that interprets the meaning of the text [15–17]. In contrast, an end-to-end (E2E) SLU system [17–24] processes speech input directly into meaning without going through an intermediate text transcript. In this paper, we demonstrate that it is possible to train an end-to-end SLU system using a set (or bag) of entities that do not match the spoken order. This may enable us to train on speech data from customer calls paired with transaction data produced by human agents. Imagine a human agent helps a client with a flight reservation, resulting in a transaction record containing the set of important entities. This record could serve as light supervision for training the model we propose. By using just the speech recording and the bag of entities in training, we can drastically reduce the cost of data collection and thus increase the amount of training data. Accurate verbatim transcription of speech data often requires 5-10× real-time for a human transcriber, not to mention additional costs for labeling entities. In contrast, the transaction record containing the bag of entities is obtained during the course of helping the customer and has no additional cost.

2. SLU use cases: what do entities look like?

For speech recognition, the training data is usually pairs of utterances and verbatim transcripts, as shown as (1) in the example below. In order to train a slot filling model, such sentences need to be further labeled with entities, as shown in (2). In this paper, we wish to train on speech that is paired with just the entities. In (3), we use the entities presented in natural spoken order for training. (3) differs from (2) simply in that all words that are not part of entities are excluded. The entities can be thought of as the more important keywords; however, it does not mean that the other words do not carry any meaning. For example, "to" and "from" clearly are important to determine whether a city is a destination or departure city. In our model, such words will not be output, but the speech signal corresponding to those words will help the model to output the correct entity label. Finally (4) makes the problem harder, but also more useful: the entities are not given in spoken order, but instead are sorted alphabetically according to entity name. This simulates the semantic frame or bag of entities concept where the order of entities does not affect the meaning: {{fromloc.city_name: RENO}, {stoploc.city_name: LAS_VEGAS }, {toloc.city_name: DALLAS}}

[*] Work performed while at IBM

1. **Transcript:** i want a flight to dallas from reno that makes a stop in las vegas
2. **Transcript+entity labels:** *i want a flight to* DALLAS B-toloc.city_name *from* RENO B-fromloc.city_name *that makes a stop in* LAS B-stoploc.city_name VEGAS I-stoploc.city_name
3. **Entities in natural spoken order:** DALLAS B-toloc.city_name RENO B-fromloc.city_name LAS B-stoploc.city_name VEGAS I-stoploc.city_name
4. **Entities in alphabetic order:** RENO B-fromloc.city_name LAS B-stoploc.city_name VEGAS I-stoploc.city_name DALLAS B-toloc.city_name

3. Adapting ASR models into SLU systems

Given the different ways in which SLU data can be transcribed, we investigate various methods to train an SLU system. Starting from a pre-trained ASR model, we explore several design choices to understand how two different kinds of E2E models behave when used to model the various kinds of SLU data. Each possible training procedure employs one or more of the following steps.

1. **ASR model adaptation to domain data (ASR-SLU adapt)**: Given that an off-the-shelf ASR model is likely trained on data that is acoustically different from the SLU data, a useful initial step is to adapt the ASR system. This step, which only uses verbatim transcripts, adapts the model to the novel acoustic conditions, words, and language constructs present in the SLU domain data. In model adaptation, one may use both the original general purpose ASR data (GP-ASR) and the domain data to provide better coverage of the ASR output units than adapting only on the domain data.
2. **Joint ASR and SLU model training (joint ASR+SLU)**: In this step, entity labels are introduced into the training pipeline along with the full transcripts. This step is a form of curriculum learning [23, 25] that gradually modifies an off-the-shelf ASR model into a full fledged SLU model. What is novel in this step is that the model is now trained to output non-acoustic entity tokens in addition to the usual graphemic or phonetic output tokens. For GP-ASR data, the targets are graphemic/phonetic tokens only, whereas for the SLU domain data, the targets also include entity labels. Although this step is a natural progression in building the final SLU model, it can be skipped if sufficient SLU resources are available.
3. **SLU model fine tuning (fine tune SLU)**: In this final step, a model from step 1 or 2 above is fine tuned on just the SLU data to create the final SLU model. As described earlier, the entities that need to be recognized by the final SLU model might take different forms: within a full transcript, entities only in spoken order, or entities only in alphabetic order.

4. Building End-to-End SLU models

Using the training procedure described above, we develop two variants of end-to-end SLU systems for the ATIS task that attempt to directly recognize entities in speech without intermediate text generation and text-based entity detection.

4.1. SLU Data and Evaluation Metric

We used the standard ATIS training and test sets for our experiments: 4978 training utterances from Class A (context independent) training data in the ATIS-2 and ATIS-3 corpora and 893 test utterances from ATIS-3 Nov93 and Dec94 data sets. The entity labeled text data is found in LDC2019T04, but there are no pointers to corresponding audio files. Only 518 (out of 893) test utterance audio files were found by [7]. We managed to find all 893 test audio files and 4976 (missing 2) training audio files, in the variety of spontaneous speaking mode, recorded with the Sennheiser microphone.

The 4976 training utterances comprise $\sim$9.64 hours of audio from 355 speakers. The 893 test utterances comprise $\sim$1.43 hours of audio from 55 speakers. The data was originally collected at 16 kHz, but we downsampled to 8 kHz to better model telephony use cases and so we could use off-the-shelf ASR models trained on conversational telephone speech. To better train the proposed E2E models, additional copies of the corpus are created using speed/tempo perturbation. The final training corpus after data augmentation is $\sim$140 hours of audio data. To simulate an additional practical use case, we create a second *noisy* ATIS corpus by adding street noise between 5-15 db SNR to the clean recordings. This $\sim$9.64 hours noisy data set is also extended via data augmentation to $\sim$140 hours. A corresponding noisy test set is also prepared by corrupting the original clean test set with additive street noise at 5 db SNR.

We measure slot filling performance with the F1 score. When using speech input instead of text, word errors can arise. The F1 score requires that both the slot label and value must be correct. For example, if the reference is *toloc.city_name:new_york* but the decoded output is *toloc.city_name:york*, then we count both a false negative and a false positive. It is not sufficient that the correct slot label is produced: no "partial credit" is given for part of the entity value (*york*) being recognized. The scoring ignores the order of entities, and is therefore suitable for the "bag-of-entities" case we study. Our scoring script was tested on text-input systems and gave identical values as the standard scoring scripts.

4.2. Evaluating CTC based SLU models

To allow the SLU model to process entities and corresponding values independent of an external language model, we first construct a word CTC model on general purpose ASR data with the recipe steps presented in [26, 27] and using 300 hours of Switchboard (SWB-300) data. We then explore different training recipes to build CTC based SLU models.

Our first experiment assumes that we have both verbatim transcripts and entity labels for the SLU data and uses all three training steps. The **ASR-SLU adapt** step is performed as follows. The output layer of the ASR model, which estimates scores for 18,324 word targets and the blank symbol, is replaced with a randomly initialized output layer that estimates scores for 18,642 word/entity targets and the blank. The weights of the remaining 6 LSTM layers, each with 640 units per direction, and a fully connected bottleneck layer with 256 units are kept the same. The model is then trained on a combined data set of 300 hours of SWB GP-ASR data and 140 hours of clean ATIS data. Note that in this step, although the output layer has units for entity labels, the training targets are only words. In the **joint ASR+SLU** step, entity labels are introduced into the training transcripts and a joint ASR-SLU model is trained on the SWB+SLU data, starting from the final weights from the **ASR-SLU adapt** step. In the third and final **fine tune SLU** step, the joint ASR-SLU model is fine tuned on just the 140 hours of ATIS SLU data.

In experiment [1A] of Table 1, we evaluate this model on the clean test ATIS data. Given that the SLU model is a word CTC model, we do not use an external LM while decoding; instead, a simple greedy decode of the output is employed. This initial model has an F1 score of 91.7 for correctly detecting

entity labels along with their values. In experiment [2A], we develop a similar SLU model with full verbatim transcripts along with entity labels, but we skip the **ASR-SLU adapt** and **joint ASR+SLU adapt** steps. We initialize the model with the pre-trained SWB ASR model and directly train the SLU model. This model also achieves 91.7 F1 score, suggesting that the curriculum learning steps may not always be required.

Table 1: ATIS *bag-of-entities slot filling F1 score for speech input using CTC and Attention based models*

Training Data	Adapt	CTC	Attention
[1A] Full transcripts	Y	91.7	92.9
[2A] Full transcripts	N	91.7	93.0
[3A] Entities, spoken order	Y	92.7	92.8
[4A] Entities, spoken order	N	91.5	92.6
[5A] Entities, alphabetic order	Y	73.5	90.9
[6A] Entities, alphabetic order	N	61.9	90.6

In the next set of experiments we investigate how important verbatim transcripts are for the training process. After the **joint ASR+SLU** step of experiment [1A], in experiment [3A], we train an SLU model that recognizes just the entity labels and their values in spoken order. We observe that the model learns to disregard words in the signal that are not entity values, while preserving just the entity values along with their labels. This model performs slightly better than full transcript model in [1A]. We extend this experiment in [4A] by removing the use of transcripts entirely in the training process. This SLU model, after being initialized with a pre-trained ASR model, is trained directly to recognize entity labels and their values without any curriculum learning steps or verbatim transcripts. The model drops slightly in performance, but remains on par with the baseline systems. Finally, we train SLU systems on the much harder task of recognizing alphabetically sorted entity labels and their values. After the **joint ASR+SLU** step of experiment [1A], in experiment [5A] we train an SLU model that recognizes just the entity labels and their values, but now in alphabetic order. In experiment [6A] a similar model is trained, but without any curriculum learning steps. On this task, the performance of the CTC model drops significantly as it is unable to learn from reordered targets. With the curriculum learning steps, the results in [5A] are better, but still much worse than the baselines.

4.3. Evaluating Attention based SLU models

The attention models for SLU are initialized with a state-of-the-art ASR model developed for standard Switchboard ASR task. This model uses an encoder-decoder architecture in which the encoder is an 8-layer LSTM stack using batch-normalization, residual connections, and linear bottleneck layers [28–31]. The decoder models the sequence of BPE units estimated on characters [32], and consists of 2 unidirectional LSTM layers. One is a dedicated language-model-like component that operates only on the embedded predicted symbol sequence, and the other jointly processes acoustic and symbol information. The decoder applies additive, location aware attention [33], and each layer has 768 unidirectional LSTM nodes. As has been shown in [34], exploiting various regularization techniques, including SpecAugment, sequence-noise injection, speed-tempo augmentation, and various dropout methods [35–40], results in state-of-the art speech recognition performance using this single-headed sequence-to-sequence model.

To recognize entities, the ASR model is adapted similar to the CTC model, following the steps of Section 3. In contrast to the CTC model, which uses word units, the attention model uses a smaller inventory of 600 BPE units and relies on the decoder LSTMs to model longer sequences — the attention based model has an inherent long-span language model. After the initial ASR model is trained on Switchboard, the subsequent adaptation and transfer learning steps used only the ATIS data without any Switchboard data. Because the attention model operates at the sub-word level, and all new words appearing in the ATIS transcripts can be modeled using these sub-word units, no extension of the output and embedding layer is needed in the first **ASR-SLU adapt** step. We skip the **joint ASR+SLU** step and proceed directly to the **fine tune SLU** step, where the output and the embedding layers of the decoder must be extended with the entity labels. The softmax layer and embedding weights corresponding to the entity labels are randomly initialized, while all other parameters, including the weights which correspond to previously known symbols in the softmax and embedding layers, are copied over from the ASR model. Having no out-of-vocabulary words, sub-word level models are ideally suited to directly start the adaptation process with step 3 of Section 3. All adaptation steps use 5 epochs of training.

The last column of Table 1 shows the slot filling F1 score for attention based SLU models. In experiment [1A], an attention based ASR model trained on Switchboard-300h is first adapted on the clean ATIS data to create a domain specific ASR model. On the test set, the word error rate (WER) using the base SWB-300 model is about 7.9% which improved to 0.6% after adaptation. This ASR model is then used as an initial model for transfer learning to create an SLU model. The F1 score is comparable to that of the CTC model. In experiment [2A], we skip the ASR adaptation step and directly use the SWB-300 ASR model to initialize the SLU model training. In this scenario, there is no degradation in F1 score. There is no difference in SLU performance whether the model is initialized with a general purpose SWB-300 ASR model (WER=7.9%) or with a domain adapted ASR model (WER=0.6%).

We next consider the effects of training transcription quality or detail. Using transcripts that contain only entities in spoken order ([4A]), we obtain F1 scores that are almost the same as using full transcripts ([1A]). When training transcripts contain entities in alphabetic order (possibly different from spoken order) ([6A]), there is a 2% degradation in F1 score, from 92.9 to 90.9. This result is much better than that for the CTC model (73.5), reflecting the re-ordering ability of attention based models. As before, adding an extra step of ASR model adaptation ([3A] and [5A]) with verbatim transcripts made little difference. This is encouraging, since we are assuming we are only given entities in the training transcripts.

Figure 1 shows the attention plots for the utterance *"i would like to make a reservation for a flight to denver from philadelphia on this coming sunday"* with three different attention models: (a) ASR model, (b) SLU in spoken order, and (c) SLU in alphabetic order. The attention for (b) is largely monotonic with attention paid on consecutive parts of the audio signal corresponding to BPE units of keywords in the entities. There are gaps reflecting skipping over non-entity words. In (c), the attention is piece-wise monotonic, where the monotonic regions cover the BPE units within a keyword. Since the entities are given in an order different from spoken order, the plot shows how the model is able to associate the correct parts of the speech signal with the entities. In addition, at around 2 seconds, attention was paid to the phrase *"make a reservation"* which is predictive of the overall intent of the sentence *"flight."* (Intent recognition is

Table 2: ATIS *bag-of-entities slot filling F1 score for speech input with additive street noise (5dB SNR)*

Training Data	Adapt	CTC	Attention
[1B] Full transcripts	Y	85.5	92.0
[2B] Full transcripts	N	79.6	91.3
[3B] Entities, spoken order	Y	88.6	91.2
[4B] Entities, spoken order	N	86.5	89.6
[5B] Entities, alphabetic order	Y	73.8	88.8
[6B] Entities, alphabetic order	N	68.5	87.7

Table 3: *Effect of different amounts of data used to pre-train the ASR model used in initializing SLU model training*

ASR Training	Attention
None	78.1
Switchboard 300h	92.6
Switchboard 2000h	93.8

left out of this paper for simplicity and due to lack of space.)

4.4. Effect of Acoustic Mismatch

In a next set of experiments (Table 2), we use the *noisy* ATIS corpus as the SLU data set and repeat the CTC based experiments conducted earlier. This set of experiments introduces additional variability to the training procedure with realistic noise in both training and test. Further, it increases the acoustic mismatch between the transferred model and the target domain. The general trends for the CTC model observed in Table 1 are also observed in Table 2: (a) ASR transcript based curriculum training is effective; and, (b) entity labels can be recognized reasonably well in spoken order, but the performance is worse when the entity order is different. In experiments like [2B], the mismatch between the SLU data and the ASR data affects the performance of models that are only initialized with mismatched pre-trained models and have no other adaptation steps. The noise distortion in general causes these systems to drop in performance compared to the performance results in matched conditions.

Looking at the results in Table 2 for attention based SLU models in more detail, we note that there is an absolute degradation of 4.3% in F1 score when we compare a model trained on full transcripts ([1B] F1=92.0) to one trained on entities in alphabetic order ([6B] F1=87.7%). While this is a significant drop in performance, it is much better than the CTC result of ([6B] F1=68.5). Compared to the clean speech condition, we also come to a different conclusion regarding the utility of ASR adaptation. We see about 1% improvement in F1 score when we are able to use an adapted ASR model instead of the base SWB-300 model to initialize SLU model training. On the noisy test set, using the base SWB-300 model results in WER=60%, whereas the ASR model adapted on noisy ATIS data gives WER=5%. It is remarkable that using these two very different ASR models to initialize the SLU model training leads to only a 1% difference in F1 scores for the final models.

4.5. Effect of the Amount of Pre-training Data

Table 3 shows how the amount of data used to train the ASR model for initializing SLU training affects the final F1 score. Here we show only results for attention-based SLU models trained on entities in spoken order for clean speech. We saw earlier that ASR adaptation on domain data does not always help. But here, using 2000h instead of 300h for the initial ASR model

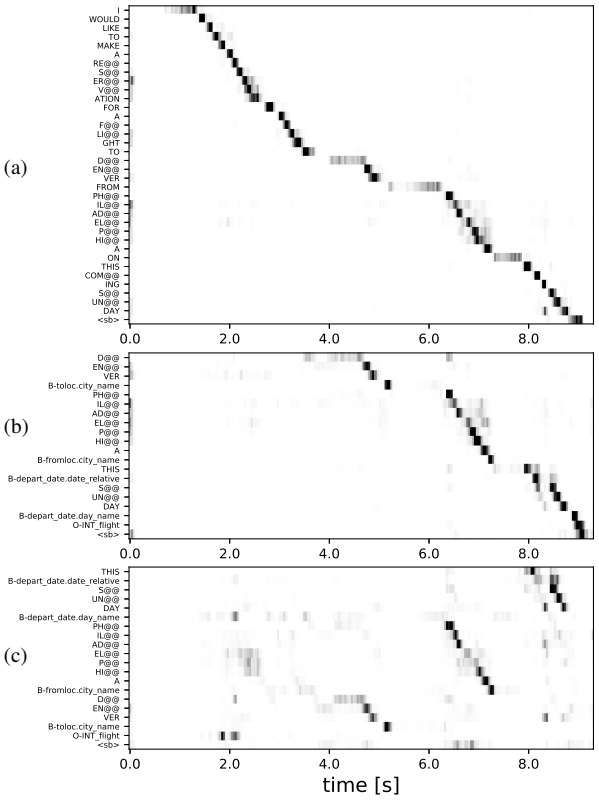

Figure 1: *Attention plots for the utterance "I would like to make a reservation for a flight to Denver from Philadelphia on this coming Sunday": (a) ASR; (b) SLU in spoken order; (c) SLU in alphabetic order.*

improves the F1 score by about 1%, most likely due to increased robustness of the model to unseen data: the unadapted WER on the ATIS test set is 3.1% (SWB2000h) vs. 7.9% (SWB300h). In contrast, when we directly train the SLU model from scratch, the best we could do was about F1=78.1. When SLU data is limited, these experiments demonstrate the importance of ASR pre-training on a broad range of speech data, not necessarily related to the final SLU task.

5. Conclusions and Future Work

In this paper we have investigated how various E2E SLU models can be built without verbatim transcripts. We have shown the importance of using pre-trained acoustic models and curriculum learning to build these systems. Using clean and noisy versions of ATIS, we explored the effects of entity order and acoustic mismatch on performance of these systems. This study shows that E2E systems can indeed be trained without verbatim transcripts and can predict entities reliably even if trained on transcripts where entities are not necessarily given in spoken order. Our results provide useful insights to building better SLU systems in practical settings where full transcripts are often not available for training and the final SLU systems need to be deployed in noisy acoustic environments. The current study was limited to a setting with context independent utterances. Future research may involve building SLU systems that operate on full conversations, rather than single utterances, where more complex linguistic phenomena like co-reference and entity linking are present.

6. References

[1] P. Price, "Evaluation of spoken language systems: The ATIS domain," in *Speech and Natural Language: Proceedings of a Workshop Held at Hidden Valley, Pennsylvania, June 24-27, 1990*, 1990.

[2] C. T. Hemphill, J. J. Godfrey, and G. R. Doddington, "The ATIS spoken language systems pilot corpus," in *Speech and Natural Language: Proceedings of a Workshop Held at Hidden Valley, Pennsylvania, June 24-27, 1990*, 1990.

[3] G. Tur, D. Hakkani-Tür, and L. Heck, "What is left to be understood in ATIS?" in *Proc. IEEE SLT Workshop*, 2010, pp. 19–24.

[4] C. Raymond and G. Riccardi, "Generative and discriminative algorithms for spoken language understanding," in *Proc. Interspeech*, 2007.

[5] B. Liu and I. Lane, "Attention-based recurrent neural network models for joint intent detection and slot filling," *arXiv preprint arXiv:1609.01454*, 2016.

[6] C.-W. Goo, G. Gao, Y.-K. Hsu, C.-L. Huo, T.-C. Chen, K.-W. Hsu, and Y.-N. Chen, "Slot-gated modeling for joint slot filling and intent prediction," in *Proc. NAACL-HLT, Vol. 2 (Short Papers)*, 2018, pp. 753–757.

[7] B. Liu and I. Lane, "Joint online spoken language understanding and language modeling with recurrent neural networks," *arXiv preprint arXiv:1609.01462*, 2016.

[8] D. Guo, G. Tur, W.-t. Yih, and G. Zweig, "Joint semantic utterance classification and slot filling with recursive neural networks," in *Proc. IEEE SLT Workshop*, 2014, pp. 554–559.

[9] G. Kurata, B. Xiang, B. Zhou, and M. Yu, "Leveraging sentence-level information with encoder LSTM for semantic slot filling," *arXiv preprint arXiv:1601.01530*, 2016.

[10] B. Liu and I. Lane, "Recurrent neural network structured output prediction for spoken language understanding," in *Proc. NIPS Workshop on Machine Learning for Spoken Language Understanding and Interactions*, 2015.

[11] G. Mesnil, Y. Dauphin, K. Yao, Y. Bengio, L. Deng, D. Hakkani-Tur, X. He, L. Heck, G. Tur, D. Yu *et al.*, "Using recurrent neural networks for slot filling in spoken language understanding," *IEEE/ACM Trans. on Audio, Speech, and Language Processing*, vol. 23, no. 3, pp. 530–539, 2014.

[12] P. Xu and R. Sarikaya, "Convolutional neural network based triangular CRF for joint intent detection and slot filling," in *Proc. IEEE ASRU Workshop*, 2013, pp. 78–83.

[13] D. Hakkani-Tür, G. Tür, A. Celikyilmaz, Y.-N. Chen, J. Gao, L. Deng, and Y.-Y. Wang, "Multi-domain joint semantic frame parsing using bi-directional RNN-LSTM," in *Proc. Interspeech*, 2016, pp. 715–719.

[14] C.-W. Huang and Y.-N. Chen, "Adapting pretrained transformer to lattices for spoken language understanding," in *Proc. IEEE ASRU Workshop*, 2019.

[15] V. Goel, H.-K. J. Kuo, S. Deligne, and C. Wu, "Language model estimation for optimizing end-to-end performance of a natural language call routing system," in *Proc. ICASSP*, vol. 1, 2005, pp. 565–568.

[16] S. Yaman, L. Deng, D. Yu, Y.-Y. Wang, and A. Acero, "An integrative and discriminative technique for spoken utterance classification," *IEEE Trans. on Audio, Speech, and Language Processing*, vol. 16, no. 6, pp. 1207–1214, 2008.

[17] 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 *Proc. IEEE SLT Workshop*, 2018, pp. 720–726.

[18] D. Serdyuk, Y. Wang, C. Fuegen, A. Kumar, B. Liu, and Y. Bengio, "Towards end-to-end spoken language understanding," in *Proc. ICASSP*, 2018, pp. 5754–5758.

[19] Y. Qian, R. Ubale, V. Ramanaryanan, P. Lange, D. Suendermann-Oeft, K. Evanini, and E. Tsuprun, "Exploring ASR-free end-to-end modeling to improve spoken language understanding in a cloud-based dialog system," in *Proc. IEEE ASRU Workshop*, 2017, pp. 569–576.

[20] Y.-P. Chen, R. Price, and S. Bangalore, "Spoken language understanding without speech recognition," in *Proc. ICASSP 2018*, 2018, pp. 6189–6193.

[21] S. Ghannay, A. Caubrière, Y. Estève, N. Camelin, E. Simonnet, A. Laurent, and E. Morin, "End-to-end named entity and semantic concept extraction from speech," in *Proc. IEEE SLT Workshop*, 2018, pp. 692–699.

[22] L. Lugosch, M. Ravanelli, P. Ignoto, V. S. Tomar, and Y. Bengio, "Speech model pre-training for end-to-end spoken language understanding," in *Proc. Interspeech*, 2019, pp. 814–818.

[23] A. Caubrière, N. Tomashenko, A. Laurent, E. Morin, N. Camelin, and Y. Estève, "Curriculum-based transfer learning for an effective end-to-end spoken language understanding and domain portability," in *Proc. Interspeech*, 2019, pp. 1198–1202.

[24] Y. Huang, H.-K. Kuo, S. Thomas, Z. Kons, K. Audhkhasi, B. Kingsbury, R. Hoory, and M. Picheny, "Leveraging unpaired text data for training end-to-end speech-to-intent systems," in *Proc. ICASSP*, 2020, pp. 7984–7988.

[25] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, "Curriculum learning," in *Proc. ICML*, 2009, pp. 41–48.

[26] K. Audhkhasi, G. Saon, Z. Tüske, B. Kingsbury, and M. Picheny, "Forget a bit to learn better: Soft forgetting for CTC-based automatic speech recognition," in *Proc. Interspeech*, 2019, pp. 2618–2622.

[27] G. Kurata and K. Audhkhasi, "Guiding CTC posterior spike timings for improved posterior fusion and knowledge distillation," *arXiv preprint arXiv:1904.08311*, 2019.

[28] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[29] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. ICML*, 2015, pp. 448–456.

[30] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. CVPR*, 2016, pp. 770–778.

[31] K. Veselý, M. Karafiát, and F. Grézl, "Convolutive bottleneck network features for LVCSR," in *Proc. IEEE ASRU Workshop*, 2011, pp. 42–47.

[32] "https://github.com/rsennrich/subword-nmt."

[33] J. K. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, "Attention-based models for speech recognition," in *Proc. NIPS*, 2015, pp. 577–585.

[34] Z. Tüske, G. Saon, K. Audhkhasi, and B. Kingsbury, "Single headed attention based sequence-to-sequence model for state-of-the-art results on Switchboard-300," *arXiv preprint arXiv:2001.07263*, 2020.

[35] D. S. Park, W. Chan, Y. Zhang, C. C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "SpecAugment: A simple data augmentation method for automatic speech recognition," in *Proc. Interspeech*, 2019, pp. 2613–2617.

[36] G. Saon, Z. Tüske, K. Audhkhasi, and B. Kingsbury, "Sequence noise injected training for end-to-end speech recognition," in *Proc. ICASSP*, 2019, pp. 6261–6265.

[37] T. Ko, V. Peddinti, D. Povey, and S. Khudanpur, "Audio augmentation for speech recognition," in *Proc. Interspeech*, 2015, pp. 3586–3589.

[38] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," *arXiv preprint arXiv:1207.0580*, 2012.

[39] L. Wan, M. Zeiler, S. Zhang, Y. L. Cun, and R. Fergus, "Regularization of neural networks using DropConnect," in *Proc. ICML*, vol. 28, no. 3, 2013, pp. 1058–1066.

[40] D. Krueger, T. Maharaj, J. Kramár, M. Pezeshki, N. Ballas, N. R. Ke, A. Goyal, Y. Bengio, A. Courville, and C. Pal, "Zoneout: regularizing RNNs by randomly preserving hidden activations," in *Proc. ICLR*, 2017.