The Sogou Spoken Language Understanding System for the NLPCC 2018 Evaluation

Neng Gong, Tongtong Shen, Tianshu Wang, Diandian Qi, Meng Li, Jia Wang, and Chi-Ho Li(✉)

Sogou Inc., Beijing, China
{gongneng,shentongtong,wangtianshu,qidiandian, lizhihao}@sogou-inc.com, lm168260@antfin.com, cdwangjia5@jd.com

Abstract. This report analyzes the problem of spoken language understanding, how the problem is simplified in the NLPCC shared task, and the properties of the official datasets. It also describes the system we developed for the shared task and provides experimental analysis that explains how promising results could be achieved by careful usage of standard machine learning and natural language processing techniques and external resources.

1 Introduction

This paper describes the architecture and the technical details for our system used for the NLPCC 2018 shared task on spoken language understanding, as well as provides experimental analysis for the choices behind our system architecture.

The structure of the paper is as follows. Section 2 examines the very problem of spoken language understanding and also the datasets used in the shared task. Two important insights found in the survey would become the guiding principles in building our SLU system, which is elaborated in Sect. 3. In Sect. 4 experimental results are provided to show the effectiveness of the various techniques used in our system. Further discussions are provided in the Sects. 5 and 6.

2 Problem Definition and Data Analysis

Spoken language understanding (SLU) comprises two tasks, intent identification and slot filling. That is, given the current query along with the previous queries in the same session, an SLU system predicts the intent of the current query and also all slots (entities or labels) associated with the predicted intent. The significance of SLU lies in that each type of intent corresponds to a particular service API and the slots correspond to the parameters required by this API. SLU helps the dialog system to decide how to satisfy the user's need by calling the right service with the right information.

The authors Meng Li and Jia Wang left the company after the shared task evaluation.

M. Zhang et al. (Eds.): NLPCC 2018, LNAI 11108, pp. 454–463, 2018.
https://doi.org/10.1007/978-3-319-99495-6_38

In real use cases of dialog systems, the difficulty of SLU depends on many factors:

1. Complexity of the intent categorization. Taking the query pattern "我要去X" ("I want to go to X") as example. In some simple design this kind of queries can only be assigned the intent of navigation or map search. But for more complicated design the query pattern may also refer to the need of booking flight ticket or train ticket. The difficulty of disambiguation goes up with the complexity of the categorization.
2. World knowledge. The intent of the last example depends, to certain extent, on whether there is an airport or train station at the place X.
3. User's situation. The intent of the last example also depends, to certain extent, on whether it is too far to drive to X from the user's current location.

In the NLPCC shared task, the last two factors are removed as it is almost infeasible to provide comprehensive world knowledge and user's situation information in a 'closed' dataset. The correct interpretation of a query depends solely on linguistic factors, i.e. the query itself and its preceding queries in the same session. Moreover, the intent categorization is confined to a system of 11 types of intent (which can be further grouped into 4 domains: MUSIC, NAVIGATION, PHONE_CALL, and OTHERS) and 15 types of slots, thereby enabling the shared task with a relatively small training dataset.

As in real use cases of dialog systems, the queries in the shared task can be roughly divided into two kinds, viz. queries with intent-indicating salient phrases and queries without. By intent-indicating salient phrase (IISP) it is meant a phrase in the query that shows the intent of the query. E.g. the phrase "我要去" in the query "我要去上海" and the phrases "打电话" in the query "打电话给李小明" are IISPs.

Our examination on the shared task training dataset shows two important insights:

A. For a query with IISP, the intent can be determined by the query itself, without considering context (the preceding queries). That is a simplification of the use case of many dialog system products, where the existence of IISP simply limits the range of possible intents, as shown by the example on last page, "我要去X", of which the correct interpretation depends not only on the IISP but also on context, world knowledge, etc. Such simplification is due to the low complexity of the intent categorization used in the shared task.
B. For a query without IISP, or 'entity-only' queries, such as "五道口", "李小明", the intent can be determined solely by the domain labels of the preceding queries in the same session. That is also a simplification of the real use case of many dialog system product, where, for example, whether the mention of a person name should be interpreted as a phone call request partly depends on whether the name exists in the user's phonebook. As all such factual knowledge cannot be provided in the shared task datasets, the organizing committee released a handful of 'data annotation principles' to interpret entity-only queries.

These two insights imply an SLU architecture for the shared task: divide queries into those with IISPs and those without; the former ones are handled by a 'context-independent' SLU model (i.e. it does not consider the preceding queries in the same session at all), while the latter ones are handled by some context-dependent rules. This SLU architecture is explained in details in the next section.

3 System Details

Figure 1 shows the framework of our SLU system, which consists of the context-dependent rules for entity-only queries and the context-independent model for queries with IISPs. The entire system feeds the query to the rules first. If the rule-based component returns null result, that means the query is judged to contain IISPs and the model-based component will continue to process it. Otherwise, it means the query is regarded as entity-only and the result of the rules is returned as the final output.

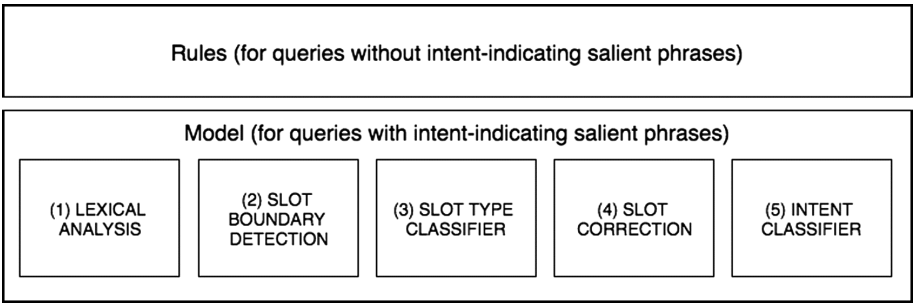

Fig. 1. The Sogou SLU system architecture for the shared task.

3.1 Rule-Based SLU

Our SLU rules apply to only six types of single entities, viz. song titles, singer names, person names, location names, phone numbers, and short commands. Song titles and singer names are defined by the lists provided by the organization committee, and phone numbers are defined by the official annotation principles. There is no official guideline on person names and location names. We used a person name list of around 9 million entries and a location name list of around 70 million entries provided by Sogou Input Method Editor (搜狗输入法) as the definitions of person names and location names respectively.[1] As to short commands like "取消" ("cancel"), we observed the training dataset and collected a dozen of short commands whose correct interpretations depend, in accordance with the annotation principles, on the domain labels of preceding queries.

We imposed altogether 12 rules based on the organizing committee's data annotation principles and our inspection on the training dataset. Each rule is of the form "if the query q is listed in a particular lexicon L, and the preceding queries and their predicted domain labels satisfy certain conditions, then q is assigned a certain intent label and, with the exception of short commands, the entire q is regarded as a slot of the type corresponding to L." The rules are arranged in sequential order in accordance with their priorities. If a query does not fire any rule then it will be processed by the subsequent

[1] If there were no such name lists, the rules could still rely on some lexical analyzer to identify person names and location names, since these two kinds of names are represented by special part-of-speech labels in most lexical analyzers. Section 4.1 will compare the value of the name lists with that of using a lexical analyzer.

model-based SLU. If a query fires a rule then the whole process ends with the result returned by the rule.

3.2 Model-Based SLU Pipeline

The SLU model comprises five components that work in a sequential manner, illustrated in Fig. 2.

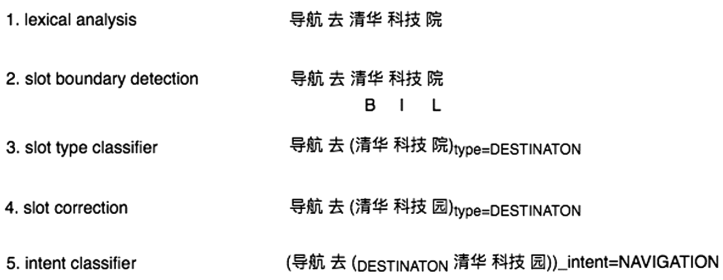

Fig. 2. An example illustrating the mechanism of the model-based SLU.

The first step is lexical analysis, i.e. word segmentation and part-of-speech (POS) tagging. The words and POS labels are used as features in the subsequent models. For the shared task we used HanLP [1] as our Chinese lexical analyzer.

Slot Boundary Detection
The second step is slot boundary detection, i.e. to find out the start and end positions of each slot in the query. This task is considered as sequence labeling using the BILOU scheme[2]. We used both character-based and word-based sequence labeling. The character-based version is a Conditional Random Field (CRF) model with window size 7, using lexical features (the characters themselves) and dictionary features, whereas the word-based version is CRF model with window size 5, using lexical features, POS features, and dictionary features. By dictionary features we mean features of the form "whether the current character/word is the prefix/infix/suffix of some entry in a dictionary of certain entity type". The dictionaries used are the same as those in the rule-based SLU. Each CRF model return n outputs[3], and all these $2n$ outputs are passed to the next step. The rationale behind the use of character-based sequence labeling in addition to the word-based version is to reduce the risk due to word segmentation errors. In Sect. 4 we will see the value of this 'combination' strategy in sequence labeling over the standard practice of doing sequence labeling on word tokens only.

[2] B stands for beginning position, I for inside, L for last, O for outside, and U for unit, i.e. both as beginning and last position.
[3] n=3 in our usage.

Slot Type Classification
The third step is slot type classification, i.e. to identify the type of slot found by the preceding step. Here we used logistic regression with L2-regularizer as the classifier, and the predicted slot, its context characters/words, and the POS labels of its context words (for the word-based input only) as features. As there are several hypotheses from slot boundary detection, where each hypothesis may contain a different number of predicted slots, the slot type classifier calculates the average score of the slots in each hypothesis[4] and chooses the hypothesis with the highest average score as output. This simple strategy of combination is based on the assumption that the best slot boundary detection hypothesis leads to the highest scored slot type classifier output.

Slot Correction
The fourth step is slot correction, i.e. to identify the 'spelling errors' of a slot due to incorrect speech recognition result, etc. A retrieval based method is used. As there is a dictionary for each slot type (c.f. Sect. 3.1), if a slot s with predicted type T is not registered in the dictionary corresponding to T, then s will be matched against the dictionary entries and the entries with lowest edit distance with s are retrieved. This process is carried out twice, one representing s and the entries as Chinese character strings and another representing s and the entries as pinyin strings. The best match is selected by a re-ranker from these two sets of retrieval results.

Intent Classifier
The last step is the intent classifier. It is based on gradient boosting, using the well-known XGBoost tool [2] with its default settings. The features used include the word tokens, query length, and the predicted slots in the preceding steps. Note that the query length feature is proposed by the observation that in the training set queries with more than 20 characters are mainly of the intent OTHERS. That is, the query length feature is a simple hack to distinguish OTHERS from other kinds of intent.

Training Sample Mining
The official training dataset contains about 21,352 samples only and it is tempting to add more samples from other sources. Although we do have two millions labeled queries for Sogou voice interface products, the categorization behind these queries are very different from that used in the shared task, and therefore these assets cannot be directly applied.

Instead we used a cautious method for introducing more samples. For intent identification, the trained classifier is applied to the training dataset itself and the error cases are believed to be under-represented by the training set. For each under-represented query q, it is then matched against our own query set using similarity metrics like ngram overlap ratio, edit distance, etc. Very harsh thresholds on the metrics are used. Those queries that satisfy the thresholds are taken as new samples and they are labeled with the same intent as q.

[4] That is, if a hypothesis contains N slots, where the i-th slot is assigned a score s_i by the slot type classifier, then the score of the hypothesis is $\frac{1}{n}\sum_i s_i$.

New samples for slot filling are introduced in a similar but a bit more complicated way. Two queries of similar structure may not be similar on their surface forms, as the slot values in the two queries could be very different from each other. Therefore, when two queries q and q' are compared their slot values should first be replaced by the slot types, thereby converting the queries into query patterns[5]. If the similarity between query patterns satisfy a very harsh threshold then q′ would be introduced as a new sample, and the slot types in q can be projected to those in q' via the query patterns.

Eventually around 500 samples are chosen as new training instances for intent classifier and around 1000 samples for slot filling.

4 Experiments

In this section we present the results of some experiments showing the effectiveness of the techniques described in the last section. The training and test datasets are the ones released by the organizing committee. The evaluation criterion for intent classifier is F1, which is the same criterion for subtask 2. The evaluation criterion used in all other experiments is precision, the same criterion for subtask 4. In the experiments on slot filling only, the precision is slightly modified so as not to take the intent label into account.

4.1 Rule-Based SLU

Among the 5350 queries in the test set, our context-dependent SLU rules are fired by 1265 of them; that is, the rules are responsible for about 23.6% of the queries. The precision of the rules for both intent identification and slot filling is 96.60%, and the F1 for intent identification only is 96.56%. Moreover, if the identification of person names and location names in our rules are not based on our own name lists but on the lexical analyzer HANLP, the F1 of intent identification drops to 93.03%, and the precision for the entire SLU drops to 92.89%. That is, our name lists reduce nearly half of the errors.

There are two kinds of errors made by the rules. The first kind is due to the incorrect domain labels predicted by SLU model for the preceding queries. The second kind is more difficult to analyze. On the one hand these errors may be explained as the noises in the dictionaries of person names and location names, yet on the other hand it may also be said that the errors are not of the rules but of the test set itself. For example, the query “楼兰” is labeled in the test set as a destination/location name while our system judges it to be a person name. It seems to be rather arbitrary to say that “楼兰” is less likely to be a person name than a location name.

[5] For example, the query in the official training set “我想去北海公园转转” and the query in extra resources “我想去蓝色港湾转一转” are not very similar to each other on their surface forms. Yet because the correct slots “北海公园” and “蓝色港湾” are already labeled in both sets, we could convert the queries into patterns “我想去<slot>转转” and “我想去<slot>转一转”. Similarity can be measured on such query patterns.

4.2 Slot Filling Models

Table 1 shows the experiment results on slot filling (combining the steps of boundary detection, type classification, and correction) of the queries that cannot be handled by rule-based SLU. In the baseline setting, the boundary detector is based only on word-based sequence labeling and it does not use any dictionary features; no extra training samples are used either. The other three settings test the values of three techniques, viz. the combination strategy in boundary detection, the dictionary features in boundary detection, and more training samples. The results show that both the combination strategy and the dictionary features are very effective techniques.

Table 1. Experiments on slot filling.

Setting	Precision (before slot correction)	Precision (after slot correction)
1: Baseline	90.58%	91.70%
2: (1) + combination strategy	91.31%	92.43%
3: (2) + dictionary features	92.39%	93.51%
4: (3) + more samples	92.73%	93.85%

Besides, our slot correction technique consistently makes an absolute improvement of about 1.1%. The errors of slot correction are mainly due to the noises in the person and location name lists.

4.3 Intent Classifier

Table 2 shows the experiment results on intent identification of the queries that cannot be handled by rule-based SLU. The baseline setting is about classifier using word token features only. The other three setting test the values of three techniques, viz. adding the slot types predicted by the slot filling models as extra features, adding query length as extra feature, and adding more training samples. As shown in Table 2, the slot features are the most useful technique, leading to an absolute improvement of nearly 4%. Moreover, only about 500 new training samples (2% of the official training set size) reduced 9% of errors. That shows the importance of data and the effectiveness of our mining techniques.

Table 2. Experiments on intent identification

Setting	F1
1: Baseline	91.36%
2: (1) + slot features	95.26%
3: (2) + query length feature	95.46%
4: (3) + more samples	95.97%

4.4 The Complete SLU System

As to intent identification only, the complete system produces an F1 score of 96.11%. Table 3 shows the experiment results of the complete system on both intent identification and slot filling. The three experiment settings are designed to show the contribution of the external resources (dictionaries and extra training samples).

Table 3. Experiments on the entire SLU system.

Setting	Precision
Rules + Model w/o extra dictionaries and samples	93.42%
Rules + Model with extra dictionaries but not samples	94.24%
Rules + Model with extra dictionaries and samples	94.49%

There are three major types of errors on queries with IISPs. The first type is about very long queries that contain multiple intent, e.g. "唱一首等你下课打电话给六三". This type of errors need a means to split up the query into shorter ones, and/or a multi-label classifier to select the prominent intent. The second type of errors are due to both speech recognition errors and data sparsity, such as the query "放手下课别走", for which the correct form should be "放首下课别走". Although we proposed a mechanism to mine similar queries from our own query log, the speech recognition errors in our log are different from those in the official training set. For this example, there are very few samples for the pattern "放手<歌曲>", and our speech recognition system seldom makes this particular mistake. It is not always successful to mine rare query patterns with certain speech recognition error.

Last but not least, the third type of errors are not of the SLU system but of the labeling consistency of the datasets. For example, it is difficult to explain why "打电话不干" is labeled as PHONE_CALL whereas "打电话差不多" is not. Similarly, it is difficult to explain, for the query "打 电 话 给 我 听 王 蓉", why the slot of CONTACT_NAME should be "我听王蓉" instead of "王蓉".

5 Related Works

There are many reviews on SLU as such [3] or as a component in a complete dialogue system [4]. The various techniques can be divided into two camps, viz. the end-to-end approaches and the pipelined approaches. An end-to-end approach completes both intent identification and slot filling in one process, using techniques like [5–7]. In contrast, a pipelined approach tackles the two tasks one by one, and there are a few choices of the pipeline architecture and various techniques in implementing the components. We decided not to use any of the advanced deep learning techniques because it is found in our pilot experiments that the training data size is too small to produce satisfying results from any deep learning technique. Similar to [8], we found that traditional methods like CRF give the optimal results.

Our slot correction method follows a general spelling correction framework using the noisy channel model [9]. That is, given a misspelled string s, the task is to seek s'

which maximizes the translation model of P($s|s'$) and the language model of P(s'). In our system, the translation model is the edit distances between s and s' on both Chinese character and Pinyin character levels, while the language model is determined by entity lexicons.

6 Summary and Discussions

In this report, we analyzed the problem of spoken language understanding, how the problem is simplified in the NLPCC shared task, and the properties of the official datasets. The data survey inspired us to adopt a hybrid approach to SLU, viz. to deal with entity-only queries with context-dependent rules whereas to deal with queries with intent-indicating salient phrases with context-independent models. A pipelined framework is used to integrate individual models to produce the final output. Although the techniques we used for the model-based SLU are all very 'old-school' ones, and yet our system achieved very promising results. The remaining problems are essentially about quality of dictionaries, long queries with multiple intent, rare query patterns with speech recognition errors, and data labeling consistency issues.

The datasets in the shared task seem to be extracted from the query log of some in-car voice interface product. The SLU problem in industrial products has three characteristics. First, users' commands/requests are mostly entity-specific. Therefore, entity knowledge (or at least lexical resources) is very important to the performance of an SLU system, and our experimental analysis proves the value of entity knowledge.

Secondly, data annotation is the key. Nowadays most speech assistant or voice interface products do not provide user feedbacks as effective as users' clicks in search engines. So, the only way to judge whether an SLU output is correct solely depends on annotators' judgments. We also showed that a few labeled samples in the official datasets are dubious.

The most important one is that the complexity of the techniques required by SLU is proportional to the complexity of intent categorization. For the categorization of only 11 intents in the shared task, we have shown that (for queries with IISPs) SLU can be reliably done with information about the current query only. For a more complicated categorization which contains, following the example in Sect. 2, navigation intent and flight/train-booking intent, the SLU models must appeal at least to context (previous queries) and perhaps even more factors, and will therefore require more complicated techniques. In general, if there are more than one intents which correspond to similar query patterns then more advanced techniques are required.

Based on these observations, we hope that the next shared task would be a more challenging one by having a more fine-grained intent categorization, and a larger and more accurately labeled dataset.

References

1. HanLP: Han Language Processing. https://github.com/hankcs/HanLP
2. Chen, T., Guestrin, C.: XGBoost: a scalable tree boosting system. arXiv:1603.02754 (2016)

3. Tur, G., De Mori, R.: Spoken Language Understanding: Systems for Extracting Semantic Information from Speec. Wiley, Hoboken (2011)
4. Chen, H., Liu, X., Yin, D., Tang, J.: A survey on dialogue systems: recent advances and new frontiers. arXiv:1711.01731 (2017)
5. Jeong, M., Lee, G.G.: Triangular-chain conditional random fields. IEEE Trans. Audio Speech Lang. Process. **16**(7), 1287–1302 (2008)
6. Xu, P., Sarikaya, R.: Convolutional neural network based triangular CRF for joint intent detection and slot filling. In: ASRU (2013)
7. Zhang, X., Wang, H.: A joint model of intent determination and slot filling for spoken language understanding. In: IJCAI (2016)
8. Vukotic, V., Raymond, C., Gravier, G.:. Is it time to switch to word embedding and recurrent neural networks for spoken language understanding? In: Interspeech (2015)
9. Kernighan, M., Church, K., Gale, W.: A spelling correction program based on a noisy channel model. In: COLING (1990)