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

Interpreting and Improving Deep Neural SLU Models via Vocabulary Importance

Yilin Shen[1], Wenhu Chen[2], Hongxia Jin[1]
[1]Samsung Research America, USA
[2]University of California, Santa Barbara, USA
[1]{yilin.shen,hongxia.jin}@samsung.com, [2]wenhuchen@cs.ucsb.edu

Abstract

Spoken language understanding (SLU) is a crucial component in virtual personal assistants. It consists of two main tasks: intent detection and slot filling. State-of-the-art deep neural SLU models have demonstrated good performance on benchmark datasets. However, these models suffer from the significant performance drop in practice after deployment due to the data distribution discrepancy between training and real user utterances. In this paper, we first propose four research questions that help to understand what the state-of-the-art deep neural SLU models actually learn. To answer them, we study the vocabulary importance using a novel *Embedding Sparse Structure Learning* (SparseEmb) approach. It can be applied onto various existing deep SLU models to efficiently prune the useless words without any additional manual hyperparameter tuning. We evaluate SparseEmb on benchmark datasets using two existing SLU models and answer the proposed research questions. Then, we utilize SparseEmb to sanitize the training data based on the selected useless words as well as the model re-validation during training. Using both benchmark and our collected testing data, we show that our sanitized training data helps to significantly improve the SLU model performance. Both SparseEmb and training data sanitization approaches can be applied onto any deep learning based SLU models.

1. Introduction

As a crucial component in emerging artificially intelligent personal assistants, spoken language understanding (SLU) system has attracted increasing research attentions. SLU aims to identify user's intent and extract semantic constituents from a natural language utterance, a.k.a. intent detection and slot filling. Existing approaches include the independent models for learning intent detection [1, 2] and slot filling [3, 4, 5, 6, 7, 8] separately as well as joint models to learn these two tasks together [9, 10, 11, 2, 12, 13].

Unfortunately, these state-of-the-art SLU models usually fail to achieve desirable performance in practice. Since the training of a SLU model requires an annotated corpus that is very expensive to collect, it is infeasible to pre-collect a corpus that covers all varieties of utterances. More importantly, each user oftentimes has his own personalized expression of intents, especially in spoken languages. As a result, the data distributions are very different between the pre-collected training corpus and real-world user utterances. Thus, such pretrained SLU models suffer from significant performance drop and fail to understand user utterances. Recently, [14, 15] developed a cold start natural language generation algorithms to enrich the training data with the hope of covering more varieties with low cost. As an alternative approach, [16, 17] proposed to leverage user and contextual information to mitigate the practical performance discrepancy. Yet, such extra information is not always available in real-world scenarios.

In this paper, our goal is to investigate and interpret what the state-of-the-art deep neural SLU models actually learn. Specifically, we propose the following four research questions:

Q1: *What enables SLU models to correctly detect the intent of an utterance?*

Q2: *When is the joint learning of intent detection and slot filling helpful?*

Q3: *Which words in an utterance are important for slot filling?*

Q4: *When do out-of-vocabulary words matter?*

In order to answer these questions, we design an embedding sparse structure learning approach, called SparseEmb, to study the vocabulary importance. SparseEmb utilizes a group lasso regularization penalty on word embedding matrix to encourage the pruning of useless words in deep neural SLU models. Moreover, we introduce a two phase training algorithm to automatically and efficiently tune the regularizer coefficient. We apply SparseEmb to benchmark datasets to answer these research questions and analyze the insights via ablation study.

Next, based on the discovered insights, we introduce a simple but efficient approach to sanitize the training set. We show that the SLU models trained on our new training set can improve the SLU performance on both benchmark test set and our collected test set. Our collected test set is crowd sourced to simulate the real user utterances in practice.

In the rest of paper, we first discuss the related work in Section 2. Next, we present our proposed SparseEmb approach in Section 3 for studying the vocabulary importance. Section 4 shows evaluation and ablation study of SparseEmb on benchmark datasets. At last, we propose a simple and effective training data sanitization method and show improved SLU performance in Section 5.

2. Related Work

Intent detection is treated as an utterance classification problem, which can be modeled using conventional classifiers such as support vector machine (SVM) [1] and RNN based models [2]. As a sequence labeling problem, slot filling can be solved using traditional machine learning approaches including maximum entropy Markov model [3] and conditional random fields (CRF) [18], as well as recurrent neural network (RNN) based approaches which takes and tags each word in an utterance one by one [4, 5, 6, 7, 8]. Recent research focuses on the joint model to learn two tasks together [2, 9, 10, 11, 12, 13, 19]. It is interesting to observe that all existing models use word embedding layer rather than character embedding layer. This is mainly because spoken language usually use simple words or even colloquial words rather than the complex words used in written language. Recently, Chen et al. [20] proposed a variational vocabulary selection algorithm for text

http://dx.doi.org/10.21437/Interspeech.2019-3184

classification.

3. SparseEmb Approach

In this section, our goal is to study the vocabulary importance in SLU models. Motivated by the utilization of word embedding in deep neural SLU models, we propose to learn sparse structure of the word embedding layer to prune the unimportant words. Our method is called *Embedding Sparse Structure Learning*, a.k.a. SparseEmb.

3.1. Embedding Group Lasso Regularization

Let E be the word embedding matrix in a deep SLU model, which consists of $|W|$ rows and d columns. W is the vocabulary in all training utterances and d is the embedding dimension. Thus, we use the word-based group Lasso regularization as follows:

$$R(E) = \sum_{i=1}^{d} \|E_i\|_2 \tag{1}$$

where E_i is the vector of all weights in row i of E which stands for the embedding of the i^{th} word; $\|\cdot\|_2$ is ℓ_2-norm (a.k.a., Euclidean length).

Since the slot filling is a sequence labeling task in which all words in an utterance are supposed to be important, we add $R(E)$ to the intent classification minimization function in order to encourage vocabulary sparsity for both single intent classification and joint learning models. The impact of $R(E)$ is controlled by the coefficient λ.

3.2. Efficient Regularizer Coefficient Tuning

To apply group Lasso regularizer, it is non trivial to select an ideal tuning coefficient hyperparameter λ. Traditional tuning criteria mainly focuses on minimizing the estimated prediction error such as cross-validation. However, it is time-consuming and oftentimes suffers from the false discovery rate.

Motivated by the recent work [21] which shows that using pseudo-features can efficiently tune the group lasso coefficient in linear model, we introduce the pseudo-words in our SLU problem. That is, we first initialize a set of pseudo-words that has the same size of the original vocabulary. For each utterance with length l, we randomly select l pseudo-words to insert after the utterance. This does not affect either the syntactic or semantic information of each utterance.

Since we know as a fact that these pseudo-words are truly useless for SLU models, we start with $\lambda = 0$ in which no pseudo words will be pruned. Then we search λ from 0 to 1 with grid interval 0.001. We stop at the minimum λ^* where all pseudo-words are pruned.

3.3. Two Phase Training

We train the SparseEmb model in two phases to accelerate the training time. In the first phase, we first train a SLU model traditionally.

The second phase is to fine tune the embedding layer. As described above, we first create the pseudo-words W_p with the same size of W and randomly initialize the embedding of each pseudo-word. Then, we create the modified training data by concatenating the same length of pseudo-words after each utterance. We start searching the minimum λ^* that can prune all pseudo-words using group Lasso regularization.

To avoid division by zero in the computation of regularization gradient, we can add a tiny number ϵ as $\|E_i\|_2 \triangleq \sqrt{\epsilon + \sum_j e_{ij}^2}$, where we choose $\epsilon = 1e - 8$. Moreover, to stabilize the sparsity during training, we zero out the weights whose absolute values are smaller than a pre-defined threshold τ. When running a specific λ, we use a threshold τ from 0 to 0.001 with grid interval 0.0001. Likewise, the minimum τ will be used for the selected λ.

Table 1: *Dataset Statistics*

	ATIS	Snips
Vocabulary Size	722	11,241
Number of Slots	127	72
Number of Intents	18	7
Training Set Size	4,478	13,084
Development Set Size	500	700
Testing Set Size	893	700

Remarks: Note that the second phase runs quite fast since the fine tuning of embedding layer on a pretrained model is fast. Also, small λ and τ are oftentimes sufficient to prune all pseudo-words.

4. Experiments of SparseEmb

4.1. Settings

We evaluate our proposed SparseEmb approach on two benchmark datasets: (1) *ATIS dataset (Airline Travel Information Systems)*: a widely used benchmark dataset in SLU research [22]; (2) *Snips dataset:* a recently collected dataset by Snips for benchmark SLU model evaluation [23]. Table 1 shows the dataset statistics.

We apply SparseEmb onto both Att-BiRNN SLU model [2] and Slot-Gated SLU model [13] for evaluation. For both models, we consider the independent intent classifier (*Int-Att-BiRNN* and *Int-Slot-Gated*) and joint learned model (*Joint-Att-BiRNN* and *Joint-Slot-Gated*).

We implement SparseEmb in TensorFlow based on the open sourced code of Att-BiRNN model [2] and Slot-Gated SLU model [13]. Since SparseEmb does no have additional hyperparameters, we use the same set of parameters as provided in original papers.

4.2. Evaluation Results

Table 2 shows the evaluation results of Att-BiRNN model. When only doing intent classification task, we find some interesting results: In ATIS dataset, the intent accuracy is improved by 0.22 by just using 32.5% of vocabulary. Similarly, only 6.4% of vocabulary in Snips dataset are needed to achieve comparable intent accuracy. With joint learning of both tasks, SparseEmb intends to use more vocabulary. As a result, the intent accuracy is further improved. Especially in Snips dataset, the accuracy is largely improved and outperforms Joint-Att-BiRNN by 0.72. On the other hand, we observe that the f1 score of slot filling task only drops within 10% despite a much smaller subset of vocabulary. This is somewhat counterintuitive since slot filling labels each word in an utterance such that all words are supposed to be equally important.

Table 3 reports the evaluation results of Slot-Gated model. We observe that the intent accuracy of both models on both datasets are improved after using our SparseEmb method, compared with original Slot-Gated model. Similar as the evaluation results of Joint-Att-BiRNN-SparseEmb in Table 2, SparseEmb enables the Slot-Gated model to better learn the

Table 2: *Evaluation Results of Att-BiRNN model*

Model	**ATIS Dataset**			**Snips Dataset**		
	Intent (Acc)	Slot (F1)	Useful Vocab (%)	Intent (Acc)	Slot (F1)	Useful Vocab (%)
Int-Att-BiRNN [2]	97.65	-	100%	97.23	-	100%
Joint-Att-BiRNN [2]	98.21	95.98	100%	97.55	87.78	100%
Int-Att-BiRNN-SparseEmb	**97.87**	-	32.5%	97.12	-	6.4%
Joint-Att-BiRNN-SparseEmb	97.91	90.72	58.2%	**98.27**	78.20	16.6%

Table 3: *Evaluation Results of Slot-Gated Model*

Model	**ATIS Dataset**			**Snips Dataset**		
	Intent (Acc)	Slot (F1)	Useful Vocab (%)	Intent (Acc)	Slot (F1)	Useful Vocab (%)
Int-Slot-Gated [13]	94.11	-	100%	96.84	-	100%
Full-Slot-Gated [13]	93.62	94.81	100%	97.03	88.82	100%
Int-Slot-Gated-SparseEmb	**94.62**	-	77.1%	**97.43**	-	33.9%
Full-Slot-Gated-SparseEmb	**94.29**	93.54	71.5%	**97.86**	83.45	23.9%

Table 4: *Intent Error Analysis: In "Intent" column, the three sub-columns represent the intent labels of Ground Truth, Int-Att-BiRNN, Int-Att-BiRNN-SparseEmb. The correctly and incorrectly predicted intents are in green and red respectively. In "Sample Testing Utterance" column, we choose some representative testing samples. The useful words selected by SparseEmb are in blue. The out-of-vocabulary words (words not in training utterances) are in grey.*

	Intent			**Sample Testing Utterance**
ATIS	atis_airport	atis_flight	atis_airport	which airport is closest to (ontario california)
	atis_airfare	atis_flight	atis_airfare	get saturday fares from (washington) to (toronto)
	atis_airline	atis_airline	atis_flight	list all alaska airlines flights
	atis_capacity	atis_capacity	atis_flight	list seating capacities of delta flights from seattle to salt lake city
	atis_meal	atis_city	atis_flight	are snacks served on tower air
	atis_day_name	atis_flight	atis_flight	what day of the week do flights from nashville to tacoma fly on
Snips	PlayMusic	SearchWork	PlayMusic	play subconscious lobotomy (from) jennifer paull
	SearchWork	SearchScreenEvent	SearchWork	find the album orphan girl at the (cemetery)
	SearchWork	SearchWork	BookRestaurant	am looking for a book with the title free to play
	SearchScreenEvent	SearchScreenEvent	SearchWork	i want to see shattered image
	GetWeather	GetWeather	GetWeather	will it be rainy in tenino
	AddToPlaylist	AddToPlaylist	AddToPlaylist	add muzika za decu to my crash course playlist

utterance structures by reducing the confusion from irrelevant words. As a result, the intent accuracy is improved by up to 0.83. On the other hand, the F1 score of slot filling is better preserved compared with Joint-Att-BiRNN-SparseEmb. This is because Slot-Gated-SparseEmb keeps more useful words for sequence labeling in slot filling task. Moreover, we find that Full-Slot-Gated-SparseEmb model intends to use less words than Int-Slot-Gated-SparseEmb on both datasets. This is because the slot attention can capture more contextual information of the utterance. Thus, Full-Slot-Gated-SparseEmb encourages more sparsity of the embedding layers such that some words become useless when the utterance context has been understood by the slot attention layer. At last, it is interesting to see that, in Snips dataset, Full-Slot-Gated-SparseEmb can achieve both better intent accuracy and better F1 score even by using less words. This is because of the relevant simpler utterance structure in Snips dataset. In ATIS dataset, Int-Slot-Gated-SparseEmb achieves better performance since the more complex utterance structures in ATIS prefers to know more original words in an utterance instead of only using latent slot attention weights information.

4.3. Error Analysis & Ablation Study

We further look into the detailed evaluation results and error analysis to answer the four research questions raised in Section 1. W.l.o.g., we provide the examples based on Att-BiRNN model [2]. The examples on Slot-Gated SLU model [13] share similar observations.

Q1: *What enables SLU models to correctly detect the intent of an utterance?*

We observe that although Int-Att-BiRNN and Int-Att-BiRNN-SparseEmb models achieve similar intent accuracy, the overlap of utterances with incorrectly predicted intents is only 38.7% and 17.2% on ATIS and Snips datasets respectively. Table 4 shows the detailed error analysis of intent classification. We find that Int-Att-BiRNN-SparseEmb improves the intent accuracy since it can better learn the utterance structures than Int-Att-BiRNN. *In the rest of this section, utterance structure is referred to as the delexicalized utterance with annotated slot type (e.g., "list all @airline_name flights" is the structure of original utterance "list all alaska flights").* The errors from Int-Att-BiRNN model could be mainly because of the confusion

Table 5: *Joint Learning Analysis: The words selected by Joint-Att-BiRNN-SparseEmb but not by Int-Att-BiRNN-SparseEmb are in orange. The green underbraces show the correct slot annotations.*

Slot Filling helps Intent Detection
list all $\underbrace{\text{alaska airlines}}_{\text{@airline\_name}}$ flights
are $\underbrace{\text{snacks}}_{\text{@meal\_description}}$ served on $\underbrace{\text{tower air}}_{\text{@airline\_name}}$
Intent Detection helps Slot Filling
the flights from $\underbrace{\text{ontario}}_{\text{@fromloc}}$ to $\underbrace{\text{westchester}}_{\text{@toloc}}$ that stop in $\underbrace{\text{chicago}}_{\text{@stoploc}}$
is there ground transportation from the $\underbrace{\text{memphis airport}}_{\text{@airport\_name}}$ into town when if i arrive at 842 in the $\underbrace{\text{morning}}_{\text{@period\_of\_day}}$

caused by other irrelevant words (words with brackets). ***A1.1:*** *SLU models mainly learn the different utterance structures associated with each intent.* On the other hand, the errors from Int-Att-BiRNN-SparseEmb are mainly because it fails to capture or disambiguate the keyword (underlined words) in an utterance. ***A1.2:*** *When the utterance structure is not sufficient for intent detection, SLU models try to capture and understand the keyword in the utterance.*

Q2: *When is the joint learning of intent detection and slot filling helpful?*

In the top part of Table 5, we find that the slot filling task enforces our Joint-Att-BiRNN-SparseEmb model to capture the keyword if the keyword is annotated as a slot. Thus, the understanding of such keyword helps to detect the correct intent. ***A2.1:*** *When the keyword in an utterance has slot annotation, the slot filling task can help to improve the intent accuracy.* On the other hand, the bottom part of Table 5 shows the cases that slots can be correctly annotated when the slot values are pruned from the vocabulary. ***A2.2:*** *When the utterance structure is sufficient for intent detection, the slot filling task can be correctly performed even without knowing the specific slot values.*

Q3: *Which words in an utterance are important for slot filling?*

Following Q2, we can naturally answer Q3. ***A3:*** *When the utterance structure is unclear (commonly used by multiple intents) or a slot value is complex (e.g., arbitrary length, overlapped words with utterance structures), the words in slot values are crucial for slot filling task.*

Q4: *When do out-of-vocabulary words matter?*

Since out-of-vocabulary (OOV) appears more in Snips dataset, the Snips results in Table 4 show the impact of OOV words (grey words). Similar as the answer of A2.2, ***A4.1:*** *OOV does not matter when they appear as slot values in an utterance with clear utterance structure learned by intent detection task.* ***A4.2:*** *When a OOV word appears to be the keyword for intent detection (either in utterance structure or keyword (underlined words)), it will lead to incorrect output for both tasks.*

5. Training Data Sanitization

In this section, we introduce a simple training data sanitization approach based on SparseEmb and show its effectiveness using both benchmark and collected testing data.

Table 6: *SLU Performance Comparison on Standard and Sanitized Training Data*

Model	Dataset	Training Data	Intent (Acc)	Slot (F1)
Att-BiRNN[2]	ATIS	standard	83.83	79.24
		sanitized	**86.34**	**82.65**
	Snips	standard	82.14	68.23
		sanitized	**85.65**	**72.75**
Slot-Gated[13]	ATIS	standard	81.54	77.39
		sanitized	**84.43**	**80.64**
	Snips	standards	81.32	67.68
		sanitized	**83.23**	**70.47**

5.1. Sanitization Approach

The key idea of training data sanitization is to combine both SparseEmb pruned vocabulary and the model re-validation during training. At first, we train a SLU model with SparseEmb on standard training data and obtain a set of useless words. Next, we create a new training data by replacing all useless words with '_BLK' token and retrain another SLU model without SparseEmb on the new training data. We then test the new model on original training data. We output the sanitized training data as follows: If both intent and slots are identified correctly for an utterance, we replace all useless words by '_BLK' token. Otherwise, if the intent is wrong, we will keep the original utterance as it is. In terms of slot values, we only keep the those which are tagged incorrectly.

5.2. Evaluation of SLU performance

5.2.1. Datasets

We evaluate our sanitized training data using the same two SLU models on ATIS and Snips benchmark datasets. Instead of using the standard testing data, we also collect an in-house testing data that simulates the real user utterances. We recruit turkers (via Amazon M-Turk) to collect a separate testing data for both datasets. We randomly select 5 intents and 10 slots in each dataset for for each turker in first group to generate 5 utterances in each intent. Each slot needs to be used at least once. To ensure the quality of generated utterances, we randomly permutate all manually generated utterances and ask the second group of turkers to select the correct utterances. We further judge the trustworthiness of second group turkers by scoring their performance on 20-30 gold-standard utterances that were internally marked correct or incorrect by experts. Each generated utterance is marked by 3 trustworthy judgers and it is considered correct only if all judgers mark it correct.

5.2.2. SLU Performance Results

Table 6 shows the performance comparison of SLU models (joint trained intent detection and slot filling tasks). First, we find that the performance of both models trained on standard training data significantly drops compared with on standard testing data from previous section. This is as what we expected due to data distribution difference even though the new testing data collection is still quite under control. Using our sanitized training data, the performance is improved on both datasets. This is because our sanitized training data removes some unimportant words that are likely to cause ambiguity in test utterances such that it leverages the joint learning of intent detection and slot filling tasks to better help each other.

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–I.

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

[3] A. McCallum, D. Freitag, and F. C. N. Pereira, “Maximum entropy markov models for information extraction and segmentation,” in *Proceedings of the Seventeenth International Conference on Machine Learning*, ser. ICML ’00. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2000, pp. 591–598.

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

[5] 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.

[6] B. Peng and K. Yao, “Recurrent neural networks with external memory for language understanding,” *arXiv preprint arXiv:1506.00195*, 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] G. Kurata, B. Xiang, B. Zhou, and M. Yu, “Leveraging sentencelevel information with encoder lstm for natural language understanding,” *arXiv preprint*, 2016.

[9] 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.

[10] 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*, Dec 2013, pp. 78–83.

[11] D. Hakkani-Tür, G. Tür, A. Celikyilmaz, Y.-N. Chen, J. Gao, L. Deng, and Y.-Y. Wang, “Multi-domain joint semantic frame parsing using bi-directional rnn-lstm.” in *INTERSPEECH*, 2016, pp. 715–719.

[12] Y. Wang, Y. Shen, and H. Jin, “A bi-model based RNN semantic frame parsing model for intent detection and slot filling,” in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 2 (Short Papers)*, 2018, pp. 309–314.

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

[14] Y. Shen, A. Ray, A. Patel, and H. Jin, “CRUISE: cold-start new skill development via iterative utterance generation,” in *Proceedings of ACL 2018, Melbourne, Australia, July 15-20, 2018, System Demonstrations*, 2018, pp. 105–110.

[15] Y. Shen, Y. Wang, A. Patel, and H. Jin, “Sliqa-i: Towards cold-start development of end-to-end spoken language interface for question answering,” in *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2019, Brighton, United Kingdom, May 12-17, 2019*, 2019, pp. 7195–7199.

[16] Y. Shen, X. Zeng, Y. Wang, and H. Jin, “User information augmented semantic frame parsing using progressive neural networks,” in *Interspeech 2018, 19th Annual Conference of the International Speech Communication Association, Hyderabad, India, 2-6 September 2018.*, 2018, pp. 3464–3468.

[17] A. Ray, Y. Shen, and H. Jin, “Learning out-of-vocabulary words in intelligent personal agents,” in *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden.*, 2018, pp. 4309–4315.

[18] C. Raymond and G. Riccardi, “Generative and discriminative algorithms for spoken language understanding,” in *INTERSPEECH*, 2007.

[19] A. Ray, Y. Shen, and H. Jin, “Robust spoken language understanding via paraphrasing,” in *Interspeech 2018, 19th Annual Conference of the International Speech Communication Association, Hyderabad, India, 2-6 September 2018.*, 2018, pp. 3454–3458.

[20] W. Chen, Y. Su, Y. Shen, Z. Chen, X. Yan, and W. Y. Wang, “How large a vocabulary does text classification need? A variational approach to vocabulary selection,” in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, 2019, pp. 3487–3497.

[21] S. Yang, J. Wen, X. Zhan, and D. Kifer, “Et-lasso: Efficient tuning of lasso for high-dimensional data,” *CoRR*, vol. abs/1810.04513, 2018.

[22] 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.

[23] Snips, https://github.com/snipsco/nlu-benchmark/tree/master/2017-06-custom-intent-engines, 2017.