Improving Slot Filling by Utilizing Contextual Information

Amir Pouran Ben Veyseh
Department of Computer and Information Science
University of Oregon
Eugene, OR, USA
apouranb@cs.uoregon.edu

Franck Dernoncourt
Adobe Research
San Jose, CA, USA
dernonco@adobe.com

Thien Huu Nguyen
Department of Computer and Information Science
University of Oregon
Eugene, OR, USA
thien@cs.uoregon.edu

Abstract

Slot Filling is the task of extracting the semantic concept from a given natural language utterance. Recently it has been shown that using contextual information, either in work representations (e.g., BERT embedding) or in the computation graph of the model, could improve the performance of the model. However, recent work uses the contextual information in a restricted manner, e.g., by concatenating the word representation and its context feature vector, limiting the model from learning any direct association between the context and the label of word. We introduce a new deep model utilizing the contextual information for each work in the given sentence in a multi-task setting. Our model enforce consistency between the feature vectors of the context and the word while increasing the expressiveness of the context about the label of the word. Our empirical analysis on a slot filling dataset proves the superiority of the model over the baselines.

1 Introduction

Slot Filling (SF) is the task of identifying the semantic concept expressed in natural language utterance. For instance, consider a request to edit an image expressed in natural language: “Remove the blue ball on the table and change the color of the wall to brown”. Here, the user asks for an “Action” (i.e., removing) on one “Object” (blue ball on the table) in the image and changing an “Attribute” (i.e., color) of the image to new “Value” (i.e., brown). Our goal in SF is to provide a sequence of labels for the given sentence to identify the semantic concept expressed in the given sentence.

Prior work have shown that contextual information could be useful for SF. They utilize contextual information either in word level representation (i.e., via contextualize embedding e.g., BERT (Chen et al., 2019)) or in the model computation graph (e.g., concatenating the context feature to the word feature (Liu et al., 2019)). However, such methods fail to capture the explicit dependence between the context of the word and its label. Moreover, such limited use of contextual information (i.e., concatenation of the feature vector and context vector) in the model cannot model the interaction between the word representation and its context. In order to alleviate these issues, in this work, we propose a novel model to explicitly increase the predictability of the word label using its context and increasing the interactivity between word representations and its context. More specifically, in our model we use the context of the word to predict its label and by doing so our model learns label-aware context for each word in the sentence. In order to improve the interactivity between the word representation and its context, we increase the mutual information between the word representations and its context. In addition to these contributions, we also propose an auxiliary task to predict which labels are expressed in a given sentence. Our model is trained in a multi-tasking framework. Our experiments on a SF dataset for identifying semantic concepts from natural language request to edit an image show the superiority of our model compared to previous baselines. Our model achieves the state-of-the-art results on the benchmark dataset by improving the F1 score by almost 2%, which corresponds to a 12.3% error rate reduction.

2 Related Work

The task of Slot Filling is formulated as a sequence labeling problem. Deep learning has been extensively employed for this task ((Zhao and Feng,
prior work has mainly utilized the recurrent neural network as the encoder to extract features per word and Conditional Random Field (CRF) (Lafferty et al., 2001) as the decoder to generate the labels per word. Recently the work (Liu et al., 2019) shows that the global context of the sentence could be useful to enhance the performance of neural sequence labeling. In their approach, they use a separate sequential model to extract word features. Afterwards, using max pooling over the representations of the words, they obtain the sentence representations and concatenate it to the word embedding as the input to the main task encoder (i.e. the RNN model to perform sequence labeling). The benefit of using the global context along the word representation is 2-fold: 1) it enhance the representations of the word by the semantics of the entire sentence thus the word representation are more contextualized 2) The global view of the sentence would increase the model performance as it contains information about the entire sentence and this information might not be encoded in word representations due to long dependencies.

However, the simple concatenation of the global context and the word embeddings would not separately ensure these two benefits of the global context. In order to address this problem, we introduce a multi-task setting to separately ensure the aforementioned benefits of utilizing contextual information. In particular, to ensure the better contextualized representations of the words, the model is encourage to learn representations for the word which are consistent with its context. This is achieved via increasing the mutual information between the word representation and its context. To ensure the usefulness of the contextual information for the final task, we introduce two novel sub-tasks. The first one aims to employ the context of the word instead of the word representation to predict the label of the word. In the second sub-task, we use the global representation of the sentence to predict which labels exist in the given sentence in a multi-label classification setting. These two sub-tasks would encourage the contextual representations to be informative for both word level classification and sentence level classification.

3 Model

Our model is trained in a multi-task setting in which the main task is slot filling to identify the best possible sequence of labels for the given sentence. In the first auxiliary task we aim to increase consistency between the word representation and its context. The second auxiliary task is to enhance task specific information in contextual information. In this section, we explain each of these tasks in more details.

3.1 Slot Filling

The input to the model is a sequence of words \(x_1, x_2, ..., x_N\). The goal is to assign each word one of the labels action, object, attribute, value or other. Following other methods for sequence labelling, we use the BIO encoding schema. In addition to the sequence of words, the part-of-speech (POS) tags and the dependency parse tree of the input are given to the model.

The input word \(x_i\) is represented by the concatenation of its pre-trained word embedding and its POS tag embedding, denoted by \(e_i\). These representations are further abstracted using a 2-layer Bi-Directional Long Short-Term Memory (LSTM) to obtain feature vector \(h_i\). We use the dependency tree of the sentence to utilize the syntactical information about the input text. This information could be useful to identify the important words and their dependents in the sentence. In order to model the syntactic tree, we utilize Graph Convolutional Network (GCN) (Kipf and Welling, 2016) over the dependency tree. This model learns the contextualized representations of the words such that the representation of each word is contextualized by its neighbors. We employ 2-layer GCN with \(h_i\) as the initial representation for the node (i.e., word) \(i\). The representations of the \(i\)th node is an aggregation of the representations of its neighbors. Formally the hidden representations of the \(i\)th word in \(l\)th layer of GCN is obtained by:

\[
\hat{h}_i^l = \text{ReLU}(SUM(h_{N(i)}^l W_l + \hat{h}_j^{l-1})/\text{deg}(i))
\]

where \(N(i)\) is the neighbors of the \(i\)th word in the dependency tree, \(W_l\) is the weight matrix in \(l\)th layer and \(\text{deg}(i)\) is the degree of the \(i\)th word in the dependency tree. The biases are omitted for
brevity. The final representations of the GCN for ith word, \( h_i \), represent the structural features for that word. Afterwards, we concatenate the structural features \( h_i \) and sequential features \( h_i \) to represent ith word by feature vector \( h_i' \):

\[
h_i' = CONCAT(h_i, \hat{h}_i)
\]

Finally in order to label each word in the sentence we employ a task specific 2-layer feed forward neural net followed by a logistic regression model to generate class scores \( S_i \) for each word:

\[
S_i = W_{LR} \ast (W_2 \ast (W_1 \ast h'_i))
\]

where \( W_{LR}, W_1 \) and \( W_2 \) are trainable parameters and \( S_i \) is a vector of size number of classes in which each dimension of it is the score for the corresponding class. Since the main task is sequence labeling we exploit Conditional Random Field (CRF) as the final layer to predict the sequence of labels for the given sentence. More specifically, class scores \( S_i \) are fed into the CRF layer as emission scores to obtain the final labeling score:

\[
Score(\hat{t}_1, \hat{t}_2, ..., \hat{t}_N|w_1, w_2, ..., w_N, \theta) = \sum_{j=1}^{N}(S_j + T_{j-1,j})
\]

where \( T \) is the trainable transition matrix and \( \theta \) is the parameters of the model to generate emission scores \( S_i \). Viterbi loss \( LV_B \) is used as the final loss function to be optimized during training. In the inference time, the Viterbi decoder is employed to find the sequence of labels with highest score.

### 3.2 Consistency with Contextual Representation

In this sub-task we aim to increase the consistency of the word representation and its context. To obtain the context of each word we perform max pooling over the all words of the sentence excluding the word itself:

\[
h'_i = MaxPooling(h_1, h_2, ..., h_N/h_i)
\]

where \( h_i \) is the representation of the ith word from the Bi-LSTM. We aim to increase the consistency between vectors \( h_i \) and \( h'_i \). One way to achieve this is by decreasing the distance between these two vectors. However, directly enforcing the word representation and its context to be close to each other would not be efficient as in long sentences the context might substantially differs from the word. So in order to make enough room for the model to represent the context of each word while it is consistent with the word representation, we employ an indirect method.

We propose to maximize the mutual information (MI) between the word representation and its context in the loss function. In information theory, MI evaluates how much information we know about one random variable if the value of another variable is revealed. Formally, the mutual information between two random variable \( X_1 \) and \( X_2 \) is obtained by:

\[
MI(X_1, X_2) = \int_{X_1} \int_{X_2} P(X_1, X_2) \log \frac{P(X_1, X_2)}{P(X_1)P(X_2)} dX_1 dX_2
\]

Using this definition of MI, we can reformulate the MI equation as KL Divergence between the joint distribution \( P_{X_1X_2} = P(X_1, X_2) \) and the product of marginal distributions \( P_{X_1 \otimes X_2} = P(X_1)P(X_2) \):

\[
MI(X_1, X_2) = D_{KL}(P_{X_1X_2}||P_{X_1 \otimes X_2})
\]

Based on this understanding of MI, we can see that if the two random variables are dependent then the mutual information between them (i.e. the KL-Divergence in equation 7) would be the highest. Consequently, if the representations \( h_i \) and \( h'_i \) are encouraged to have large mutual information, we expect them to share more information. The mutual information would be introduced directly into the loss function for optimization.

One issue with this approach is that the computation of the MI for such high dimensional continuous vectors as \( h_i \) and \( h'_i \) is prohibitively expensive. In this work, we propose to address this issue by employing the mutual information neural estimation (MINE) in (Belghazi et al., 2018) that seeks to estimate the lower bound of the mutual information between the high dimensional vectors via adversarial training. To this goal, MINE attempts to compute the lower bound of the KL divergence between the joint and marginal distributions of the given high dimensional vectors/variables. In particular, MINE computes the
lower bound of the Donsker-Varadhan representation of KL-Divergence:

$$D_{KL}(P_{X_1} \| P_{X_1} \otimes P_{X_2}) = \sup_{T: \Omega \rightarrow \mathbb{R}} \mathbb{E}_{P_{X_1} X_2}[T] - \log(\mathbb{E}_{P_{X_1} \otimes X_2}[e^T])$$  \hspace{1cm} (8)

However, recently, it has been shown that other divergence metrics (i.e., the Jensen-Shannon divergence) could also be used for this purpose (Hjelm et al., 2019; Ravanelli and Bengio, 2018), offering simpler methods to compute the lower bound for the MI. Consequently, following such methods, we apply the adversarial approach to obtain the MI lower bound via the binary cross entropy loss to estimate the mutual information between the word representation and its context representation. This is desirable as the word representation and its context, increasing the task specific information in contextual representations would help the model’s final performance.

In order to increase task-specific information in contextual representation, we train the model on two auxiliary tasks. The first one aims to use the context of each word to predict the label of that word and the goal of the second auxiliary task is to use the global context information to predict sentence level labels. We describe each of these tasks in more details in the following sections.

3.3 Prediction by Contextual Information

In this sub-task we use the context representations of each word to predict its label. It will increase the information encoded in the context of the word about the label of the word. We use the same context vector $h^c_i$ for the $i$th word as described in the previous section. This vector is fed into a 2-layer feed forward neural network with a softmax layer at the end to output the probabilities for each class:

$$P_{t}(\cdot|x_1, x_2, ..., x_n)/x_i) = \text{softmax}(W_2 * (W_1 * h^c_i))$$  \hspace{1cm} (10)

Where $W_2$ and $W_1$ are trainable parameters. Biases are omitted for brevity. Finally we use the following cross-entropy loss function to be optimized during training:

$$L_{WP} = \frac{1}{N} \sum_{i=1}^{N} - \log(P_t(l_i|x_1, x_2, ..., x_n)/x_i))$$  \hspace{1cm} (11)

where $N$ is the length of the sentence and $l_i$ is the label of the $i$th word.

3.3.2 Predicting Sentence Labels

The word label prediction enforces the context of each word to contain information about its label but it would not ensure the contextual information to capture the sentence level patterns for expressing intent. In other words, the word level prediction lacks a general view about the entire sentence. In order to increase the general information about the sentence in the representation of the words, we aim to predict the labels existing in a sentence from the representations of its words. More specifically, we introduce a new sub-task to predict which labels exit in the given sentence (Note that sentences might have only a subset of the labels; e.g. only action and object). We formulate this
task as a multi-class classification problem. Formally, given the sentence \( X = x_1, x_2, ..., x_N \) and label set \( S = \{ \text{action, attribute, object, value} \} \) our goal is to predict the vector \( L^s = l^s_1, l^s_2, ..., l^s_{|S|} \) where \( l^s_i \) is one if the sentence \( X \) contains \( i \)th label from the label set \( S \) otherwise it is zero.

First, we find representation of the sentence from the word representations. To this end, we use max pooling over all words of the sentence to obtain vector \( H \):

\[
H = \text{MaxPooling}(h_1, h_2, ..., h_N)
\]  

(12)

Afterwards, the vector \( H \) is further abstracted by a 2-layer feed forward neural net with a sigmoid function at the end:

\[
P(.|x_1, x_2, ..., x_N) = \sigma(W_2 * (W_1 * H))
\]  

(13)

where \( W_2 \) and \( W_1 \) are trainable parameters. Note that since this task is a multi-class classification the number of neurons at the final layer is equal to \( |S| \). We optimize the following binary cross entropy loss function:

\[
L_{SP} = \frac{1}{|S|} \sum_{k=1}^{|S|} - (l_k * \log(P(l_k|x_1, x_2, ..., x_N)) + (1 - l_k) * \log(1 - P(l_k||x_1, x_2, ..., x_N)))
\]  

(14)

where \( l_k \) is one if the sentence contains the \( k \)th label otherwise it is zero. Finally, to train the model we optimize the following loss function:

\[
L = L_{VB} + \alpha L_{discr} + \beta L_{WP} + \gamma L_{SP}
\]  

(15)

where \( \alpha, \beta \) and \( \gamma \) are all set to 0.1; We use Adam optimizer with learning rate 0.003 to train the model. We use micro-averaged F1 score on all labels as the evaluation metric.

We compare our method with the models trained using Adobe internal NLU tool, Pytext \((\text{Aly et al., 2018})\) and Rasa \((\text{Bocklisch et al., 2017})\) NLU tools. Table 2 shows the results on Test set. Our model improves the F1 score by almost 2%, which corresponds to a 12.3% error rate reduction. This improvements proves the effectiveness of using contextual information for the task of slot filling.

In order to analyze the contribution of the proposed sub-tasks we also evaluate the model when we remove one of the sub-task and retrain the model. The results are reported in Table 3. This table shows that all sub-tasks are required for the model to have its best performance. Among all sub-tasks the word level prediction using the contextual information has the major contribution to the model performance. This fact shows that context-

| Model  | F1   |
|--------|------|
| Adobe  | 77.9 |
| Pytext | 78.3 |
| Rasa   | 85.4 |
| Ours   | 87.2 |

Table 2: Performance of the model and baselines on the Test set.

| Label    | Train | Dev | Test |
|----------|-------|-----|------|
| Action   | 1562  | 448 | 479  |
| Object   | 4676  | 1447| 1501 |
| Attribute| 1437  | 403 | 462  |
| Value    | 507   | 207 | 155  |
| Sentences| 1737  | 497 | 559  |

Table 1: Label Statistics

4 Experiments

In our experiments, we use Onsei Intent Slot dataset. Table 1 shows the statics of this dataset. We use the following hyper parameters in our model: We set the word embedding and POS embedding to 768 and 30 respectively; The pre-trained BERT \((\text{Devlin et al., 2018})\) embedding are used to initialize word embeddings; The hidden dimension of the Bi-LSTM, GCN and feed forward networks are 200; the hyper parameters \( \alpha, \beta \) and \( \gamma \) are all set to 0.1; We use Adam optimizer with learning rate 0.003 to train the model. We use micro-averaged F1 score on all labels as the evaluation metric.

We compare our method with the models trained using Adobe internal NLU tool, Pytext \((\text{Aly et al., 2018})\) and Rasa \((\text{Bocklisch et al., 2017})\) NLU tools. Table 2 shows the results on Test set. Our model improves the F1 score by almost 2%, which corresponds to a 12.3% error rate reduction. This improvements proves the effectiveness of using contextual information for the task of slot filling.

In order to analyze the contribution of the proposed sub-tasks we also evaluate the model when we remove one of the sub-task and retrain the model. The results are reported in Table 3. This table shows that all sub-tasks are required for the model to have its best performance. Among all sub-tasks the word level prediction using the contextual information has the major contribution to the model performance. This fact shows that context-

| Model  | F1   |
|--------|------|
| Full   | 87.2 |
| Full - MI | 84.2 |
| Full - WP | 83.2 |
| Full - SP | 84.1 |

Table 3: Performance of the model when the loss function of each sub-task has been removed from the final loss function. MI, WP and SP stand for Mutual Information, Word Prediction and Sentence Prediction respectively.
textual information trained to be informative about the final sub-task is necessary to obtain the representations which could boost the final model performance.

5 Conclusion & Future Work

In this work we introduce a new deep model for the task of Slot Filling. In a multi-task setting, our model increase the mutual information between word representations and its context, improve the label information in the context and predict which concepts are expressed in the given sentence. Our experiments on an image edit request corpus shows that our model achieves state-of-the-art results on this dataset.

References

Ahmed Aly, Kushal Lakhota, Shicong Zhao, Mridul Mohit, Barlas Oguz, Abhinav Arora, Sonal Gupta, Christopher Dewan, Stef Nelson-Lindall, and Rushin Shah. 2018. Pytext: A seamless path from nlp research to production. arXiv preprint arXiv:1812.08729.

Mohamed Ishmael Belghazi, Aristide Baratin, Sairajeswar, Sherjil Ozair, Yoshua Bengio, Aaron Courville, and R Devon Hjelm. 2018. Mine: mutual information neural estimation. arXiv preprint arXiv:1801.04062.

Tom Bocklisch, Joey Faulkner, Nick Pawlowski, and Alan Nichol. 2017. Rasa: Open source language understanding and dialogue management. arXiv preprint arXiv:1712.05181.

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

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

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, New Orleans, Louisiana. Association for Computational Linguistics.

Daniel Guo, Gokhan Tur, Wen-tau Yih, and Geoffrey Zweig. 2014. Joint semantic utterance classification and slot filling with recursive neural networks. In 2014 IEEE Spoken Language Technology Workshop (SLT), pages 554–559. IEEE.

Dilek Hakkani-Tür, Gökhan Tür, Asli Celikyilmaz, Yun-Nung Chen, Jianfeng Gao, Li Deng, and Ye-Yi Wang. 2016. Multi-domain joint semantic frame parsing using bi-directional rnn-lstm. In Inter-speech, pages 715–719.

R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. 2019. Learning deep representations by mutual information estimation and maximization. In International Conference on Learning Representations.

Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907.

Gakuto Kurata, Bing Xiang, Bowen Zhou, and Mo Yu. 2016. Leveraging sentence-level information with encoder lstm for semantic slot filling. arXiv preprint arXiv:1601.01530.

John Lafferty, Andrew McCallum, and Fernando CN Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. Proceedings of the Eighteenth International Conference on Machine Learning.

Bing Liu and Ian Lane. 2016. Attention-based recurrent neural network models for joint intent detection and slot filling. arXiv preprint arXiv:1609.01454.

Yijin Liu, Fandong Meng, Jinchao Zhang, Jinao Xu, Yufeng Chen, and Jie Zhou. 2019. Gcdt: A global context enhanced deep transition architecture for sequence labeling. arXiv preprint arXiv:1906.02437.

Baolin Peng and Kaisheng Yao. 2015. Recurrent neural networks with external memory for language understanding. arXiv preprint arXiv:1506.00195.

Mirco Ravanelli and Yoshua Bengio. 2018. Learning speaker representations with mutual information. arXiv preprint arXiv:1812.05181.

Ngoc Thang Vu. 2016. Sequential convolutional neural networks for slot filling in spoken language understanding. arXiv preprint arXiv:1606.07783.

Puyang Xu and Ruhi Sarikaya. 2013. Convolutional neural network based triangular crf for joint intent detection and slot filling. In 2013 IEEE Workshop on Automatic Speech Recognition and Understanding, pages 78–83. IEEE.

Kaisheng Yao, Baolin Peng, Yu Zhang, Dong Yu, Geoffrey Zweig, and Yangyang Shi. 2014. Spoken language understanding using long short-term memory neural networks. In 2014 IEEE Spoken Language Technology Workshop (SLT), pages 189–194. IEEE.

Lin Zhao and Zhe Feng. 2018. Improving slot filling in spoken language understanding with joint pointer and attention. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 426–431.