Strategies for Statistical Spoken Language Understanding with Small Amount of Data – an Empirical Study

Ye-Yi Wang

Microsoft Research, Microsoft Corporation
One Microsoft Way, Redmond, WA 98052, USA
yeyiwang@microsoft.com

Abstract

The semantic frame based spoken language understanding involves two decisions – frame classification and slot filling. The two decisions can be made either separately or jointly. This paper compares the different strategies and presents some empirical results in the conditional model framework when only a small amount of training data is available. It is found that while the two pass classification/slot filling solution has resulted in the much better frame classification accuracy, the joint model has yielded better results for slot filling. Application developers need to carefully choose the strategy appropriate to the application scenarios.

Index Terms: spoken language understanding, classification, maximum entropy model, conditional random fields

1. Introduction

Two tasks need to be performed in the semantic-frame based spoken language understanding [1] – *frame classification* picks up the correct frame (corresponding to a task or a domain in a specific application) for an input utterance, and *slot filling* finds the values for the frame specific attributes from the input utterance. For example, in a local-domain application, users may use spoken queries to obtain information about different sub-domains (movies, restaurants, hotels, etc) and for each sub-domain, they may specify the domain specific attribute information (e.g., movie titles and theaters; cuisine and opening hours for restaurants; star rating and check-in dates for hotels.)

Different strategies can be applied in making the classification and slot filling decisions. The one-pass strategy makes both decisions jointly with a single model, while the two-pass strategy performs frame classification first and follows it with slot filling. It has been found that the one-pass joint approach performs better in general. In [2], systematic studies have been performed to compare the two different strategies under the conditional model framework, and it has been found that the one-pass approach outperformed the two-pass strategy with the different factorizations of the proposed "triangular chain CRFs". Similar approach has been adopted in the work in [3].

In this paper, we compare the two different strategies when the amount of training data is very limited. This is a practically important problem. While a statistical SLU model has the advantages of robustness to noise and learnability, it often requires a large amount of labeled training data, which are expensive to obtain. Bootstrapping is a common practice in building such a model, where a small set of labeled training data is collected to train an initial model. The initial model is then deployed to a small population to collect more data – users can correct the understanding errors made by the initial system in the process of performing a task. Such corrections can be logged and used in the feedback loop to improve the model – this process can be repeated for multiple iterations. In such a scenario, the quality of the initial model trained with a small amount of data is essential to reduce the cost (making corrections) for using the system, hence to attract more usage of the system, which will result in more training data and better improved model.

We have discovered, in contrast to what we and other have found previously when training data were abundant, that while the one-pass approach has better slot filling accuracy, it suffers from much higher classification error. The two-pass approach has much improved classification accuracy when the training data is limited, which slightly degrades the slot filling performance. It is thus important to adopt different strategies for the frame-based SLU according to the different application scenarios.

2. Conditional Model in Spoken Language Understanding

We have conducted the study in the conditional model framework. Given an observation $\mathbf{x}$, a conditional model directly models the conditional posterior probability $P(\mathbf{y} \mid \mathbf{x})$ of a possible label $\mathbf{y}$ for $\mathbf{x}$ based on a set of feature functions. Here the label $\mathbf{y}$ can take different forms that yield different models. The conditional probability follows an exponential (log-linear) form in Eq. (1), which is defined with respect to a set of features. A feature $f_k(\mathbf{y}, \mathbf{x})$ in the set is a function of the observation sequence $\mathbf{x}$ and the associated label $\mathbf{y}$.

$$P(\mathbf{y} \mid \mathbf{x}; \Lambda) = \frac{1}{Z(\mathbf{x}; \Lambda)} \exp\left\{\sum_k \lambda_k f_k(\mathbf{y}, \mathbf{x})\right\} \quad (1)$$

Here $\Lambda = \{\lambda_k\}$ is a set of parameters. The value of λ_k determines the impact of the feature $f_k(\mathbf{y}, \mathbf{x})$ on the conditional probability. $Z(\mathbf{x}; \Lambda) = \sum_{\mathbf{y}} \exp\left\{\sum_k \lambda_k f_k(\mathbf{y}, \mathbf{x})\right\}$ is a partition function that normalizes the distribution. Given a set of m labeled training examples $(\mathbf{x}_1, \mathbf{y}_1) \ldots (\mathbf{x}_m, \mathbf{y}_m)$, the model is trained to optimize the following objective function:

$$\begin{aligned} L(\Lambda) &= \frac{1}{m}\sum_{i=1}^{m} \log P(\mathbf{y}_i \mid \mathbf{x}_i; \Lambda) - \frac{1}{2\sigma^2}\|\Lambda\|^2 \\ &= \mathbb{E}_{\hat{P}(\mathbf{x},\mathbf{y})} \log P(\mathbf{y} \mid \mathbf{x}; \Lambda) - \frac{1}{2\sigma^2}\|\Lambda\|^2 \end{aligned} \quad (2)$$

where $\hat{P}(\mathbf{x}, \mathbf{y})$ stands for the empirical distribution of the labeled training samples.

Fancy	French	restaurants	in	Los	Angeles
PriceRange	*Cuisine*	*Type*	*Unl*	*Location*	*Location*

Figure 1: SLU as a sequential labeling problem.

The second term in Eq. (2) regularizes the parameters to keep them from taking extreme values, thus prevents the model from over-fitting the training data. Note that the objective function is a convex function, so a single global optimum exists.

Given a model in Eq. (1), the optimal label $\hat{\mathbf{y}}$ can be obtained according to the following decision rule:

$$\hat{\mathbf{y}} = \arg\max_{\mathbf{y}} P(\mathbf{y} \mid \mathbf{x}) \tag{3}$$

The maximum entropy model (MaxEnt) and the conditional random fields (CRFs) are the two commonly applied conditional models. The former has been broadly used in categorical classification; and the latter has been widely adopted for sequential labeling tasks like part-of-speech tagging, named entity extraction and slot filling for SLU.

2.1. Maximum Entropy Model for Frame Classification

MaxEnt is a special form of the conditional model, in which the label $\mathbf{y}$ is a single random variable whose value represents the target class in a classification task. It is called the maximum entropy model for the following reason: as it was originally formulated, it is a conditional model that satisfies the constraints that the expected value of a feature predicted according to the conditional distribution equals to the empirical value of the feature observed in the training data:

$$\begin{aligned} \mathbb{E}_{\tilde{P}(\mathbf{x})P(\mathbf{y} \mid \mathbf{x})} f_k(\mathbf{x}, \mathbf{y}) &= \sum_{\mathbf{x},\mathbf{y}} \tilde{P}(\mathbf{x}) P(\mathbf{y} \mid \mathbf{x}) f_k(\mathbf{x}, \mathbf{y}) \\ = \mathbb{E}_{\tilde{P}(\mathbf{x},\mathbf{y})} f_k(\mathbf{x}, \mathbf{y}) &= \sum_{\mathbf{x},\mathbf{y}} \tilde{P}(\mathbf{x}, \mathbf{y}) f_k(\mathbf{x}, \mathbf{y}), \end{aligned} \tag{4}$$

for $\forall f_k(\mathbf{x}, \mathbf{y}) \in \mathcal{F}$. Here $\tilde{P}$ stands for the empirical distributions over a training set. There can be many such distributions $P(\mathbf{y} \mid \mathbf{x})$ that satisfies Eq. (4). The maximum entropy principle states that the target distribution should have the maximum entropy subject to the condition of Eq. (4). In other words, the model should make no more assumptions other than those about the expected feature values in Eq. (4).

It has been proven that the maximum entropy distribution subject to the constraints in Eq. (4) has the exponential (log-linear) form [4] shown in Eq. (1). And such a model can be trained by optimizing the objective function in Eq. (2).

2.2. Linear-Chain and Semi-Markov Conditional Random Fields for Slot Filling

The slot filling task can be viewed as a sequential labeling problem, where each word is assigned a label indicating the slot it belongs to, as illustrated by the example in Figure 1.

Such a sequential labeling task can be modeled by the CRFs. The difference between a CRF and a MaxEnt classifier lies in the form of the label $\mathbf{y}$. In the CRF, $\mathbf{y}$ is a sequence of random variables that may be inter-dependent, while in the MaxEnt model, $\mathbf{y}$ is a single random variable representing the target class. The CRF in the original form as in Eq. (1) is unconstrained in the sense that the feature functions are defined on the entire label sequence $\mathbf{y}$. Because the number of all possible label sequences is combinatorial, the model training and inference of an unconstrained CRF is very inefficient. Because of that, it is common to restrict attention to the linear-chain CRFs [5]. The linear chain CRFs impose a Markov constraint on the model topology, and as a consequence, restrict the feature functions to depend only on the labels assigned to the current and the immediately previous states, in the form $f(y_{t-1}, y_t, \mathbf{x}, t)$. The restrictions enable the application of efficient dynamic programming algorithms in model training and inference – yet still support the use of potentially interdependent features defined on the entire observation sequence $\mathbf{x}$.

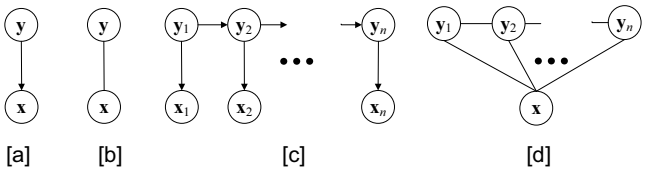

Figure 2: Graphical model representation of generative and conditional models. [a] Generative classification model, such as the Naive Bayesian Classifier; [b] MaxEnt Classifier; [c] HMMs for sequential labeling; and [d] the Linear-chain CRFs.

Figure 2 compares the conditional models to the generative models well known in the speech communities with graphic model representations. Figure 2[a] shows a generative Naive Bayesian classifier, and 2[b] shows the conditional MaxEnt classifier. The difference lies in the directness of the graph – the generative model is a directed graph with edges pointing to the direction of the generative process, while the MaxEnt model is an undirected graph. The implication is that the emission parameters must be a statistical distribution in the generative model. Such a constraint does not apply to the conditional model, which only has a single decision level constraint – $P(\mathbf{y} \mid \mathbf{x})$ must be a statistical distribution properly normalized over all possible values of $\mathbf{y}$. While the constraint in a generative model restricts its expressing power, the decision level constraint in a conditional model makes it discriminative by nature – its objective function can only be optimized by maximizing the posterior probability of the correct state sequence while minimizing the posterior of the competing hypotheses. Figure 2[c] shows the graphic model representation of an HMM, while 2[d] shows that of the linear chain CRF. Here the difference lies in not only the directness of the graph but also the topology of the graph – since the entire observation $\mathbf{x}$ is given (does not need to be generated following the generative process) and observable from all time frames, each state is linked to the entire observation sequence instead of the corresponding frame as in the HMM. Because of this, it is possible to incorporate interdependent, overlapping features defined on the entire observation, which is impossible to do in a generative model.

The Markov assumption in the linear chain CRF restricts it from modeling the long distance dependency over model states. To relax this restriction, the semi-Markov CRF is introduced in [6]. It models the variable-length segmentation of the observation sequence by introducing a random variable $\mathbf{s}$, which is a vector of variable length $k \leq n$, where k stands for the number of segments in the segmentation and n is the length of $\mathbf{x}$. s_i indicates the end position of the ith segment in the segmentation ($s_1 < s_2 < \cdots < s_k = n$). Each segment in the segmentation is assigned a label (model state), and the Markov constraint is imposed so that only the first degree state

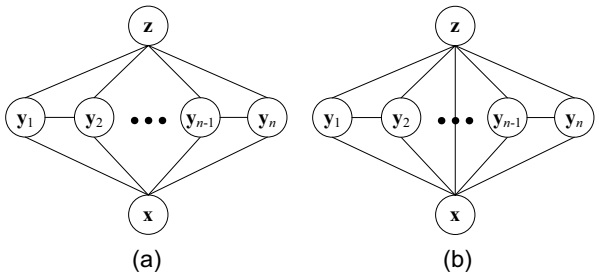

Figure 3: Graphical model representation of the joint model for frame classification and slot filling with differently factorized CRFs. (a) A factorization indirectly models the dependency between the observation **x** and the class **z** via **y**; and (b) a general "triangular chain CRF" directly models the dependency between **x** and **z**.

dependency is modeled between the adjacent segments. Because of this constraints, the features in the model are functions of the form $f(y_{t-1}, y_t, \mathbf{x}, s_{t-1}, s_t)$. Even with this assumption, the Markov constraint is imposed on larger units of segments instead of single tokens/frames, therefore the long distance dependency can be modeled to a certain degree. The semi-Markov CRFs to the linear chain CRFs is like the segment models [7] to the HMMs used in speech recognition.

2.3. Triangular-Chain CRFs for One-Pass Joint Classification/Slot Filling

While CRF is often characterized as a sequential labeling model, it is important to note that such a model can perform non-sequential classification as well. This is done by assigning class specific label sequences to the observation sequences. For example, in [8], a phone-specific state sequence is assigned to the feature vectors at each frame and the phone associated with the state sequence that has the highest posterior probability according to a Hidden State CRF (HCRF) is picked as the target in a phone classification task. Similar idea has been adopted in the application of CRFs to the spoken language understanding tasks, where the state sequences in a CRF represent the class specific slot assignment [3]. This effectively results in the graphic model illustrated by Figure 3(a), where the dependency between the random variable for the class **z** and the observation **x** is indirectly modeled via **y**, the class specific slot label sequence. In [2], a general "triangular chain CRF" has been introduced for joint classification/slot filling for SLU, as illustrated by Figure 3(b). Here the dependency between **z** and **x** is directly modeled with features that are functions of **x** and **z**. Different factorizations of this general model have been investigated in [2]. They were found to have the superior performance over a two-pass solution.

2.4. Strategies for Two-Pass Solutions

In the two-pass solution, the MaxEnt model is applied first to find the target domain of an utterance. Following that, the CRF is used to tag the utterance to extract the values of related slots.

The CRF for slot filling can be trained in two different ways. In the *intra-class training*, the CRF is trained to maximize the conditional probability $P(\mathbf{y} \mid \mathbf{x}, z)$, where z is the known, labeled domain class for **x**. The intra-class training boosts the posterior probability of the labeled state sequences by discrimination against the competing state sequences associated with the same labeled class. In the *inter-class training*, the CRF is trained to maximize the conditional probability $P(\mathbf{y} \mid \mathbf{x})$, in the same way as in the one-pass solution. The inter-class training discriminates against all the competing state sequences associated with all domain classes.

3. Experiments

We conducted experiments for a prototype dialog system in the local domain, which currently includes three sub-domains – restaurant, hotel and weather. We intend to include more sub-domains in the future for the application. The restaurant and hotel sub-domain are quite complicated. They contain more than 15 slots. At the early stage in building the system, we have very limited amount of labeled data. The statistics of the data is show in Table 1.

Table 1: *Statistics about the domain and the data.*

# Samples	303
# Tokens	2070
# Samples w/ Slot	294
# Slot Types	36 (R: 15, H: 18, W: 3)
# Slot Occurrence	763
# Slot Tokens	1278

3.1. Experimental Setup

Four different models have been trained for the task – two for the one-pass approach and two for the two-pass solution. The first one-pass solution (1-pass [a]) adopted the triangular chain model depicted by Figure 3(a), and the second one (1-pass [b]) used the model in Figure 3(b), with features defined in terms of **x** and **z**. Both of the two-pass models used the MaxEnt model for the first-pass classification. One (2-pass [a]) applied the intra-class training for the second-pass CRFs and the other (2-pass [b]) applied the inter-class training.

For slot filling, we applied the semi-Markov CRFs [6]. The features used in the model include the lexical unigram and bigram features, the lexicon features (if a segment of an utterance is in the lexicon of a predefined database of business entities [e.g., hotel names] or not), the lexical features within the 2-word windows of a segment to its left or right, the membership trigram count features (how often a trigram string occurred in a database of business entities) for robust matching against entity names, and the CFG coverage features (to check if a segment matches a chunk covered by a CFG rule for a concept, e.g., date/time expressions, small numbers). The features used by the MaxEnt classifier in the 2-pass solutions are also used in 1-pass [b], which include the lexical unigram and bigram features, the lexicon features, the membership trigram count features and the CFG coverage features. Note that the classifier related features are defined in terms of **x** and **z**, while the slot-filling features are defined in terms of **x** and **y**. Since the number of classes (3) is far less than the number of slots (36), the classifier is a much simpler model.

Since we only have a very small set of data, a 5-fold cross training/testing scheme has been adopted: the data were split into 5 partitions of equal size. Five models were trained with different combinations of four partitions and tested on the remaining partition. The results averaged over the five runs are reported here. We used the classification error rate (CER) for the classification performance, and the word level precision, re-

call and F1 scores for the slot filling performance. The word level precision and recall are defined as follows:

$$Pr = \frac{\text{\# words tagged with the correct slot labels}}{\text{\# words tagged as part of a slot by the model}} \quad (5)$$

Here the words that are not tagged as a slot (the words tagged as "Unl" for "unlabeled") are not counted in the word level precision. And the word level recall is defined as

$$Re = \frac{\text{\# words tagged with the correct slot labels}}{\text{\# words tagged as part of a slot in reference}} \quad (6)$$

3.2. Experimental Results

Table 2 compares the performance of the different models. The two pass solutions have significantly better sub-domain classification accuracy. This is contrary to most of the conclusions in prior art, drawn from the experimental results when training data for statistical models were abundant. This is due to the fact that the decisions made in slot filling are much more complicated than those made in classification, and the classification model is much simpler ($\sim$3,500 parameters) than the slot filling model ($\sim$25,000 parameters). Hence separating the simpler decision from the complicated one makes the classification performance much improved. While the introduction of the classification related features into 1-pass [b] has improved the classification performance to 0.17 CER, it is still significantly worse than the CER of the two-pass approach.

Unfortunately, the much improved classification doesn't help improving the slot filling performance. The one-pass solutions have slightly better slot filling F1 score. Preliminary error analysis reveals that the sentences that had failed the classification are the hard ones that the slot filling CRFs also have trouble with. Hence even though the class was picked correctly, the slot filling precision is much lower on these sentences, which hurt the overall F1 score. This is supported by the separate precision and recall scores in Table 2 – while the recall has been improved slightly (statistically insignificantly), the precision has dropped significantly in the two-pass solutions. It is also interesting to note that the interclass training has resulted in slightly better performance, which reveals the power of the discriminative model – the competition from the state sequences of a class can help reduce the posterior probability of the incorrect state sequences of another class.

Table 2: *Classification error (CER) and word level precision, recall and F1 scores for slot filling for the four different models.*

Model	CER	Precision	Recall	F1
1-pass (a)	0.21	0.67	0.57	0.61
1-pass (b)	0.17	0.67	0.58	0.62
2-pass (a)	0.11	0.62	0.58	0.60
2-pass (b)	0.11	0.63	0.59	0.61

To investigate the impact of more data on the classification/slot filling performance, we have also synthesized training data with a simple grammar and manually checked/corrected the synthetic data. This adds 500 additional labeled data to each training sets in the 5-fold cross-validation experiments. The experiments were repeated and the results are shown in Table 3. Here the CER of 1-pass [a] has much improved as more training data become available. The improvement is larger than those in the two-pass approaches. This tendency is in accordance with the previous findings that the one-pass solution is more promising with more data. Here the difference between the intra-class training and the inter-class training has vanished.

Table 3: *Classification error (CER) and word level precision, recall and F1 scores for slot filling for the four different models.*

Model	CER	Precision	Recall	F1
1-pass (a)	0.17	0.67	0.59	0.63
1-pass (b)	0.17	0.66	0.59	0.62
2-pass (a)	0.10	0.62	0.58	0.60
2-pass (b)	0.10	0.62	0.58	0.60

4. Discussions and Conclusions

Aiming at answering the practical questions about the strategies to be taken at the early stage development of a dialog system, we compared the one-pass and two-pass solutions for domain classification and slot filling in the semantic-frame based spoken language understanding, under the scenario of very limited amount of training data. In contrast to most of the findings in the literature from the experiments when training data are abundant – the two-pass strategy has significantly better classification performance while suffering from a minor degradation in the slot-filling performance. Hence dialog system developers should carefully choose the strategy appropriate to their application scenarios. In many cases, a misclassification is more costly than a slot filling error when there are easy ways to pick the alternative values of a slot (for example, via SLU's n-best results). In such a case, the two-pass strategy should be preferred in the early stage of system deployment when only a small amount of labeled data was used to train the statistical models.

5. References

[1] Y.-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, 2005.

[2] M. Jeong and G. Geunbae Lee, "Triangular-chain conditional random fields," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 16, no. 7, pp. 1287–1302, Sept. 2008.

[3] Y.-Y. Wang, J. Lee, M. Mahajan, and A. Acero, "Combining statistical and knowledge-based spoken language understanding in conditional models." in *Proceedings of the Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, July 2006 2006, pp. 882–889.

[4] A. L. Berger, S. A. Della Pietra, and V. J. Della Pietra, "A maximum entropy approach to natural language processing." *Computational Linguistics*, vol. 22, no. 1, pp. 39–72, 1996.

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

[6] S. Sarawagi and W. W. Cohen, "Semi-Markov conditional random fields for information extraction." in *Advances in Neural Information Processing Systems*, December 5 December 10 2005.

[7] M. Ostendorf, V. Digalakis, and O. Kimball, "From HMMs to segment models: A unified view of stochastic modeling for speech recognition." *IEEE Transactions on Speech and Audio Processing*, vol. 4, no. 5, pp. 360–378, 1996.

[8] A. Gunawardanaand, M. Mahajanand, A. Acero, and J. C. Platt, "Hidden conditional random fields for phone classification," in *Proceedings of INTERSPEECH*. ISCA, September 4-8, 2005 2005, pp. 1117–1120.