

# A MULTICHANNEL CONVOLUTIONAL NEURAL NETWORK FOR CROSS-LANGUAGE DIALOG STATE TRACKING

*Hongjie Shi, Takashi Ushio, Mitsuru Endo, Katsuyoshi Yamagami and Noriaki Horii*

Interactive AI Research Group, Panasonic Corporation, Osaka, Japan

## ABSTRACT

The fifth Dialog State Tracking Challenge (DSTC5) introduces a new cross-language dialog state tracking scenario, where the participants are asked to build their trackers based on the English training corpus, while evaluating them with the unlabeled Chinese corpus. Although the computer-generated translations for both English and Chinese corpus are provided in the dataset, these translations contain errors and careless use of them can easily hurt the performance of the built trackers. To address this problem, we propose a multichannel Convolutional Neural Networks (CNN) architecture, in which we treat English and Chinese language as different input channels of one single CNN model. In the evaluation of DSTC5, we found that such multichannel architecture can effectively improve the robustness against translation errors. Additionally, our method for DSTC5 is purely machine learning based and requires no prior knowledge about the target language. We consider this a desirable property for building a tracker in the cross-language context, as not every developer will be familiar with both languages.

**Index Terms**— Convolutional neural networks, multichannel architecture, dialog state tracking, dialog systems

## 1. INTRODUCTION

Dialog state tracking is one of key sub-tasks of dialog management, whose goal is to transfer human utterances into a slot-value representation (dialog state) that is easy for computer to process and track the information that appeared in the dialog. To provide a common testbed for this task, the series of Dialog State Tracking Challenges (DSTC) was initiated [1]. This challenge has already been held for four times, during which it provided a very valuable shared recourse for the research in this field and helped to improve the state-of-the-art. Since the fourth challenge (DSTC4 2015), the target of dialog state tracking has been shifted from human-machine dialog to human-human dialog, which significantly increases the difficulty of the dialog state tracking task because of the variety and ambiguity in the human-human dialog. One lesson we learned from DSTC4 is the difficulty of building a high performance tracker for human-human dialog with very limited training corpus, no matter whether using machine learning or

hand-crafted rule-based approaches [2, 3]. This is a very unfavorable situation because building hand-annotated training corpus is very expensive, time-consuming and requires human experts. Not to mention the collection of a new corpus for each language other than English if we need to build trackers for a new language.

The DSTC5 proposed a new challenge based on using the rapidly advancing machine translation (MT) technology, one may be able to adapt the built tracker to a new language with limited training or development corpus in that language. We find this idea very attractive because not only it can reduce the cost of new language adaptation, but also it provides the possibility of building a tracker with cross-language corpus. For example it can be very useful for developing the tourist information systems because one may have corpus collected from different language speakers (i.e. tourists from different countries): for each language, the amount of corpus may be very limited, but together it can be large enough for a good training. On the other hand, although the machine translation technology has achieved great progress recently, the translation quality is still not satisfactory [4]. A conventional monolingual tracker trained on the computer-generated translations may lead to an imperfect model, and it can only accept the translations from other languages as input which will also degrade the performance.

To address these problems, we propose a model that can be trained with different languages at the same time, and use both original utterances and their translations as input source for the dialog state tracking. In such way, we can avoid building the tracker only based on computer-generated translations, and maximize the use of all possible input languages to increase the robustness to translation errors. This paper is organized as follows. Sect.2 briefly describes the dataset and the dialog state tracking problem; Sect.3 presents an overview of our method and explains in detail our multichannel CNN model. Sect.4 presents the evaluation results with analysis and discussion. Sect.5 concludes our work and proposes future improvements.

## 2. DATASET AND PROBLEM DESCRIPTION

The fifth Dialog State Tracking Challenge (DSTC5) uses the whole dataset (including train/dev/test datasets) from the

**Table 1.** Example of a transcription and dialog state label for a sub-dialog segment in the topic of ‘Accommodation’.

Transcription	Dialog state label
<b>Guide:</b> Let’s try this one, okay? <b>Tourist:</b> Okay. <b>Guide:</b> It’s InnCrowd Backpackers Hostel in Singapore. If you take a dorm bed per person only twenty dollars. If you take a room, it’s two single beds at fifty nine dollars <b>Tourist:</b> Um. Wow, that’s good. <b>Guide:</b> Yah, the prices are based on per person per bed or dorm. But this one is room. So it should be fifty nine for the two room. So you’re actually paying about ten dollars more per person only. <b>Tourist:</b> Oh okay. That’s- the price is reasonable actually. It’s good.	<b>INFO:</b> Pricerange  <b>PLACE:</b> InnCrowd Back- packers Hostel

DSTC4 as the training dataset. This dataset contains 35 dialog sessions on tourist information for Singapore collected from English speakers. Besides the training dataset, a development set which includes 2 dialog sessions collected from Chinese speakers is provided for testing and tuning the trackers’ cross-language performance before the final evaluation. Both the training and development sets are labelled with the dialog state tags and come with 5-best hypothesis of English or Chinese translations by a machine translation systems. In the evaluation phase of the challenge, a Test set including 8 unlabeled Chinese dialogs is distributed to each participant, and all prediction results submitted by each participant are evaluated by comparing with the true labels. The test dataset also includes 5-best English translations which are generated by the same machine translation system as the training/development dataset.

The dialog state in this challenge is defined by the same ontology used in DSTC4, which contains 5 topic branches with different slot sets (Table2). These topic-slot combinations indicate the most important information mentioned in that topic, for example the ‘CUISINE’ slot under the topic of ‘FOOD’ refers to the cuisine types, while the ‘STATION’ slot for the topic of ‘TRANSPORTATION’ refers to the train stations. In total there are 30 such topic-slot combinations, and all possible values for each topic-slot are given as a list in the ontology. The main task of DSTC5 is to predict the proper value(s) for each slot given the current utterance and its topic, with all dialog history prior to that turn (e.g. Table1).

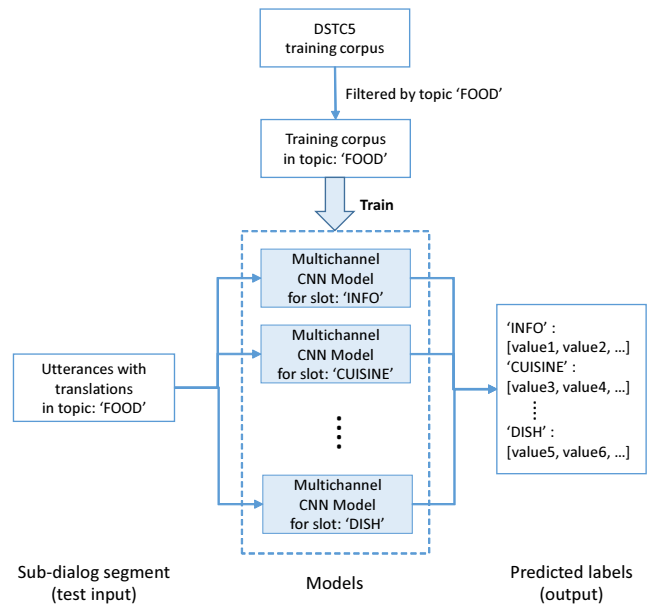
**Table 2.** List of slots for each topic.

Topic	SLOT
Food	INFO, CUISINE, TYPE_OF_PLACE, DRINK, PLACE, MEAL_TIME, DISH, NEIGHBOURHOOD
Attraction	INFO, TYPE_OF_PLACE, ACTIVITY, PLACE, TIME, NEIGHBOURHOOD
Shopping	INFO, TYPE_OF_PLACE, PLACE, NEIGHBOURHOOD, TIME
Transportation	INFO, FROM, TO, STATION, LINE, TYPE, TICKET
Accommodation	INFO, TYPE_OF_PLACE, PLACE, NEIGHBOURHOOD

### 3. METHOD

In DSTC4 we proposed a method which is based on the convolutional neural networks originally proposed by Kim [5]. By this method we were able to achieve the best performance for tracking the INFO slot. The CNN model we used in this method was modified from the origin by adding a structure of multi-topic convolutional layer, so that it can better handle the information presented in different dialog topics. This model is characterized by its high performance for limited training data, because it can be trained across various topics. More details about this multi-topic model can be found in [2].

In DSTC5 the training data is 75% more than DSTC4, therefore the situation of limited training data is improved. In order to focus more on the new cross-language problem and keep our method simple, instead of using the more complex multi-topic model we proposed last time, we trained individual CNN model for each slot-topic combination. That is, for example the ‘INFO’ slot in the topic of ‘FOOD’ and the same ‘INFO’ slot in the topic of ‘SHOPPING’ will be trained by two independent models. This is the major difference from the last time, where we trained one single model for all topics. With this new scheme, we can set the hyperparameters in each model for each slot/topic to be exactly the same, so that our method is scalable, universally applicable and easy to tune. Fig 1 is a simple diagram illustrating our method.



**Fig. 1.** Overview of our method (for the ‘FOOD’ topic).

#### 3.1. Motivation

The biggest challenge of DSTC5 is that the training and test corpora are originally collected in different languages. Since

both computer-generated Chinese and English translation are provided in the training and test dataset respectively, one straightforward approach is to train a model with English corpus and use it for the English translation in the test data. Alternatively, a model trained on Chinese translations in the training dataset can be used for Chinese utterances in the test data. However, both methods will waste the originally collected utterances either in the training or the test data. In order to fully utilize the corpus resource in both English and Chinese languages, we proposed the following multi-channel model which can be regarded as a combination of both English and Chinese models.

### 3.2. Model architecture

Our model is inspired by the multichannel convolutional neural networks commonly used in the image processing [6]. Instead of RGB channels used for color images, we apply each input channel to each different language source. In this model, the input of each channel is a two dimensional matrix, each row of which is the embedding vector of the corresponding word:

$$\mathbf{s} = \begin{bmatrix} - \mathbf{w}_1 - \\ - \mathbf{w}_2 - \\ \vdots \\ - \mathbf{w}_n - \end{bmatrix}, \quad (1)$$

where  $\mathbf{w}_i \in \mathbb{R}^k$  is the embedding vector for the  $i$ -th word in the input text. This 2-dimensional array  $\mathbf{s}$  is a matrix representation of the input text. We used three different word embedding in our model — two for Chinese and one for English. The details of these embedding will be explained later in the Sect. 3.3. For each channel, a feature map  $\mathbf{h} \in \mathbb{R}^{n-d+1}$  is obtained by convolving a trainable filter  $\mathbf{m} \in \mathbb{R}^{d \times k}$  with the embedding matrix  $\mathbf{s} \in \mathbb{R}^{n \times k}$  using the following equation:

$$\mathbf{h} = f(\mathbf{m} * \mathbf{s} + \mathbf{b}). \quad (2)$$

Here  $f$  is a non-linear activation function<sup>1</sup>;  $*$  is the convolution operator and  $\mathbf{b} = (b, \dots, b) \in \mathbb{R}^n$  is a bias term. The maximum value of this feature map  $\hat{h} = \max\{\mathbf{h}\}$  is then selected by the max-pooling layer. This is the process how one filter extracts one most important feature from the input matrix. In this model, multiple filters are used in each channel to extract multiple features. These features then form the pooling layer which are passed to a fully connected layer for prediction.

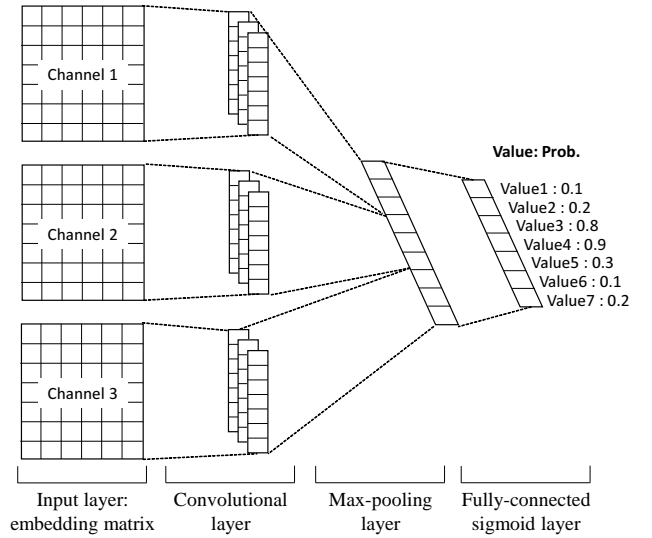
The idea of multichannel model is to connect those extracted features from different channels before the final output, so that the model can use richer information obtained from different channels. The fully connect layer in multichannel model follows the equation:

$$\mathbf{y} = S(\mathbf{w} \cdot (\hat{\mathbf{h}}_{\text{ch1}} \oplus \hat{\mathbf{h}}_{\text{ch2}} \oplus \hat{\mathbf{h}}_{\text{ch3}}) + \mathbf{b}), \quad (3)$$

<sup>1</sup>We used rectified linear unit (ReLU) for this activation function.

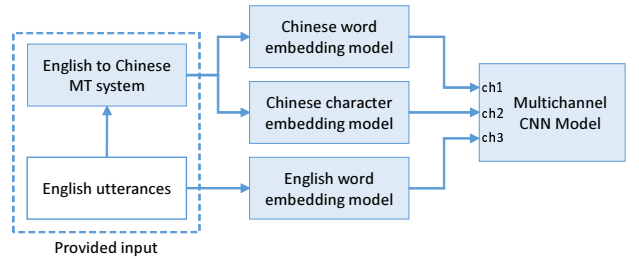
where  $S$  is the sigmoid function;  $\oplus$  is the concatenation operator and  $\hat{\mathbf{h}}_{\text{chn}} = (\hat{h}_1, \dots, \hat{h}_m)$  is the penultimate layer of the  $n$ -th channel.

Notice that in the original paper of Kim, a multi-channel architecture has also been proposed. The main difference between our model and their model is that we use different sets of filters for each channel, while in their model the same filter set are applied to all channels. The reason for this modification is that the word embedding for different languages can

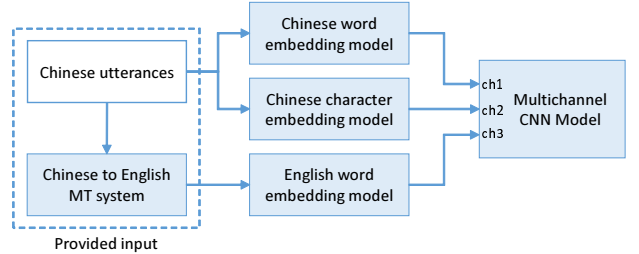


**Fig. 2.** Multichannel CNN model architecture for three input channels.

**For training corpus (English):**



**For development/test corpus (Chinese):**



**Fig. 3.** Pre-processing of the input utterances.

vary greatly, for example the same (or nearly the same) embedding vector in different language models may correspond to irrelevant words with very different meanings. Using different sets of filters ensures that proper features can be extracted in each channel no matter how the word embedding varies among different languages.

### 3.3. Embedding models

The word2vec [7] is one of the most common methods for producing word embeddings. In DSTC5, we applied this method and trained three different models with different training corpus. The details of these models are listed as below:

1. English word model: 200-dimension word2vec model trained on English Wikipedia, with all text split by space and all letters lowercased. This model contains 253854 English words.
2. Chinese word model: 200-dimension word2vec model trained on Chinese Wikipedia, with all text split by word boundary using ‘jieba’ module<sup>2</sup>. This model contains 457806 Chinese words and 53743 English words appeared in the Chinese Wikipedia.
3. Chinese character model: 200-dimension word2vec model trained on Chinese Wikipedia with all text split into single Chinese character. This model contains 12145 Chinese characters and 53743 English words appeared in the Chinese Wikipedia.

The reason why we trained two models for Chinese language is because identifying word boundaries in Chinese is not a trivial task. For Chinese, the smallest element with meaning (word) varies from one single Chinese character to several concatenated Chinese characters, and the task for Chinese word splitting usually involves parsing the sentence and the state-of-the-art method still cannot achieve perfect accuracy. For this reason the Chinese word model may contain incorrect vocabularies and is not capable of handling unseen Chinese character combinations. On the other hand, the Chinese character model does not rely on word segmentation so that the model is error-free, and also it can easily deal with unseen words. However, since the Chinese character model ignores the word boundaries, the resulting embedding vector may not be able to reflect the precise meaning of each word.

## 4. RESULTS

The results of the proposed method along with the scores of other teams are shown in the Table 3. Our multichannel CNN model achieves the best score among all 9 teams: the result of entry-3 outperforms the second best team by 50%

<sup>2</sup><https://github.com/fxsjv/jieba>

(0.0956/0.0635) in Accuracy and 15% (0.4519/0.3945) in F-measure with the sub-dialog evaluation. Our submitted five entries are the results of 5 different hyperparameters settings which are determined by a rough grid search<sup>3</sup>, and those settings are summarized in Table 4. Compared these results with each other, one can easily tell that among these hyperparameters the dropout rate is a key factor. The dropout is known as a technique for reducing overfitting in neural networks [9], and in our case reducing the dropout rate always improves the Precision while degrading the Recall score. One explanation for this is that an over-fitted model only outputs the same labels for the data which are very similar to the training data, and therefore decreases its generalization to unseen data. On the other hand, further decreasing the dropout rate does not improve the overall performance, whose results and parameter settings are also shown in the table as ‘Additional Expt. #5&6’.

**Table 3.** Evaluation results (subdialog-level) on DSTC5 test dataset.<sup>4</sup>

Tracker	Accuracy	Precision	Recall	F-measure
Multichannel #3	0.0956	<b>0.5643</b>	0.3769	<b>0.4519</b>
Multichannel #4	0.0872	0.5427	0.3842	0.4499
Multichannel #0	<b>0.0964</b>	0.5217	0.3849	0.4430
Multichannel #1	0.0712	0.4340	0.4196	0.4267
Multichannel #2	0.0681	0.4216	<b>0.4303</b>	0.4259
Team4-entry2	0.0635	0.3768	0.4140	0.3945
Team1-entry4	0.0612	0.3811	0.3548	0.3675
Team6-entry1	0.0383	0.4063	0.3124	0.3532
Team5-entry0	0.0520	0.3637	0.3044	0.3314
baseline 2	0.0222	0.1979	0.1774	0.1871
Additional Expt #5	0.0949	0.5786	0.3689	0.4505
Additional Expt #6	0.0888	0.5677	0.3712	0.4489

**Table 4.** Main hyperparameter settings for each entry.

Entry # in Table3	#3	#4	#0	#1	#2	#5	#6
Dropout rate	0.4	0.5	0.6	0.8	0.8	0.2	0
Number of filters $\mathbf{m} \in \mathbb{R}^{1 \times k}$ for each channel	1000	600	1000	1400	1000	1000	1000
Number of filters $\mathbf{m} \in \mathbb{R}^{2 \times k}$ for each channel	1000	600	1000	1400	1000	1000	1000
Learning rate	0.001						
Weight decay coefficient for L2 regularization	0.0005						
Training data	Training corpus + 1-best Chinese translation & Development corpus + 1-best English translation						
Training epochs	100						
Test input	Test corpus + 1-best English translation						

<sup>3</sup>A guide for setting these hyperparameters can be found in [8]

<sup>4</sup>These are the evaluation results using ‘Schedule 2’ described in the challenge handbook [1].

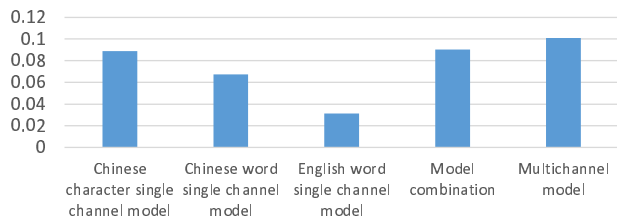
**Table 5.** Example of predicted labels by different models.

Transcription (translation)	Chinese character model	Chinese word model	English word model	Model combination	Multichannel model (= True label)
<b>Guide:</b> 你就可以步行到我们的植物园去散步。 (you can walk to the botanical garden where we go for a walk .) 或者赏花。(or the flowers .)  <b>Tourist:</b> 呃。(er .) 呃植物-(er , plants)	<b>PLACE:</b> Singapore Botanic Gardens	<b>INFO:</b> Exhibit  <b>PLACE:</b> Singapore Botanic Gardens, Bukit Timah Nature Reserve, Hort Park, National Orchid Garden	<b>INFO:</b> Exhibit  <b>PLACE:</b> Singapore Botanic Gardens, Bukit Timah Nature Reserve, Orchard Road, National Orchid Garden  <b>ACTIVITY:</b> Walking	<b>INFO:</b> Exhibit  <b>PLACE:</b> Singapore Botanic Gardens, National Orchid Garden	<b>PLACE:</b> Singapore Botanic Gardens  <b>ACTIVITY:</b> Walking

#### 4.1. Multichannel model & single channel model & model combination

To investigate by how much the proposed multichannel architecture contributes to these results, we compared the performance between the multichannel and ordinary single channel CNN models. For this comparison, we trained three different monolingual single channel CNN models using each of the embedding models mentioned in Section 3.3. These models used the same parameter setting as ‘multichannel #3’ in the Table 4, and were trained only on the 1-best machine translation results. Fig.4 shows the comparing results: the Chinese character model achieves the best overall accuracy among single channel models, while the multichannel model outperforms all three single channel models.

In the earlier DSTC, a simple model combination technique has been used to further improve the predictive performance, where the final output is computed by averaging the scores output by different models [10]. We also applied this method to combine the output from all three single channel models, and the result is also shown in the Fig.4. This simple model combination method does not perform as good as the multichannel model, but considering its simplicity, we still consider it as a good alternative to improve the performance.



**Fig. 4.** Overall predictive accuracy of different models.

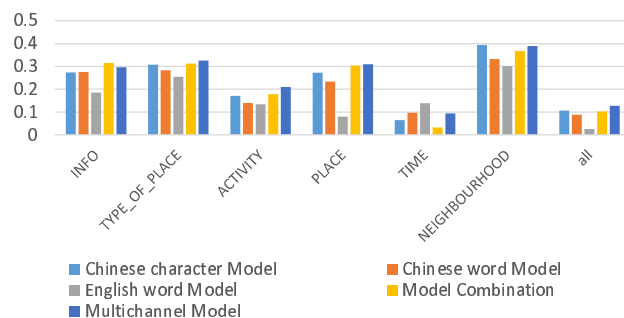
#### 4.2. Discussion

We think the above results can be partially explained from the point of view of ensemble learning. In a multichannel model, each channel provides a different view of the data, and an

example is described using different feature sets that provide different, complementary information about the instance. The fully connected layer in the multichannel model further provides an optimization to use this information for the prediction, and therefore the resulting model can in principle better deal with the translation errors appeared in different channels.

Table 5 is one of the examples that demonstrate this idea. In this particular sub-dialog segment, none of the 3 single channel models is able to output the correct labels, while the multichannel model gives the correct prediction. As seen in this example, the model combination behaves like a simple voting, which means it only picks up the labels that are supported by majority of the single channel models. The multichannel model, on the other hand, is able to selectively choose which language source to trust more for each particular slot or value. As a result, the label of ‘Walking’ is correctly predicted despite it only appearing in the English model’s output, while the ‘Exhibit’ label is correctly rejected even though it is supported by two single channel models out of three.

However the real situation is more complex. When we look at the overall predictive accuracy for each slot (Fig.5), we can find that the performance for each model varies on slots. We consider this is due to the ambiguity caused by machine translation which varies on different subjects. For example, as a time expression in English, 96% of the word “evening” and 43% of the word “night” are translated into



**Fig. 5.** Comparison of accuracy for each slot in the topic of ‘Attraction’.

the same Chinese word “wan shang”. Although this Chinese word does have both meanings of “evening” and “night” in Chinese, there are more precise Chinese terms representing each word. This English to Chinese translation ambiguity immediately increases the difficulty of identifying the values of EVENING and NIGHT in Chinese, which leads to the poor performance of the Chinese model in the slot of TIME.

Another problem is that the translation quality often varies by reversing the translation direction, due to the difference in inflections, word order and grammars [11]. Since the training corpus only contains one translation direction (English to Chinese), the multichannel model is by no means optimized for the reverse translation direction. This may cause the multichannel model to have bias on certain channels, and it can explain why in certain slots the model combination that treats each channel equally works better. A more sophisticated way to train our multichannel model should be firstly training the model with one translation direction and then fine-tuning the model with the other. Unfortunately this is difficult in DSTC5 because the development dataset that can be used for the fine tuning is too limited.

## 5. CONCLUSION

We proposed a multichannel convolutional neural network, in which we treat multiple languages as different input channels. This multichannel model is found to be robust against the translation errors and outperforms any of the single channel models. Furthermore, our method does not require prior knowledge about new languages, and therefore can be easily applied to available corpus resources of different languages. This not only can reduce the cost for the adaption to a new language, but also offers the possibility to build multilingual dialog state trackers with large-scale cross-language corpora.

In this work we applied three different embedding models, while there is one more we did not try — the English character model. There are several character-aware language models proposed recently, which are superior in dealing with subword information, rare words and misspelling [12, 13]. We believe that integrating them into the multichannel model is a promising research direction.

On the other hand, since our method is purely machine learning based, it cannot handle unseen labels in the test data. This is a very important issue especially for a large ontology, because of the difficulty in obtaining large training corpus that covers all concepts. To overcome this disadvantage, future work should include combining machine learning with other approaches, such as hand-craft rules, data argumentation and so on.

## 6. REFERENCES

- [1] Seokhwan Kim, Luis Fernando D’Haro, Rafael E. Banchs, Jason Williams, Matthew Henderson, and Koichiro Yoshino, “The Fifth Dialog State Tracking Challenge,” in *Proceedings of the 2016 IEEE Workshop on Spoken Language Technology (SLT)*, 2016.
- [2] Hongjie Shi, Takashi Ushio, Mitsuru Endo, Katsuyoshi Yamagami, and Noriaki Horii, “Convolutional neural networks for multi-topic dialog state tracking,” *Proceedings of the 7th International Workshop on Spoken Dialogue Systems (IWSDS)*, 2016.
- [3] Franck Dernoncourt, Ji Young Lee, Trung H Bui, and Hung H Bui, “Robust dialog state tracking for large ontologies,” *arXiv preprint arXiv:1605.02130*, 2016.
- [4] Ilya Sutskever, Oriol Vinyals, and Quoc V Le, “Sequence to sequence learning with neural networks,” in *Advances in neural information processing systems*, 2014, pp. 3104–3112.
- [5] Yoon Kim, “Convolutional neural networks for sentence classification,” *arXiv preprint arXiv:1408.5882*, 2014.
- [6] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton, “Imagenet classification with deep convolutional neural networks,” in *Advances in neural information processing systems*, 2012, pp. 1097–1105.
- [7] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean, “Efficient estimation of word representations in vector space,” *arXiv preprint arXiv:1301.3781*, 2013.
- [8] Ye Zhang and Byron Wallace, “A sensitivity analysis of (and practitioners’ guide to) convolutional neural networks for sentence classification,” *arXiv preprint arXiv:1510.03820*, 2015.
- [9] Geoffrey E Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R Salakhutdinov, “Improving neural networks by preventing co-adaptation of feature detectors,” *arXiv preprint arXiv:1207.0580*, 2012.
- [10] Matthew Henderson, Blaise Thomson, and Steve Young, “Word-based dialog state tracking with recurrent neural networks,” in *Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, 2014, pp. 292–299.
- [11] David Vilar, Jia Xu, Luis Fernando dHaro, and Hermann Ney, “Error analysis of statistical machine translation output,” in *Proceedings of LREC*, 2006, pp. 697–702.
- [12] Yoon Kim, Yacine Jernite, David Sontag, and Alexander M Rush, “Character-aware neural language models,” *arXiv preprint arXiv:1508.06615*, 2015.
- [13] Xiang Zhang, Junbo Zhao, and Yann LeCun, “Character-level convolutional networks for text classification,” in *Advances in Neural Information Processing Systems*, 2015, pp. 649–657.