STRUCTURES FOR SPOKEN LANGUAGE UNDERSTANDING: A TWO-STEP APPROACH

Minwoo Jeong and Gary Geunbae Lee

Department of Computer Science and Engineering,
Pohang University of Science & Technology (POSTECH),
San 31, Hyoja-Dong, Pohang, 790-784, Korea
{stardust, gblee}@postech.ac.kr

ABSTRACT

Spoken language understanding (SLU) aims to map a user's speech into a semantic frame. Since most of the previous works use the semantic structures for SLU, we verify that the structure is valuable even for noisy input. We apply a structured prediction method to SLU problem with comparison to unstructured one. In addition, we present a combined method to embed long-distance dependency between entities in a cascaded manner. On air travel data, we show that our approach improves performance over baseline models.

Index Terms— Spoken language understanding, named entity recognition from speech, structured prediction, long-distance dependency, two-step cascaded approach

1. INTRODUCTION

Spoken language understanding (SLU) addresses the problem of mapping natural language speech into semantic frame to provide a natural human-computer interface. In most dialog systems, a semantic frame is a formal structure of predicted meanings consisting of slot/value pairs [1]. By using the statistical approach, the SLU system automatically learns a semantic mapping from training examples in a supervised learning manner.

Most of the statistical SLU approaches implicitly or explicitly utilize "structures." [2] used the semantic parse trees to extract a meaning of sentence, but it is expensive to build the full-annotated parse tree data. To reduce this problem, [3] presented a method to incorporate the hierarchical structure into the hidden Markov model (HMM) without the semantic parse tree. And [1] proposed a composite model to integrate a HMM with context-free grammar for putting a semantic prior for the semantic structure to the statistical model. All of these previous works assume that the semantic structure is a crucial component of understanding process.

A statistical SLU is based on classification techniques. Recently, structured prediction methods are successfully applied to many text-based natural language applications [4, 5, 6]. Unlike written text, however, spoken language is much noisier and more ungrammatical; hence, we can expect that noise would affect the structures. Therefore, we wish to compare structured prediction with unstructured methods in noisy SLU environments. Our goal is to verify that the structure is also important for SLU problem and structure-based SLU system is robust for noisy inputs.

In this paper, we evaluate unstructured and structured models for SLU, and examine the use of structures for statistical SLU. Moreover, we describe a TwoStep approach for improving both accuracy and speed by exploiting long-distance dependency and reducing the redundancy in structured prediction. A key idea of our method is to filter out the outside or non-entity words before the structured prediction algorithm is applied. We compare our method to baseline models, and demonstrate that our TwoStep approach improves performance and reduces time complexity cost on air travel data.

Show me flights from Denver to New York on Nov. 18th
`FROMLOC.CITY_NAME` = Denver
`TOLOC.CITY_NAME` = New York
`MONTH_NAME` = Nov.
`DAY_NUMBER` = 18th

Fig. 1. An example of a semantic frame representation

2. TASK DEFINITION AND DATA SET

To understand the user's speech, most SLU systems define a semantic frame, i.e. a formal structure of predicted meanings consisting of slot/value pairs [1]. Figure 1, for example, shows semantic frame representations for air travel information system domain. In this example, the slot labels are two-level hierarchical; such as `FROMLOC.CITY_NAME`. This hierarchy differentiates the semantic frame extraction problem from the conventional named entity recognition (NER) problem in information extraction fields. Regardless of the fact that there are some differences between hierarchical SLU and NER, we can still apply the well-known techniques used in NER to an SLU problem. Following [7], the slot labels are drawn from a set of classes constructed by extending each label by three additional symbols, Beginning/Inside/Outside (B/I/O), and a two-level hierarchical slot can be considered as an integrated flattened slot. For example, `FROMLOC.CITY_NAME` and `TOLOC.CITY_NAME` are different in this slot definition scheme.

To study the effect of structure for SLU task, we use the CU-Communicator corpus (air travel) [8]. It consists of almost 2,300 dialogues (approximately 40,000 user utterances) in total. Following [3], most of the single word responses are removed and 13,983 utterances are used for our experiments. The semantic categories in this data set correspond to city names, time-related information, airlines and other miscellaneous entities. The semantic labels are automatically generated by a rule-based parser and are manually corrected. In the data set, the semantic category has a two-level hierarchy: 31 first level classes and 7 second level classes, for 62 class combinations. Figure 2 shows the statistics of hierarchical classes of air travel data set. The data set is 630k words with 29k entities. Roughly, half of the entities are time-related information, a quarter of the entities are city names, a tenth are state and country names, and a fifth are airline

1-4244-0728-1/07/$20.00 ©2007 IEEE

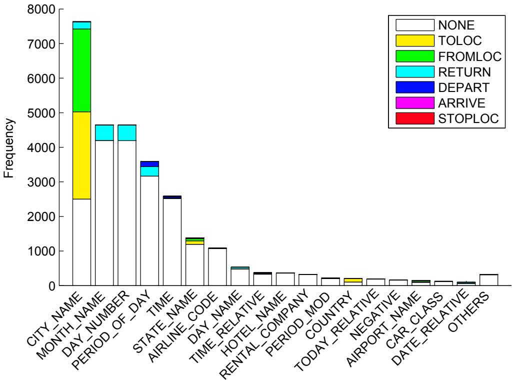


Fig. 2. A corpus statistics of air travel data set

and airport names. For the second level hierarchy, approximately three quarters of the entities are NONE, a tenth are TOLOC, a tenth are FROMLOC, and the remaining are RETURN, DEPART, ARRIVE, and STOPLOC.

3. UNSTRUCTURED AND STRUCTURED MODELS

3.1. MaxEnt and CRF

In the statistical framework, the SLU problem can be stated as a supervised classification problem. Define the input feature vector of discrete random variables as $\mathbf{x} = (x_1, \ldots, x_T)$, and the a B/I/O label sequence for the state variables as $\mathbf{y} = (y_1, \ldots, y_T)$. Then the problem of the SLU tasks is to take a finite set of the observed training examples and construct a classifier $f : \mathbf{x} \rightarrow \mathbf{y}$ that achieves small misclassification errors on test examples. In the probabilistic model, we can evaluate the test examples by using the search operation $\mathbf{y}^* = \arg\max_{\mathbf{y}} p(\mathbf{y}|\mathbf{x})$. Note that the input vector $\mathbf{x}$ includes high-dimensional linguistic features.

If we assume that the output variables $\mathbf{y}$ are all-independent of each other, we can construct a simple unstructured model. In this paper, we use a maximum entropy (MaxEnt) classifier as

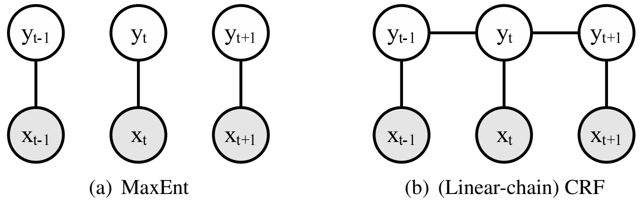
$$p(y_t|\mathbf{x}_t) = \frac{1}{Z(\mathbf{x}_t)} \exp\left(\sum_k \mu_k g_k(y_t, \mathbf{x}_t)\right) \tag{1}$$

where $Z(\mathbf{x}_t)$ is the normalization factor and $g_k(y_t, \mathbf{x})$ is the feature function associated with μ_k. A MaxEnt classifier assigns words to labels one by one, that is, dependencies between labels are ignored. Figure 3 (a) shows the graphical representation of the MaxEnt classifier.

To build a structured model, we use a linear-chain conditional random fields (CRF) which is naturally extended from MaxEnt. It assigns a joint probability distribution over labels which is conditional on the input sequences, where the distribution respects the relations between labels encoded in a linear-chain structure [4]. A linear-chain CRF (Figure 3 (b)) is defined as follows:

$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \exp\left(\sum_{t=1}^{T} \sum_k \lambda_k f_k(y_{t-1}, y_t) + \mu_k g_k(y_t, \mathbf{x}_t)\right) \tag{2}$$

where $Z(\mathbf{x})$ is the normalization factor that makes the probability of all state sequences sum to one. $f_k(y_{t-1}, y_t)$ encodes any aspect of a state transition, $y_{t-1} \rightarrow y_t$ and $g_k(y_t, \mathbf{x})$ encodes the observation (a set of input features) feature, $\mathbf{x}$, centered at the current time, t. λ_k and μ_k are trained parameters associated with the features f_k and g_k.



(a) MaxEnt

(b) (Linear-chain) CRF

Fig. 3. Graphical representations

Note that the key difference between MaxEnt and CRF is the term of $f_k(y_{t-1}, y_t)$ which encodes a linear-structure of output variables, i.e., input sequence $\mathbf{x}$ and output sequence $\mathbf{y}$ are significantly correlated by a left-to-right structure in a linear-chain CRF. In general, a liner-chain CRF outperforms MaxEnt classifier on natural language processing tasks, because it provides a structural nature of language to a statistical method.

3.2. TwoStep Prediction Approach

In this section, we describe a new method for improving the performance by combining MaxEnt and CRF in a cascaded manner. The basic idea is to predict labels twice to reduce the redundancy of the structured model and to exploit long-distance dependency. First, we reduce the multi-class labeling problem to binary-class one. Then, we decide whether to classify the entity (E) or not (O), given word sequence, and filter out the outsides words (O). Next, we connect candidate entity labels, and perform the structured prediction algorithm. We call this strategy a TwoStep approach.

In practice, however, using the true-labeled training data to prune the outside words cannot be robust for new test data; because a noisy word which does not appear in training data seriously affects the result in this case. Alternatively, we used the predicted outputs of the first prediction in training step, and use a cross-validation (CV) technique (not required for test step). We reproduce the binary-labeled training data by using 10-fold CV, and use it to estimate a second model of the system. This is an elaborate technique to reduce the training-test mismatch problem [9].

Figure 4 shows an example, in which the TwoStep approach is applied for SLU problem. In this example, we extract the entities (july, fifth, and morning) and classify them to domain-specific labels via TwoStep method. In first prediction, the system prunes the outside words using binary MaxEnt classifier. In second prediction, we only focus on the candidate entities to classify the labels, which leads us to reduce the redundancy of negative examples and naturally exploit long-distance dependency. Using an original CRF-based system, for example, morning was ambiguously labeled to NONE.PERIOD or DEPART.PERIOD. In the TwoStep method, however, the previous label RETURN.DAY can directly influence the next state; hence, morning is correctly recognized to RETURN.PERIOD. Moreover, we can expect that the computation time would be decreased in both training and inference step.

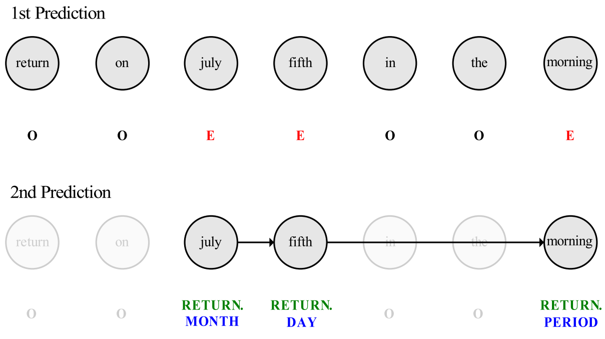


Fig. 4. An example of the TwoStep method

4. RESULTS AND DISCUSSION

4.1. Experimental Setup

For spoken inputs, we use the HTK-based recognition system. We train the recognizer with only the domain-specific speech corpus (not only the acoustic model but also the language model). The reported accuracy for air travel set is approximately 85% [8], but the accuracy of our speech recognizer is 78.42%; because we only used a subset of the data without tuning and the sentences of this subset are longer and more complex than those of the removed ones, most of which are single-word responses. Next, we normalize the transcripts and speech recognition results. After pre-processing some disfluencies and digits (e.g. year, month, day number, and time), we can apply our method to both the text and the spoken inputs in a uniform manner.

Then, all of our results are averaged over 5-fold cross-validation with an 80/20 split of the data. For evaluation, precision (P) and recall (R) are evaluated on a per-entity basis and combined into an F1 score ($F_1 = 2PR/(P + R)$). All models are trained for 100 iterations with a Gaussian prior ($\sigma = 20$). All experiments are implemented in C++ and executed on Linux with XEON 2.8 GHz dual processors and 2.0 Gbyte of main memory.

4.2. Effects of Structures: Comparison

We compare MaxEnt, CRF and our TwoStep method. Our primary question is how structures help the SLU problem even under noisy environments. Figure 5 represents the difference in performance of MaxEnt, CRF, and TwoStep method as a function of training set size. This result shows that structural information improves the performance of SLU in both Text (a) and automatic speech recognition (ASR) input (b). Compared to the unstructured model (MaxEnt; 93.18 (±0.32) for Text and 74.97 (±0.55) for ASR), structured predictions (CRF and TwoStep method) are significantly better in both text and spoken input (CRF; 94.87 (±0.21) for Text and 76.48 (±0.68) for ASR, TwoStep; 95.62 (±0.14) for Text and 77.13 (±0.68) for ASR). Our analysis has found that the noisy context (features) affects the decision of current state, and it is common for both unstructured and structured prediction. In structured models, however, the previous label information plays an important role in discriminating an ambiguous label. Therefore, we conclude that the structural nature of language should be encoded for robust SLU.

The result shows that TwoStep method significantly improves the performance of the system for both Text (a) and ASR inputs (b). The difference in F_1 between TwoStep method and CRF is statistically significant (McNemar's test [10]; $p < 0.001$ for both Text and ASR). Many errors in CRF models are related with hierarchical classes discussed in section 3.2. TwoStep approach explicitly models the long-distance relationship, which efficiently imitates the hierarchical structure, hence it can be an alternative solution for hierarchical models.

However, TwoStep approach performs worse than CRF model with small training examples. The reason is that the first prediction trained with a small amount of data is less accurate (due to out-of-vocabulary (OOV) problem), and prediction errors of first step are propagated to the next stage of the cascaded system. Figure 5 (c) shows the comparison between first and second predictions. For practical spoken dialog application, we assume that the speech recognizer's vocabulary will be limited, hence we believe that it is a conquerable problem. Also our current system only uses lexical-based features to estimate both first and second classifiers, and we can extend the model to exploit more elaborated features (e.g. dictionary for entity lists or acoustic scores) to improve the first step classifier.

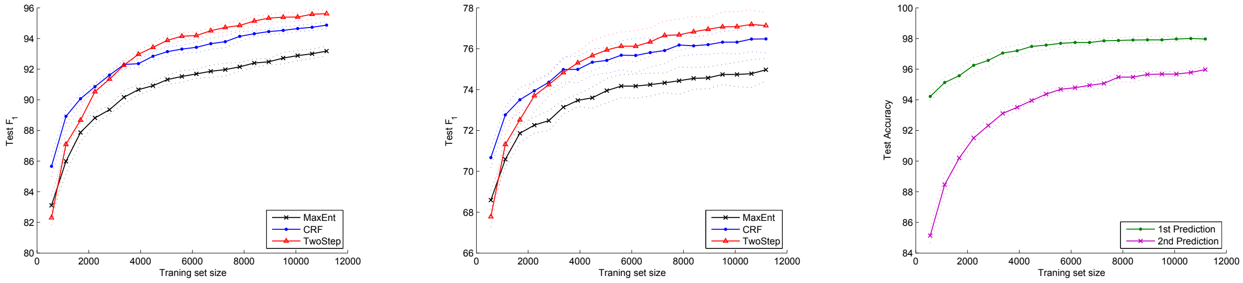
4.3. Robustness for Errors: Simulated ASR

The second goal of this paper is to evaluate the robustness of the TwoStep method for noisy inputs. To verify this, we performed an experiment with synthesis data. We generate synthetic ASR errors; insertion, substitution, deletion, and combined errors. We assume that no error contains content words (named entity) and all errors are uniformly generated. In real ASR system, errors sometimes include the content words and can be captured by n-gram language models; however, this assumption is acceptable to test the model on noisy inputs, because it is reasonable that content words are recognized by named entity even if these words are errors. In this experiment, we also control the word error rate (WER) as a parameter, so we call it as a simulated ASR.

Figure 6 shows that increasing WER decreases the performance for each type of errors. Our TwoStep approach is remarkably robust for insertion errors (a), because our method tries to remove the noisy words in first prediction step and remaining words are highly confident. For substitution errors, performance is lower than the others (b), because replaced words break the dependency between entities. In addition, deletion errors equivalently affect to all models (c). Nevertheless, TwoStep approach is robust to synthetic errors in general (d). We believe that our simulation result is similar to real ASR situation (see section 4.2), and TwoStep method can be extended to word confusion networks or lattices to improve the performance by using acoustic information.

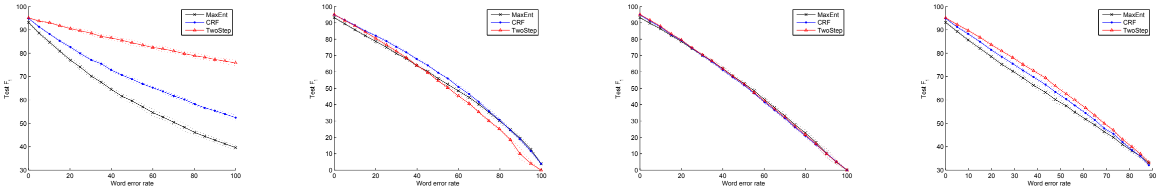
4.4. Efficiency of Training: Time Complexity Cost

In this section, we show that our method dramatically reduces the time complexity for both training and prediction compared with CRF. We measure the training time and corresponding test F_1 in Figure 7. Learning a MaxEnt, CRF and TwoStep model take 36.20 (±0.02), 2941.92 (±128.84) and 1084.27 (±13.07) seconds, respectively. Compared to the full structured model, TwoStep approach requires one third of training time for CRF. In testing with about 2,800 utterances, it takes 0.34 (±0.01), 6.13 (±0.17) and 3.13 (±0.09) for each MaxEnt, CRF and TwoStep model. In air travel data, TwoStep approach successfully achieves two goals; performance and speed.



(a) Learning curve for text input (b) Learning curve for ASR input (c) TwoStep: 1st vs. 2nd prediction

Fig. 5. Results for air travel data



(a) Insertion errors (b) Substitution errors (c) Deletion errors (d) All errors

Fig. 6. Results for synthetic errors

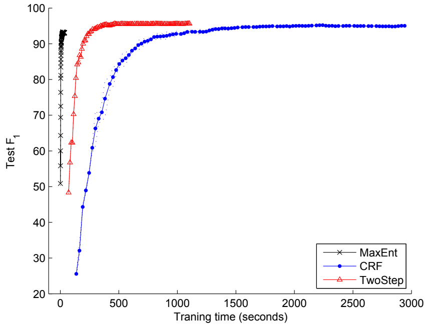


Fig. 7. Training computation time (second) vs. performance (F_1)

5. SUMMARY AND FUTURE WORKS

We have compared unstructured and structured classifiers for SLU task, and discovered that the structural nature of language plays an important role even for a noisy input. To our knowledge, no previous work have presented that structures improve performance over unstructured method under noisy inputs. Moreover, we have presented a method for improving performance by encoding long-distance dependency as an imitation of hierarchical structures. Our method outperforms a full linear-chain CRF model for both accuracy and time complexity cost on air travel data.

In this paper, we have focused on a sequential model such as a linear-chain structure. However, our method can also be naturally applied to arbitrary structured models, thus the alternative is to combine our methods with a hierarchical structure. Applying and extending our approach to other data set is a topic of our future works.

6. ACKNOWLEDGEMENTS

This work was supported in part by MIC & IITA through IT Leading R&D Support Project, and also funded by second BK21 program by the Ministry of Education of Korea.

7. REFERENCES

[1] Y. Wang, L. Deng, and A. Acero, "Spoken language understanding: An introduction to the statistical framework," *IEEE Signal Processing Magazine*, vol. 22, no. 5, pp. 16–31, September 2005.

[2] R. Schwartz, S. Miller, S. Stallard, and J. Makhoul, "Hidden understanding models for statistical sentence understanding," in *Proceedings of the ICASSP*, Washington, DC, USA, 1997.

[3] Y. He and S. Young, "Semantic processing using the hidden vector state model," *Computer Speech & Language*, vol. 19, no. 1, pp. 85–106, January 2005.

[4] J. Lafferty, A. McCallum, and F. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *Proceedings of the ICML*, 2001, pp. 282–289.

[5] B. Taskar, C. Guestrin, and D. Koller, "Max-margin markov networks," in *Proceedings of the NIPS 2003*, 2003.

[6] T. G. Dietterich, "Machine learning for sequential data: A review," *T. Caelli (Ed.) Structural, Syntactic, and Statistical Pattern Recognition; Lecture Notes in Computer Science*, vol. 2396, pp. 15–30, 2002.

[7] L. A. Ramshaw and M. P. Marcus, "Text chunking using transformation-based learning," in *3rd Workshop on Very Large Corpora*, 1995, pp. 82–94.

[8] W. Ward and B. Pellom, "The cu-communicator system," in *Proceedings of IEEE Workshop on Automatic Speech Recognition and Understanding*, 1999.

[9] W. W. Cohen and V. R. de Carvalho, "Stacked sequential learning," in *Proceedings of 19th IJCAI*, 2005, pp. 671–676.

[10] L. Gillick and S. Cox, "Some statistical issues in the comparison of speech recognition algorithms," in *Proceedings of the ICASSP*, 1989, pp. 532–535.