INTERSPEECH 2019
September 15–19, 2019, Graz, Austria

A Comparison of Deep Learning Methods for Language Understanding

Mandy Korpusik[1], *Zoe Liu*[2], *James Glass*[2]

[1]Loyola Marymount University, Los Angeles, CA 90045, USA
[2]MIT Computer Science and Artificial Intelligence Laboratory, Cambridge, MA 02139, USA
korpusik@csail.mit.edu, liuxy.zoe2016@outlook.com, glass@mit.edu

Abstract

In this paper, we compare a suite of neural networks (recurrent, convolutional, and the recently proposed BERT model) to a CRF with hand-crafted features on three semantic tagging corpora: the Air Travel Information System (ATIS) benchmark, restaurant queries, and written and spoken meal descriptions. Our motivation is to investigate pre-trained BERT's transferability to the domains we are interested in. We demonstrate that neural networks without feature engineering outperform state-of-the-art statistical and deep learning approaches on all three tasks (except written meal descriptions, where the CRF is slightly better) and that deep, attention-based BERT, in particular, surpasses state-of-the-art results on these tasks. Error analysis shows the models are less confident when making errors, enabling the system to follow up with the user when uncertain.
Index Terms: BERT, Semantic Tagging, CNN, RNN, CRF

1. Introduction

The first step in a dialogue system, after recognizing the speech of the user utterance, is spoken language understanding (SLU). Specifically, SLU entails identifying relevant slots and their associated values, also known as semantic tagging or slot filling. Given these slots and their values, the system then decides which action to take next and how best to respond to the user.

In our prior work, we explored convolutional neural network (CNN) models for semantic tagging and mapping of natural language meal descriptions to a structured food database [1, 2, 3], as well as for dialogue state tracking [4, 5, 6]. In this work, we demonstrate that our CNN generalizes to other domains beyond nutrition, outperforming prior state-of-the-art on the benchmark ATIS task, as well as a restaurant query dataset. In addition, we compare to recurrent neural networks (RNNs) and the recent BERT model [7]. We establish that prior to BERT, ensembles of RNNs and CNNs performed best in general, with boosts from pre-trained word vectors, but that now a single pre-trained BERT model with fine-tuning of a token classification layer on top outperforms them all. In the remainder of this paper, we explain the three tasks in detail, outline the deep learning models, show results, and analyze the types of errors the model makes, as well as what the CNN filters learn.

2. Semantic Tagging Tasks

We evaluated deep learning models on three spoken language understanding datasets—Air Travel Information System (ATIS), restaurant booking, and food logging. ATIS[1], which has been used since the 1990s, and the restaurant task[2] are both publicly available. See Table 1 for a summary of the data.

Table 1: *The data statistics for each corpus. The spoken meal description data uses the same training set as the written.*

Dataset	# Train Data	# Test Data	# Tags
ATIS	4,978	893	127
Restaurants	7,659	1,520	17
Written Meals	35,130	3,412	5
Spoken Meals	35,130	476	5

2.1. ATIS Corpus

The Air Travel Information System (ATIS) task involves dialogues between users and automated spoken dialogue systems for booking flights. The goal is to label each token in a user utterance with the correct semantic tags, which are in the standard B-I-O format (e.g., `B-fromloc.city_name` refers to the beginning of the departure city's name, `I-fromloc.city_name` is inside the departure city phrase, and `O` is `Other`). We show an example user utterance with its corresponding gold standard semantic tags in Fig. 1. In our work, we start from the same dataset as prior work [8]: the training set consists of 4,978 utterances selected from the Class A (context independent) training data in the ATIS-2 and ATIS-3 corpora, and the ATIS test set contains both the ATIS-3 NOV93 and DEC94 datasets.

Figure 1: *Semantic tagging on a user utterance in ATIS, where BOS and EOS refer to the beginning and end of a sentence.*

2.2. Restaurant Corpus

The restaurant corpus was collected on Amazon Mechanical Turk (AMT), where workers were hired to write queries about restaurants, given a set of keywords [9].

2.3. Written and Spoken Meal Corpus

In our prior work [10], we collected written meal descriptions on AMT, which were then tagged by Turkers in a second round of labeling tasks [11]. To generate spoken meal descriptions, we asked Turkers to verbally record a subset of the written meal descriptions. In total, we collected 2,962 spoken utterances (from 37 speakers, totaling 2.74 hours), which we used in prior work to train a Kaldi [12] recognizer with a decoder word error rate (WER) of 7.98% on a held-out test set. AMT workers annotated the semantic tags of the recognizer's output.

This research was sponsored by a grant from Quanta Computing, Inc., and by the Department of Defense (DoD) through the National Defense Science & Engineering Graduate Fellowship (NDSEG) Program.

[1]https://github.com/yvchen/JointSLU/tree/master/data

[2]https://groups.csail.mit.edu/sls/downloads/restaurant/

http://dx.doi.org/10.21437/Interspeech.2019-1262

3. Related Work

Substantial work has been devoted to spoken language understanding, specifically semantic tagging of the ATIS corpus for flights and air travel. Early work explored generative and discriminative models, including finite-state transducers (FSTs), support vector machines (SVMs), and conditional random field models (CRFs) [13]. Other work boosted the performance of statistical models by extracting keywords based on a dependency parse of the user utterance [14]. More recently, deep learning models have been shown to outperform CRFs, such as convolutional neural networks (CNNs) [15], recurrent neural networks (RNNs) [16], and jointly trained RNNs for slot filling and intent detection [17, 18, 19, 20]. For the restaurant and meal description tasks, prior state-of-the-art used CRFs with carefully hand-crafted features, such as semantic dependency features from query dependency parses [21], and word vector and distributional prototype similarity features [10].

4. Models

We investigated a collection of deep learning models for semantic tagging: RNNs, CNNs, the recently proposed attention-based BERT [7], conditional random field (CRF) models trained on logits from the hidden layer of RNNs, and a feed-forward (FF) baseline. We also ensembled neural networks together, and initialized the CNN with pre-trained word embeddings.

4.1. RNN

In our PyTorch implementation of the RNN, we built a bidirectional gated recurrent unit (GRU) on top of a word embedding layer, with a linear layer on top for the final prediction. We used embeddings of dimension 128, hidden layers of size 512, batches with 50 samples each, and trained with the Adam optimizer on cross-entropy loss for 1,000 steps of randomly selected batches. The maximum length was set based on the longest sample in each batch. The FF baseline replaced the recurrent layer with a linear layer of size 512. We implemented the RNN-CRF as in [22] by extracting logits from the trained RNN's hidden layer and feeding these as features to the CRF. Therefore, the RNN-CRF does not require any manual feature engineering, unlike the CRFs discussed in the related work.

4.2. CNN

As in our prior work [23], we built a CNN tagger composed of a word embedding layer followed by three stacked 1D convolutional layers, with kernel windows spanning lengths of five, five, and three tokens, respectively. We learned 150-dimension embeddings, 64 filters per convolutional layer, applied ReLU activation, and trained with the Adam optimizer on cross-entropy loss for up to 15 epochs with early stopping determined by no performance gain on the validation set (20% split). We experimented with pre-trained word embeddings from 200-dimension Glove [24] (trained on Wikipedia and Gigaword) and 300-dimension word2vec [25] (trained on Google News).

4.3. BERT

Within the past year, several papers have come out that learn *contextual* representations of sentences, where the entire sentence is used to generate embeddings. ELMo (Embeddings from Language Models) [26] uses a linear combination of vectors extracted from intermediate layer representations of a bidirectional LSTM trained on a large text corpus as a language model (LM). The OpenAI GPT (Generative Pre-trained Transformer) [27] is a fine-tuning approach, where they first pre-train a multi-layer Transformer [28] as a LM on a large text corpus, and then conduct supervised fine-tuning on the specific task of interest, with a linear and softmax layer on top of the pre-trained Transformer. Finally, Google's BERT (Bidirectional Encoder Representations from Transformers) [7] is a fine-tuning approach similar to GPT, but with the key difference that instead of combining separately trained forward and backward Transformers, they instead use a *masked* LM for pre-training. They demonstrate state-of-the-art performance on 11 NLP tasks, including the CoNLL 2003 entity recognition task.

For our BERT model, we used the base pre-trained BERT (parameters were frozen) with a fine-tuned softmax token classification layer added on top (no CRF), tuned hyperparameters on 10% validation data (i.e., batch of 32, uncased tokenizer, 3×10^{-5} learning rate, and four epochs). Since BERT uses word pieces, but the data is pre-tokenized, we use only the first sub-token's predicted label during evaluation. In Fig. 2, sub-word tokens labeled `X` are omitted in evaluation. In future work, it would be interesting to compare the performance of several BERT models—fine-tuning multiple layers, using more than just the first sub-token, and extracting contextual embedding features from multiple layers for classification.

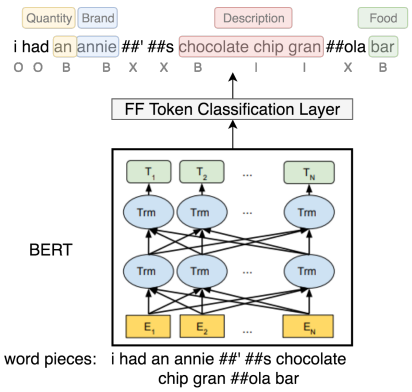

Figure 2: *An illustration of how BERT is used to generate contextualized word embeddings, which are then fed into a fine-tuned linear token classification layer on top.*

5. Experiments

For each of the language understanding tasks, we evaluated our deep learning models on a held-out test set and compared our performance to state-of-the-art models. For ATIS, restaurants, and spoken meals, we demonstrate that deep learning models surpass prior state-of-the-art, in terms of weighted average precision, recall, and F-score. For meals, we break down F-score by tag since there are only five tags. Although the CRF performs better than the deep learning models on the written meals, it requires many carefully hand-crafted features, while the deep learning models do not require any feature engineering.

Overall, BERT performs best on all tasks (see Tables 2, 3, 4, and 5). On ATIS, the RNN ensemble is second best; on restaurants, the ensemble of RNNs and CNNs is next; on written meals, the CRF trained on n-grams, part-of-speech (POS) tags, entities, food and brand lexicons, and pre-trained word embeddings slightly outperforms BERT; on *spoken* meals, a CNN trained on word2vec is second; and, in general, the FF baseline is worst. Finally, pre-training the CNN with word vectors improves performance, with Glove slightly better than word2vec.

Table 2: *F1 scores on ATIS, for our models versus prior work.*

Model	Precision	Recall	F-score
FF baseline	85.8	86.7	85.7
Transformer encoder [28]	96.6	96.6	96.4
1 RNN	97.9	97.9	97.7
RNN-CRF	97.8	97.9	97.7
4 RNNs	98.0	98.1	97.8
1 CNN	96.4	97.0	96.5
CNN + Glove	97.5	97.6	97.3
CNN + word2vec	97.1	97.4	97.1
4 Glove CNNs	97.5	97.7	97.4
4 RNNs + 4 Glove CNNs	97.8	97.9	97.6
BERT	**98.1**	**98.3**	**98.1**
FST [13]	91.6	91.9	91.7
CRF [14]	–	–	95.0
R-CRF [16]	–	–	96.5
Joint seq. bLSTM [17]	–	–	94.3
Attention-based [18]	–	–	94.2
Slot-gated bLSTM [20]	–	–	95.2

Table 3: *Precision, recall, and F-scores on the restaurant test set [21].The gain from Glove and word2vec is not from using larger dimensions (200 and 300, respectively). Without pre-trained embeddings, using larger dimensions decreases the F-score (from 88.1 to 87.5 and 87.1, respectively).*

Model	Precision	Recall	F-score
FF baseline	82.3	80.7	81.0
1 RNN	87.1	87.4	87.2
RNN-CRF	87.1	87.4	87.2
4 RNNs	89.1	89.3	89.1
1 CNN	88.3	88.2	88.1
CNN + Glove	88.8	88.8	88.7
CNN + word2vec	88.9	88.7	88.6
4 Glove CNNs	89.7	89.7	89.7
4 RNNs + 4 Glove CNNs	89.9	90.0	89.8
BERT	**91.6**	**91.6**	**91.6**
Best CRF [21]	85.3	83.9	84.6

Table 4: *Per-label F1 scores on written meals [29]. The CRF performs best, but it requires hand-crafted features, whereas the neural models are competitive without feature engineering. Although BERT is not the best overall, it does particularly well on brands and descriptions, which is hard for the other models, and even the AMT workers, to distinguish.*

Model	Food	Brand	Num	Descrip	Avg
FF baseline	85.1	69.0	91.0	74.4	85.3
1 RNN	94.3	77.2	94.2	87.1	92.1
RNN-CRF	94.4	76.8	93.1	86.9	91.7
4 RNNs	**95.1**	80.5	94.5	88.4	92.9
1 CNN	91.9	79.5	95.1	87.1	92.4
CNN + Glove	94.4	84.1	94.7	89.5	93.9
CNN + wd2vc	93.6	83.6	91.0	88.0	92.1
4 Glove CNNs	94.4	84.4	91.7	89.0	92.7
4 RNN + 4 CNN	76.9	78.3	94.5	89.1	85.3
BERT	94.6	**87.0**	94.7	**90.4**	94.2
CRF (unigram)	92.3	78.5	93.9	86.6	92.4
CRF (+ bigram)	94.1	80.3	95.1	88.9	93.7
Best CRF [29]	94.6	85.7	**95.1**	90.3	**94.4**

Table 5: *Per-label F1 scores on spoken meal data [29]. All the neural networks outperform the CRF, demonstrating that they are more robust to speech recognition errors.*

Model	Food	Brand	Num	Descrip	Avg
FF baseline	87.9	61.7	93.7	78.3	90.1
1 RNN	94.0	81.5	95.9	89.1	94.9
RNN-CRF	94.1	80.6	95.3	87.9	94.5
4 RNNs	94.4	80.9	97.2	89.8	95.3
1 CNN	93.9	78.2	97.5	89.1	95.1
CNN + Glove	94.9	80.9	97.5	91.1	95.5
CNN + wd2vc	95.4	80.6	97.2	90.9	95.7
4 Glove CNNs	95.1	82.2	97.0	91.4	95.6
4 RNN + 4 CNN	**95.5**	77.4	97.8	**91.7**	95.6
BERT	94.4	**85.3**	**97.8**	91.1	**95.8**
Best CRF [29]	93.3	79.0	96.6	87.7	94.2

6. Analysis

We hypothesize that the models will be less confident in their predictions when making mistakes. This has important applications for real-world tasks, in which the model could learn from human feedback when it is not confident in its prediction. For example, in a diet tracking application, if the model is uncertain about the tag for "oatmeal" in the food description "oatmeal cookie," it might ask, "Was the oatmeal a *description* of cookie, or a separate *food*?" Thus, the model could learn online from users without asking an overwhelming number of questions, only for clarification on those for which it is least certain. In addition, we could use the confidence of the model to discover errors in the test set labels if, for example, the model is very confident when it makes a mistake. Here we show that the models are indeed less confident when making errors, and that high confidence can be used to discover errors in the test data.

We see in Fig. 3 that as we increase the confidence threshold[3] from 0.5 to 0.999, the weighted average F-score on the written meal test set increases. In particular, during evaluation, we omit the examples where the model is uncertain, since the predicted tag's probability is below the specified confidence threshold. The performance improvement from eliminating examples where the model is less certain indicates that the model is more confident when its predicted tag is correct, and less confident when it makes errors, as hypothesized.

Error analysis reveals that the model tends to have less certainty in its predictions when it is mistaken, and that high-confidence predictions may identify errors in the data annotation. We see in Fig. 4 that the high-confidence prediction for "syrup" as a `Food` (i.e., $p = 0.999$) is actually correct, whereas the gold standard tag for `Brand` from AMT is a mistake. We also note that the predicted probabilities for the mistakes are lower, illustrating the model's uncertainty (i.e., $p = 0.69$ and $p = 0.73$ for incorrectly tagged tokens "medium sized" and "butter," respectively).

We see a common type of error made by the model in Fig. 5, where multiple adjacent food items cause the model to incorrectly predict the first food token as a `Description` rather than a `Food`. In addition, Fig. 6 illustrates the difficulty of distinguishing between brands, foods, and descriptions, especially in the presence of OOV words (i.e., out-of-vocabulary tokens not seen during training are `<UNK>`). One advantage of BERT is that it uses word-pieces, which mitigates the OOV problem.

[3]Note that we define "confidence" as the probability which the model assigns to the top-predicted label (unrelated to ASR confidence).

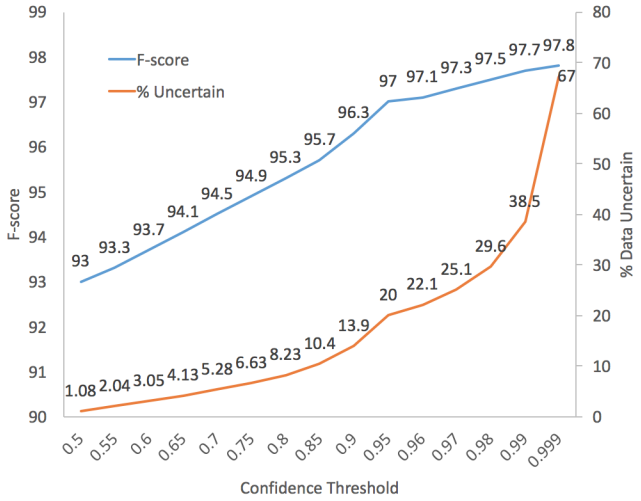

Figure 3: *The F-score as a function of the confidence threshold, on the written meal description task, where tags that fall below the threshold are omitted from the recompute of the F-score. Performance improves as more data is omitted, since the model is only evaluated on tags where it is more certain (i.e., the predicted tag's probability is above the confidence threshold). While the percent of data omitted increases quite a bit as the confidence threshold goes up from 0.9 to 0.999, the F-score gain is incremental in this region of the plot.*

Figure 4: *Incorrectly predicted semantic tags for a sample meal description, where "syrup" is actually tagged correctly by the model. Thus, this data annotation error can be discovered by identifying incorrect predictions that have high confidence (e.g., $p = 0.999$).*

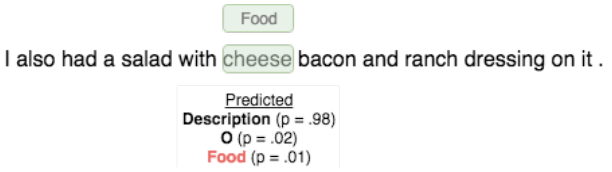

Figure 5: *The model incorrectly predicts that "cheese" is a description of "bacon" instead of a separate food item.*

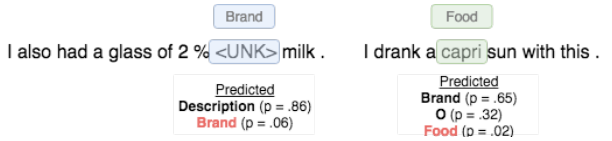

Figure 6: *The model mistakes brands for descriptions or foods.*

In addition to error analysis, we also analyzed what specifically the neural network models were learning, since a common criticism is that these networks are "black boxes" and hard for humans to interpret. To do this, we select the tokens that have the highest activation for each of the 64 learned filters in the top layer of the trained three-layer CNN. We observe that individual filters seem to pick out *semantically similar* words. For example, in Table 6, filter 11 identifies numbers, 12 focuses on flight attributes such as airline and number of stops, and filter 22 finds cities. We see a similar trend in Table 7 on the restaurant data (i.e., filter 6 identifies cuisines, filter 56 focuses on time, and filter 35 finds tokens related to ratings), and Table 8 on the meal description task (i.e., filter 34 identifies quantities, filter 5 picks out brands, and filter 20 focuses on food).

Finally, we discuss the tradeoff between accuracy and model speed/size. When deploying a real-world system for interacting with users, efficiency is critical. While BERT performs best, it has 110M parameters (1.1GB), and takes 8s per GPU to test on ATIS; the CNN is only 3.1M parameters (12MB), and takes 1.6s on ATIS with 1 CPU.

Table 6: *Top-10 activated tokens for ATIS CNN filters.*

Filter	Top-10 Highest Activation Tokens
11	12, 5, 230, 4, 7, to, 6, round, 10, 8
12	round, nonstop, us, american, northwest, delta, United, twa, daily, continental
22	phoenix, houston, pittsburgh, denver, detroit, milwaukee, cincinnati, chicago, charlotte, toronto

Table 7: *Top-10 activated tokens for restaurant CNN filters.*

Filter	Top-10 Highest Activation Tokens
6	brazilian, authentic, asia, italian, tex, mexican, sandwich, mex, mediterranean, spanish
56	till, until, after, open, at, past, before, a, is, every
35	4, highest, 3, 5, star, 1, starving, three, get, five

Table 8: *Top-10 activated tokens for meal CNN filters.*

Filter	Top-10 Highest Activation Tokens
34	8, 16, 14, 2, had, a, 6, an, ., 12
44	Coke, coke, Water, Mountain, Kraft, cheese, Eggs, Dr. Miracle, Mt.
20	chile, tuna, egg, crab, chicken, cottage, butter, oranges, cauliflower, coconut

7. Conclusion

In this paper, we have demonstrated that BERT outperforms prior state-of-the-art approaches on three spoken language understanding tasks: ATIS, restaurants, and spoken meal descriptions (with the exception of *written* meal descriptions, where a hand-crafted CRF slightly outperforms BERT). Our analysis of the trained CNN establishes that its filters are learning semantically meaningful categories related to the semantic tags, as well as predicting tags with lower confidence when making mistakes. In the future, we aim to incorporate a feedback mechanism into the dialouge system so that the model will ask for clarification when it is uncertain about the semantic tags, thus learning online from users.

8. References

[1] M. Korpusik, Z. Collins, and J. Glass, "Semantic mapping of natural language input to database entries via convolutional neural networks," *Proceedings of IEEE Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5685–5689, 2017.

[2] M. Korpusik, Z. Collins, and J. Glass, "Character-based embedding models and reranking strategies for understanding natural language meal descriptions," *Proceedings of Interspeech*, pp. 3320–3324, 2017.

[3] M. Korpusik and J. Glass, "Deep learning for database mapping and asking clarification questions in dialogue systems," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2019.

[4] M. Korpusik and Glass, "Convolutional neural networks for dialogue state tracking without pre-trained word vectors or semantic dictionaries," in *Proceedings of 2018 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2018, pp. 884–891.

[5] M. Korpusik and J. Glass, "Convolutional neural encoder for the 7th dialogue system technology challenge," in *Proceedings of 2019 AAAI 7th Dialogue System Technology Challenge Workshop (DSTC7)*. AAAI, 2019.

[6] M. Korpusik and Glass, "Dialogue state tracking with convolutional semantic taggers," in *Proceedings of IEEE Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019.

[7] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[8] Y. He and S. Young, "A data-driven spoken language understanding system," in *Proceedings of 2003 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*. IEEE, 2003, pp. 583–588.

[9] J. Liu, P. Pasupat, S. Cyphers, and J. Glass, "Asgard: A portable architecture for multilingual dialogue systems," in *Proceedings of IEEE Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2013, pp. 8386–8390.

[10] M. Korpusik, C. Huang, M. Price, and J. Glass, "Distributional semantics for understanding spoken meal descriptions," *Proceedings of 2016 IEEE Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6070–6074, 2016.

[11] M. Korpusik, N. Schmidt, J. Drexler, S. Cyphers, and J. Glass, "Data collection and language understanding of food descriptions," *Proceedings of 2014 IEEE Spoken Language Technology Workshop (SLT)*, pp. 560–565, 2014.

[12] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Vesely, "The kaldi speech recognition toolkit," in *IEEE 2011 workshop on automatic speech recognition and understanding (ASRU)*, Dec. 2011.

[13] C. Raymond and G. Riccardi, "Generative and discriminative algorithms for spoken language understanding," in *Proceedings of the Eighth International Conference on Spoken Language Processing (Interspeech)*, 2007, pp. 1605–1608.

[14] G. Tur, D. Hakkani-Tür, L. Heck, and S. Parthasarathy, "Sentence simplification for spoken language understanding," in *Proceedings of the 2011 IEEE Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2011, pp. 5628–5631.

[15] P. Xu and R. Sarikaya, "Convolutional neural network based triangular CRF for joint intent detection and slot filling," in *2013 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*. IEEE, 2013, pp. 78–83.

[16] 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 Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 3, pp. 530–539, 2015.

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

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

[19] M. Ma, K. Zhao, L. Huang, B. Xiang, and B. Zhou, "Jointly trained sequential labeling and classification by sparse attention neural networks," *arXiv preprint arXiv:1709.10191*, 2017.

[20] C. Goo, G. Gao, Y. Hsu, C. Huo, T. Chen, K. Hsu, and Y. Chen, "Slot-gated modeling for joint slot filling and intent prediction," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT), Volume 2 (Short Papers)*, vol. 2, 2018, pp. 753–757.

[21] J. Liu, P. Pasupat, Y. Wang, S. Cyphers, and J. Glass, "Query understanding enhanced by hierarchical parsing structures," in *Proceedings of 2013 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*. IEEE, 2013, pp. 72–77.

[22] K. Yao, B. Peng, G. Zweig, D. Yu, X. Li, and F. Gao, "Recurrent conditional random field for language understanding," in *Proceedings of 2014 IEEE Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2014, pp. 4077–4081.

[23] M. Korpusik and J. Glass, "Convolutional neural networks and multitask strategies for semantic mapping of natural language input to a structured database," in *Proceedings of IEEE Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 6174–6178.

[24] J. Pennington, R. Socher, and C. Manning, "GloVe: Global vectors for word representation," *Proceedings of 2014 Conference on Empirical Methods on Natural Language (EMNLP)*, vol. 12, 2014.

[25] T. Mikolov, K. Chen, and J. Dean, "word2vec (2013)."

[26] M. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," *arXiv preprint arXiv:1802.05365*, 2018.

[27] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving language understanding by generative pre-training," *URL https://s3-us-west-2. amazonaws. com/openai-assets/research-covers/languageunsupervised/language understanding paper. pdf*, 2018.

[28] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems (NIPS)*, 2017, pp. 5998–6008.

[29] M. Korpusik and J. Glass, "Spoken language understanding for a nutrition dialogue system," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 25, pp. 1450–1461, 2017.