Attention Based Joint Model with Negative Sampling for New Slot Values Recognition

Mulan Hou, Xiaojie Wang, Caixia Yuan, Guohua Yang, Shuo Hu, and Yuanyuan Shi

Abstract Natural Language Understanding (NLU) is an important component of a task oriented dialogue system, which obtains slot values in user utterances. NLU module is often required to return standard slot values and recognize new slot values at the same time in many real world dialogue such as restaurant booking. Neither previous sequence labeling models nor classifiers can satisfy both requirements by themselves. To address the problem, the paper proposes an attention based joint model with negative sampling. It combines a sequence tagger with a classifier by an attention mechanism. The tagger helps in identifying slot values in raw texts and the classifier simultaneously maps them into standard slot values or the symbol of new values. Negative sampling is used for constructing negative samples of existing values to train the model. Experimental results on two datasets show that our model outperforms the previous methods. The negative samples contribute to new slot values identification, and the attention mechanism discovers important information and boosts the performance.

1 Introduction

Task oriented dialogue system, which has been widely used in a variety of different applications, is designed to accomplish a specific task through natural language interactions. One of its most important components is Natural Language Understanding(NLU). NLU aims at collecting information related to the task.

Semantic frames are commonly applied in NLU [11], each of which contains different slots. One of the goals of NLU is to fill in the slots with values extracted from

Mulan Hou, Xiaojie Wang, Caixia Yuan, Guohua Yang
Center of Intelligence Science and Technology, Bejing University of Posts and Telecommunications e-mail: {houmulan, xjwang, yuancx,yangguohua}@bupt.edu.cn

Shuo Hu, Yuanyuan Shi
Beijing Samsung Telecom R&D Center e-mail: {shuo.hu,yy.shi}@samsung.com

the user utterances. In previous work, sequence labeling models are usually used for slot values recognition. For example, Tur et al. [10] used Conditional Random Field (CRF) with domain-specific features for the task. With the success of deep neural networks, Kaisheng Yao et al. [14] proposed a RNN model with Named Entities(NER) as features. They also used Long Short-Term Memory (LSTM) [13] and some other deeper models. Ma et al. [4] combined Convolutional Neural Network (CNN), LSTM and CRF in a hierarchical way, where features extracted by a CNN are fed to a LSTM, a CRF in top level is used to label slot values.

Nevertheless, only the labeling of the slot values is not enough in some applications. The slot values labeled in utterances should be normalized to some standard values for database search. For example, in a restaurant booking system, there are standard values of slot 'food' like 'Asian oriented'. If a user wondered a restaurant which serves 'pan Asian' food, the system should normalize the 'pan Asian' in utterance into the standard value of 'Asian oriented' in database. There were two different ways for addressing this problem. One is two-stage methods. Lefvévre [3] proposed a 2+1 model. It used a generative model consisted of two parts, namely semantic prior model and lexicalization model, to determine the best semantic structure and then treated the normalized slot values as hidden variables to figure it out. Peter Z. Yeh [15] employed fuzzy matching in Apache Solr system for the normalization. Two-stage methods are either prone to accumulating errors or too complicated to compute. The other way is directly mapping an utterance to one of the standard values instead of identifying the values in raw texts. A lot of classifiers were used for building the mappings. Rahul Bhagat et al. [1] tried several different models including Vote model, Maximum Entropy, Support Vector Machine (SVM). Mairesse et al. [5] proposed a two-step method: a binary classifiers was first used to determine if a slot appears in the utterance or not, and then a series classifiers were used to map the utterance to standard values of that slot. Pedro Mota et al. [7] built different classifiers for different slots respectively.

There is an important problem in above classification based methods however. These models failed in dealing with the situation where a slot value out of the standard value set is mentioned in an utterance. This value should not be classified into any existing standard values and should be recognized as a new value. To our knowledge, there is no research on this problem in classification based NLU.

The problem might be thought as one type of zero-shot problems in word sense or text classification and others. But there is a significant difference between new slot values and other zero-shot problems. The sense of a new word might be very different from that of other known words. But a new slot value is still a value of the same slot. It should share some important similarities with other known slot values. That is the starting point for us to construct training samples for unknown new values. We first distinguish two different types of samples of the standard values of a specific slot S. Utterances including any known standard value or its variants of the slot S are positive samples, and the others are negative ones. We further divide the negative samples into two types, the first is negative samples of S, i.e. samples including values of other slots or including no value of any slot, and the second is negative samples of any known standard values of S. The latter is therefore can be used to build a classifier (together with positive samples of the standard values of S) for identifying if an utterance includes a known standard value or a new value of S. The paper proposes a negative sampling based method to construct samples of the latter.

Meanwhile, sequence labeling is able to locate slot values in original utterances even if they are unseen in standard value set. The slot values themselves are also important information for classification. The paper proposes a joint model of sequence labeling and classification by attention mechanism, which focuses on important information automatically and takes advantage of the raw texts at the same time. Sequence labeling here aims at slot-value detection and classification is used to obtain the standard values directly.

Overall, we propose an attention based joint model with negative sampling. Our contributions in this work are two-fold: (1) negative sampling for existing values for a certain slot S enables our model to effectively recognize new slot values; (2) joint model collaborated by attention mechanism promotes the performance. We evaluate our work on a public dataset DSTC and a dataset Service from an enterprise. All the results demonstrate that our model achieves impressive improvements on new slot values with less damage on other sub-datasets. The F1 score evaluated on new slot values raises up to 0.8621 in DSTC and 0.7759 in Service respectively.

This paper is organized as follows: Sect. 2 details on our attention based joint model with negative sampling. We explain experiment settings in Sect. 3, then evaluate and analyse our model in Sect. 4. In Sect. 5 we will conclude our work.

2 Attention Based Joint Model with Negative Sampling(AJM_NS)

We assume that slots are independent of each other so they can be handled separately. A vocabulary of values for slot S is defined as $R^S = \{S_{old}\} \bigcup \{NEW, NULL\}$, where $S_{old} = \{s_0, s_1, ...s_k\}$ refers to the set of standard values for which there is some labeled data in training set. NEW refers to a new slot value. It will be assigned to an utterance providing a new slot value for slot S which is outside S_{old}, and $NULL$ refers to no value in an utterance. For a user input x_i, the aim of the model is to map the x_i into one of values in R^S. Since there is no training data for a new slot value (if we have some training samples for a value, it belongs to S_{old}), classification based models on the dataset are unable to address the problem, while sequence taggers need another step to normalize the labels.

We describe our attention based joint model, followed by the negative sampling methods.

2.1 Attention based joint model

A sequence tagger and a classifier complement each other. A sequence tagger recognizes units of a slot value in an utterance, while a classifier map an utterance as a whole into a slot value. In order to benefit from both of them, we combine them into a joint model.

Specifically, we adopt the bi-directional LSTM [2] as a basic structure. The output of each timestep is used to output a slot tag by a softmax operation on a linear layer as shown in Eq. 1:

$$\hat{s}_t = \mathrm{softmax}(\mathbf{W}_s h_t + b_s) \tag{1}$$

$h_t = (\overrightarrow{h_t}, \overleftarrow{h_t})$ refers to the hidden state of time t by concatenating the hidden state in forward and backward direction. In each direction of LSTM, like in forward LSTM, hidden state $\overrightarrow{h_t}$ is a function of the current input and the inner memory, as defined in Eq. 2

$$\overrightarrow{h_t} = f(\overrightarrow{h_{t-1}}, w_t, \overrightarrow{C_{t-1}}) \tag{2}$$

where w_t denotes the input word at time t and $\overrightarrow{C_{t-1}}$ is the previous cell state. We compute function f using the LSTM cell architecture in [16]. So as on backward direction.

The hidden state of the last timestep T is used to output the class label according to Eq. 3:

$$\hat{y} = \mathrm{softmax}(\mathbf{W}_c h_T + b_c) \tag{3}$$

We further combine them by attention mechanism[13]. Fig. 1 illustrates the procedure.

Upon attention mechanism, the model automatically focuses on locations of important information and constructs a context vector H which is defined in Eq. 4.

$$H = \sum_t^T \alpha_t v_t \tag{4}$$

where $v_t = (e_t, h_t)$ concatenates word embeddings and hidden states of LSTM and α_t is defined in Eq. 5.

$$\alpha_t = \frac{\exp(q_t)}{\sum_k^T \exp(q_k)} \tag{5}$$

Our model computes q_t by an align function in Eq. 6 which is the same way as [9]:

$$q_t = (\tanh(\mathbf{W} v_t))^\top h_T \tag{6}$$

It is regarded as a similarity score of the utterance representation h_T and the information v_t of each timestep.

Finally we concatenate context vector H and the sentence embedding h_T, and feed it into a softmax layer as shown in Eq. 7 to predict the class label of standard slot values.

$$\hat{y} = \mathrm{softmax}(\mathbf{W}(H, h_T) + b) \tag{7}$$

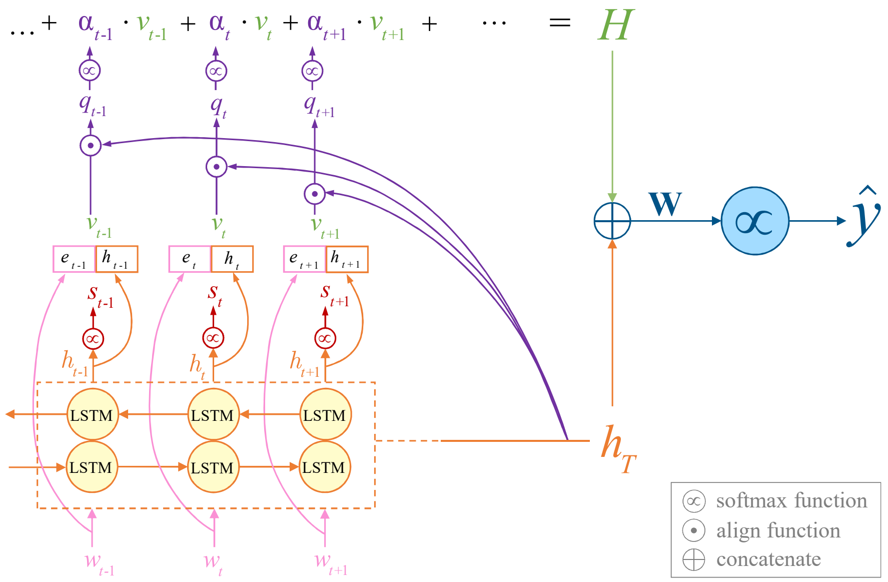

Fig. 1 In this figure, attention based joint model combines sequence tagging and classifying and adopts attention mechanism for further improvements. Legend in the right corner shows the meaning of operations.

All parameters are learned simultaneously to minimize a joint loss function shown in Eq. 8, i.e. the weighted sum of two losses for sequence tagging and classification respectively.

$$L = \gamma L_{tagging} + (1-\gamma) L_{classification} \tag{8}$$

$$L_{tagging} = \frac{1}{N}\sum_{i}^{N}\frac{1}{T_i}\sum_{t}^{T_i} L(\hat{s}_t^i, s_t^i) \tag{9}$$

$$L_{classification} = \frac{1}{N}\sum_{i}^{N} L(\hat{y}_i, y_i) \tag{10}$$

γ is a hyperparameter to balance the sequence tagging and classifying module. N in Eq. 9 refers to the size of training data and T_i is the length of the i-th input. $L(\cdot)$ is cross-entropy loss function.

2.2 Negative sampling

Model fails in recognizing new slot values without training data for them as mentioned before. If we regard all the samples for new slot values of a specific slot as

	我	的	手	机	可	以	边	打	电	话	边	视	频	吗	?
	O	O	O	O	O	O	B-func	I-func	I-func	I-func	I-func	I-func	I-func	O	O
negative sampling:	**手**	**速**	**录**	**支**	**无**	**用**	**间**								
	我	的	手	机	可	以	**手**	**速**	**录**	**支**	**无**	**用**	**间**	吗	?
	O	O	O	O	O	O	B-func	I-func	I-func	I-func	I-func	I-func	I-func	O	O

	Can	I	have	a	video	chat	on	my	phone	?
	O	O	B-func	I-func	I-func	I-func	O	O	O	O
negative sampling:	**You**	**use**	**battery**	**let**						
	Can	I	**You**	**use**	**battery**	**let**	on	my	phone	?
	O	O	B-func	I-func	I-func	I-func	O	O	O	O

Fig. 2 Negative sampling for Service dataset. Lower part is a translation of the example.

the negative samples of existing ones, construction of samples for new slot values can then convert to the construction of negative samples of old ones.

As mentioned in Sect. 1, a new slot value is still a value of the same slot. It should share some important similarities with other known slot values. Here we think the similarities are hidden in contexts of the value, i.e. the contexts are shared among different values of a same slot. It is therefore a possible way to construct a negative sample by just replacing the slot values in a positive sample with a non-value. But there are so many choices for non-value, how to choose a proper one?

Mikolov et al. [6] have already used negative sampling in CBOW and Skip-gram models. They investigated a number of choices for distribution of negative samples and found that the unigram distribution $U(word)$ raised to the $3/4$rd power (i.e., $U(word)^{3/4}/Z$) outperformed significantly the unigram and the uniform distributions. Z is the normalization constant and $U(word)$ is the word frequency in another word, which is calculated by $U(word) = count(word)/|Data|$. We use the same method but leave the word frequency alone. In our work a negative sample is a complete slot value that sometimes consists of several words, different from the negative samples of a single word in [6]. That results in repeating sampling until a segment of the same length as the existing value is formed. Fig. 2 shows the construction of a negative example for Service dataset.

3 Experiments Setting

3.1 Dataset

We evaluate our model on two dataset: Dialogue State Tracking Challenge (DSTC) and a dataset from an after-sale service dialogue system of an enterprise(Service).

DSTC is an English dataset from a public contest [12] and we use DSTC2 and DSTC3 together. It collects 5510 dialogues about hotels and restaurants booking. Each of the utterance in dialogues gives the standard slot values, according to which slot tags can be assigned to word sequence. Based on the independency assumption, we build datasets for each slot: keep all B- or I- tags of the slot labels and reset the rest to ’O’. However we find out that not all slots are suitable for the task, since there are too few value types of the slot. At last we choose the dataset for slot ’food’ only in our experiments.

Service is a Chinese dialogue dataset which is mainly about consultation for cell phones and contains a single slot named ’function’. It has both sequence tags and slot values on each utterance.

We divide two datasets into training, dev and test set respectively, and then construct some negative samples into training set for both of them. All of the utterances corresponding to infrequent slot values in training set are put into test set to form corpus of new slot values. These values thus have no samples in training data. Table 1 shows the statistics of the final experimental data and Table 2 tells about the diversity of slot values.

Table 1 Statistics of two dataset

Corpus		DSTC			Service		
		train	dev	test	train	dev	test
Original data	old	2805	937	917	3682	514	1063
	new	0	113	275	0	15	64
	null	2244	840	953	427	64	109
negative samples		561	0	0	736	0	0
overall size		5610	1890	2145	4845	593	1236

Table 2 Value types

Corpus	DSTC			Service		
	train	dev	test	train	dev	test
old	66	64	65	80	55	67
new	0	21	21	0	13	44

3.2 Evaluation measurements

We take weighted $F1$ score as the evaluation criterion in our experiments. It is defined as in Eq. 11 and Eq. 12.

$$F1 = \sum_{i}^{N} \omega_i F1_{s_i} \tag{11}$$

with

$$\omega_i = \frac{n_{s_i}}{n} \quad , \quad F1_{s_i} = 2\frac{P_{s_i} \times R_{s_i}}{P_{s_i} + R_{s_i}} \tag{12}$$

where n refers to the size of the test set and n_{s_i} denotes the size of class s_i. P and R is precision score and recall score defined in [8].

We also evaluate on the sub-dataset of old values by Eq. 13.

$$F1_{old} = \sum_{i=0}^{k} \omega_i^{old} F1_{s_i} \tag{13}$$

where $\omega_i^{old} = \frac{n_{s_i}}{n_{old}}$.

For sequence tagging we still consider F1 score as criterion which can be calculated by running the official script conlleval.pl[1] of CoNLL conference.

3.3 Baseline

There are no previous models and experimental results reported especially on new slot values recognition. We compare our model to existing two types of NLU methods for the task.

(1)The pipeline method:labeling the words with slot value tags first and then normalizing them into standard values. Here, a bi-directional LSTM as same as that in our model is used for tagging, and the fuzzy matching[2] is then used to normalize extracted tags like that in [15]. The model is denoted by LSTM_FM.

(2)The classification: classifying the utterance to standard values directly. A bi-directional LSTM is used to encode user input, a full-connected layer is then used for the classification. The model is denoted by LSTM_C.

3.4 Hyperparameters

We adopt bi-directional LSTM as the basic structure. Hyperparameter γ is 0.1. The longest input is 30, size of LSTM cell is 64, and dimension of word embedding is 100. We use minibatch stochastic gradient descent algorithm with Adam to update parameters. Learning rate is initialized as 0.005. Each batch keeps 512 pieces of training data. We choose the model performs best in dev set as the test one.

4 Result and Analyses

4.1 Comparisons among different models

We evaluate our model on two dataset described in Sect. 3.1. Our model can output both classification results of a utterance and the labeled tags in a utterance. Table 3 and Table 4 shows the results of classification and labeling respectively.

As we can see in Table 3, our model outperforms both baseline models significantly in classification task. It achieves 13.44% and 16.17% improvements compared to LSTM_FM and LSTM_C model on DSTC dataset, and achieves 8.55% and 5.85% improvements on Service dataset. Especially, it shows big advantage on new

[1] https://www.clips.uantwerpen.be/conll2000/chunking/output.html

[2] http://chairnerd.seatgeek.com/fuzzywuzzy-fuzzy-string-matching-in-python

Table 3 Classification results of different models

	DSTC				Service			
	all	NEW	S_{old}	NULL	all	NEW	S_{old}	NULL
LSTM_FM	0.8491	0.3063	0.9670	0.8923	0.8981	0.5693	0.9320	0.5693
LSTM_C	0.8290	0.0000	0.9249	0.9761	0.9210	0.0000	0.9720	**0.9643**
AJM_NS(ours)	**0.9632**	**0.8621**	**0.9738**	**0.9822**	**0.9749**	**0.7759**	**0.9881**	0.9633

Table 4 Tagging Results of different models

	DSTC			Service		
	all	NEW	S_{old}	all	NEW	S_{old}
LSTM_FM	0.8546	0.2363	0.9837	0.8850	0.2615	0.9269
LSTM_FM_NS	0.8289	0.2844	0.9709	0.8386	**0.4853**	0.8655
AJM_NS(ours)	**0.9024**	**0.5684**	**0.9946**	**0.9132**	0.3399	**0.9573**

slot values recognition, where the $F1$ scores achieve at least 20% raises on both DSTC and Service data.

Similar to the performance in the classification, our model also achieves best results in slot value tagging as we can see in Table 4. It performs significant better than the pipeline method, especially for the new value. We also give the tagging results of LSTM_FM trained by adding negative samples used in our model (denoted by LSTM_FM_NS in Table 4). We find negative samples are helpful to NEW slot values significantly, but they hurt the performance of old values. We give more details of negative samples and attention mechanism in our model and baseline models in next subsection.

4.2 Analyses

In order to analyze our model, we compare it to the model dropping out attention mechanism only and the other dropping negative samples only. We refer to the former model as JM_NS and the latter as AJM.

Table 5 Comparison inside the model with F1 scores for classification.

	DSTC				Service			
	all	NEW	S_{old}	NULL	all	NEW	S_{old}	NULL
Full(AJM_NS)	**0.9632**	**0.8621**	0.9738	**0.9822**	**0.9749**	**0.7759**	**0.9881**	**0.9633**
-Attention only(JM_NS)	0.9515	0.8129	**0.9739**	0.9699	0.9700	0.7207	0.9862	0.9585
-NS only(AJM)	0.8247	0.0000	0.9426	0.9492	0.9234	0.0000	0.9761	0.9511

From Table 5 we can find out that the one dropping out negative samples(AJM) failed in dealing with new slot values recognition. It shows that the negative sampling is the key for the success in new slot values recognition. The negative samples actually enables the model to distinguish old and new slot values. For more details, the changes of confusion matrices are shown in Table 6 and Table 7. The left part of '⇒' in the table is the confusion matrix of the model without negative samples(AJM), and the right part is from the original full model(AJM_NS). With the training of negative samples, classification results related to NEW value change better significantly, while change little on other classes, i.e. negative samples bring less damage to other classes.

Table 6 Confusion matrix of DSTC

DSTC								
	NEW	S_{old}	NULL			NEW	S_{old}	NULL
NEW	0	184	91	⇒	NEW	225	33	17
S_{old}	0	916	1		S_{old}	9	908	0
NULL	0	9	944		NULL	13	4	936

Table 7 Confusion matrix of Service

Service								
	NEW	S_{old}	NULL			NEW	S_{old}	NULL
NEW	0	55	9	⇒	NEW	45	17	2
S_{old}	0	1063	0		S_{old}	4	1057	2
NULL	0	2	107		NULL	3	1	105

We also add same negative samples for training other models. The result in Table 8 shows that LSTM_C_NS(LSTM_C with negative samples) now achieve good performance of recognizing new slot values. As for LSTM_FM_NS, the $F1$ score drops a lot for old values while for new slot values it raises up on the contrary. It shows that, although negative samples still work, they damage other classes significantly in pipeline model. We can also find out that our model AJM_NS still beats the rest models on the whole dataset even if all of them use negative samples.

When we abandon attention mechanism(JM_NS), the model is slightly inferior to the full one(AJM_NS), i.e. the attention mechanism can further improve the performance by focusing on the important subsequences. Since it introduces the original word embeddings at the same time, it corrects some mistakes in the model dropping out attention mechanism(JM_NS) in which the final label is wrongly classified even with correct sequence tags. We visualize a sample of attention in Fig. 3.

Table 8 Classification results based on negative samples

	DSTC				Service			
	all	NEW	S_{old}	NULL	all	NEW	S_{old}	NULL
LSTM_FM_NS	0.8572	0.3536	0.9286	0.9241	0.8642	0.6203	0.9009	0.6488
LSTM_C_NS	0.9543	0.8261	0.9637	**0.9822**	0.9684	0.7103	0.9825	**0.9815**
JM_NS	0.9515	0.8129	**0.9739**	0.9699	0.9700	0.7207	0.9862	0.9585
AJM_NS	**0.9632**	**0.8621**	0.9738	**0.9822**	**0.9749**	**0.7759**	**0.9881**	0.9633

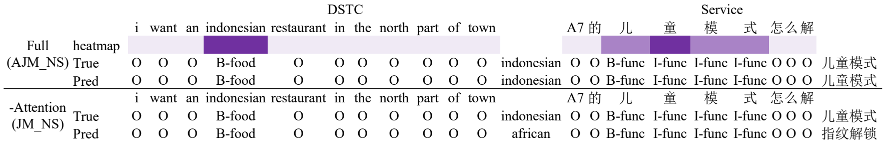

Fig. 3 Comparison between the full model(AJM_NS) and the one dropping out attention mechanism(JM_NS). The heatmap in full model is the visualization of weights for different words. The deeper color means a larger weight.

5 Conclusion

In lots of industrial or commercial applications, it is necessary for a NLU module to not only fill the slot with predefined standard values but also recognize new slot values due to the diversity of users linguistic habits and business update.

The paper proposes an attention based joint model with negative sampling to satisfy the requirement. The model combines a sequence tagger with a classifier by an attention mechanism. Negative sampling is used for constructing negative samples for training the model. Experimental results on two datasets show that our model outperforms the previous methods. The negative samples contributes to new slot values identification, and the attention mechanism improves the performance.

We may try different methods of negative sampling to further improve the performance in following works, such as introducing prior knowledge. At the same time, scenario of multiple slot in an utterance will also be explored as it happens a lot in daily life.

References

1. Rahul Bhagat, Anton Leuski, and Eduard Hovy. Statistical shallow semantic parsing despite little training data. In *Proceedings of the Ninth International Workshop on Parsing Technology*, pages 186–187. Association for Computational Linguistics, 2005.
2. Alex Graves, Navdeep Jaitly, and Abdel-rahman Mohamed. Hybrid speech recognition with deep bidirectional lstm. In *Automatic Speech Recognition and Understanding (ASRU), 2013 IEEE Workshop on*, pages 273–278. IEEE, 2013.

3. F Lefévre. Dynamic bayesian networks and discriminative classifiers for multi-stage semantic interpretation. In *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, volume 4, pages IV–13. IEEE, 2007.
4. Xuezhe Ma and Eduard Hovy. End-to-end sequence labeling via bi-directional lstm-cnns-crf. *arXiv preprint arXiv:1603.01354*, 2016.
5. François Mairesse, Milica Gasic, Filip Jurcícek, Simon Keizer, Blaise Thomson, Kai Yu, and Steve Young. Spoken language understanding from unaligned data using discriminative classification models. In *Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on*, pages 4749–4752. IEEE, 2009.
6. Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.
7. Ana Mendes Pedro Mota, Luísa Coheur. Natural language understanding as a classification process: report of initial experiments and results. In *INForum*, 2012.
8. James W Perry, Allen Kent, and Madeline M Berry. Machine literature searching x. machine language; factors underlying its design and development. *Journal of the Association for Information Science and Technology*, 6(4):242–254, 1955.
9. Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. Bidirectional attention flow for machine comprehension. *arXiv preprint arXiv:1611.01603*, 2016.
10. Gokhan Tur, Dilek Hakkani-Tür, Larry Heck, and Sarangarajan Parthasarathy. Sentence simplification for spoken language understanding. In *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, pages 5628–5631. IEEE, 2011.
11. Ye-Yi Wang, Li Deng, and Alex Acero. Semantic frame-based spoken language understanding. *Spoken Language Understanding: Systems for Extracting Semantic Information from Speech*, pages 41–91, 2011.
12. Jason Williams, Antoine Raux, Deepak Ramachandran, and Alan Black. The dialog state tracking challenge. In *Proceedings of the SIGDIAL 2013 Conference*, pages 404–413, 2013.
13. Kaisheng Yao, Baolin Peng, Yu Zhang, Dong Yu, Geoffrey Zweig, and Yangyang Shi. Spoken language understanding using long short-term memory neural networks. In *Spoken Language Technology Workshop (SLT), 2014 IEEE*, pages 189–194. IEEE, 2014.
14. Kaisheng Yao, Geoffrey Zweig, Mei-Yuh Hwang, Yangyang Shi, and Dong Yu. Recurrent neural networks for language understanding. In *Interspeech*, pages 2524–2528, 2013.
15. Peter Z Yeh, Benjamin Douglas, William Jarrold, Adwait Ratnaparkhi, Deepak Ramachandran, Peter F Patel-Schneider, Stephen Laverty, Nirvana Tikku, Sean Brown, and Jeremy Mendel. A speech-driven second screen application for tv program discovery. In *AAAI*, pages 3010–3016, 2014.
16. Wojciech Zaremba, Ilya Sutskever, and Oriol Vinyals. Recurrent neural network regularization. *arXiv preprint arXiv:1409.2329*, 2014.