Wide and Deep Learning for Spoken Language Understanding

Anonymous ACL submission

Abstract

Spoken language understanding (SLU) is one of the essential parts in smart voice assistants, which typically includes intent classification (IC) and slot filling (SF) tasks to interpret user utterances. Deep models jointly trained for the two tasks show more promising results compared with single-task models. However, these models always learn semantic representations for tokens and utterances but ignore their lexical information. Although they can generalize better to unseen tokens and utterances from low-dimensional dense semantic features, they also suffer from over-generalization when training data is limited. On the other hand, sparse lexical features such as word ngrams are good to memorize existing data correlations but fail for generalization. In this paper, we propose an approach leveraging lexical and semantic features to jointly learn IC and SF. The aim is to combine the benefits of memorization and generalization for SLU. Evaluating on a couple of domains from a large-scale smart voice assistant, results show our approach significantly improves IC and SF compared with several strong baselines.

1 Introduction

Smart voice assistants (SVA) such as Amazon Alexa, Google Assistant and Apple Siri are becoming ubiquitous by providing voice-enabled applications built by third-party developers to fulfill customer requirements. The essential part of SVAs is the spoken language understanding (SLU) system, where intent classification (IC) and slot filling (SF) are two major tasks to parse utterances into semantic frames and capture utterance core meanings (Tur and De Mori, 2011). Table 1 demonstrates how an utterance is assigned with one intent and a sequence of slots with In-Out-Begin (IOB) format.

Traditionally, intent classification is treated as a sequence classification problem, and slot filling is defined as a sequence tagging problem. Current research shows promising results by jointly learning the two tasks (Weld et al., 2021; Kim et al., 2017). Given the advantages of deep neural networks, convolutional neural networks (CNN) and recurrent neural networks (RNN) have been widely used to construct joint models along with conditional random fields (CRF) (Kane et al., 2021; Niu et al., 2019; Kumar and Baghel, 2021). More advanced techniques are utilized to further improve prediction accuracy such as pre-trained language models (Chen et al., 2019), capsule neural networks (Zhang et al., 2018), and attention-based models (Goo et al., 2018; Chen et al., 2021; Wu et al., 2021). Some other works focus on solving label sparsity issue in the two tasks from meta learning (Bhathiya and Thayasivam, 2020), transfer learning (Soto and Arkoudas, 2021), and few shot learning (Yu et al., 2021) perspectives.

Sentence	Pay	my	electricity	bill
Intent	PayBillIntent			
Slots	O	O	B-BillType	O

Table 1: An example utterance with its annotated intent and semantic slots (IOB format).

One challenge in SLU, similar to the recommendation task (Cheng et al., 2016), is to achieve both memorization and generalization. Memorization can be loosely defined as learning the frequent co-occurrence of features and exploiting their correlation, which can be achieved by learning linear relationship over sparse lexical features such as word ngrams. While generalization is based on transitivity of correlation and explores new feature combinations from unseen tokens and utterances, which is more topical and semantic. Current models mostly represent tokens and utterances as low-dimensional dense vectors to capture utterance semantics for generalization but ignore utterance lexical information for memorization, which may over-generalize when the training data is limited. For instance, if the model is trained to recognize that both Seattle and San Francisco are labeled as "*US City*" slot type, it may over-generalize at infer-

ence time and mistakenly recognize Berlin, Cairo or Beijing as slot values given that all these cities' semantic representations could be similar.

Previous works (Cheng et al., 2016; Yang et al., 2013) notice that combining both lexical and semantic features can achieve better results than single type features in recommendation and SLU tasks. Given that, we propose a joint model combining semantic features (extracted from jointly trained encoders) and hand-crafted lexical features for IC and SF. The goal is to combine the benefits of memorization and generalization for reducing model errors. Meanwhile, in Table 1, the annotated slot "*B-BillType*" is highly correlated with intent "*PayBillIntent*", indicating that slot information will inherently benefit intent classification. Therefore, to enhance the connection between IC and SF, we merge predicted slots with utterance context to construct lexical features for intent classification.

2 Method

At high level, our proposed model consists of two head blocks as shown in Figure 1: an utterance intent classification head which is a sequence-level softmax layer and a slot filling head which is a conditional random field (CRF) layer on top of bi-directional LSTM (BiLSTM) and token-level softmax layer. The model is trained jointly to minimize the linear combination of the two task losses.

2.1 Feature Engineering

Inspired from the Wide and Deep model (Cheng et al., 2016), we combine two types of features for intent classification: lexical features for memorization and semantic features for generalization. As lexical features are represented as multi-hot embeddings on utterance level, they cannot support the sequence tagging problem. Therefore only semantic features are employed for slot filling.

2.1.1 Lexical Features

The lexical features require more feature engineering effort. To enhance connections between the two tasks, we also use predicted slots to construct lexical features for intent classification. In the end, there are three types of lexical features: utterance length, slot-mixed features and token features.

Given an utterance $u = \{w_1, w_2, ..., w_k\}$, lexical features include word unigrams and bigrams. Utterance length is treated as a categorical feature. Slot-mixed features include predicted slot unigrams and slot-mixed bigrams where predicted slots are used to replace the original tokens for bigram construction. Specifically, in training stage, utterance ground truth slots are used to construct slot-mixed features. While in testing stage, SF predicted slots are used to construct slot-mixed features instead. For example in Table 1, the slot-mixed bigrams will be "*pay_my, my_B-BillType, B-BillType_bill*".

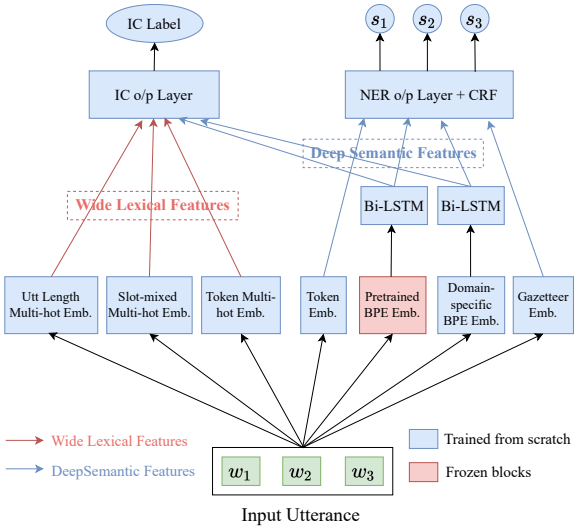

Figure 1: The pipeline of the proposed joint model. Red lines indicate wide lexical features, and blue lines indicate deep semantic features.

2.1.2 Semantic Features

The Byte-Pair encoding (BPE) subword tokenization (Sennrich et al., 2015) is applied to split words into subwords (tokens). For semantic features, the concatenation of the Bi-LSTM hidden states of first and last token is regarded as utterance semantic representation. To incorporate both general and domain-specific information, we train two BPE embedding layers (in Figure 1): the first embedding layer is pre-trained on public Wikipedia data and the second is trained from scratch using domain data. Then we train a separate Bi-LSTM block on top of each BPE embedding layer.

2.2 Intent Classification

We calculate multi-hot embeddings of lexical features for model memorization, including utterance length e_l, unigram/bigram tokens e_t and slot-mixed unigram/bigram tokens e_s. We also have dense semantic features for model generalization, including pre-trained BPE encoder output h_p and domain-specific BPE encoder output h_d. Similar to (Cheng et al., 2016), we concatenate both types of features together and pass it to a non-linear layer to predict intent I with the largest probability score.

$$h_I = [e_l; e_t; e_s; h_p; h_d]$$
$$I = \underset{I}{\operatorname{argmax}}\, \operatorname{softmax}(W h_I + b) \quad (1)$$

2.3 Slot Filling

Besides the two shared BPE components (in Figure 1) capturing token's sequential semantics, we involve two augmented features including token embeddings and gazetteer embeddings to capture token's individual semantics. Different from token multi-hot embeddings used for intent classification, we hereby use dense vectors for token embeddings. Gazetteer features are mappings from tokens to named entities through our pre-owned gazetteer dictionary, e.g., "*The Beatles*" would be mapped to "*ArtistName*". It is a preprocessing step to generate an extra slot signal for each token. Gazetteer features are excluded in intent classification as they contain duplicated information with slot-mixed features. In the end, each token is associated with a gazetteer embedding as well.

The i^{th} word w_i in utterance u may contain multiple BPE tokens, the last token is empirically used to represent the word itself. We first concatenate its token embedding e^i_w, i^{th} step pre-trained BPE' BiLSTM encoder output h^i_p, i^{th} step domain-specific BPE's BiLSTM encoder output h^i_d and gazetteer embedding e^i_g to pass to a softmax layer. A CRF layer is finally employed on all step outputs to predict the slot sequence $S = \{s_1, ..., s_k\}$.

$$\begin{aligned} h^i_S &= \text{softmax}([e^i_w; h^i_p; h^i_d; e^i_g]) \\ S &= \text{CRF}(h^1_S, ..., h^k_S) \end{aligned} \tag{2}$$

2.4 Model Training

Our proposed model is trained by jointly minimizing the two task losses. Intent classification loss $\mathcal{L}_I$ is cross entropy loss with L1 and L2 regularization on weight matrix W. C is the number of intents, y_i is the ground truth score and $\hat{y}_i$ is the predicted score for the i_{th} intent. Slot filling loss $\mathcal{L}_S$ is standard CRF loss aiming to find the slot sequence S with the highest score. The score$(\cdot)$ function measures the slot sequence likelihood given utterance tokens. $\log(\sum_{\widetilde{S}} e^{\text{score}(u,\widetilde{S})})$ is the sum over all possible slot sequences $\widetilde{S}$. The final loss $\mathcal{L}$ is the weighted sum of $\mathcal{L}_I$ and $\mathcal{L}_S$.

$$\begin{aligned} \mathcal{L}_I &= -\sum_{i=1}^{C} y_i \log(\hat{y}_i) + \beta_1 ||W||^1_F + \beta_2 ||W||^2_F \\ \mathcal{L}_S &= -\text{score}(u, S) + \log(\sum_{\widetilde{S}} e^{\text{score}(u,\widetilde{S})}) \\ \mathcal{L} &= \mathcal{L}_I + \alpha \mathcal{L}_S \end{aligned} \tag{3}$$

In training stage, two tasks are learned jointly. In inference stage, SF prediction is first retrieved to construct slot-mixed features for IC prediction.

3 Experiments

3.1 Data

The data is collected from 5 domains (third-party applications) of Amazon Alexa: Talking Tom, Trivia Battle, CL Vocab Game, WikiHow and Plex. Detailed data statistics are reported in Table 2. Both training and development sets consist of synthetic utterances provided by skill developers. The testing set consists of manually-annotated real utterances.

Dataset	Training	Development	Testing	Intents	Slot Types
Talking Tom	10,000	1,000	788	4	2
Trivia Battle	7,542	1,000	2,070	9	8
CL Vocab Game	7,542	1,000	1,863	9	4
WikiHow	10,000	1,000	541	7	3
Plex	10,000	1,000	1,784	19	9

Table 2: Dataset Statistics.

3.2 Settings

We use four different metrics: SemER (Semantic Error Rate), SER (Slot Error Rate), IRER (Interpretation Recognition Error Rate), and ICER (Intent Classification Error Rate) (Su et al., 2018). SemER combines IC and SF errors into a single score. It computes a modified edit distance that takes into account the number of substitutions (S), incorrect predictions (I), and deletions (D) in intent and slot prediction. For a sequence of L tokens, SemER is defined as (S + I + D) / (L + 1). SER is similar to SemER but only measures slot accuracy. IRER is the fraction of utterances not correctly recognized on both intents and slots. ICER is the the fraction of utterances not correctly recognized on intents.

Our proposed model is compared with three baseline models: 1) **Linear-CRF** model currently serves as Alexa production model which contains a generalized linear model for intent classification and a conventional CRF model for slot filling. 2) **Wide-BiLSTM-CRF** model uses only lexical features for intent classification. It uses the same structure as our proposed model for slot filling. 3) **Deep-BiLSTM-CRF** model uses only semantic features for intent classification. It uses the same structure as our proposed model for slot filling. The intention to choose these three baselines is that first we want to compare our proposed model with production model, second we want to explore the effectiveness of wide and deep components.

3.3 Model Comparison

Table 3 reports the relative improvements of baselines and our models compared with the production Linear-CRF model. The three DNN based models all outperform Linear-CRF on all evaluation metrics, reflecting the advantages of deep models.

Wide-BiLSTM-CRF beats Deep-BiLSTM-CRF on all metrics. As their slot filling component are with same structure, SER is relatively similar. Higher ICER means designed lexical features are more powerful than semantic features for intent classification. By combining lexical and semantic features, our model achieves the best results. It means that the two types of features complement with each other and combining them can reduce errors.

Model Δ%	SemER	SER	IRER	ICER
Wide-BiLSTM-CRF	5.50	2.03	4.77	5.33
Deep-BiLSTM-CRF	4.74	1.97	3.94	4.27
Our Model	7.78	2.04	6.39	7.41

Table 3: Model comparison results. Relative improvement values are computed with respect to the Linear-CRF baseline model.

3.4 Feature Effectiveness Validation

To validate the effectiveness of each input feature, we conduct experiments to remove each type of features from the proposed full model and keep the rest components fixed. The relative performance results are summarized in Table 4. "–" sign means the corresponding features are removed from inputs. For example, "–BPE" means two BPE semantic features are removed for IC and SF prediction. All features have positive impact on the two tasks as removing each of them will downgrade model performance. BPE is the most important one among all features. Removing it will hugely hurt model slot prediction capability, which in return will also affect intent classification performance as BPE and predicted slots are both used for intent classification. Token features are important because they include lexical features and token embeddings for both tasks. Slot-mixed features also have significant impact, indicating the usefulness to directly import predicted slots for intent classification.

Model Δ%	SemER	SER	IRER	ICER
–BPE	-8.21	-9.83	-7.63	-3.37
–Token	-2.74	-1.05	-2.60	-1.27
–Utt Length	-1.09	-0.04	-0.98	-0.76
–Gazetteer	-2.17	-1.37	-2.05	-0.49
–Slot-mixed	-1.65	-0.04	-1.63	-1.42

Table 4: Relative improvement comparison results if we remove each type of input features from full model.

3.5 Hyper Parameter Tuning

We report the tuning results of four hyper-parameters: L1 and L2 regularization, batch size and number of epochs. The visualization results in Figure 2 help us determine the default settings:

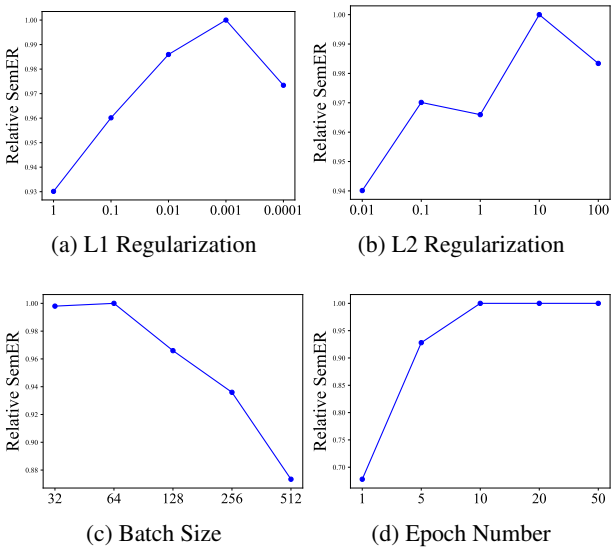

Figure 2: Parameter tuning results. Reported values are relative SemER scores compared with default settings.

L1 regularization is 1e-3, L2 regularization is 10, batch size is 64 and number of epochs is 10.

Figure 2 shows relative SemER scores compared with default settings. Selecting appropriate L1 and L2 regularization values both have significant impact on model performance. Large batch sizes will degrade model performance. As the parameters are updated based on average gradients in each batch, gradients might be blurred if averaged by large batch size. But small batch size will slow down the training speed, which is a trade-off for batch size selection. We also observe that model validation results stay unchanged after 10 epochs, meaning that training more epochs is not necessary as the best model has already been achieved.

4 Conclusion

In this paper, we presented a wide and deep multi-task model to address the disadvantages of the widely adopted deep learning architecture for most SLU systems. Although it is jointly trained to perform intent classification and slot filling, it combines semantic and lexical features for IC but only uses semantic features for SF. The experimental results on five domains of a commercial voice assistant, Amazon Alexa, have shown that the combined features have significantly improved the quality of IC but with minor improvement to the SF quality. Eventually, the wide and deep model reported average relative improvement on SEMER and IRER by 7.78% and 6.39%, respectively. In the future work, we will study the impact of combining semantic and lexical features on the slot filling task as well.

References

Hemanthage S Bhathiya and Uthayasanker Thayasivam. 2020. Meta learning for few-shot joint intent detection and slot-filling. In *Proceedings of the 2020 5th International Conference on Machine Learning Technologies*, pages 86–92.

Dongsheng Chen, Zhiqi Huang, Xian Wu, Shen Ge, and Yuexian Zou. 2021. Towards joint intent detection and slot filling via higher-order attention. *arXiv preprint arXiv:2109.08890*.

Qian Chen, Zhu Zhuo, and Wen Wang. 2019. Bert for joint intent classification and slot filling. *arXiv preprint arXiv:1902.10909*.

Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, et al. 2016. Wide & deep learning for recommender systems. In *Proceedings of the 1st workshop on deep learning for recommender systems*, pages 7–10.

Chih-Wen Goo, Guang Gao, Yun-Kai Hsu, Chih-Li Huo, Tsung-Chieh Chen, Keng-Wei Hsu, and Yun-Nung Chen. 2018. 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)*, pages 753–757.

Bamba Kane, Fabio Rossi, Ophélie Guinaudeau, Valeria Chiesa, Ilhem Quénel, and Stéphane Chau. 2021. Joint intent detection and slot filling via cnn-lstm-crf. In *2020 6th IEEE Congress on Information Science and Technology (CiSt)*, pages 342–347. IEEE.

Young-Bum Kim, Sungjin Lee, and Karl Stratos. 2017. Onenet: Joint domain, intent, slot prediction for spoken language understanding. In *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 547–553. IEEE.

Niraj Kumar and Bhiman Kumar Baghel. 2021. Smart stacking of deep learning models for granular joint intent-slot extraction for multi-intent slu. *IEEE Access*, 9:97582–97590.

Peiqing Niu, Zhongfu Chen, Meina Song, et al. 2019. A novel bi-directional interrelated model for joint intent detection and slot filling. *arXiv preprint arXiv:1907.00390*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*.

Victor Soto and Konstantine Arkoudas. 2021. Combining weakly supervised ml techniques for low-resource nlu. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Industry Papers*, pages 288–295.

Chengwei Su, Rahul Gupta, Shankar Ananthakrishnan, and Spyros Matsoukas. 2018. A re-ranker scheme for integrating large scale nlu models. In *2018 IEEE Spoken Language Technology Workshop (SLT)*, pages 670–676. IEEE.

Gokhan Tur and Renato De Mori. 2011. *Spoken language understanding: Systems for extracting semantic information from speech*. John Wiley & Sons.

HENRY Weld, Xiaoqi Huang, SIQU Long, Josiah Poon, and SOYEON CAREN Han. 2021. A survey of joint intent detection and slot-filling models in natural language understanding. *arXiv preprint arXiv:2101.08091*.

Jie Wu, Ian Harris, and Hongzhi Zhao. 2021. Spoken language understanding for task-oriented dialogue systems with augmented memory networks. In *Annual Conference of the North American Chapter of the Association for Computational Linguistics*.

Lili Yang, Chunping Li, Qiang Ding, and Li Li. 2013. Combining lexical and semantic features for short text classification. *Procedia Computer Science*, 22:78–86.

Dian Yu, Luheng He, Yuan Zhang, Xinya Du, Panupong Pasupat, and Qi Li. 2021. Few-shot intent classification and slot filling with retrieved examples. In *NAACL-HLT*.

Chenwei Zhang, Yaliang Li, Nan Du, Wei Fan, and Philip S Yu. 2018. Joint slot filling and intent detection via capsule neural networks. *arXiv preprint arXiv:1812.09471*.