Learning Conditional Random Fields from Unaligned Data for Natural Language Understanding

Deyu Zhou[1] and Yulan He[2]

[1] School of Computer Science and Engineering
Southeast University, China
[2] Knowledge Media Institute, Open University
Milton Keynes MK7 6AA, UK
d.zhou@seu.edu.cn, y.he@open.ac.uk

Abstract. In this paper, we propose a learning approach to train conditional random fields from unaligned data for natural language understanding where input to model learning are sentences paired with predicate formulae (or abstract semantic annotations) without word-level annotations. The learning approach resembles the expectation maximization algorithm. It has two advantages, one is that only abstract annotations are needed instead of fully word-level annotations, and the other is that the proposed learning framework can be easily extended for training other discriminative models, such as support vector machines, from abstract annotations. The proposed approach has been tested on the DARPA Communicator Data. Experimental results show that it outperforms the hidden vector state (HVS) model, a modified hidden Markov model also trained on abstract annotations. Furthermore, the proposed method has been compared with two other approaches, one is the hybrid framework (HF) combining the HVS model and the support vector hidden Markov model, and the other is discriminative training of the HVS model (DT). The proposed approach gives a relative error reduction rate of 18.7% and 8.3% in F-measure when compared with HF and DT respectively.

1 Introduction

One of the key tasks in natural language understanding is semantic parsing which maps natural language sentences to complete formal meaning representations. For example, the following sentence could be represented by the predicate formula (also called abstract semantic annotation) as shown below:

I want to return to Dallas on Thursday.
`RETURN(TOLOC(CITY(Dallas)) ON(DATE(Thursday)))`

Early approaches to semantic parsing rely on hand-crafted semantic grammar rules to fill slots in semantic frames using word pattern and semantic tokens. Such rule-based approaches are typically domain-specific and often fragile. In contrast, statistical approaches are able to accommodate the variations found in real data and hence can in principle be more robust. They can be categorized into three types: generative approaches, discriminative approaches and a hybrid of the two.

Generative approaches learn the joint probability model, $P(W, C)$, of input sentence W and its semantic tag sequence C, compute $P(C|W)$ using the Bayes rule,

and then take the most probable semantic tag sequence C. The hidden Morkov model (HMM), being a generative model, has been predominantly used in statistical semantic parsing. It models sequential dependencies by treating a semantic parse sequence as a Markov chain, which leads to an efficient dynamic programming formulation for inference and learning. The hidden vector state (HVS) model [4] is a discrete HMM model in which each HMM state represents the state of a push-down automaton with a finite stack size. State transitions are factored into separate stack pop and push operations constrained to give a tractable search space. The result is a model which is complex enough to capture hierarchical structure but which can be trained automatically from only lightly annotated data. Discriminative approaches directly model posterior probability $P(C|W)$ and learn mappings from W to C. Conditional random fields (CRFs), as one representative example, define a conditional probability distribution over label sequence given an observation sequence, rather than a joint distribution over both label and observation sequences [5]. Another example is the hidden Markov support vector machines (HM-SVMs) [1] which combine the flexibility of kernel methods with the idea of HMMs to predict a label sequence given an input sequence. However, such discriminative methods require fully annotated corpora for training which are difficult to obtain in practical applications. On the other hand, the HVS model can be easily trained from only lightly annotated corpora. However, unlike discriminative models such as the CRFs, it cannot include a large number of correlated lexical or syntactic features in input sentences. It is thus interesting to explore the feasibility to train CRFs from abstract semantic annotations. It is a highly challenge task since the derivation from each sentence to its abstract semantic annotation is not annotated in the training data and is considered hidden.

In this paper, we propose a learning approach based on expectation maximization (EM) to train the CRFs from abstract annotations. This approach works as follows, the CRFs compute expectation based on initial parameters in first step. Based on the expectation results, the CRFs are then constrainedly trained using some general learning algorithms such as stochastic gradient descent (SGD). With re-estimated parameters, the CRFs go to the next iteration until no more improvements could be achieved. Our proposed learning approach has two advantages, one is that the CRFs can be trained from abstract semantic annotations without expensive treebank style annotation data, and the other is that the learning approach is applicable to other discriminative models such as SVMs. To evaluate the performance of the proposed approach, we conducted experiments on the DARPA Communicator Data. Experimental results show that our proposed approach outperforms the HVS model trained also on abstract annotations. Furthermore, the proposed approach outperforms the other two approaches, one is the hybrid framework (HF) combing HVS and HM-SVMs, and the other is discriminative training of the HVS model (DT). The proposed approach gives a relative error reduction rate of 18.7% and 8.3% in F-measure when compared with HF and DT respectively.

The rest of this paper is organized as follows. Section 2 introduces CRFs and the parameter estimation and inference procedures of training CRFs. Our proposed learning procedure to train CRFs from abstract annotations is presented in Section 3. Experimental setup and results are discussed in Section 4. Finally, Section 5 concludes the paper.

2 Conditional Random Fields

Linear-chain conditional random fields (CRFs), as a discriminative probabilistic model over sequences of feature vectors and label sequences, have been widely used to model sequential data. This model is analogous to maximum entropy models for structured outputs. By making a first-order Markov assumption on states, a linear-chain CRF defines a distribution over state sequence $\mathbf{y} = y_1, y_2, \ldots, y_T$ given an input sequence $\mathbf{x} = x_1, x_2, \ldots, x_T$ (T is the length of the sequence) as

$$p(\mathbf{y}|\mathbf{x}) = \frac{\Pi_t \Phi_t(y_{t-1}, y_t, \mathbf{x})}{Z(\mathbf{x})} \tag{1}$$

where the partition function $Z(\mathbf{x})$ is the normalization constant that makes the probability of all state sequences sum to one.

2.1 Inference

The most probable labeling sequence can be calculated by $\text{argmax}_Y P(Y|X;\Theta)$. It can be efficiently calculated using the Viterbi algorithm. Similar to the forward-backward procedure for HMM, the marginal probability of states at each position in the sequence can be computed as,

$$P(y_t = s|\mathbf{x}) = \frac{\alpha_t(y_t = s|\mathbf{x})\beta_t(y_t = s|\mathbf{x})}{Z(\mathbf{x})} \tag{2}$$

where $Z(\mathbf{x}) = \sum_y \alpha_t(y|\mathbf{x})$.

The forward values $\alpha_t(y_t = s|\mathbf{x})$ and backward values $\beta_t(y_t = s|\mathbf{x})$ are defined in iterative form as follows,

$$\alpha_t(y_t = s|\mathbf{x}) = \sum_{y'} \alpha_{t-1}(y_{t-1} = y'|\mathbf{x}) \exp \sum_k \theta_k f_k(y_{t-1} = y', y_t = s, \mathbf{x}) \tag{3}$$

$$\beta_t(y_t = s|\mathbf{x}) = \sum_{y'} \beta_{t+1}(y_{t+1} = y'|\mathbf{x}) \exp \sum_k \theta_k f_k(y_{t+1} = y', y_t = s, \mathbf{x}) \tag{4}$$

3 Training CRFs from Abstract Annotations

To train CRFs from abstract annotations, the expectation maximization (EM) algorithm can be extended to efficiently estimate model parameters. The EM algorithm is an efficient iterative procedure to compute the maximum likelihood (ML) estimate in the presence of missing or hidden data [3]. The EM algorithm is divided into two-step iterations: The E-step, and the M-step. The missing data are estimated given the observed data and current estimate of the model parameters in E-step. In the M-step, the likelihood function is maximized under the assumption that the missing data are known. We now explain how to train CRFs from abstract annotations.

Given a sentence labeled with an abstract semantic annotation as shown in Table 1, we first expand the annotation to the flattened semantic tag sequence as in Table 1(a).

Table 1. Abstract semantic annotation.

Sentence:	I want to return to Dallas on Thursday.
Abstract annotation:	`RETURN(TOLOC(CITY(Dallas)) ON(DATE(Thursday)))`
(a) Flattened semantic tag list:	
`RETURN RETURN+TOLOC RETURN+TOLOC+CITY(Dallas) RETURN+ON RETURN+ON+DATE(Thursday)`	
(b) Expanded semantic tag list:	
`RETURN RETURN+DUMMY RETURN+TOLOC RETURN+TOLOC+DUMMY RETURN+TOLOC+CITY(Dallas) RETURN+TOLOC+CITY(Dallas)+DUMMY RETURN+ON RETURN+ON+DUMMY RETURN+ON+DATE(Thursday) RETURN+ON+DATE(Thursday)+DUMMY`	

In order to cater for irrelevant input words, a DUMMY tag is allowed everywhere in preterminal positions. Hence, the flattened semantic tag sequence is finally expanded to the semantic tag sequence as in Table 1(b).

We first calculate the log likelihood of $L(\Theta)$ with expectation over the abstract annotation as follows,

$$\begin{aligned} L(\Theta;\Theta^t) &= \sum_i^M \sum_{Y_i^u} P(Y_i^u|X_i;\Theta^t) \log P(Y_i^u|X_i;\Theta) \\ &= \sum_i^M \sum_{Y_i^u} P(Y_i^u|X_i;\Theta^t) \sum_t \sum_k \theta_k f_k(y', y, X_i) - \sum_i^k \log Z(X_i) \end{aligned}$$

, where Y_i^u is the unknown semantic tag sequence of the i-th word sequence, and

$$Z(X_i) = \sum_y exp(\sum_t \sum_k \theta_k f_k(y_{t-1}, y_t, X_i)) \tag{5}$$

. It can be optimized using the same optimization method as in standard CRFs training.

Then, to infer the word-level semantic tag sequences based on abstract annotations, Equations 3 and 4 are modified as shown in Equations 6 and 7,

$$\alpha_t(y_t = s|\mathbf{x}) = \begin{cases} 0, & \text{when } g(s, x_t) = 1 \\ \sum_{y'} \Big\{ \alpha_{t-1}(y_{t-1} = y'|\mathbf{x}) & \\ \quad \exp \sum_k \theta_k f_k(y_{t-1} = y', y_t = s, \mathbf{x}) \Big\}, & \text{otherwise} \end{cases} \tag{6}$$

$$\beta_t(y_t = s|\mathbf{x}) = \begin{cases} 0, & \text{when } g(s, x_t) = 1 \\ \sum_{y'} \Big\{ \beta_{t+1}(y_{t+1} = y'|\mathbf{x}) & \\ \quad \exp \sum_k \theta_k f_k(y_{t+1} = y', y_t = s, \mathbf{x}) \Big\}, & \text{otherwise} \end{cases} \tag{7}$$

where $g(s, x_t)$ is defined as follows,

$$g(s, x_t) = \max \begin{cases} 1, \ s \text{ is not in the allowable semantic tag list of } \mathbf{x} \\ 1, \ s \text{ is not of class type and } x_t \text{ is of class type} \\ 0, \text{ otherwise} \end{cases} \tag{8}$$

$g(s, x_t)$ in fact encodes the two constraints implied from abstract annotations. Firstly, state transitions are only allowed if both incoming and outgoing states are listed in the

semantic annotation defined for the sentence. Secondly, if there is a lexical item attached to a preterminal tag of a flattened semantic tag, that semantic tag must appear bound to that lexical item in the training annotation.

4 Experiments

Experiments have been conducted on the DARPA Communicator data [2] which are available for public download. The data contain utterance transcriptions and the semantic parse results from the rule-based Phoenix parser. After cleaning up the data, the training set consist of 12702 utterances while the test set contains 1178 utterances. The abstract annotation used for training and the reference annotation needed for testing were derived by hand correcting the Phoenix parse results. For example, for the sentence "Show me flights from Boston to New York", the abstract annotation would be `FLIGHT(FROMLOC(CITY) TOLOC(CITY))`. Such an annotation need only list a set of valid semantic concepts and the dominance relationships between them without considering the actual realized concept sequence or attempting to identify explicit word/concept pairs. Thus, it avoids the need for expensive tree-bank style annotations.

In all the subsequent experiments, the proposed learning approach is implemented by modifying the source code of the CRF suite[3]. The features such as word features (current word, previous word, next word and so on) and POS features (current POS tag, previous one, next one and so on) are employed. To estimate the parameters of CRFs, the stochastic gradient descent (SGD) iterative algorithm [6] was employed. As discussed in Section 1 that while CRFs can easily incorporate arbitrary features into training, HVS model cannot include a large number of correlated lexical or syntactic features in input sentences. It would be interested to see how CRFs compared to HVS when both are trained from abstract annotations. The proposed CRFs learning approach achieved 92.37% of F-measure, which significantly outperforms HVS. Employing SGD gives a relative error reduction of 36.6%, when compared with the performance of the HVS model where only 87.97% was achieved.

We further compare our proposed learning approach with two other methods. One is a hybrid generative/discriminative framework (HF) [7] which combines HVS with HM-SVMs so as to allow the incorporation of arbitrary features as in CRFs. The same features as listed above were used in HF training. The other is a discriminative approach (DT) based on parse error measure to train the HVS model [8]. The generalized probabilistic descent (GPD) algorithm was employed for adjusting the HVS model to achieve minimum parse error rate. Table 2 shows that our proposed learning approach outperforms both HF and DT. Training CRFs on abstract annotations allows the calculation of conditional likelihood and hence results in direct optimization of the objective function to reduce the error rate of semantic labeling. In the contrary, the hybrid framework firstly uses the HVS parser to generate full annotations for training HM-SVMs. This process involves the optimization of two different object functions (one for HVS and another for HM-SVMs). Although DT also uses an objective function which aims to reduce the semantic parsing error rate. It is in fact employed for supervised re-ranking where the input is the N-best parse results generated from the HVS model.

[3] `http://www.chokkan.org/software/crfsuite/`

Table 2. Performance comparison between the proposed approach and the two other approaches.

	Performance			Relative Error Reduction	
Measurement	HF	DT	Our Approach	Compared with HF	Compared with DT
Recall (%)	90.99	91.47	92.27	14.2	9.4
Precision (%)	90.25	91.87	92.48	22.9	7.5
F-measure (%)	90.62	91.68	92.37	18.7	8.3

5 Conclusions and Future Work

In this paper, we proposed an effective learning approach which can train the conditional random fields without the expensive treebank style annotation data. Instead, it trains the CRFs from only abstract annotations in a constrained way. Experimental results show that 36.6% relative error reduction in F-measure was obtained using the proposed approach on the DARPA Communicator Data when compared with the performance of the HVS model. Furthermore, the proposed learning approach also outperforms two other methods, one is the hybrid framework (HF) combining both HVS and HM-SVMs, and the other is discriminative training (DT) of the HVS model, with a relative error reduction rate of 18.7% and 8.3% being achieved when compared with HF and DT respectively.

References

1. Y. Altun, I. Tsochantaridis, and T. Hofmann. Hidden markov support vector machines. In *International Conference in Machine Learning*, pages 3–10, 2003.
2. CUData. Darpa communicator travel data. university of colorado at boulder. Avaiable from http://communicator.colorado.edu/phoenix, 2004.
3. A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *JOURNAL OF THE ROYAL STATISTICAL SOCIETY, SERIES B*, 39(1):1–38, 1977.
4. Y. He and S. Young. Semantic processing using the hidden vector state model. *Computer Speech and Language*, 19(1):85–106, 2005.
5. J. D. Lafferty, A. McCallum, and F. C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML '01: Proceedings of the Eighteenth International Conference on Machine Learning*, pages 282–289, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc.
6. Y. S. Shai Shalev-Shwartz and N. Srebro. Pegasos: Primal estimated sub-gradient solver for svm. In *Proceedings of the 24th International Conference on Machine Learning (ICML 2007)*, pages 807–814, 2007.
7. D. Zhou and Y. He. A Hybrid Generative/Discriminative Framework to Train a Semantic Parser from an Un-annotated Corpus. In *Proceedings of 22nd International Conference on Computational Linguistics (COLING2008)*, pages 1113–1120, Manchester, UK, 2008.
8. D. Zhou and Y. He. Discriminative Training of the Hidden Vector State Model for Semantic Parsing. *IEEE Transaction on Knowledge and Data Engineering*, 21(1):66–77, 2008.