Joint Intent Detection and Slot Filling with Rules

Shiya Ren[1,2], Huaming Wang[1,2], Dongming Yu[1,2],
Yuan Li[1,2], and Zhixing Li[1,2]

[1] Chongqing Key Lab of Computation Intelligence
[2] Chongqing University of Posts and Telecommunications
Chongqing, China 40065
mingyates@163.com

Abstract. Command understanding, the basis of dialogue system and human-computer interaction, roughly involves two sub tasks: intent detection and slot filling. Traditional methods conduct these two tasks in a pipeline fashion, suffering from error propagation. To overcome this problem, this paper proposes a method to solve intent detection and slot filling jointly by back introducing the result of slot filling into intent detection step. Moreover, rules extracted on the training set are used to ease the noise and imbalance problems. The final result are generated by fusing the results of rule-based method and model-based method. Experiments show that our method is effective and achieves best performance among all teams.

Keywords: Command understanding · Intent detection · Slot filling · Dialogue systems.

1 Introduction

Dialogue system is a significant research direction in natural language processing, whose main purpose is to allow the computer to execute commands or answer questions. Command understanding, the basis of dialogue system, involves two sub tasks: intent detection and slot filling. Intent detection task is generally regarded as a sentence classification problem. Recently, the most popular way to accomplish this task is to use the deep learning framework[1, 2]. Slot filling is to find the semantic class labels corresponding to characters, which is typically treated as sequence labeling problem. Conditional Random Fields (CRFs)[3] and Recurrent Neural Networks (RNNs)[4, 5] are usually utilized to fulfill this task.

Existing models follow a pipeline to conduct these two tasks. Intent detection is first conducted on natural language utterances. Only ones classified as positive are delivered to extracted slots. The error of intent detection will propagate to slot filling and harm the final performance. Actually, the result of slot filling can also help to improve the performance of intent detection task. For example, even given that ”陈周” is a singer, this example ”陈周的怎么说？” could be classified into negative. But the truth that ”怎么说” is a song may be uncovered by the

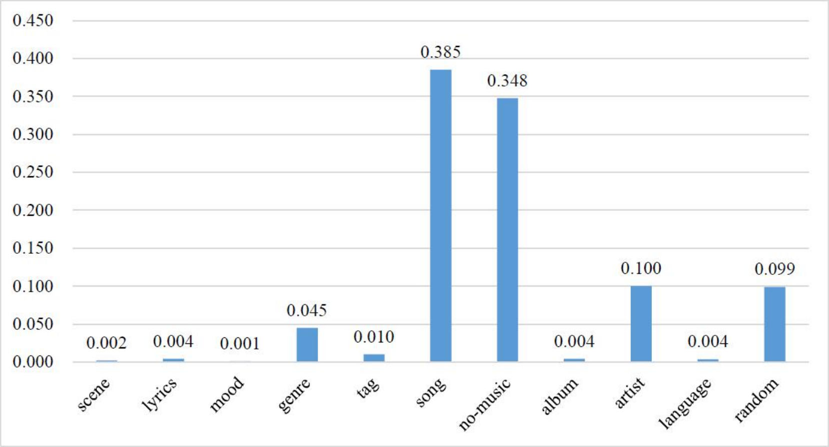

Fig. 1. The proportion of intent and slots.

Table 1. Task illustration

	Command	Time	Objective
s_{-2}	来首刘德华的歌	2017-10-17 19:41:53	-
s_{-1}	你叫什么名字	2017-10-17 19:42:51	-
s_0	请播放周杰伦的稻香	2017-10-17 19:43:53	{”song”:”稻香”,”artist”:”周杰伦”}

slot filling model. If been told that, its intent can be determined more easily and correctly.

To ease the error propagation problem of pipeline model, this paper adopts a piggybacking step which feeds the results of slot extraction into intent classification model. Moreover, there have challenges of lacking labeled data and unbalance as Figure 1 says. To cope with the these problems, we also extract rules adopting statistical technique for slot filling task. Finally, the results of rule-based and model-based slot extraction methods are fused to generate the final result.

This paper proceeds as follows. Section 2 presents the problem restatement in this task formally. Section 3 proposes our model framework for intent detection and slot filling. Section 4 covers the experimental studies. Section 5 concludes the work.

2 Problem Restatement

As Table 1 illustrates, given an utterance $x = (s_{-2}, s_{-1}, s_0)$ where s_0 is current command and s_{-2}, s_{-1} are related histories. The objectives of this task are (1) detecting the intention of s_0. (2) extracting the values of probable slots of s_0 if it is considered as positive.

3 Framework

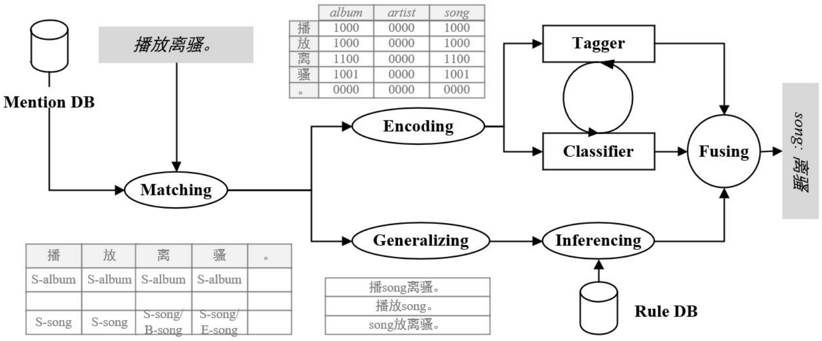

Fig. 2. The framework of proposed method

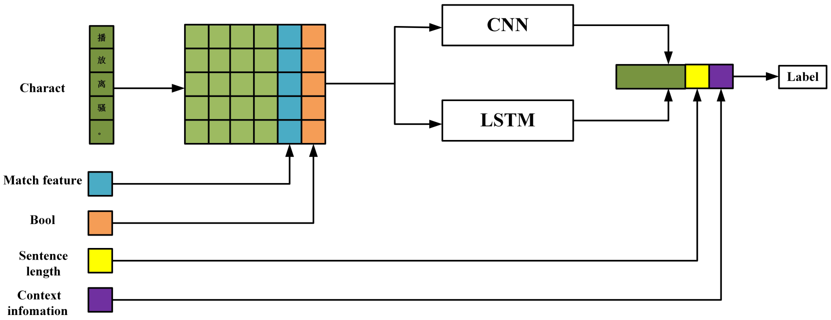

Fig. 3. The model of intent detection

Figure 2 shows the framework of proposed method. It first matches the mentions of songs, artists and albums in the given sentence. Then, the results of matching are encoded into lexical features to feed into classification model for intent detection and tagging model for slot extraction. Besides, the sentence is generalized to a bag of instances by replacing mentions with their types. Rules extracted on the generalized instances are applied on the bag to extract probable slots and values. Finally, the results of rule-based and model-based method are fused.

3.1 Models

Classifier Intuitively, intent detection can be solved as a classification problem. In this paper, it is viewed as a multi-class classification problem. There are

Table 2. Feature information

Names	Functions
Unigram	w[i], w[i - 1], w[i - 2], w[i + 1], w[i + 2]
Bigram	w[i - 1:i + 1], w[i - 2:i], w[i:i + 2], w[i + 1:i + 3],[w[i - 1], w[i + 1]], [w[i - 1], w[i + 2]], [w[i - 2], w[i + 1]], [w[i - 2], w[i + 2]]
Trigram	w[i -2:i+1], w[i:i+3], [w[i - 1], w[i + 1], w[i + 2]], [w[i - 2], w[i - 1], w[i + 1]]
Highgram	[w[i - 2], w[i - 1], w[i + 1], w[i + 2]]
Bool	is_letter, is_punctuation
match_feat(song)	is_S, is_B, is_M, is_E
match_feat(artist)	is_S, is_B, is_M, is_E
match_feat(album)	is_S, is_B, is_M, is_E

Table 3. Examples of rules for open slots

Rule.r	Rule.v	Conf.
点一首song	{song}	1.0
听artist的歌	{artist}	1.0
artist的歌，song	{song,artist}	1.0
听artist	{artist}	0.8
song	{song}	0.74

three coarse classes (*No-music*, *random* and *other*) instead of 9 classes(9 kinds of intentions) for the unbalance problem in the given data. In addition, a hybrid model of Convolutional Neural Networks(CNNs)[6, 7] and Long Short-Term Memory Networks(LSTMs) is designed to predict the class of given utterance. The details of the model can be seen in Figure 3. The embedding weights are pre-trained on the Chinese Wikipedia dump. Context information features are the classification results of prior two sentences and details of other features can be seen in Table 2.

Tagger Slot filling is typically considered as a sequence labeling problem. This paper adopts conditional random fields(CRF), a state-of-the-art model solution for labeling, to extract the values of *song*, *artist* and *album*. We use the python-crfsuite[3] library to train the CRF model. Besides, We utilize character itself, character shape and some information from nearby characters as features. Some mention information such as *artist*, *album* and *song* in the utterance is also used. More details of features are shown in Table 2.

3.2 Rules

All rules are extracted from labeled training set. Considering that there has label noise in the training set, we also check the rules manually. In this paper, *song*, *artist* and *album* are considered as open slots whose value domain is open. The

[3] https://github.com/scrapinghub/python-crfsuite

Table 4. Examples of rules for closed slots

Slot	Value	Positive	Negative
language	ja	{日文的歌,日语歌}	{}
genre	嘻哈	{嘻哈}	{中国有嘻哈,嘻嘻哈哈}
mood	悲伤	{悲伤的歌}	{}
scene	起床歌	{起床歌}	{快乐起床歌}
language	古筝	{古筝}	{古筝曲}

Table 5. The experimental results for this task

Method	$\mathrm{F1}_l$	$\mathrm{F1}_E$	ACC	Score
$\mathrm{C}_{pipeline}$+T	0.8481	0.7686	0.9662	1.2871
C+T	0.8626	0.7931	0.9737	1.3135
C+T+R	0.8667	0.8106	0.9737	1.3256

other 5 slots are treated as closed slot, assuming that their probable values all be presented in the training set. Two kinds of rules are extracted, rules for open slots and rules for closed slots respectively.

Rules for open slots To extract rules for open slots, all sentences in training set are generalized by replacing matched values with its slot type, as Figure 2 shows. Rules are exactly the generalized sentences and their types.

The confidence of rules $s = (Rule.r, Rule.v)$ are defined as

$$Conf = \frac{Freq^P(s)}{Freq^P(s) + Freq^N(s)} \tag{1}$$

where $Freq^P$ is the frequency of s be positive and $Freq^N$ is the frequency of s be negative. The rules whose $Conf$ is lower than 0.5 are discarded. Some of rules are shown in Table 3.

Rules for closed slots To extract rules for closed slots, we start with its values which are treated as key words, sentences contain these key words are collected. Then, related words or phrases are categorized into positive and negative as Table 4 shows. When prediction, an utterance contains no negative phrases but positive phrases is assigned with corresponding slots and values.

4 Experiments

4.1 Datasets and evaluation metrics

Datasets The datasets is mainly from the real user's utterance in the music field and the non-music field of the human-machine dialogue system. The training set consists of 12,000 samples, each sample consists of the three consecutive user utterance and the label of the sample. The testing set consists of 3,000 unlabeled samples.

Table 6. Examples of results

Utterance	Classifier	Tagger	Rule	Result
播放田馥甄的痒。	Other	artist: 田馥甄	artist: 田馥甄, song: 痒	artist: 田馥甄, song: 痒
我要听歌。	Random	-	-	
放一首中文歌。	Other	-	language: zh	language: zh
给高春艳，唱一首丢手绢之歌。	Other	song: 丢手绢之歌	-	song: 丢手绢之歌

Evaluation metrics The intent detection is a classification task. Precision, Recall, and F1-score(F1$_l$) are utilized as evaluation metrics. The slot filling is a sequence labeling task. Two types of performance evaluation are discussed: for the utterances containing entities, Precision, Recall, and F1-score(F1$_E$) are used as evaluation metrics; for the utterances without entity, Accuracy(ACC) is used as the evaluation metric. Overall, we evaluate the quality of our method as follows:

$$Score = \frac{num_l * F1_l + num_E * F1_E + num_a * ACC}{num_t} \tag{2}$$

Where num$_E$ represents the total number of utterances which need to judge the intent, num$_E$ represents the total number of utterances containing the entities, num$_a$ represents the total number of utterances without entity, num$_t$ represents the total number of utterances in dataset.

4.2 Results

Table 5 shows the experimental results on the dataset in this task. We summarize several methods and give their performance. C$_{pipline}$+T and C+T do not use rules and C$_{pipline}$+T does not use the match features when classifying. C+T+R represents the rules be utilized. As shown in Table 5, C+T is better than C$_{pipline}$+T and C+T+R has a significant improvement compared to the first two methods, which prove the importance of the match features and the prior knowledge.

4.3 Case study

Table 6 shows the examples of the results. We can find that combining rules and models can mine more slot values.

5 Conclusion

This paper proposes a solution solving intent detection and slot filling jointly to overcome the error propagation problem of pipeline mechanism. Moreover, rules are extracted from the training set and used to help to produce better result. Experiments show that joint learning significantly improves performances. Besides, rules help to produce better results. The adopted strategies of joint learning and rules fusing are naive. In future, more advanced and effective mechanisms for joint intent detection and slot filling are worth of study.

References

1. Lai S, Xu L, Liu K, et al. Recurrent Convolutional Neural Networks for Text Classification[C]//AAAI. 2015, 333: 2267-2273.
2. Liu B, Lane I. Attention-based recurrent neural network models for joint intent detection and slot filling[J]. arXiv preprint arXiv:1609.01454, 2016.
3. Raymond C, Riccardi G. Generative and discriminative algorithms for spoken language understanding[C]//Eighth Annual Conference of the International Speech Communication Association. 2007
4. Mesnil G, Dauphin Y, Yao K, et al. Using recurrent neural networks for slot filling in spoken language understanding[J]. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 2015, 23(3): 530-539.
5. Liu B, Lane I. Recurrent neural network structured output prediction for spoken language understanding[C]//Proc. NIPS Workshop on Machine Learning for Spoken Language Understanding and Interactions. 2015.
6. P. Xu and R. Sarikaya, “Convolutional neural network based triangular crf for joint intent detection and slot filling,” in Automatic Speech Recognition and Understanding (ASRU), 2013 IEEE Workshop on. IEEE, 2013, pp. 78–83.
7. X. Zhang, J. Zhao, and Y. LeCun, “Character-level convolutional networks for text classification,” in Advances in Neural Information Processing Systems, 2015, pp. 649–657.