Attention-Based Recurrent Neural Network Models for Joint Intent Detection and Slot Filling

Bing Liu[1], *Ian Lane*[1,2]

[1]Electrical and Computer Engineering, Carnegie Mellon University
[2]Language Technologies Institute, Carnegie Mellon University

liubing@cmu.edu, lane@cmu.edu

Abstract

Attention-based encoder-decoder neural network models have recently shown promising results in machine translation and speech recognition. In this work, we propose an attention-based neural network model for joint intent detection and slot filling, both of which are critical steps for many speech understanding and dialog systems. Unlike in machine translation and speech recognition, alignment is explicit in slot filling. We explore different strategies in incorporating this alignment information to the encoder-decoder framework. Learning from the attention mechanism in encoder-decoder model, we further propose introducing attention to the alignment-based RNN models. Such attentions provide additional information to the intent classification and slot label prediction. Our independent task models achieve state-of-the-art intent detection error rate and slot filling F1 score on the benchmark ATIS task. Our joint training model further obtains 0.56% absolute (23.8% relative) error reduction on intent detection and 0.23% absolute gain on slot filling over the independent task models.

Index Terms: Spoken Language Understanding, Slot Filling, Intent Detection, Recurrent Neural Networks, Attention Model

1. Introduction

Spoken language understanding (SLU) system is a critical component in spoken dialogue systems. SLU system typically involves identifying speaker's intent and extracting semantic constituents from the natural language query, two tasks that are often referred to as intent detection and slot filling.

Intent detection and slot filling are usually processed separately. Intent detection can be treated as a semantic utterance classification problem, and popular classifiers like support vector machines (SVMs) [1] and deep neural network methods [2] can be applied. Slot filling can be treated as a sequence labeling task. Popular approaches to solving sequence labeling problems include maximum entropy Markov models (MEMMs) [3], conditional random fields (CRFs) [4], and recurrent neural networks (RNNs) [5, 6, 7]. Joint model for intent detection and slot filling has also been proposed in literature [8, 9]. Such joint model simplifies the SLU system, as only one model needs to be trained and fine-tuned for the two tasks.

Recently, encoder-decoder neural network models have been successfully applied in many sequence learning problems such as machine translation [10] and speech recognition [11]. The main idea behind the encoder-decoder model is to encode input sequence into a dense vector, and then use this vector to generate corresponding output sequence. The attention mechanism introduced in [12] enables the encoder-decoder architecture to learn to align and decode simultaneously.

In this work, we investigate how an SLU model can benefit from the strong modeling capacity of the sequence models. Attention-based encoder-decoder model is capable of mapping sequences that are of different lengths when no alignment information is given. In slot filling, however, alignment is explicit, and thus alignment-based RNN models typically work well. We would like to investigate the combination of the attention-based and alignment-based methods. Specifically, we want to explore how the alignment information in slot filling can be best utilized in the encoder-decoder models, and on the other hand, whether the alignment-based RNN slot filling models can be further improved with the attention mechanism that introduced from the encoder-decoder architecture. Moreover, we want to investigate how slot filling and intent detection can be jointly modeled under such schemes.

The remainder of the paper is organized as follows. In section 2, we introduce the background on using RNN for slot filling and using encoder-decoder models for sequence learning. In section 3, we describe two approaches for jointly modeling intent and slot filling. Section 4 discusses the experiment setup and results on ATIS benchmarking task. Section 5 concludes the work.

2. Background

2.1. RNN for Slot Filling

Slot filling can be treated as a sequence labeling problem, where we have training examples of $\left\{(\mathbf{x}^{(n)}, \mathbf{y}^{(n)}) : n = 1, ..., N\right\}$ and we want to learn a function $f : \mathcal{X} \rightarrow \mathcal{Y}$ that maps an input sequence $\mathbf{x}$ to the corresponding label sequence $\mathbf{y}$. In slot filling, the input sequence and label sequence are of the same length, and thus there is explicit alignment.

Sentence	first	class	fares	from	boston	to	denver
Slots	B-class_type	I-class_type	O	O	B-fromloc	O	B-toloc
Intent	airfare						

Figure 1: *ATIS corpus sample with intent and slot annotation.*

RNNs have been widely used in many sequence modeling problems [6, 13]. At each time step of slot filling, RNN reads a word as input and predicts its corresponding slot label considering all available information from the input and the emitted output sequences. The model is trained to find the best parameter set θ that maximizes the likelihood:

$$\arg\max_{\theta} \prod_{t=1}^{T} P(y_t | y_1^{t-1}, \mathbf{x}; \theta) \tag{1}$$

where $\mathbf{x}$ represents the input word sequence, y_1^{t-1} represents the output label sequence prior to time step t. During inference, we want to find the best label sequence $\mathbf{y}$ given an input sequence $\mathbf{x}$ such that:

$$\hat{\mathbf{y}} = \arg\max_{\mathbf{y}} P(\mathbf{y}|\mathbf{x}) \tag{2}$$

2.2. RNN Encoder-Decoder

The RNN encoder-decoder framework is firstly introduced in [10] and [14]. The encoder and decoder are two separate RNNs. The encoder reads a sequence of input $(x_1, ..., x_T)$ to a vector c. This vector encodes information of the whole source sequence, and is used in decoder to generate the target output sequence. The decoder defines the probability of the output sequence as:

$$P(\mathbf{y}) = \prod_{t=1}^{T} P(y_t|y_1^{t-1}, c) \tag{3}$$

where y_1^{t-1} represents the predicted output sequence prior to time step t. Comparing to an RNN model for sequence labeling, the RNN encoder-decoder model is capable of mapping sequence to sequence with different lengths. There is no explicit alignment between source and target sequences. The attention mechanism later introduced in [12] enables the encoder-decoder model to learn a soft alignment and to decode at the same time.

3. Proposed Methods

In this section, we first describe our approach on integrating alignment information to the encoder-decoder architecture for slot filling and intent detection. Following that, we describe the proposed method on introducing attention mechanism from the encoder-decoder architecture to the alignment-based RNN models.

3.1. Encoder-Decoder Model with Aligned Inputs

The encoder-decoder model for joint intent detection and slot filling is illustrated in Figure 2. On encoder side, we use a bidirectional RNN. Bidirectional RNN has been successfully applied in speech recognition [15] and spoken language understanding [6]. We use LSTM [16] as the basic recurrent network unit for its ability to better model long-term dependencies comparing to simple RNN.

In slot filling, we want to map a word sequence $\mathbf{x} = (x_1, ..., x_T)$ to its corresponding slot label sequence $\mathbf{y} = (y_1, ..., y_T)$. The bidirectional RNN encoder reads the source word sequence forward and backward. The forward RNN reads the word sequence in its original order and generates a hidden state fh_i at each time step. Similarly, the backward RNN reads the word sequence in its reverse order and generate a sequence of hidden states $(bh_T, ..., bh_1)$. The final encoder hidden state h_i at each time step i is a concatenation of the forward state fh_i and backward state bh_i, i.e. $h_i = [fh_i, bh_i]$.

The last state of the forward and backward encoder RNN carries information of the entire source sequence. We use the last state of the backward encoder RNN to compute the initial decoder hidden state following the approach in [12]. The decoder is a unidirectional RNN. Again, we use an LSTM cell as the basic RNN unit. At each decoding step i, the decoder state s_i is calculated as a function of the previous decoder state s_{i-1}, the previous emitted label y_{i-1}, the aligned encoder hidden state h_i, and the context vector c_i:

$$s_i = f(s_{i-1}, y_{i-1}, h_i, c_i) \tag{4}$$

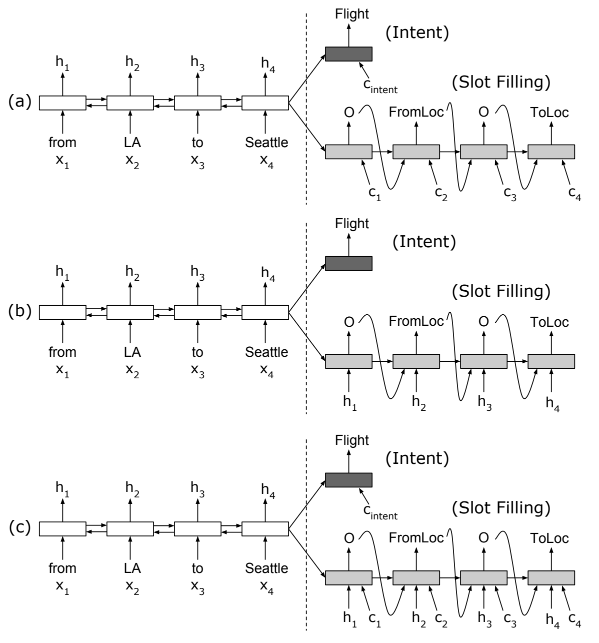

Figure 2: *Encoder-decoder model for joint intent detection and slot filling. (a) with no aligned inputs. (b) with aligned inputs. (c) with aligned inputs and attention. Encoder is a bidirectional RNN. The last hidden state of the backward encoder RNN is used to initialize the decoder RNN state.*

where the context vector c_i is computed as a weighted sum of the encoder states $\mathbf{h} = (h_1, ..., h_T)$ [12]:

$$c_i = \sum_{j=1}^{T} \alpha_{i,j} h_j \tag{5}$$

and

$$\begin{aligned} \alpha_{i,j} &= \frac{\exp(e_{i,j})}{\sum_{k=1}^{T} \exp(e_{i,k})} \\ e_{i,k} &= g(s_{i-1}, h_k) \end{aligned} \tag{6}$$

g a feed-forward neural network. At each decoding step, the explicit aligned input is the encoder state h_i. The context vector c_i provides additional information to the decoder and can be seen as a continuous bag of weighted features $(h_1, ..., h_T)$.

For joint modeling of intent detection and slot filling, we add an additional decoder for intent detection (or intent classification) task that shares the same encoder with slot filling decoder. During model training, costs from both decoders are back-propagated to the encoder. The intent decoder generates only one single output which is the intent class distribution of the sentence, and thus alignment is not required. The intent decoder state is a function of the shared initial decoder state s_0, which encodes information of the entire source sequence, and the context vector c_{intent}, which indicates part of the source sequence that the intent decoder pays attention to.

3.2. Attention-Based RNN Model

The attention-based RNN model for joint intent detection and slot filling is illustrated in Figure 3. The idea of introducing attention to the alignment-based RNN sequence labeling model

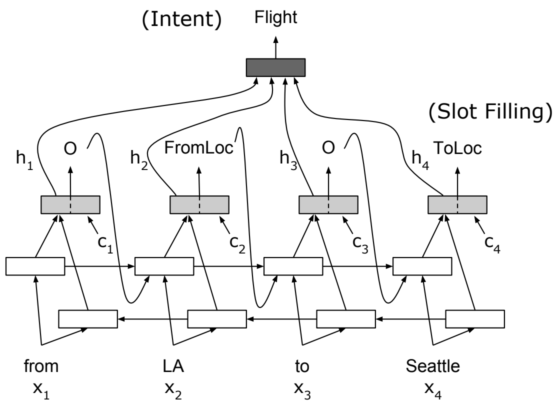

Figure 3: *Attention-based RNN model for joint intent detection and slot filling. The bidirectional RNN reads the source sequence forward and backward. Slot label dependency is modeled in the forward RNN. At each time step, the concatenated forward and backward hidden states is used to predict the slot label. If attention is enabled, the context vector c_i provides information from parts of the input sequence that is used together with the time aligned hidden state h_i for slot label prediction.*

is motivated by the use of attention mechanism in encoder-decoder models. In bidirectional RNN for sequence labeling, the hidden state at each time step carries information of the whole sequence, but information may gradually lose along the forward and backward propagation. Thus, when making slot label prediction, instead of only utilizing the aligned hidden state h_i at each step, we would like to see whether the use of context vector c_i gives us any additional supporting information, especially those require longer term dependencies that is not being fully captured by the hidden state.

In the proposed model, a bidirectional RNN (BiRNN) reads the source sequence in both forward and backward directions. We use LSTM cell for the basic RNN unit. Slot label dependencies are modeled in the forward RNN. Similar to the encoder module in the above described encoder-decoder architecture, the hidden state h_i at each step is a concatenation of the forward state fh_i and backward state bh_i, $h_i = [fh_i, bh_i]$. Each hidden state h_i contains information of the whole input word sequence, with strong focus on the parts surrounding the word at step i. This hidden state h_i is then combined with the context vector c_i to produce the label distribution, where the context vector c_i is calculated as a weighted average of the RNN hidden states $\mathbf{h} = (h_1, ..., h_T)$.

For joint modeling of intent detection and slot filling, we reuse the pre-computed hidden states $\mathbf{h}$ of the bidirectional RNN to produce intent class distribution. If attention is not used, we apply mean-pooling [17] over time on the hidden states $\mathbf{h}$ followed by logistic regression to perform the intent classification. If attention is enabled, we instead take the weighted average of the hidden states $\mathbf{h}$ over time.

Comparing to the attention-based encoder-decoder model that utilizes explicit aligned inputs, the attention-based RNN model is more computational efficient. During model training, the encoder-decoder slot filling model reads through the input sequence twice, while the attention-based RNN model reads through the input sequence only once.

4. Experiments

4.1. Data

ATIS (Airline Travel Information Systems) data set [18] is widely used in SLU research. The data set contains audio recordings of people making flight reservations. In this work, we follow the ATIS corpus[1] setup used in [6, 7, 9, 19]. The training set contains 4978 utterances from the ATIS-2 and ATIS-3 corpora, and the test set contains 893 utterances from the ATIS-3 NOV93 and DEC94 data sets. There are in total 127 distinct slot labels and 18 different intent types. We evaluate the system performance on slot filling using F1 score, and the performance on intent detection using classification error rate.

We obtained another ATIS text corpus that was used in [9] and [20] for SLU evaluation. This corpus contains 5138 utterances with both intent and slot labels annotated. In total there are 110 different slot labels and 21 intent types. We use the same 10-fold cross validation setup as in [9] and [20].

4.2. Training Procedure

LSTM cell is used as the basic RNN unit in the experiments. Our LSTM implementation follows the design in [21]. Given the size the data set, we set the number of units in LSTM cell as 128. The default forget gate bias is set to 1 [22]. We use only one layer of LSTM in the proposed models, and deeper models by stacking the LSTM layers are to be explored in future work.

Word embeddings of size 128 are randomly initialized and fine-tuned during mini-batch training with batch size of 16. Dropout rate 0.5 is applied to the non-recurrent connections [21] during model training for regularization. Maximum norm for gradient clipping is set to 5. We use Adam optimization method following the suggested parameter setup in [23].

4.3. Independent Training Model Results: Slot Filling

We first report the results on our independent task training models. Table 1 shows the slot filling F1 scores using our proposed architectures. Table 2 compares our proposed model performance on slot filling to previously reported results.

Table 1: *Independent training model results on ATIS slot filling.*

Model	F1 Score	Average
(a) Encoder-decoder NN with no aligned inputs	81.64	79.66 ± 1.59
(b) Encoder-decoder NN with aligned inputs	95.72	95.38 ± 0.18
(c) Encoder-decoder NN with aligned inputs & attention	**95.78**	95.47 ± 0.22
BiRNN no attention	95.71	95.37 ± 0.19
BiRNN with attention	**95.75**	95.42 ± 0.18

In Table 1, the first set of results are for variations of encoder-decoder models described in section 3.1. Not to our surprise, the pure attention-based slot filling model that does not utilize explicit alignment information performs poorly. Letting the model to learn the alignment from training data does not seem to be appropriate for slot filling task. Line 2 and line 3 show the F1 scores of the non-attention and attention-based encode-decoder models that utilize the aligned inputs. The

[1]We thank Gokhan Tur and Puyang Xu for sharing the ATIS data set.

attention-based model gives slightly better F1 score than the non-attention-based one, on both the average and best scores. By investigating the attention learned by the model, we find that the attention weights are more likely to be evenly distributed across words in the source sequence. There are a few cases where we observe insightful attention (Figure 4) that the decoder pays to the input sequence, and that might partly explain the observed performance gain when attention is enabled.

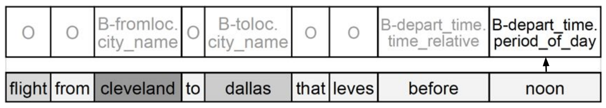

Figure 4: *Illustration of the inferred attention when predicting the slot label for the last word "noon" in the given sentence. Darker shades indicate higher attention weights. When word "noon" is fed to the model as the aligned input, the attention mechanism tries to find other supporting information from the input word sequence for the slot label prediction.*

The second set of results in Table 1 are for bidirectional RNN models described in section 3.2. Similar to the previous set of results, we observe slightly improved F1 score on the model that uses attentions. The contribution from the context vector for slot filling is not very obvious. It seems that for sequence length at such level (average sentence length is 11 for this ATIS corpus), the hidden state h_i that produced by the bidirectional RNN is capable of encoding most of the information that is needed to make the slot label prediction.

Table 2 compares our slot filling models to previous approaches. Results from both of our model architectures advance the best F1 scores reported previously.

Table 2: *Comparison to previous approaches. Independent training model results on ATIS slot filling.*

Model	F1 Score
CNN-CRF [9]	94.35
RNN with Label Sampling [7]	94.89
Hybrid RNN [6]	95.06
Deep LSTM [5]	95.08
RNN-EM [24]	95.25
Encoder-labeler Deep LSTM [25]	95.66
Attention Encoder-Decoder NN (with aligned inputs)	**95.78**
Attention BiRNN	**95.75**

4.4. Independent Training Model Results: Intent Detection

Table 3 compares intent classification error rate between our intent models and previous approaches. Intent error rate of our proposed models outperform the state-of-the-art results by a large margin. The attention-based encoder-decoder intent model advances the bidirectional RNN model. This might be attributed to the sequence level information passed from the encoder and additional layer of non-linearity in the decoder RNN.

4.5. Joint Model Results

Table 4 shows our joint training model performance on intent detection and slot filling comparing to previous reported results. As shown in this table, the joint training model using

Table 3: *Comparison to previous approaches. Independent training model results on ATIS intent detection.*

Model	Error (%)
Recursive NN [8]	4.60
Boosting [19]	4.38
Boosting + Simplified sentences [26]	3.02
Attention Encoder-Decoder NN	**2.02**
Attention BiRNN	**2.35**

encoder-decoder architecture achieves 0.09% absolute gain on slot filling and 0.45% absolute gain (22.2% relative improvement) on intent detection over the independent training model. For the attention-based bidirectional RNN architecture, the join training model achieves 0.23% absolute gain on slot filling and 0.56% absolute gain (23.8% relative improvement) on intent detection over the independent training models. The attention-based RNN model seems to benefit more from the joint training. Results from both of our joint training approaches outperform the best reported joint modeling results.

Table 4: *Comparison to previous approaches. Joint training model results on ATIS slot filling and intent detection.*

Model	F1 Score	Intent Error (%)
RecNN [8]	93.22	4.60
RecNN+Viterbi [8]	93.96	4.60
Attention Encoder-Decoder NN (with aligned inputs)	**95.87**	**1.57**
Attention BiRNN	**95.98**	**1.79**

To further verify the performance of our joint training models, we apply the proposed models on the additional ATIS data set and evaluate them with 10-fold cross validation same as in [9] and [20]. Both the encoder-decoder and attention-based RNN methods achieve promising results.

Table 5: *Joint training model results on the additional ATIS corpus using 10-fold cross validation.*

Model	F1 Score	Intent Error (%)
TriCRF [20]	94.42	6.93
CNN TriCRF [9]	95.42	5.91
Attention Encoder-Decoder NN (with aligned inputs)	**95.62**	**5.86**
Attention BiRNN	**95.78**	**5.60**

5. Conclusions

In this paper, we explored strategies in utilizing explicit alignment information in the attention-based encoder-decoder neural network models. We further proposed an attention-based bidirectional RNN model for joint intent detection and slot filling. Using a joint model for the two SLU tasks simplifies the dialog system, as only one model needs to be trained and deployed. Our independent training models achieved state-of-the-art performance for both intent detection and slot filling on the benchmark ATIS task. The proposed joint training models improved the intent detection accuracy and slot filling F1 score further over the independent training models.

6. References

[1] P. Haffner, G. Tur, and J. H. Wright, "Optimizing svms for complex call classification," in *Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03). 2003 IEEE International Conference on*, vol. 1. IEEE, 2003, pp. I–632.

[2] R. Sarikaya, G. E. Hinton, and B. Ramabhadran, "Deep belief nets for natural language call-routing," in *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*. IEEE, 2011, pp. 5680–5683.

[3] A. McCallum, D. Freitag, and F. C. Pereira, "Maximum entropy markov models for information extraction and segmentation." in *ICML*, vol. 17, 2000, pp. 591–598.

[4] C. Raymond and G. Riccardi, "Generative and discriminative algorithms for spoken language understanding." in *INTERSPEECH*, 2007, pp. 1605–1608.

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

[6] 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," *Audio, Speech, and Language Processing, IEEE/ACM Transactions on*, vol. 23, no. 3, pp. 530–539, 2015.

[7] 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.

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

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

[10] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Advances in neural information processing systems*, 2014, pp. 3104–3112.

[11] W. Chan, N. Jaitly, Q. V. Le, and O. Vinyals, "Listen, attend and spell," *arXiv preprint arXiv:1508.01211*, 2015.

[12] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[13] T. Mikolov, S. Kombrink, L. Burget, J. H. Černockỳ, and S. Khudanpur, "Extensions of recurrent neural network language model," in *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*. IEEE, 2011, pp. 5528–5531.

[14] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," *arXiv preprint arXiv:1406.1078*, 2014.

[15] A. Graves, N. Jaitly, and A.-r. Mohamed, "Hybrid speech recognition with deep bidirectional lstm," in *Automatic Speech Recognition and Understanding (ASRU), 2013 IEEE Workshop on*. IEEE, 2013, pp. 273–278.

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

[17] X. Zhang, J. Zhao, and Y. LeCun, "Character-level convolutional networks for text classification," in *Advances in Neural Information Processing Systems*, 2015, pp. 649–657.

[18] C. T. Hemphill, J. J. Godfrey, and G. R. Doddington, "The atis spoken language systems pilot corpus," in *Proceedings, DARPA speech and natural language workshop*, 1990, pp. 96–101.

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

[20] M. Jeong and G. Geunbae Lee, "Triangular-chain conditional random fields," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 16, no. 7, pp. 1287–1302, 2008.

[21] W. Zaremba, I. Sutskever, and O. Vinyals, "Recurrent neural network regularization," *arXiv preprint arXiv:1409.2329*, 2014.

[22] R. Jozefowicz, W. Zaremba, and I. Sutskever, "An empirical exploration of recurrent network architectures," in *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, 2015, pp. 2342–2350.

[23] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[24] B. Peng and K. Yao, "Recurrent neural networks with external memory for language understanding," *arXiv preprint arXiv:1506.00195*, 2015.

[25] G. Kurata, B. Xiang, B. Zhou, and M. Yu, "Leveraging sentence-level information with encoder lstm for natural language understanding," *arXiv preprint arXiv:1601.01530*, 2016.

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