Interspeech 2018
2-6 September 2018, Hyderabad

Slot Filling with Delexicalized Sentence Generation

Youhyun Shin, Kang Min Yoo, Sang-goo Lee

Department of Computer Science and Engineering, Seoul National University, Seoul, Korea
`{shinu89,kangminyoo,sglee}@europa.snu.ac.kr`

Abstract

We introduce a novel approach that jointly learns slot filling and delexicalized sentence generation. There have been recent attempts to tackle slot filling as a type of sequence labeling problem, with encoder-decoder attention framework. We further improve the framework by training the model to generate delexicalized sentences, in which words according to slot values are replaced with slot labels. Slot filling with delexicalization shows better results compared to models having a single learning objective of filling slots. The proposed method achieves state-of-the-art slot filling performance on ATIS dataset. We experiment different variants of our model and find that delexicalization encourages generalization by sharing weights among the words with same labels and helps the model to further leverage certain linguistic features.

Index Terms: spoken language understanding, slot filling, delexicalization, encoder-decoder, attention model

1. Introduction

The main components of Spoken Language Understanding (SLU) are intention classification and slot filling. We focus on the latter, slot filling, where semantic concepts that we call *slot* embedded in the sentences are extracted. The slot filling task can be regarded as a sequence labeling problem where IOB-based (In-Out-Begin) labels are to be predicted for a given set of sentences with word alignment. As shown in Figure 1, we take input X and predict Y, where X={show, flights, from, baltimore, to, dallas} and Y={O, O, O, B-FromCity, O, B-ToCity}. There have been numerous approaches to the slot filling problem, such as maximum entropy Markov models (MEMMs) [1], CRF [2, 3], and neural network models such as CNN [4] and RNN [5, 6, 7, 8].

More recently, the encoder-decoder framework [9, 10], which are known to perform well in machine translation, have been applied to slot filling, achieving state-of-the-art performances [11, 12, 13, 14]. When the framework is trained end-to-end, the encoder learns to encode sentence-level information [11], and the decoder learns to generate sequences with information extracted by the encoder. However, the framework does not enforce perfect alignment between the input sentence and the generated labels. To overcome this, [12, 13] propose an improved framework where the encoder's hidden states are fed to the decoder and the decoder decodes as long as there are aligned encoder's hidden states to be passed.

In a related theme, there have been a number of works to improve performances of natural language processing (NLP) tasks by jointly predicting multiple NLP tasks [15, 16, 17]. These studies show promising results in leveraging common NLP features across different tasks to improve individual performances. However, [15] shows that slot filling performance degrade when jointly trained with intention classification and language modeling. This is somewhat counter-intuitive since it seems that jointly predicting language features and slot labels should improve each task due to information sharing. We explore the possibility of improving slot filling performance through joint learning with linguistic features.

Sentences

X_1	show	flights	from	baltimore	to	dallas
X_2	show	flights	from	philadelphia	to	boston

Slot Filling

Y	O	O	O	B-FromCity	O	B-ToCity

Delexicalization

Y^{word}	show	flights	from	B-FromCity	to	B-ToCity

Figure 1: *Slot filling and delexicalization example.*

Generally, different words that correspond to the same slot play a similar semantic and syntactic role in the sentence. If the words that make up slot values in the sentence are replaced by the corresponding slot labels, this delexicalized sentence becomes an abstraction for all possible slot value with respect to the slot label. For example, in Figure 1, the two sentences "I want to fly from ***baltimore*** to ***dallas***" and "I want to fly from from ***philadelphia*** to ***boston***" share semantically and syntactically identical words ***baltimore*** and ***philadelphia***. These words can be replaced with "***B-FromCity***", allowing the slot filling model to more easily learn common words that appear nearby the slot label.

In this paper, we introduce a novel model that jointly learns to generate delexicalized sentences and predict slot labels. We base our method on the attentive encoder-decoder framework with input alignment method that has been applied to the sequence labeling problem [12]. Our model sequentially encodes words of a sentence, and using the encoded representation the model simultaneously decodes slot labels, delexicalized input sentence, and binary entity classification. We use ATIS and MIT Corpus[1] as the benchmark dataset for the slot filling task, and show that the proposed model outperforms previous models.

2. Related Work

2.1. Encoder-decoder Attention Framework

In recent years, an array of problems have been tackled by the encoder-decoder attention framework. However, there is a problem when applying the framework to word-by-word alignment problem between input and output. Because there is no explicit

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

10.21437/Interspeech.2018-1808

alignment between input and output. To solve this problem, [12, 13] gives encoder's hidden state at time step i to an input of decoder at time step i. [12] and [13] show F1-scores 95.78% and 95.79%, respectively. We try to solve the slot filling problem by using the same model as used in [12] and improve the model with generating delexicalized sentence.

2.2. Delexicalization

Delexicalized sentence is a sentence whereby each word corresponding to a slot entity has been replaced by a symbol (e.g., slot label). [18] introduces a delexicalized RNN to improve the generality of the training data for dialog state tracking. [19] uses delexicalized sentences in learning dialogue response generation according to dialog act. Previous works [18, 19] use delexicalized sentence as a model input. Moreover, they already know the information for delexicalization such as which words are slot values in a sentence and their slot labels. However, we use delexicalized sentence as one of our model's output to be predicted. We aim to generate delexicalized sentence from raw sentence where there is no given information for delexicalization. In other words, we make our delexicalized sentence generator to learn identifying slot values and appropriate slot labels to be replaced.

2.3. Joint Learning

There are many studies on joint learning that various tasks are learned by sharing model parameters. Joint learning improves the performance of individual task by leveraging common information from each task when learning jointly. [16, 17] describes a model to learn linguistic resources such as part-of-speech tagging, named entity recognition, or parse tree with machine translation by sharing model parameters. In this paper, we propose a model that learns delexicalized sentence generation and slot filling using the same model which shares parameters. We assume that joint slot filling and delexicalization helps to learn each task because delexicalization encourage the capability of learning generalized sentence representation while slot filling clarifies the slot label to be replaced.

3. Proposed Model

3.1. Encoder-decoder Attention Framework

3.1.1. Encoder

In this paper, we use the encoder-decoder attention model with aligned input to apply it to the problem of sequence labeling with word-by-word alignment introduced in [12]. Input alignment is achieved by providing the hidden states of the encoder as input of the decoder for all time steps. Slot filling task takes input sentence $X = (x_1, ..., x_T)$ and maps X to output slot labels $Y = (y_1, ..., y_T)$. We use GRU as the basic RNN unit, to model function f. The encoder is Bi-GRU. Each word of the input sentence is expanded from embedding matrix to be fed in as the input value of the encoder. $\overrightarrow{h_i}$ is the hidden state of the GRU encoder after word embeddings have passed through the forward GRU f_f. $\overleftarrow{h_i}$ is the hidden state of the backward GRU f_b. The h_i delivered to the decoder is produced by concatenating $[\overrightarrow{h_i}, \overleftarrow{h_i}]$. There are several ways to initialize the decoder's hidden state. In [11, 12], the last state of the encoder is used as an initial state. Recently, max pooling is used in [20] to encode sentence representation. Experimental results showed that there was no significant difference between the two methods, but the max pooling method was marginally better.

$$h_i = [\overrightarrow{h_i}, \overleftarrow{h_i}] \tag{1}$$

$$\overrightarrow{h_i} = f_f(\overrightarrow{h_{i-1}}, x_i) \tag{2}$$

$$\overleftarrow{h_i} = f_b(\overleftarrow{h_{i+1}}, x_i) \tag{3}$$

3.1.2. Decoder

We use GRU decoder which receives a total of four inputs for each time step. Decoder at time i is s_i, which is the concatenation of s_{i-1} (previous decoder hidden state), y_{i-1} (previous slot label), c_i (attention vector), and h_i (hidden state of the encoder at time step i). g is a feed-forward neural network. $\alpha_{i,j}$ is a alignment probability between input x_j and output at time step i. $e_{i,k}$ implies the importance of the encoder hidden state h_k and previous decoder hidden state s_{i-1} when deciding next state s_i and generating output at time step i.

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

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

$$\alpha_{i,j} = \frac{exp(e_{i,j})}{\sum_{k=1}^{T} exp(e_{i,k})} \tag{6}$$

$$e_{i,k} = g(s_{i-1}, h_k) \tag{7}$$

In decoder, slot filling and delexicalized sentence generation share the same model parameters. The hidden state of the decoder GRU is passed through two MLP layers to predict each output values which are slot label y_i and delexicalized word y_i^{word} [2].

The prediction of the y values of the next step in each step is as follows:

$$P(y_i \mid y_{<i}; x) = SlotLabelDist(s_i) \tag{8}$$

$$P(y_i^{word} \mid y_{<i}; x) = WordDist(s_i) \tag{9}$$

We also explicitly learn about binary class value z to decide the word is replaced to slot label when generating delexicalized sentence. Also, in view of slot filling, z recognizes given input is slot entity or not. This z value is learned to have a value of 1 for an entity and 0 for others. We also predict z_i from the same hidden state of GRU decoder at time step i.

$$P(z_i \mid y_{<i}; x) = BinaryClassDist(s_i) \tag{10}$$

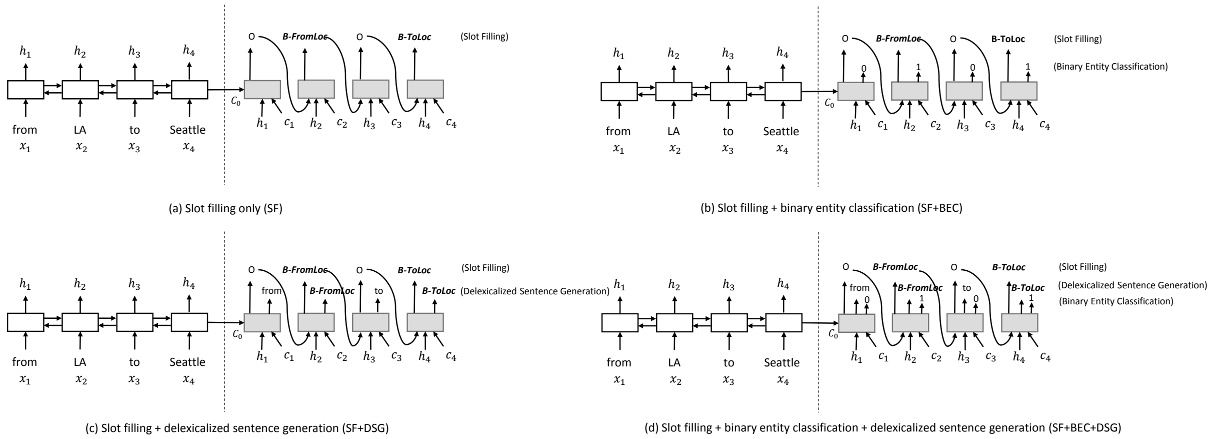
3.2. Training

The basic slot filling objective function is Eq. (11). The objective function of delexicalized sentence generation is Eq. (12).

$$\max_{\theta} \sum_{i=0}^{T} \log P(y_i \mid y_{<i}; x, \theta) \tag{11}$$

$$\max_{\theta} \sum_{i=0}^{T} \log P(y_i^{word} \mid y_{<i}; x, \theta) \tag{12}$$

[2] We do not train the model to deliver y_{i-1}^{word} to decoder hidden state s_i. Experimental results show that it performs worse (F1-score of 95.60). We guess that 1) y_{i-1}^{word} has the possibility to perform as noise because of the sparseness and 2) encoder hidden state h_i implies sufficient information to predict y_i^{word}.



h_1 h_2 h_3 h_4
from LA to Seattle
x_1 x_2 x_3 x_4
c_0
h_1 c_1 h_2 c_2 h_3 c_3 h_4 c_4
O B-FromLoc O B-ToLoc
(Slot Filling)
0 1 0 1
(Binary Entity Classification)
from B-FromLoc to B-ToLoc
(Delexicalized Sentence Generation)

(a) Slot filling only (SF)

(b) Slot filling + binary entity classification (SF+BEC)

(c) Slot filling + delexicalized sentence generation (SF+DSG)

(d) Slot filling + binary entity classification + delexicalized sentence generation (SF+BEC+DSG)

Figure 2: *Encoder-decoder attention model (a) with aligned inputs used in [12]. Joint learning for slot filling (b) with binary entity classification (c) with delexicalized sentence generation (d) with delexicalized sentence generation and binary entity classification. Sequence of input words is X={from, LA, to, Seattle} and sequence of output labels is Y={O, B-FromLoc, O, B-ToLoc}. Also, delexicalized words are Y^{word}={from, B-FromLoc, to, B-ToLoc} and binary classes are Z={0, 1, 0, 1}*

Therefore, we train the parameters θ for joint model that maximize the objective function as follows:

$$\max_{\theta} \sum_{i=0}^{T} [\alpha_s \log P(y_i \mid y_{<i}; x, \theta) + \alpha_w \log P(y_i^{word} \mid y_{<i}; x, \theta)] \quad (13)$$

where α_s and α_w are the weights trained by z_i.

4. Experiments

4.1. Experimental Setup

We experimented with two different datasets that are mainly used for slot filling performance evaluation. First, we use ATIS (Airline Travel Information Systems) dataset [21] which is a representative benchmark dataset in SLU. Training set consists of 4978 sentences, and test set consists of 893 sentences. Our dataset settings are identical to those in [4, 12, 22], and our experimental setups to those in [4, 22] (90% as the train, and 10% as the validation data). The number of slot labels are 127.

As in [11] and [14], we also use a combination of ATIS and MIT Corpus dataset. It is a large dataset with 3 different domains and various queries. We call this dataset "LARGE" as called in [14]. The MIT dataset consists of the MIT Restaurant Corpus, which has queries related to restaurant search and reservation, and the MIT Movie Corpus, composed of simplex and complex queries related to movies. LARGE dataset has 30,229 training sentences and 6,810 test sentences. Similar to previous works [11, 14], we use 80% for train and 20% for development. There are 191 slot labels. For the LARGE dataset, we replace rare words (i.e., frequency less than 3) with *UNK*. We employ IOB (In-Out-Begin) scheme for slot labels. The F1-score was calculated using the *conlleval script*[3].

Each parameter was initialized by sampling from standard distribution. Dimensionality of word embeddings is 100. We use GRU as the recurrent unit. The hidden dimension of the GRU is 128. Mini-batch size is 16 and dropout rate is 0.5 for both recurrent and fully connected layers. Gradient clipping is set to 1, and the optimization method is adam [23].

Table 1: *Slot filling results on ATIS dataset. Left most column references models in Figure 2. We use shorthands SF, BEC and DSG to denote slot filling, binary entity classification and delexicalized sentence generation, respectively. Reported F1-score is the best case.*

Variants	**Methods**	**F1-score**	**Average**$(\pm std)$
(a)	SF	95.82	$95.65(\pm0.11)$
(b)	SF+BEC	95.86	$95.72(\pm0.07)$
(c)	SF+DSG	95.96	$95.76(\pm0.11)$
(d)	SF+BEC+DSG	96.08	$95.92(\pm0.09)$

4.2. Results on ATIS dataset

We assume that there are three similar tasks and suggest to learn these tasks jointly:

- **Binary entity classification**, to determine whether the word is a slot value or not;
- **Slot filling**, to identify the slot labels for the word sequences;
- **Sentence generation**, to create a delexicalized sentence.

. To compare how varying joint tasks in addition to slot label prediction affect the performance, we compare several variants of our models (Figure 2). Variant (a) is the baseline (slot filling only). [12] reports slot filling performance of (a) as 95.78% and we achieved a 95.82%, similarly reproducing with our experimental settings. We guess that our result is slightly better because we use max pooling compared with using last hidden state of the encoder when extracting the context from the encoders hidden states. Variant (b) predicts whether the word in the current time step is a slot entity z in addition to the task in (a). At each time step, $z = 1$ if the word is a slot entity and $z = 0$ otherwise. Variant (c) predicts language, but the target sentence has been delexicalized at words to which slot

[3] https://www.clips.uantwerpen.be/conll2000/chunking/conlleval.txt

Table 2: *Slot filling performance comparison with published results on ATIS dataset.*

Methods	**F1-score**
RNN [5]	94.11
CNN-CRF [4]	94.35
Joint SLU-LM [15]	94.64
Bi-RNN [6]	94.73
LSTM [7]	94.85
Hybrid RNN [6]	95.06
Deep LSTM [7]	95.08
Bi-LSTM	95.23
Encoder-labeler Deep LSTM(W) [11]	95.66
Attention Encoder-decoder NN (with aligned inputs) [12]	95.78
BiLSTM-LSTM (focus) [13]	95.79
Model III [14]	95.86
Slot filling with delexicalization (Ours)	**96.08**

labels correspond. The size of the vocabulary set used in delexicalization is smaller than the word vocabulary because words corresponding to slot labels are replaced as its label. Variant (d) is identical to variant (c), except that it additionally predicts z. Our best model is variant (d).

The results are shown in Table 1. The results of slot filling only (a) is 95.82%. If we jointly learn slot filling with binary entity classification (b), there is no significant improvement which is 0.04%. However, based on our investigation, binary entity classification helps to distinguish label *O* from other labels as its purpose. When the delexicalized sentence generation (c) is added as a secondary objective function, there is an absolute value improvement of 0.14. Finally, we get an absolute value improvement of 0.26 when the model learns all of the three jointly (d). In case of variant (d), its superior performance (96.08%) seems to suggest that the addition of binary classification of z to the prediction tasks helps the model to leverage common knowledge between delexicalization and slot label prediction. From these results, we find that the following two tasks are beneficial to joint slot prediction: 1) predicting whether the current word is a slot entity and 2) determining the slot type given context.

Delexicalized sentence generation is a task to predict language while replacing words which are slot value to corresponding slot label. It allows to learn lexical patterns (or contexts) nearby the slot labels (e.g., show me flights from *fromloc.city_name* to *toloc.city_name*). For example, our model correctly labels the phrase *san francisco* with *fromloc.city_name* for the sentence “show me flights from $[san\ francisco]_{fromloc.city\_name}$ to baltimore”. Moreover, the model does a better job in correctly labeling the same word appearing in different contexts compared to the baseline. For example, while the baseline predicts a sentence in the following manner, “how far is $[san francisco]_{fromloc.city\_name}$ international from downtown”, our model correctly predicts the label *airport_name* for the sentence as “how far is $[$*san francisco international*$]_{airport\_name}$ from downtown”.

In Table 2, we report that the best of our model outperforms the baseline (slot filling only) [12] which is based on encoder-decoder framework with encoder’s hidden states attention. The result is 0.22% higher than the state-of-the-art model [14], which leverages segmentation result for labeling.

4.3. Results on LARGE dataset

For the LARGE dataset, the reported F1-scores are 74.41% in [11] and 78.49% in [14], respectively. [11] uses the encoder-decoder model to encode sentence-level representation and decode labels for the slot value with the encoded information. [14] also uses encoder-decoder, but they first find phrases and then do labeling. The result of our slot filling with delexicalization method is 76.21%, which is better than 74.41% [11] and the baseline 75.90% (slot filling only). However, it is worse than the result 78.49% [14]. One of frequently observed error cases is segmentation error. For example, for the gold sentence “movies are made with $[$*video game*$]_{PLOT}$ plot”, our model labels it as “movies are made with $[$*video game plot*$]_{PLOT}$” including the word *plot* in label. Another example is “whats the movie with the trailer that has a $[teenage girl flashing a crowd]_{PLOT}$ is a gold result, but the result from our model is “whats the movie with the trailer that has $[a\ teenage\ girl\ flashing\ a\ crowd]_{PLOT}$” including the word *a* in label. In MIT Corpus, there are slot labels which have the word length over 3. Especially, the average word length for label *Plot* is 10.602 and *Quote* is 7.484. We may consider adding a segmentation phase to cover these long length labels in the future.

5. Conclusion

In this paper, we propose a novel approach based on encoder-decoder framework with input alignment that jointly learns slot filling and delexicalized sentence generation. Delexicalized sentence generation is a task that generates sentence that is equivalent to a given input sentence but replaces words which are slot value to corresponding slot label. The best result obtained when the model learns slot filling, delexicalized sentence generation and binary entity classification simultaneously. Binary entity classification is a task that distinguishes words other than slot entities and slot entities, which is implicit common subtask in both slot filling and delexicalized sentence generation. As a result, slot filling with delexicalization helps to learn common knowledge from tasks jointly learned based on their similarities. Our results on ATIS dataset show that training the model with delexicalized sentences improves slot prediction performance, outperforming previous state-of-the-art models. We conjecture that our model learns frequently appearing patterns in a sentence. Moreover, it is able to predict slot labels more precisely for cases where a word has multiple labels depending on its context. However, our result on the LARGE dataset, which have labels of length 3 or more, is worse than the model first identify scope of the chunk and then labeling. Still, our model performs better than baseline (slot filling only) on this dataset. In the future, we intend to explore joint training opportunities with other tasks such as phrase segmentation.

6. Acknowledgements

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (NRF2016M3C4A7952587).

7. References

[1] 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.

[2] C. Raymond and G. Riccardi, “Generative and discriminative al-

gorithms for spoken language understanding," in *Eighth Annual Conference of the International Speech Communication Association*, 2007.

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

[4] 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.

[5] K. Yao, G. Zweig, M.-Y. Hwang, Y. Shi, and D. Yu, "Recurrent neural networks for language understanding." in *Interspeech*, 2013, pp. 2524–2528.

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

[7] 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.

[8] 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.

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

[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] G. Kurata, B. Xiang, B. Zhou, and M. Yu, "Leveraging sentence-level information with encoder lstm for semantic slot filling," in *In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 2016, pp. 2077–2083.

[12] B. Liu and I. Lane, "Attention-based recurrent neural network models for joint intent detection and slot filling," in *InterSpeech*, 2016.

[13] S. Zhu and K. Yu, "Encoder-decoder with focus-mechanism for sequence labelling based spoken language understanding," in *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*. IEEE, 2017, pp. 5675–5679.

[14] F. Zhai, S. Potdar, B. Xiang, and B. Zhou, "Neural models for sequence chunking." in *AAAI*, 2017, pp. 3365–3371.

[15] B. Liu and I. Lane, "Joint online spoken language understanding and language modeling with recurrent neural networks," in *17th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, 2016, p. 22.

[16] J. Niehues and E. Cho, "Exploiting linguistic resources for neural machine translation using multi-task learning," *arXiv preprint arXiv:1708.00993*, 2017.

[17] M.-T. Luong, Q. V. Le, I. Sutskever, O. Vinyals, and L. Kaiser, "Multi-task sequence to sequence learning," *arXiv preprint arXiv:1511.06114*, 2015.

[18] M. Henderson, B. Thomson, and S. Young, "Robust dialog state tracking using delexicalised recurrent neural networks and unsupervised adaptation," in *Spoken Language Technology Workshop (SLT), 2014 IEEE*. IEEE, 2014, pp. 360–365.

[19] T.-H. Wen, M. Gasic, D. Kim, N. Mrksic, P.-H. Su, D. Vandyke, and S. Young, "Stochastic language generation in dialogue using recurrent neural networks with convolutional sentence reranking," *arXiv preprint arXiv:1508.01755*, 2015.

[20] A. Conneau, D. Kiela, H. Schwenk, L. Barrault, and A. Bordes, "Supervised learning of universal sentence representations from natural language inference data," *arXiv preprint arXiv:1705.02364*, 2017.

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

[22] X. Zhang and H. Wang, "A joint model of intent determination and slot filling for spoken language understanding." in *IJCAI*, 2016, pp. 2993–2999.

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