Hindawi Publishing Corporation
The Scientific World Journal
Volume 2014, Article ID 121650, 11 pages
http://dx.doi.org/10.1155/2014/121650

Research Article

Semi-Supervised Learning of Statistical Models for Natural Language Understanding

Deyu Zhou[1] and Yulan He[2]

[1] *School of Computer Science and Engineering, Key Laboratory of Computer Network and Information Integration, Ministry of Education, Southeast University, Nanjing 210096, China*
[2] *School of Engineering and Applied Science, Aston University, Birmingham B4 7ET, UK*

Correspondence should be addressed to Deyu Zhou; d.zhou@seu.edu.cn

Received 26 March 2014; Revised 27 May 2014; Accepted 21 June 2014; Published 20 July 2014

Academic Editor: Alessandro Moschitti

Natural language understanding is to specify a computational model that maps sentences to their semantic mean representation. In this paper, we propose a novel framework to train the statistical models without using expensive fully annotated data. In particular, the input of our framework is a set of sentences labeled with abstract semantic annotations. These annotations encode the underlying embedded semantic structural relations without explicit word/semantic tag alignment. The proposed framework can automatically induce derivation rules that map sentences to their semantic meaning representations. The learning framework is applied on two statistical models, the conditional random fields (CRFs) and the hidden Markov support vector machines (HM-SVMs). Our experimental results on the DARPA communicator data show that both CRFs and HM-SVMs outperform the baseline approach, previously proposed hidden vector state (HVS) model which is also trained on abstract semantic annotations. In addition, the proposed framework shows superior performance than two other baseline approaches, a hybrid framework combining HVS and HM-SVMs and discriminative training of HVS, with a relative error reduction rate of about 25% and 15% being achieved in F-measure.

1. Introduction

Given a sentence such as "I want to fly from Denver to Chicago," its semantic meaning can be represented as `FROMLOC(CITY(Denver)) TOLOC(CITY(Chicago)).`

Natural language understanding can be considered as a mapping problem where the aim is to map a sentence to its semantic meaning representation (or abstract semantic annotation) as shown above. It is a *structured classification* task which predicts output labels (semantic tag or concept sequences) from input sentences where the output labels have rich internal structures.

Early approaches rely on hand-crafted semantic grammar rules to fill slots in semantic frames using word pattern and semantic tokens [1, 2]. 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, $P(C, S)$, of input sentence S and its semantic tag sequence C, then compute $P(C \mid S)$ using Bayes' rule, and finally take the most probable semantic tag sequence C. The hidden Markov model (HMM), a generative model, has been predominantly employed 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. Discriminative approaches directly model posterior probability $P(C \mid S)$ and learn mappings from S 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 [3]. Another example is the hidden Markov support vector machines (HM-SVMs) [4]

which combine the flexibility of kernel methods with the idea of HMMs to predict a label sequence given an input sequence.

Nevertheless, statistical models mentioned above require fully annotated corpora for training which are difficult to obtain in practical applications. It thus motivates the investigation of train statistical models on abstract semantic annotations without the use of expensive token-style annotations. This is a highly challenging problem because the derivation from each sentence to its abstract semantic annotation is not annotated in the training data and is considered hidden.

A hierarchical hidden state structure could be used to model embedded structural context in sentences, such as the hidden vector state (HVS) model [5], which learns a probabilistic pushdown automaton. However, it cannot incorporate a large number of correlated lexical or syntactic features in input sentences and cannot handle any arbitrary embedded relations since it only supports right-branching semantic structures.

In this paper, we propose a novel learning framework to train statistical models from unaligned data. Firstly, it generates semantic parses by computing expectations using initial model parameters. Secondly, parsing results are then filtered based on a measure describing the level of agreement with the sentence abstract semantic annotations. Thirdly, the filtered parsing results are fed into model learning. With the reestimated parameters, the learning of statistical models goes to the next iteration until no more improvements could be achieved. The proposed framework has two advantages: one is that only abstract semantic annotations are required for training without the explicit word/semantic tag alignment; and another is that the proposed learning framework can be easily extended for training any discriminative models on abstract semantic annotations.

We apply the proposed learning framework on two statistical models, CRFs and HM-SVMs. Experimental results on the DARPA communicator data show that the framework on both CRFs and HM-SVMs outperforms the baseline approach, the previously proposed HVS model. In addition, the proposed framework shows superior performance than two other approaches, a hybrid framework combining HVS and HM-SVMs and discriminative training of HVS, with a relative error reduction rate of about 25% and 15% being achieved in F-measure.

The rest of this paper is organized as follows. Section 2 gives a brief introduction of CRFs and HM-SVMs, followed by a review on the existing approaches for training semantic parsers on abstract annotations. The proposed framework is presented in Section 3. Experimental setup and results are discussed in Section 4. Finally, Section 5 concludes the paper.

2. Related Work

In this section, we first briefly introduce CRFs and HM-SVMs. Then, we review the existing approaches for training semantic parsers on abstract semantic annotations.

2.1. Statistical Models. Given a set of training data $D = \{(S_i, C_i), i = 1, \ldots, N\}$, to learn a function that assigns to a sequence of words $S = \{s^1, s^2, \ldots, s^T\}$, $s^i \in \mathbf{s}$, $i = 1, \ldots, T$, a sequence of semantic concepts or tags $C = \{c^1, c^2, \ldots, c^T\}$, $c^i \in \mathbf{c}$, $i = 1, \ldots, T$, a common approach is to find a discriminant function $F : \mathcal{S} \times \mathcal{C} \rightarrow \mathbb{R}$ that assigns a score to every input $S \in \mathcal{S}$ and every semantic tag sequence $C \in \mathcal{C}$. In order to obtain a prediction $f(S) \in \mathcal{C}$, the function is maximized with respect to $f(S) = \arg\max_{C \in \mathcal{C}} F(S, C)$.

2.1.1. Conditional Random Fields (CRFs). Linear-chain 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 $C = \{c^1, c^2, \ldots, c^T\}$ given an input sequence $S = \{s^1, s^2, \ldots, s^T\}$ (T is the length of the sequence) as

$$p(C \mid S) = \frac{\Pi_t \Phi_t \left(c^{t-1}, c^t, S\right)}{Z(S)}, \tag{1}$$

where the partition function $Z(S)$ is the normalization constant that makes the probability of all state sequences sum to one and is defined as $Z(S) = \Sigma_c \Pi_t \Phi_t(c^{t-1}, c^t, S)$.

By exploiting the Markov assumption, $Z(S)$ can be calculated efficiently by variants of the standard dynamic programming algorithms used in HMM instead of summing over the exponentially many possible state sequences c. $\Phi(c^{t-1}, c^t, S)$ can be factorized as

$$\Phi\left(c^{t-1}, c^t, S\right) = \exp\left(\Sigma_k \theta_k f_k\left(c^{t-1}, c^t, S, t\right)\right), \tag{2}$$

where θ_k is the real weight for each feature function $f_k(c^{t-1}, c^t, S, t)$. The feature functions describe some aspect of a transition from c^{t-1} to c^t as well as c^t and the global characteristics of S. For example, f_k may have value 1 when $\mathrm{POS}(s^{t-1}) = \mathrm{DT}$ and $\mathrm{POS}(s^t) = \mathrm{NN}$, which means that the previous word s^{t-1} has the POS tag "DT" (determiner) and the current word s^t has the POS tag "NN" (noun, singular common). The final model parameters for CRFs are a set of real weights $\Theta = \{\theta_k\}$, one for each feature.

2.1.2. Hidden Markov Support Vector Machines (HM-SVMs). For HM-SVMs [4], the function $F(S, C)$ is assumed to be linear in some combined feature representation of S and C; $F(S, C) := \langle w, \Phi(S, C)\rangle$. The parameters w are adjusted so that the true semantic tag sequence C_i scores higher than all other tag sequences $C \in \mathcal{C}_i := \mathcal{C} \backslash C_i$ with a large margin. To achieve the goal, the following optimization problem is solved:

$$\begin{aligned} \min_{\xi_i \in \mathbb{R}, w \in \mathcal{F}} \quad & \mathrm{Cons} \sum_i \xi_i + \frac{1}{2}\|w\|^2 \\ \text{s.t.} \quad & \langle w, \Phi\left(S, C_i\right)\rangle - \langle w, \Phi\left(S, C\right)\rangle \geq 1 - \xi_i, \\ & \forall i = 1, \ldots N, \quad C \in \mathcal{C} \setminus C_i, \end{aligned} \tag{3}$$

where ξ_i is nonnegative slack variables allowing one to increase the global margin by paying a local penalty on some

outlying examples and Cons dictates the desired tradeoff between margin size and outliers. To solve (3), the dual of the equation is solved instead. The solution $\widehat{w}$ can be written as

$$\widehat{w} = \sum_{i=1}^{N} \sum_{C \in \mathcal{C}} \alpha_i (C)\, \Phi\left(S_i, C\right), \tag{4}$$

where $\alpha_i(C)$ is the Lagrange multiplier of the constraint associated with example i and C_i.

2.2. Training Statistical Models from Lightly Annotated Data. Semantic parsing can be viewed as a pattern recognition problem and statistical decoding can be used to find the most likely semantic representation. The majority of statistical approaches to semantic parsing rely on fully annotated corpora. There have been some prior works on learning semantic parsers that map natural language sentences into a formal meaning representation such as first-order logic [6–10]. However these systems either require a hand-built, ambiguous combinatory categorical grammar template to learn a probabilistic semantic parser [11] or assume the existence of an unambiguous, context-free grammar of the target meaning representations [6, 7, 9, 12, 13]. Furthermore, they have only been studied in two relatively simple tasks, GeoQuery [14] for US geography query and RoboCup (http://www.robocup.org/) where coaching instructions are given to soccer agents in a simulated soccer field.

He and Young [5] proposed the hidden vector state (HVS) model based on the hypothesis that a suitably constrained hierarchical model may be trainable without treebank data whilst simultaneously retaining sufficient ability to capture the hierarchical structure needs to robustly extract task domain semantics. Such a constrained hierarchical model can be conveniently implemented using the HVS model which extends the *flat-concept* HMM model by expanding each state to encode the stack of a pushdown automaton. This allows the model to efficiently encode hierarchical context, but because stack operations are highly constrained it avoids the tractability issues associated with full context-free stochastic models such as the hierarchical HMM. Such a model is trainable using only lightly annotated data and it offers considerable performance gains compared to the flat-concept model.

Conditional random fields (CRFs) have been extensively studied for sequence labeling. Most applications require the availability of fully annotated data, that is, an explicit alignment of sentence and word-level labels. There have been some attempts to train CRFs from a small set of labeled data and a large set of unlabeled data. In these approaches, a training objective is redefined to combine the conditional likelihood of labeled data and unlabeled data. Jiao et al. [15] extended the minimum entropy regularization framework to the structured prediction case so a training objective that combines unlabeled conditional entropy with labeled conditional likelihood is yielded. Mann and McCallum [16] augmented the traditional conditional likelihood objective function with an additional term that aims to minimize the predicted label entropy on unlabeled data. Entropy regularization was employed for semisupervised learning. In [17], a training objective combining the conditional likelihood on labeled data and the mutual information on unlabeled data is proposed. It is based on the rate distortion theory in information theory. Mann and Mccallum [18] used labeled features instead of fully labeled instances to train linear-chain CRFs. Generalized expectation criteria are used to express a preference for parameter settings in which the model distribution on unlabeled data matches a target distribution. They tested their approach on the classified advertisements data set (Classified) [19] consisting of classified advertisements for apartment rentals in the San Francisco Bay Area with 12 fields being labeled for each of the advertisements, including size, rent, neighborhood, and features. With only labeled features, their approach gave a mediocre result with 68.3% accuracy being achieved. With an additional inclusion of 100 labeled instances, the accuracy is increased to 80%. The DARPA communicator data used in our experiment appear to be more complex than the Classified data since semantic annotations in the DARPA communicator data describe embedded structural context in sentences while semantic labels in the Classified data do not represent any hierarchical relations.

3. The Proposed Framework

Given the training data $D = \{(S_1, A_1), \ldots, (S_N, A_N)\}$, where A_i is the abstract annotation for sentence S_i, the parameters Θ will be estimated through a maximum likelihood procedure. The log-likelihood of $L(\Theta)$ with expectation over the abstract annotation is calculated as follows:

$$L(\Theta) = \sum_{i}^{N} \sum_{C_i^u} P\left(C_i^u \mid S_i\right) \log P\left(C_i^u \mid S_i\right), \tag{5}$$

where C_i^u is the unknown semantic tag sequence of the ith word sequence. To learn statistical models, we extended the use of expectation maximization (EM) algorithm to estimate model parameters. The EM algorithm [20] is widely employed in statistical models for parameter estimation when the model depends on unobserved latent variables. Given a set of observed data D, a set of unobserved latent data, or missing values D^u, the EM algorithm seeks to find the maximum likelihood estimation of the marginal likelihood

$$L(D \mid \theta) = \sum_{D^u} p\left(D, D^u, \theta\right) \tag{6}$$

by alternating between performing an *expectation* step and a *maximization* step.

(i) E-step: given the current estimate of the parameters, calculate the expected value for unobserved latent variables or data.

(ii) M-step: find the parameter that maximizes this quantity. These parameter estimates are then used to determine the distribution of the latent variables in the next E-step.

We propose a learning framework based on EM to train statistical models from abstract semantic annotations as

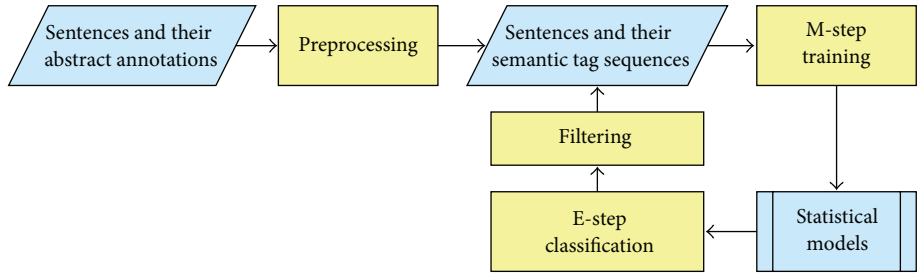


Figure 1: The proposed learning framework of training statistical models from abstract semantic annotations.

Table 1: Abstract semantic annotation and its flattened semantic tag sequence.

Sentence	I want to return to Dallas on Thursday.
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+ON RETURN+ON+DUMMY RETURN+ON+DATE (Thursday) RETURN+ON+DATE (Thursday)+DUMMY`	

illustrated in Figure 1. The whole procedure works as follows. Given a set of sentences $\mathbf{S} = \{S_i, i = 1, \ldots, N\}$ and their corresponding semantic annotations $\mathbf{A} = \{A_i, i = 1, \ldots, N\}$, each annotation A_i is expanded to the flattened semantic tag sequence C_i at initialization step. Based on the flattened semantic tag sequences, the initial model parameters are estimated. After that, the semantic tag sequence $\widehat{C}_i$ is generated for each sentence using the current model, $\widehat{\mathbf{C}} = \{\widehat{C}_i, i = 1, \ldots, N\}$. Then, $\widehat{\mathbf{C}}$ is filtered based on a score function which measures the agreement of the generated semantic tag sequences with the actual flattened semantic tag sequences. In the *maximization* step, model parameters are reestimated using the filtered $\widehat{\mathbf{C}}$. The iteration continues until convergence. The details of each step are discussed in Figure 1.

3.1. Preprocessing. 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). The provision of abstract annotations implies that the semantics encoded in each sentence need not be provided in expensive token style. Obviously, there are some input words such as articles, which have no specific semantic meanings. In order to cater for these irrelevant input words, a `DUMMY` tag is introduced in the preterminal position. Hence, the flattened semantic tag sequence is finally expanded to the semantic tag sequence as in Table 1(b).

3.2. Expectation with Constraints. During the *expectation* step, that is, calculating the most likely semantic tag sequence given a sentence, we need to impose the following two constraints which are implied from abstract semantic annotations.

(1) Considering the calculated semantic tag sequence as a hidden state sequence, state transitions are only allowed if both current and next states are listed in the semantic annotation defined for the sentence.

(2) If a lexical item is attached to a preterminal tag of a flattened semantic tag, the semantic tag must appear bound to that lexical item in the training annotation.

To illustrate how these two constraints are applied, the sentence "I want to return on Thursday to Dallas" with its annotation "RETURN(TOLOC(CITY(Dallas)) ON(DATE(Thursday)))" is taken as an example. The transition from `RETURN+TOLOC+CITY` to `RETURN` is allowed since both states can be found in the semantic annotation and follows constraint 1. However, the transition from `RETURN` to `FLIGHT` is not allowed as it does not follow constraint 1 and `FLIGHT` is not listed in the semantic annotation. Also, for the lexical item `Dallas` in the training sentence, the only valid semantic tag is `RETURN+TOLOC+CITY` because to apply constraint 2 `Dallas` has to be bound with the preterminal tag `CITY`.

We further describe how these two constraints can be imposed into two different models, CRFs and HM-SVMs:

$$\alpha_t\left(c^t = c \mid S\right) = \sum_{c'} \alpha_{t-1}\left(c^{t-1} = c' \mid S\right) \exp \sum_k \theta_k f_k\left(c^{t-1} = c', c^t = c, S\right),$$

$$\beta_t\left(c^t = c \mid S\right) = \sum_{c'} \beta_{t+1}\left(c^{t+1} = c' \mid S\right) \exp \sum_k \theta_k f_k\left(c^{t+1} = c', c^t = c, S\right) \quad (7)$$

$$\alpha_t\left(c^t = c \mid S\right) = \begin{cases} 0, & \text{when } g\left(c^t, c, s^t\right) = 1, \\ \sum_{c'} \left\{ \alpha_{t-1}\left(c^{t-1} = c' \mid S\right) \times \exp \sum_k \theta_k f_k\left(c^{t-1} = c', c^t = c, S\right) \right\}, & \text{otherwise,} \end{cases}$$

$$\beta_t\left(c^t = c \mid S\right) = \begin{cases} 0, & \text{when } g\left(c^t, c, s^t\right) = 1, \\ \sum_{c'} \left\{ \beta_{t+1}\left(c^{t+1} = c' \mid S\right) \times \exp \sum_k \theta_k f_k\left(c^{t+1} = c', c^t = c, \mathbf{x}\right) \right\}, & \text{otherwise.} \end{cases} \tag{8}$$

3.2.1. Expectation in CRFs. The most probable labeling sequence in CRFs 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

$$P\left(c^t = c \mid S\right) = \frac{\alpha_t\left(c^t = c \mid S\right) \beta_t\left(c^t = c \mid S\right)}{Z\left(S\right)}, \tag{9}$$

where $Z(S) = \sum_c \alpha_t(c \mid S)$.

The forward values $\alpha_t(c^t = c \mid S)$ and backward values $\beta_t(c^t = c \mid S)$ are defined in iterative form as (7).

Given the training data $D = \{(S_1, C_1), \ldots, (S_N, C_N)\}$, the parameter Θ can be estimated through a maximum likelihood procedure. To calculate the log-likelihood of $L(\Theta)$ with expectation over the abstract annotation as follows,

$$\begin{aligned} L\left(\Theta; \Theta^t\right) &= \sum_i^N \sum_{C_i^u} P\left(C_i^u \mid S_i; \Theta^t\right) \log P\left(C_i^u \mid S_i; \Theta\right) \\ &= \sum_i^N \sum_{C_i^u} P\left(C_i^u \mid S_i; \Theta^t\right) \sum_t \sum_k \theta_k f_k\left(c', c, S_i\right) \\ &\quad - \sum_i^k \log Z\left(S_i\right), \end{aligned} \tag{10}$$

where C_i^u is the unknown semantic tag sequence of the ith word sequence and $Z(S_i) = \sum_c \exp(\sum_t \sum_k \theta_k f_k(c^{t-1}, c^t, S_i))$. It can be optimized using the same optimization method as in standard CRFs training.

To infer the word-level semantic tag sequences based on abstract annotations, (7) are modified as shown in (8), where $g(c^t, c, s^t)$ is defined as follows:

$$g\left(c^t, c, s^t\right) = \max \begin{cases} 1, & c \text{ is not in the allowable semantic tag list of } S, \\ 1, & c \text{ is not of class type and } s^t \text{ is of class type,} \\ 0, & \text{otherwise.} \end{cases} \tag{11}$$

3.2.2. Expectation in HM-SVM. To calculate the most likely semantic tag sequence $\widehat{C}$ for each sentence S, $\widehat{C} = \arg\max_{C \in \mathscr{C}} F(S, C)$, we can decompose the discriminant function $F : \mathcal{S} \times \mathscr{C} \to \mathbb{R}$ into two components, $F(S, C) = F_1(S, C) + F_2(S, C)$, where

$$\begin{aligned} F_1\left(S, C\right) &= \sum_{\sigma \in \mathbf{c}, \tau \in \mathbf{c}} \delta\left(\sigma, \tau\right) \sum_{l=1}^{T} \left[\left[c^{l-1} = \sigma \wedge c^l = \tau\right]\right], \\ F_2\left(S, C\right) &= \sum_{\sigma \in \mathbf{c}} \sum_{l=1}^{T} \gamma\left(s^l, \sigma\right) \left[\left[c^l = \sigma\right]\right]. \end{aligned} \tag{12}$$

Here, $\delta(\sigma, \tau)$ is considered as the coefficient for the transition from state (or semantic tag) σ to state τ while $\gamma(s^l, \sigma)$ can be treated as the coefficient for the emission of word s^l from state σ. They are defined as follows:

$$\begin{aligned} \delta\left(\sigma, \tau\right) &= \sum_{i, \overline{C}} \alpha_i\left(\overline{C}\right) \sum_{m=1}^{|\overline{C}|} \left[\left[\overline{c}^{m-1} = \sigma \wedge \overline{c}^m = \tau\right]\right], \\ \gamma\left(s^l, \sigma\right) &= \sum_{i,m} \sum_{C} \left[\left[c^m = \sigma\right]\right] \alpha_i\left(C\right) k\left(s^l, s_i^m\right), \end{aligned} \tag{13}$$

where $k(s^l, s_i^m) = \langle \Psi(s^l), \Psi(s_i^m) \rangle$ describes the similarity of the input patterns Ψ between word s^l and word s_i^m, the mth word in the training example i, and $\alpha_i(C)$ is a set of dual parameters or Lagrange multiplier of the constraint associated with example i and semantic tag sequence C as in (4). Using the results derived in (13), Viterbi decoding can be performed to generate the best semantic tag sequence.

To incorporate the constraints as defined in the abstract semantic annotations, the values of $\delta(\sigma, \tau)$ and $\gamma(s^l, \sigma)$ are modified for each sentence:

$$\begin{aligned} \delta\left(\sigma, \tau\right) &= \begin{cases} 0, & \text{when } g\left(\sigma, \tau\right) = 1, \\ \sum_{i, \overline{C}} \alpha_i\left(\overline{C}\right) \sum_m \left[\left[\overline{c}^{m-1} = \sigma \wedge \overline{c}^m = \tau\right]\right], & \\ & \text{otherwise,} \end{cases} \\ \gamma\left(s^l, \sigma\right) &= \begin{cases} 0, & \text{when } h\left(\sigma, s^l\right) = 1, \\ \sum_{i,m} \sum_{C} \left[\left[c^m = \sigma\right]\right] \alpha_i\left(C\right) k\left(s^l, s_i^m\right), & \\ & \text{otherwise,} \end{cases} \end{aligned} \tag{14}$$

where $g(\sigma, \tau)$ and $h(\sigma, s^l)$ are defined as follows:

$$g\left(\sigma, \tau\right) = \begin{cases} 1, & \tau \text{ is not in the allowable semantic tag list,} \\ 0, & \text{otherwise,} \end{cases}$$

$$h\left(\sigma, s^l\right) = \begin{cases} 1, & \sigma \text{ is not of class type and } s^l \text{ is of class type,} \\ 0, & \text{otherwise,} \end{cases} \tag{15}$$

where $g(\sigma, \tau)$ and $h(\sigma, s^l)$ in fact encode the two constraints implied from abstract annotations.

3.3. Filtering. For each sentence, the semantic tag sequences generated in the *expectation* step are further processed based on a measure on the agreement of the semantic tag sequence $T = \{t_1, t_2, \ldots, t_n\}$ with its corresponding abstract semantic annotation A. The score of T is defined as

$$\text{Score}(T) = 2 * \frac{S_{\text{recall}} * S_{\text{precision}}}{S_{\text{recall}} + S_{\text{precision}}}, \tag{16}$$

where $S_{\text{precision}} = N_r/n$, $S_{\text{Recall}} = N_r/p$. Here, N_r is the number of the semantic tags in T which also occur in A, n is the number of semantic tags in T, and p is the number of semantic tags in the flattened semantic tag sequence for A. The score is similar to the F-measure which is the harmonic mean of precision and recall. It essentially measures the agreement of the generated semantic tag sequence with the abstract semantic annotation. We filter out sentences with their score below certain predefined threshold and the remaining sentences together with their generated semantic tag sequences are fed into the next *maximization* step. In our experiments, we empirically set the threshold to 0.1.

3.4. Maximization. Given the filtered training examples from the *filtering* step, the parameters Θ are adjusted using the standard training algorithms.

For CRFs, the parameter Θ can be estimated through a maximum likelihood procedure. The model is traditionally trained by maximizing the conditional log-likelihood of the labeled sequences, which is defined as

$$L(\Theta) = \sum_{i=1}^{N} \log\left(P\left(C_i \mid S_i; \Theta\right)\right), \tag{17}$$

where N is the number of sequences.

The maximization can be achieved gradient ascent where the gradient of the likelihood is

$$\begin{aligned}\frac{\partial}{\partial \theta_k} = &\sum_{i=1}^{N} \sum_{t} f_k\left(c_i^{t-1}, c_i^t, S_i, t\right) \\ &- \sum_{i=1}^{N} \sum_{S} p_\theta\left(C \mid S_i\right) \sum_{t} f_k\left(c^{t-1}, c^t, S_i, t\right).\end{aligned} \tag{18}$$

For HM-SVMs, the parameters $\Theta = w$ are adjusted so that the true semantic tag sequence C_i scores higher than all the other tag sequences $C \in \mathcal{C}_i := \mathcal{C} \setminus C_i$ with a large margin. To achieve the goal, the optimization problem as stated in (3) is solved using an online learning approach as described in [4]. In short, it works as follows: a pattern sequence S_i is presented and the optimal semantic tag sequence $\widehat{C}_i = f(S_i)$ is computed by employing Viterbi decoding. If $\widehat{C}_i$ is correct, no update is performed. Otherwise, the weight vector w is updated based on the difference from the true semantic tag sequence $\Delta\Phi = \Phi(S_i, \widehat{C}_i) - \Phi(S_i, C_i)$.

4. Experimental Results

Experiments have been conducted on the DARPA communicator data (http://www.bltek.com/spoken-dialog-systems/cu-communicator.html/) which were collected in 461 days. From these, 46 days were randomly selected for use as test set data and the remainders were used for training. After cleaning up the data, the training set consists of 12702 utterances while the test set contains 1178 utterances.

Table 2

I wanna travel from Denver to San Diego on March sixth.	
Frame	AIR
Slots	FROMLOC · CITY = Denver
	TOLOC · CITY = San Diego
	MONTH = March
	DAY = sixth

The abstract semantic annotations used for training only list a set of valid semantic tags and the dominance relationships between them without considering the actual realized semantic tag sequence or attempting to identify explicit word/concept pairs. Thus, it avoids the need for expensive treebank style annotations. For example, for the sentence "I wanna go from Denver to Orlando Florida on December tenth," the abstract annotation would be `FROMLOC(CITY) TOLOC(CITY(STATE)) MONTH(DAY)`.

To evaluate the performance of the model, a reference frame structure was derived for every test set sentence consisting of slot/value pairs. An example of a reference frame is shown in Table 2.

Performance was then measured in terms of F-measure on slot/value pairs, which combines the precision (P) and recall (R) values with equal weight and is defined as $F = 2 * P * R/(P + R)$.

We modified the open source of the CRF suite (http://www.chokkan.org/software/crfsuite/) and SVM$^{\text{HMM}}$ (http://www.cs.cornell.edu/people/tj/svm_light/svm_hmm.html/) to implement our proposed learning framework. We employed two algorithms to estimate the parameters of CRFs, the stochastic gradient descent (SGD) iterative algorithm [21], and the limited-memory BFGS (L-BFGS) method [22]. For both algorithms, the regularization parameter was empirically set in the following experiments.

4.1. Overall Comparison. We first compare the time consumed in each iteration using HM-SVMs or CRFs as shown in Figure 2. The experiments were conducted on the Intel(R) Xeon(TM) model Linux server equipped with 3.00 Ghz processor and 4 GB RAM. It can be observed that, for CRFs, the time consumed in SGD is almost doubled compared to that in L-BFGS in each iteration. However, since SGD converges much faster than L-BFGS, the total time required for training is almost the same. As SGD gives balanced precision and recall values, it should be preferred more than L-BFGS in our proposed learning procedure. On the other hand, as opposed to CRFs which consume much less time after iteration 1, HM-SVMs take almost the same run time for all the iterations. Nevertheless, the total run time until convergence is almost the same for CRFs and HM-SVMs.

Figure 3 shows the performance of our proposed framework for CRFs and HM-SVMs at each iteration. At each word position, the feature set used for both statistical models

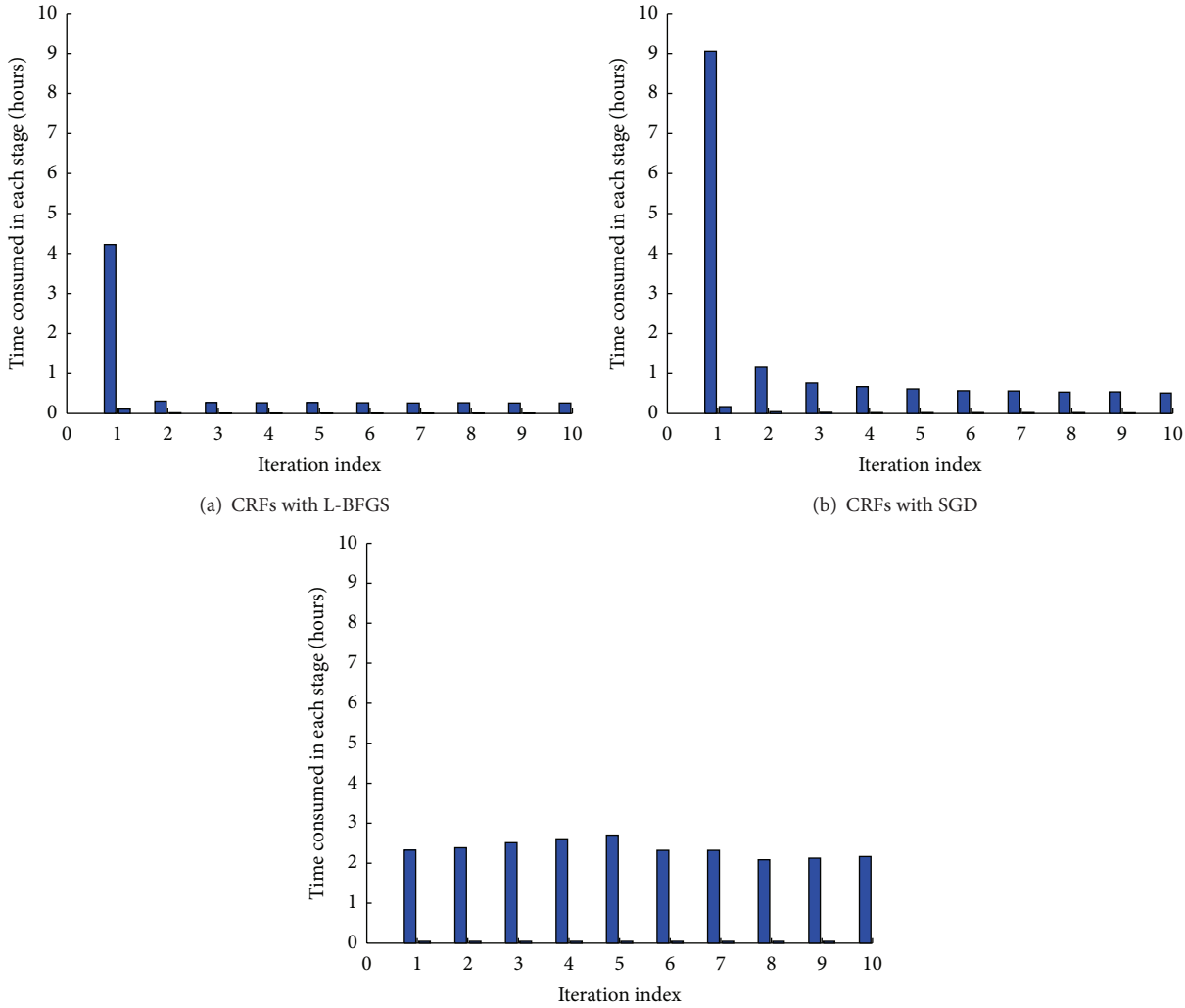


Figure 2: Time consumed in each iteration by CRFs and HM-SVMs.

consists of the current word and the current part-of-speech (POS) tag. It can be observed that both models achieve the best performance at iteration 8 with an F-measure of 92.95% and 93.18% being achieved using CRFs and HM-SVMs, respectively.

4.2. Results with Varied Features Set. We employed word features (such as current word, previous word, and next word) and POS features (such as current POS tag, previous one, and next one) for training. To explore the impact of the choices of features, we explored with feature sets comprised of words or POS tags occurring before or after the current word within some predefined window size.

Figure 4 shows the performance of our proposed approach with the window size varying between 0 and 3. Surprisingly, the model learned with feature set chosen by setting window size 0 gives the best overall performance. Varying window size between 1 and 3 only impacts the convergence rate and does not lead to any performance difference at the end of the learning procedure.

4.3. Performance with or without Filtering Step. In a second set of experiments, we compare the performance with or without the *filtering* step as discussed in Section 3.3. Figure 5 shows that the *filtering* step is indeed crucial as it boosted the performance by nearly 4% for CRFs with L-BFGS and 3% for CRFs with SGD and HM-SVMs.

4.4. Comparison with Existing Approaches. We compare the performance of CRFs and HM-SVMs with HVS, all trained on abstract semantic annotations. While it is hard to incorporate arbitrary input features into HVS learning, both CRFs and HM-SVMs have the capability of dealing with overlapping features. Table 3 shows that they outperform HVS with a relative error reduction of 36.6% and 43.3% being achieved, respectively. In addition, the superior performance of HM-SVMs over CRFs shows the advantage of HM-SVMs

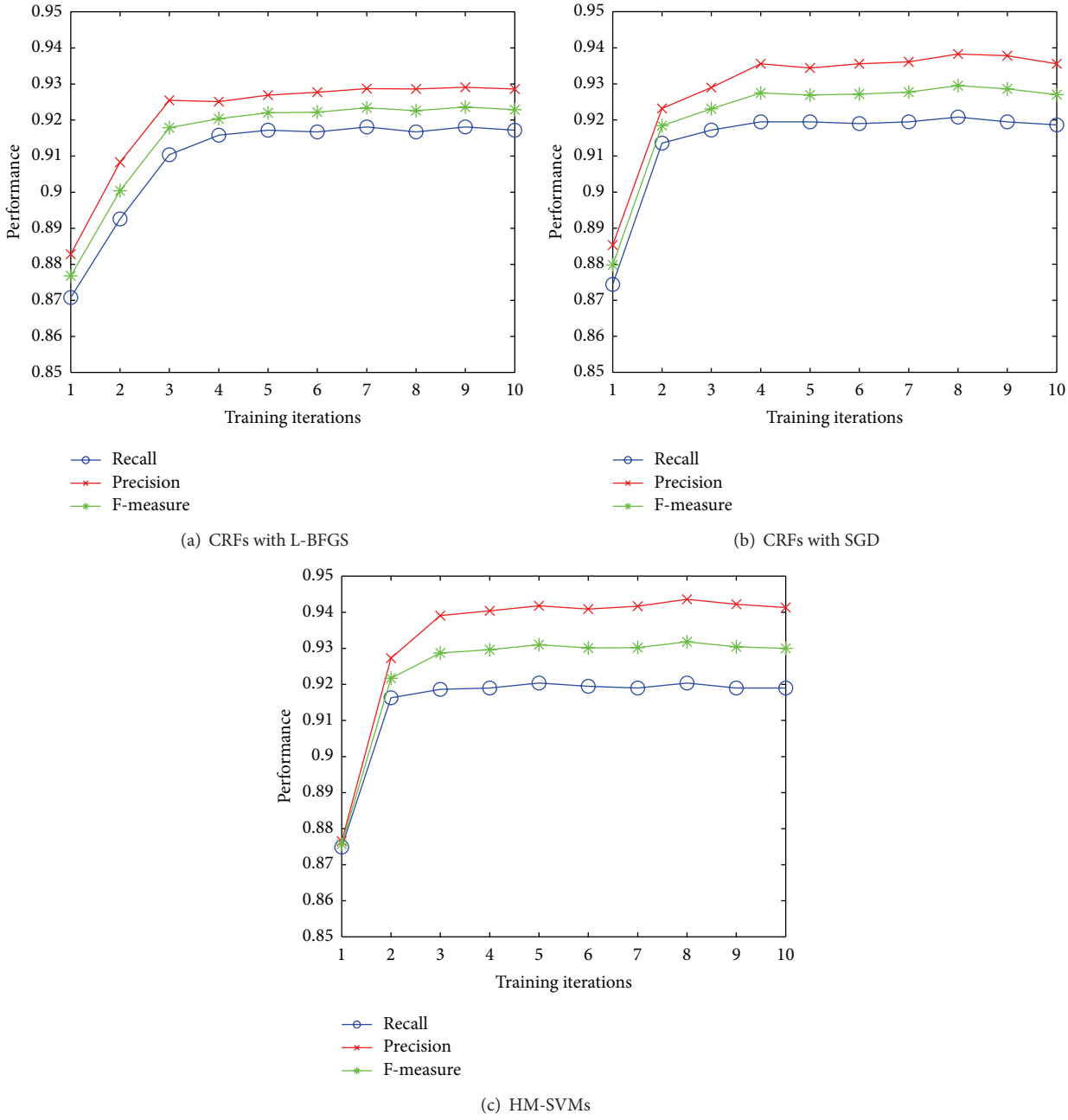


Figure 3: Performance for CRFs and HM-SVMs at each iteration.

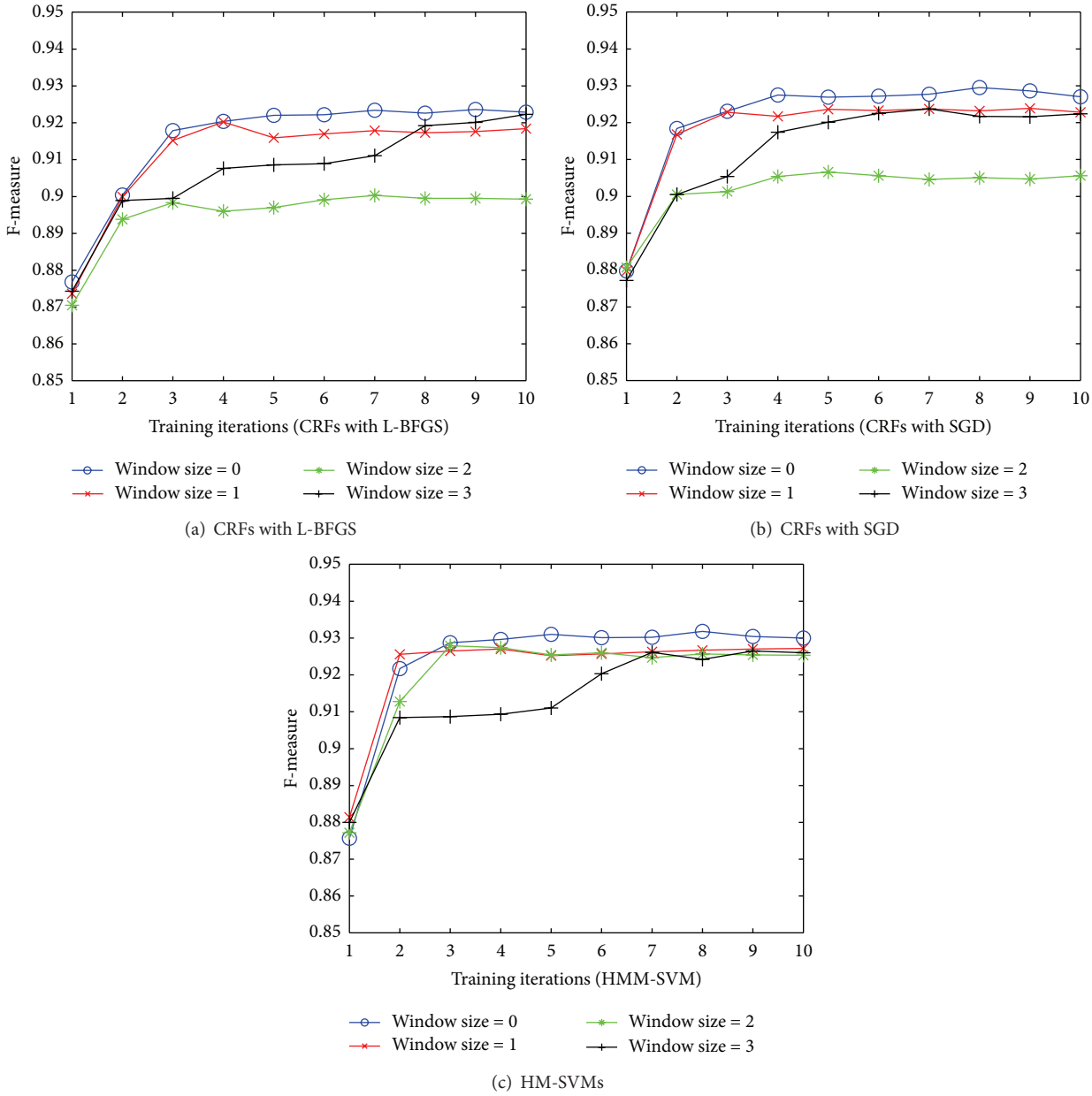
Table 3: Performance comparison between the proposed framework and three other approaches (HF denotes the hybrid framework and DT denotes discriminative training the HVS model.

Measurement	HVS	HF	DT	Proposed framework	
				CRFs	HM-SVMs
Recall (%)	87.81	90.99	91.49	92.08	92.04
Precision (%)	88.13	90.25	91.87	93.83	94.36
F-measure (%)	87.97	90.62	91.68	92.95	93.18

on learning nonlinear discriminant functions via kernel functions.

We further compare our proposed learning approach with two other methods. One is a hybrid generative/discriminative framework (HF) [23] which combines HVS with HM-SVMs so as to allow the incorporation of arbitrary features as in CRFs. The other is a discriminative approach (DT) based on parse error measure to train the HVS model [24]. The generalized probabilistic descent (GPD) algorithm [25] was employed to adjust the HVS model to achieve the minimum parse error rate.

Table 3 shows that our proposed learning approach outperforms both HF and DT. Training statistical models on abstract annotations allows the calculation of conditional likelihood and hence results in direct optimization of the



(a) CRFs with L-BFGS

(b) CRFs with SGD

(c) HM-SVMs

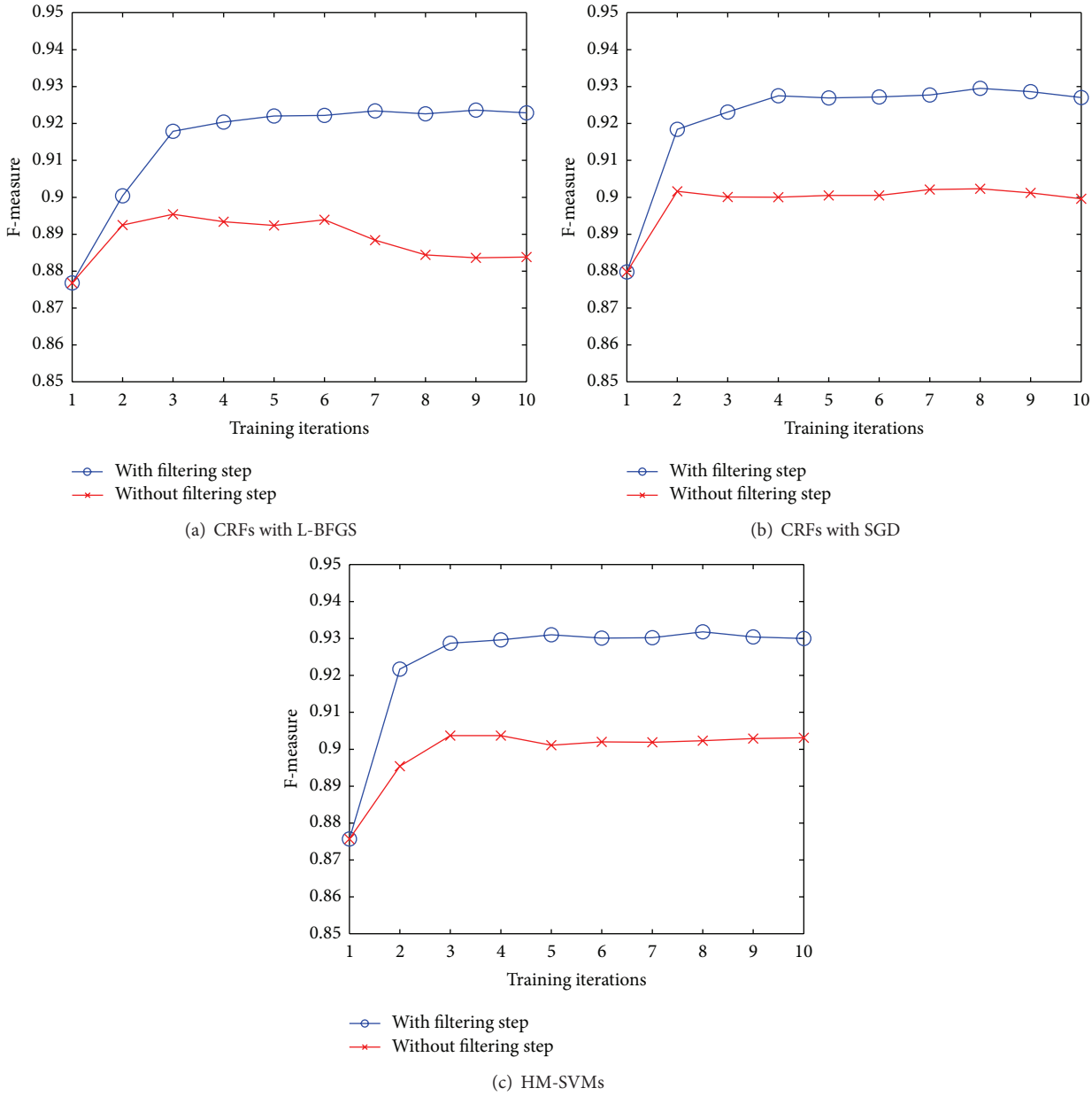
Figure 4: Comparison of performance on models learned with feature sets chosen based on different window sizes.

objective function to reduce the error rate of semantic labeling. On 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 objective 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 reranking where the input is the N-best parse results generated from the HVS model.

5. Conclusions

In this paper, we have proposed an effective learning approach which can train statistical models such CRFs and HM-SVMs without using the expensive treebank style annotation data. Instead, it trains the statistical models from only abstract annotations in a constrained way. Experimental results show that, using the proposed learning approach, both CRFs and HM-SVMs outperform the previously proposed HVS model on the DARPA communicator data. Furthermore, they also show superior performance than the 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 about 25% and 15% being achieved when compared with HF and DT, respectively.

In future work, we will explore other score functions in *filtering* step to describe the precision of the parsing results.



(a) CRFs with L-BFGS

(b) CRFs with SGD

(c) HM-SVMs

Figure 5: Comparisons of performance with or without the *filtering* stage.

Also, we plan to apply the proposed framework in some other domains such as information extraction and opinion mining.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

The submitted paper is the extended version of the conference paper for CIKM 2011 with the title "A novel framework of training hidden Markov support vector machines from lightly-annotated data." The authors thank the anonymous reviewers for their insightful comments. This work was funded by the National Natural Science Foundation of China (61103077), Ph.D. Programs Foundation of Ministry of Education of China for Young Faculties (20100092120031), Scientific Research Foundation for the Returned Overseas Chinese Scholars, State Education Ministry, and the Fundamental Research Funds for the Central Universities (the Cultivation Program for Young Faculties of Southeast University).

References

[1] J. Dowding, R. Moore, F. Andry, and D. Moran, "Interleaving syntax and semantics in an efficient bottom-up parser," in *Proceedings of the 32th Annual Meeting of the Association for*

Computational Linguistics, pp. 110–116, Las Cruces, NM, USA, 1994.

[2] W. Ward and S. Issar, "Recent improvements in the cmu spoken language understanding system," in *Proceedings of the Workshop on Human Language Technology*, pp. 213–216, Plainsboro, NJ, USA, 1994.

[3] J. D. Lafferty, A. McCallum, and F. C. N. Pereira, "Conditional random fields: probabilistic models for segmenting and labeling sequence data," in *Proceedings of the 18th International Conference on Machine Learning (ICML '11)*, pp. 282–289, 2001.

[4] Y. Altun, I. Tsochantaridis, and T. Hofmann, "Hidden markov support vector machines," in *Proceedings of the International Conference in Machine Learning*, pp. 3–10, 2003.

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

[6] R. J. Kate and R. J. Mooney, "Using string-kernels for learning semantic parsers," in *Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics (ACL '06)*, pp. 913–920, 2006.

[7] Y. W. Wong and R. J. Mooney, "Learning synchronous grammars for semantic parsing with lambda calculus," in *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (ACL '07)*, pp. 960–967, June 2007.

[8] W. Lu, H. Ng, W. Lee, and L. Zettlemoyer, "A generative model for parsing natural language to meaning representations," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP '08)*, pp. 783–792, Stroudsburg, PA, USA, October 2008.

[9] R. Ge and R. Mooney, "Learning a compositional semantic parser using an existing syntactic parser," in *Proceedings of the 47th Annual Meeting of the ACL*, pp. 611–619, 2009.

[10] M. Dinarelli, A. Moschitti, and G. Riccardi, "Discriminative reranking for spoken language understanding," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 20, no. 2, pp. 526–539, 2012.

[11] L. S. Zettlemoyer and C. Michael, "Learning to map sentences to logical form: structured classification with probabilistic categorial grammars," in *Proceedings of the 21st Conference on Uncertainty in Artificial Intelligence (UAI '05)*, pp. 658–666, July 2005.

[12] A. Giordani and A. Moschitti, "Syntactic structural kernels for natural language interfaces to databases," in *Machine Learning and Knowledge Discovery in Databases*, W. Buntine, M. Grobelnik, D. Mladeni, and J. Shawe-Taylor, Eds., vol. 5781 of *Lecture Notes in Computer Science*, pp. 391–406, Springer, Berlin, Germany, 2009.

[13] A. Giordani and A. Moschitti, "Translating questions to SQL queries with generative parsers discriminatively reranked," in *Proceedings of the 24th International Conference on Computational Linguistics*, pp. 401–410, 2012.

[14] J. Zelle and R. Mooney, "Learning to parse database queries using inductive logic programming," in *Proceedings of the AAAI*, pp. 1050–1055, 1996.

[15] F. Jiao, S. Wang, C.-H. Lee, R. Greiner, and D. Schuurmans, "Semi-supervised conditional random fields for improved sequence segmentation and labeling," in *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics (COLING/ACL '06)*, pp. 209–216, July 2006.

[16] G. S. Mann and A. McCallum, "Efficient computation of entropy gradient for semi-supervised conditional random fields," in *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL '07)*, pp. 109–112, 2007.

[17] Y. Wang, G. Haffari, S. Wang, and G. Mori, "A rate distortion approach for semi-supervised conditional random fields," in *Proceedings of the 23rd Annual Conference on Neural Information Processing Systems (NIPS '09)*, pp. 2008–2016, December 2009.

[18] G. S. Mann and A. Mccallum, "Generalized expectation criteria for semi-supervised learning of conditional random fields," in *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics*, pp. 870–878, June 2008.

[19] T. Grenager, D. Klein, and C. D. Manning, "Unsupervised learning of field segmentation models for information extraction," in *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL '05)*, pp. 371–378, Ann Arbor, Mich, USA, June 2005.

[20] J. A. Bilmes, "A gentle tutorial of the em algorithm and its application to parameter estimation for gaussian mixture and hidden markov models," in *Proceedings of the International Conference on Systems Integration*, 1997.

[21] S. Shalev-Shwartz, Y. Singer, and N. Srebro, "Pegasos: primal estimated sub-gradient solver for svm," in *Proceedings of the 24th International Conference on Machine Learning (ICML '07)*, pp. 807–814, June 2007.

[22] J. Nocedal, "Updating quasi-newton matrices with limited storage," *Mathematics of Computation*, vol. 35, no. 151, pp. 773–782, 1980.

[23] D. Zhou and Y. He, "A hybrid generative/discriminative framework to train a semantic parser from an un-annotated corpus," in *Proceeding of the 22nd International Conference on Computational Linguistics (COLING '08)*, pp. 1113–1120, Manchester, UK, August 2008.

[24] D. Zhou and Y. He, "Discriminative training of the hidden vector state model for semantic parsing," *IEEE Transactions on Knowledge and Data Engineering*, vol. 21, no. 1, pp. 66–77, 2009.

[25] H. K. J. Kuo, E. Fosler-Lussier, H. Jiang, and C.-H. Lee, "Discriminative training of language models for speech recognition," in *Proceedings of the IEEE International Conference on Acustics, Speech, and Signal Processing (ICASSP '02)*, vol. 1, pp. 325–328, IEEE, Merano, Italy, May 2002.