JAIST Repository
https://dspace.jaist.ac.jp/

Title	Online Structured Learning for Semantic Parsing with Synchronous and λ-Synchronous Context Free Grammars
Author(s)	Nguyen, Le-Minh; Shimazu, Akira; Phan, Xuan Hieu; Nguyen, Phuong Thai
Citation	20th IEEE International Conference on Tools with Artificial Intelligence, 2008. ICTAI '08.: 135-142
Issue Date	2008-11
Type	Conference Paper
Text version	publisher
URL	http://hdl.handle.net/10119/8496
Rights	Copyright (C) 2008 IEEE. Reprinted from 20th IEEE International Conference on Tools with Artificial Intelligence, 2008. ICTAI '08., 135-142. This material is posted here with permission of the IEEE. Such permission of the IEEE does not in any way imply IEEE endorsement of any of JAIST's products or services. Internal or personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution must be obtained from the IEEE by writing to pubs-permissions@ieee.org. By choosing to view this document, you agree to all provisions of the copyright laws protecting it.
Description	

JAIST
JAPAN
ADVANCED INSTITUTE OF
SCIENCE AND TECHNOLOGY

Japan Advanced Institute of Science and Technology

2008 20th IEEE International Conference on Tools with Artificial Intelligence

Online Structured Learning for Semantic Parsing with Synchronous and $\lambda-$Synchronous Context Free Grammars

Le-Minh Nguyen and Akira Shimazu
GSIS, JAIST
1-1 Asahidai, Nomi Ishikawa, 923-1292, Japan
{nguyenml,shimazu}@jaist.ac.jp

Xuan Hieu Phan
GIS, Tohoku University
Aobayama 6-3-09 Sendai, 980-8579, Japan
hieuxuan@ecei.tohoku.ac.jp

Phuong Thai Nguyen
College of Technology, VNUH
thainp@vnu.edu.vn

Abstract

We formulate semantic parsing as a parsing problem on a synchronous context free grammar (SCFG) which is automatically built on the corpus of natural language sentences and the representation of semantic outputs. We then present an online learning framework for estimating the synchronous SCFG grammar. In addition, our online learning methods for semantic parsing problems are also extended to deal with the case, in which the semantic representation could be represented under $\lambda - calculus$. Experimental results in the domain of semantic parsing show advantages in comparison with previous works.

1 Introduction

Semantic parsing is the task of mapping a natural language sentence into a complete, formal meaning representation. This task is an interesting problem in Natural Language Processing (NLP) as it would very likely be part of any interesting NLP applications [2]. For example, the necessity of semantic parsing for most NLP applications and the ability to map natural language to a formal query or command language, are critical for developing more user-friendly interfaces.

There have been a significant amount of previous works on learning to map sentences to semantic representation. Zelle and Mooney [22] and Tang [19] proposed the empirically-based method using a corpus of natural language(NL) sentences and their formal representation for learning by inductive logic programming (ILP). The disadvantage of the ILP approach is that it is quite complex, and slow to acquire parsers for mapping long sentences to logical form such as the Robocup corpus.

To overcome this problem, [12] proposed a method that used transformation rules estimating from the corpus of NL and logical form, to transform NL sentences to logical forms. This method does not use any prior knowledge about language, but its performance is still not high enough for using in a real application.

In order to improve semantic parsing accuracy, Ge and Mooney (2005) presented a statistical method [9] by merging syntactic and semantic information. However the distinction of this method in comparison with the other methods is that the training data is required to have Semantic Augmented Parsing trees (SAPTs) annotation [1]

Similar to Ge and Mooney (2005), the approach by Nguyen, Shimazu, and Phan (2006) also uses the corpus of SAPTs tree to estimate their semantic parsing model. This approach uses structured SVMs [11] and learns ensemble learning of semantic parsers.

Unlike those methods using SAPTs, the works proposed by [23][24] map an NL sentence to its logical form using a Combinatory Categorial Grammar (CCG) with structured learning models. They have indicated that using online structured prediction along with CCG could lead to the state of the art results on several domains (i.e ATIS and Query language). However, their methods still require NL-specific templates for building CCG grammars. Kate and Mooney (2006) proposes a method using string kernel function to learn the transformation rules automatically from NL and logical forms. Wong and Mooney (2006) proposed a synchronous context free grammar frame work (SCFG)(Aho and Ullman, 1972) to transform NL language sentence to semantic representation. The system was extended to work

[1] SAPT is a syntactic tree with semantic augmented at each non-terminal node.

1082-3409/08 $25.00 © 2008 IEEE
DOI 10.1109/ICTAI.2008.96

IEEE computer society

when the formal language are in λ-calculus (Wong and Mooney, 2007) to deal with logical variables.

A major advantage of [21] over methods such as [9][15] and [24] is that it does not require any prior knowledge of the NL syntax for training. This is easily employed the method to other languages than English.

Our goal is to construct a semantic parser which does not require any prior knowledge for training, while ensure the accurate of semantic parsing's performance. In order to do that we propose a method by combining the advantage of online structured prediction and the use synchronous context free grammar. We then formulate the process of transforming an NL sentence into its logical form as a structured prediction, which assigns a given NL sentence to a semantic representation. To adapt the structured learning to the problem of semantic parsing domain, there are two fundamental problems as follows.

- How to represent the structure of the semantic parsing problem.
- How to select a suitable structured prediction algorithm for that representation.

Even if we have a good structure for representing the input and output of semantic parsing, we might have a problem in that we do not know exactly the correspondence of each word in the input to each token in the semantic output. This is very different from syntactic parsing and tagging problems.

To solve the problem mentioned above, we formulate the representation of semantic parsing problem using a probabilistic synchronous context free grammar (SCFG) [1] which is built on the grammar of natural language sentences and the grammar for representation of semantic output. For the semantic output using logical variable such as prolog representation, we used the $\lambda-$calculus synchronous grammar ($\lambda-$SCFG) which have introduced for semantic parsing problem [21].

We then present a novel online learning algorithm to estimate the SCFG. The method can be used for any representation of NL input sentence and its semantic representation in the the synchronous grammars. Our method has the advantage that it can fully learn SCFG and $\lambda-$SCFG from the corpus of sentences and their semantic representation. In addition, the use of large-margin methods which directly reduce error by using a loss function are sensitive to the performance of our semantic parsing's systems. Our method is different from [21] and [24] at this point.

The rest of this paper is organized as follows: Section 2 describes an online-large margin learning method for semantic parsing using SCFG and $\lambda-$SCFG grammars. Section 3 shows experimental results and Section 4 discusses the advantage of our method and describes future work.

2 Online Large-Margin Structured Learning for Semantic Parsing

This section will present an online structured prediction method for a semantic parsing problem. First, we consider the semantic parsing problem as structured classification using online large-margin learning. We then exploit it in the structured domain of SCFG and λ-SCFG grammars.

2.1 Semantic Parsing as structured classification

Semantic parsing can be seen as a structured classification task, in which the goal is to learn a mapping from an input (NL) sentence x to a meaning representation (MR) y in a meaning representation language (MRL) . Given this setup, discriminative methods allow us to define a broad class of features Φ that operate on (x, y).

However, the semantic parsing task in this framework differs from traditional applications of discriminative structured classification such as POS tagging. Whereas in POS tagging, there is a one-to-one correspondence between the words x and the tags y, the correspondence between x and y in semantic parsing is unknown. This is due to the fact that one word in x might correspond to some lexical meaning in y and vice visa.

To deal with this problem, we introduce a hidden correspondence structure h and work with the feature over input sentences and output semantic representation.

In this model the correspondence h is a synchronous parse tree including the scores of various productions used in the tree. Given features Φ and a corresponding set of parameters w, a standard classification rule f is to return the highest scoring output sentence y, maximizing over correspondences h:

$$f(x; w) = \arg\max_{y,h}(w_t \cdot \Phi(x, y, h))$$

The learning task is to induce a set of SCFG rules (or $\lambda-$SCFG rules), which we call a lexicon, and a probabilistic model for derivations. A lexicon defines the set of derivations that are possible, so the induction of a probabilistic model first requires a lexicon. Therefore, the learning task can be separated into two sub-tasks:

- The induction of a lexicon
- The induction of a probabilistic model

The learning task requires a training set, $\{x_i, y_i\}$, where each training example $< x_i, y_i >$ is an NL sentence, x_i, paired with its correct MR, y_i. Lexical induction also requires an unambiguous Context Free Grammar(CFG) of the MRL.

2.2 Lexical Induction

Given the training set $S = (x_i; y_i), i = 1, 2, ..., N$, the main learning task is to construct a synchronous grammar using the training set. Similar to [20][21], we would like to use a word alignment to obtain a bilingual lexicon consisting of NL substrings coupled with their translations in the target MRL. Unlike [20, 21], (they used the word alignment system, GIZA++ [18]), in this paper we used the Berkeley Word Aligner [8] as it is claimed that more accurate than GIZA++.

As formal languages, MRLs frequently contain many purely syntactic tokens such as parentheses or brackets, which are difficult to align with words in NL. In order to solve this problem, the lexical induction is conducted as follows.

- Parse each semantic representation into a tree using the MRL grammar.
- Extract a tree to a sequence of MRL productions.
- Using word alignment technique to obtain N to 1 alignment between the word within NL sentence and a sequence of MRL productions using corresponds to the top-down, left-most derivation of an MR.
- Generate a set synchronous rules using a bottom-up strategy as follows. The process starts with productions whose right hand side (RHS) is all terminals. Then those productions whose RHS contains non-terminals, i.e. predicates with arguments.

For our experiment, we obtained K-best (10 best) output of the word alignment, and these outputs are intergraded into the estimating procedure for each rules within our synchronous grammar model.

Once a lexicon is acquired, the next task is to learn a discriminative model for the semantic parser. For the purpose of easy understanding, we show an alignment example in Figure 1.

2.3 Semantic Parsing using Synchronous Grammars

Aho and Ullman (1972) originally developed a theory of compilers in which syntax analysis and code generation are combined into a single phase. According to this theory, a semantic parser defines a translation, as a set of pairs of strings in which each pair is an NL sentence coupled with its MR. To finitely specify a potentially infinite translation, we use a synchronous context-free grammar (SCFG) for generating the pairs in a translation. Each SCFG rule consists of a single non-terminal on the left-hand side (LHS). The right-hand side (RHS) of an SCFG rule is a pair of strings,

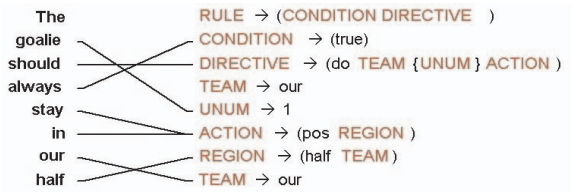

Figure 1. Alignment between NL sentence and its MR. Partial parse trees for the CLANG statement and its English gloss

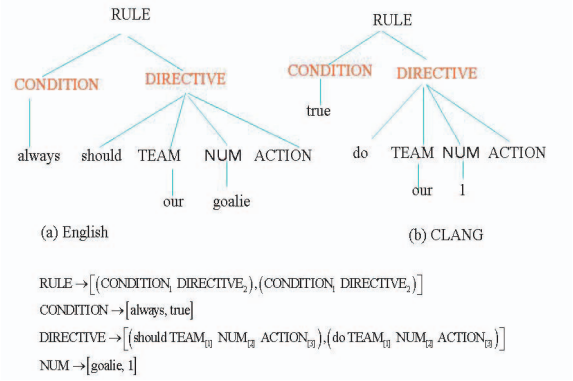

Figure 2. Synchronous rules are extracted using the top-down and left-most strategy. Given the alignment in Figure 1, we can extract synchronous rules.

$< \alpha, \beta >$, where the non-terminals in β are a permutation of the non-terminals in α. Figure 1 shows some rules for generating the parse trees.

Each production rule $X \rightarrow < \alpha, \beta >$ is the combination of a semantic rule $X \rightarrow \alpha$ and a production of an MRL grammar $X \rightarrow \beta$. Following their terminology, we call the string α a pattern, and the string β a template. Non-terminals are indexed to show their association between a pattern and a template. Each step of a derivation involves the rewriting of a pair of associated non-terminals in both the NL and MRL streams.

The conventional SCFG works well for target MRLs that are free of logical variables such as CLANG (Wong and Mooney, 2006), however it cannot easily handle various kinds of logical forms used in computational semantics, such as predicate logic [21]. Wong and Mooney (2007) claimed that WASP lacks a principled mechanism to handle logical variables. In order copy with the problem, they have extended the WASP algorithm by adding a variable-binding mechanism based on $\lambda-$calculus. The details of using $\lambda-$SCFG are sketched in [21]. The formulation of $\lambda-$SCFG is different from SCFG is that each rule in the grammar has a form

$$X \rightarrow < \alpha, \lambda x_1 ... \lambda x_k . \beta > \quad (1)$$

where α is an NL phrase and β is the MR translation of α. The variable-binding operator λ binds occurrences of the logical variables $x_1, .., x_k$ in α, which makes $x_1 ... x_k$ in β a λ-function of arity k. The interesting point of using λ-SCFG grammar is that when using it to represent outputs of parsing NL sentences, we have several constraints to check the correctness of an output.

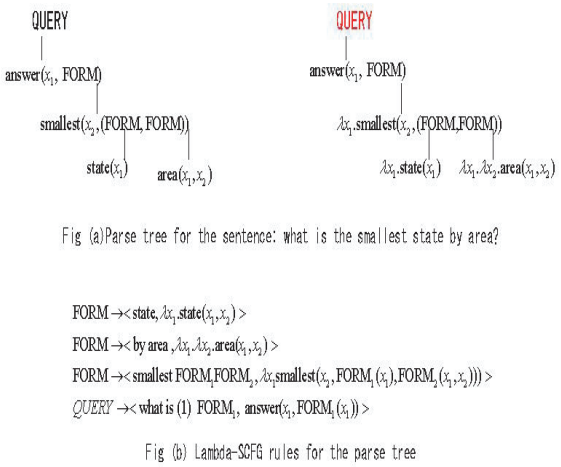

Figure 3. Parse trees for NL sentence in Query language using Lambda-SCFG rules.

Figure 3 showed an example of using λ-SCFG grammar to parse the NL sentence in query language.

2.4 Online Large-Margin structured Learning

This section introduce an online-large margin learning for semantic parsing. It is an extended version of the passive-aggressive learning (PA) [4] to structure prediction using synchronous grammar representation. We also show that Perceptron learning [5] and MIRA learning [14] could be used in our framework in a straightforward manner.

We introduce a hidden correspondence structure h and work with the feature over NL sentences and semantic representation. Here, in this model the correspondence h is a synchronous parse tree including the scores of various productions used in the tree. We modify the PA algorithm for the problem of structured prediction by learning a discriminant function that maps an input sentence x to a semantic output y with a hidden parameter h. In addition, we will address the problem of how PA learning algorithm can estimate the weight w_i associated with a feature f_i in the training data.

Assume that we are given a set of sentences x_i and their semantic representation y_i where $i = 1, ..., N$. Let the feature mapping between a sentence x and a semantic representation y be: $\Phi(x, y, h) = \Phi_1(x, y, h), \Phi_2(x, y, h), ..., \Phi_d(x, y, h)$ where each feature mapping Φ_j maps (x, y, h) to a real value. For example $\Phi_1(x, y, h)$ is the number of a synchronous rules for mapping x to y. We assume that each feature $\Phi(x, y, h)$ is associated with a weight value w. The goal of PA learning for semantic parsing is to obtain a parameter w that minimizes the hinge-loss function and the margin of learning data.

1 Input: $S = (x_i; y_i), i = 1, 2, ..., N$ in which x_i is the sentence and y_i is a semantic representation
2 Alignment information and a set of SCFG (or λ-SCFG) rules
3 Output: the PA learning model
4 Initialize: $w_1 = (0, 0, ..., 0)$
5 **for** *t=1, 2... T* **do**
6 **for** *i=1, 2... N* **do**
7 Receive a sentence and its semantic representation x_i, y_i
8 Predict
$$h_i = \arg\max_h (w_t \cdot \Phi(x_i, y_i, h))$$
$$y_i^*, h_i^* = \arg\max_{y,h} (w_t \cdot \Phi(x_i, y, h))$$
9 Set: $l_t = w_t \cdot \Phi(x_i, y_i^*, h_i^*) - w_t \cdot \Phi(x_i, y_i, h_i) + \sqrt{\rho(y_i, y_i^*)}$
10 $Set : \tau_t = \frac{l_t}{||\Phi(x_i, y_i^*, h_i^*) - \Phi(x_i, y_i, h_i)||^2}$
11 Update:
$$w_{t+1} = w_t + \tau_t(\Phi(x_i, y_i, h_i) - \Phi(x_i, y_i^*, h_i^*))$$
12 **end**
13 $v = v + w_{t+1}$
14 **end**
15 $w = \frac{v}{T}$
16 return w

Algorithm 1: The Passive-Aggressive algorithm for Semantic Parsing.

Algorithm 1 shows the PA learning for semantic parsing. It shows (line 8) the prediction methods for computing argmax as well as an update formulations in line 11. Line 9 shows $p(y_i, y_i^*)$ is a real-valued loss for the tree y_i^* relative to the correct tree y_i. This loss function is similar to one-zero loss function by computing the difference of each node in y_t in comparison with a corresponding node in y. An auxiliary weight vector v is maintained that accumulates the values of w after each iteration, and the returned weight vector is the average of all the weight vectors throughout training to help reduce overfitting [5].

Like the method for statistical machine translation de-

scribed in [17], the significant difference for our semantic parsing problem is that for a given example (the pair of NL sentence and its logical form) we might have some different ways to induce an NL sentence to a logical form. In supervised training, there is only one to one mapping between the input and the output. Therefore the updating process in the online learning mechanism is simpler than our consideration here. Algorithm 1 shows we need to determine the correspondence h_i for each update step. This is easy for supervised learning since we can obtain it from the training data. In this case, we obtain h_i by finding the correspondence that gave the induction from x_i to y_i with the highest score.

$h_i = \arg\max_h(w_t \cdot \Phi(x_i, y_i, h))$

As mentioned above, the difference of PA learning in comparison with Perceptron learning are an update procedure. For Perceptron learning, the update score sketched in Algorithm 1 is replaced by using the formulation as below.

$$Update : w_{t+1} = w_t + (\Phi(x_i, y_i, h_i) - \Phi(x_i, y_i^*, h_i^*))$$

So, we do not need to compute the τ_t in the Perceptron learning. For the MIRA, the line from 8 to line 10 are replaced by solving the optimization as below.

$$\begin{aligned} &\min \|w_{t+1} - w_t\| \\ &s.t. w_{t+1} \circ \Phi(x_i, y_i, h_i) - w_{t+1} \circ \Phi(x_i, y_i^*, h_i^*) \\ &\geq \rho(y_i, y_i^*) \end{aligned}$$

However, to solve the optimization we used an optimization procedure, namely Hidreth's algorithm [7]. Generaly speaking, the behavior of PA algorithm is similar to the MIRA algorithm, which is based on the online mechanism of Perceptron learning, and maximizes the margin of the training data. Each online update of MIRA requires solving a complex optimization problem, but each update of PA has a simple closed-form expression, and is thus much faster and easier to implement.

The main drawback of the Perceptron-style algorithm is that it uses a simple update function to approximate the global optimization (i.e attaining the maximize margin of the training data). So, it may be difficult to obtain high accuracy in dealing with hard learning data. The structured support vector machine [11] and the maximize margin model [3] can gain a maximize margin value for a given training data by solving an optimization problem (i.e quadratic programming). It is obvious that using such an optimization algorithm requires much computational time. In contrast to the previous method, this paper presents an online algorithm for semantic parsing in which we can attain the maximize margin of the training data without using an optimization technique. It is thus much easier to implement.

2.5 Features

In this part, we would like to present the features we used in our online learning framework. We describe the way to represent $\Phi(x, y, h)$ in the feature space. The following set of features are carefully selected from our online models.

- The first set of features is simply using a set of rules generated by using the synchronous grammar. In addition, similar to [20] for each word w within NL sentence we used a feature function that returns the number of times w is generated from word gaps. Generation of unseen words is modelled using an extra feature whose value is the total number of words generated from word gaps.
- The second set of features is two-level rules which are borrowed from Collins and Koo (2005). These two-level rules give the number of times a given rule is used to expand a non-terminal in a given parent rule.
- For the $\lambda - calculus$ representation we used the features of type checking such as the type mismatch errors in the MR translation.

For example in the type of mismatch feature, a "state" cannot possibly be a "river" in the semantic parsing of the sentence.

What is the entity that is a state and also a major river, that traverses something that is the largest?

answer(x1,largest(x2,(state(x1),major(x1),river(x1),traverse(x1,x2))))

Figure 4. Example of type mismatch.

2.6 Argmax Algorithm

One of the important components for applying online structured prediction is how well we design an argmax algorithm. Algorithm 1 sketches a kind of argmax algorithms. As shown in Algorithm 1, there are two argmax algorithms we need to design as follows.

$$y^*, h^* = \arg\max_{y,h}(w \cdot \Phi(x_i, y, h)) \quad (2)$$

$$h_i = \arg\max_{h}(w \cdot \Phi(x_i, y_i, h)) \quad (3)$$

As mentioned in previous subsection, for each x_i and y_i we might have several set of correspondent features h for mapping x_i to y_i. The argmax algorithm finds an appropriate correspond h_i among all possible h to transform x_i to

y_i. To solve this matter, we used a bottom up strategy that finds a maximal score in each sub-tree up to the full tree.

In the equation (2), we need to find the one with highest the score of $(w \cdot \Phi(x, y, h))$. To solve this matter, we simply used a version of Earley parsing algorithm [1].

Note that the argmax decoding could be used for others online prediction learning algorithms such as Perceptron and MIRA leaning. Like MIRA, our PA model can utilize the strength of k-best argmax by a simple strategy that finds among k-best output τ_t that minimize the loss function l_t.

3 Experimental Results

For the purpose of testing our models for semantic parsing, we used the CLANG corpus which is the RoboCup Coach Language (www.robocup.org). In the Coach Competition, teams of agents compete on a simulated soccer field and receive advice from a team coach in a formal language. The average length of an NL sentence in the CLANG [13] corpus is 22.52 words.

We also evaluated our online learning framework on the λ-SCFG grammar in the GEOQUERY domain. The larger GEOQUERY corpus consists of 880 English questions gathered from various sources (Wong and Mooney, 2006). The questions were manually translated into Prolog logical forms. The average length of a sentence is 7.57 words. The statistics information of these two corpora are shown in Table 1.

Table 1. Statistics on CLANG and GEO880 corpus. The average length of the MRs is large in the CLANG corpus.

Statistics	CLANG	GEO880
No.of. Examples	300	880
Avg. NL sentence length	22.5	7.48
Avg. MR length (tokens)	13.42	6.47
No. of non-terminals	37	44
No. of productions	102	133
No. of unique NL tokens	337	270

In the scope of this paper, we used the prediction based argmax algorithm and the PA update strategy for training model. We believe that other kinds of prediction base algorithms such as in [16] can be used in our semantic parsing framework.

We take into account the use of the standard 10-fold cross validation test for evaluating the methods (we used a single run in our experiment.). We used 270 sentences for training and 30 sentences for testing. To evaluate the proposed methods in parsing NL sentences to logical form, we measured the number of test sentences that produced complete logical forms, and the number of those logical forms that were correct. For the CLANG corpus, a logical form is correct if it exactly matches the correct representation, up to reordering of the arguments of commutative operators like "and". We used the evaluation method presented in [12], in which the performance of the parser was then measured in terms of precision and recall as in the formula below.

$$Precision = \frac{\#\text{correct}-\text{representation}}{\#\text{completed}-\text{representation}}$$
$$Recall = \frac{\#\text{correct}-\text{representation}}{\#\text{sentences}}$$

$$\text{F} - \text{measure} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

Table 2. Experimental results with CLANG corpus. The $\lambda - SCFG$ could not be used because the CLANG is free logic variable. We set the k-best parameter used in Algorithm 1 equal to 5, and the number of iterations is 10.

Methods	Precision	Recall	F-measure
KRIP (2006)	85.2%	61.9%	71.7
WASP	88.9%	61.39%	73.0
PA-structured	89.2%	68.9%	75.1

Table 2 shows the results of the KRIP system [13], the WASP system [20], and the PA-structured system on the CLANG corpus, respectively. We only compare our method with some other methods which does not use SAPT annotation like the method of [9][15]. Since the CCG methods of [23, 24] were not tested on the CLANG corpus, so we could not compare our methods with their methods. Table 2 shows that our system (PA-structured) is better than the KRIP system. It is comparable to the WASP system in terms of precision. But, it significantly improves previous works (WASP and KRIP) in terms of recall. This shows that the use of loss functions directly with the mechanism of online-large margin, can lead to a better performance. Our experimental results and other systems on CLANG showed that precision is high, while the recall is not. After observing the outputs, we found a problem that some NL sentences are not able to produce semantic representation. This seems that improving generation of learning models could lead a better performance. We believe that this problem will be solved when the size of training corpus becomes large.

For the purpose of testing our model in Prolog logical form, we used a 10-fold cross validation, and measured the performance of the learned parsers using precision and recall measurements. We used the same evaluation method described in [13] and [21] to compare our system with others. Note that in this evaluation method, a translation is considered correct if it retrieves the same answer as the correct logical form [21].

The results of testing our online learning framework shown in Table 3 indicate that the proposed framework works well on the GEOQUERY domain. Our method archived the best F-measure result in comparison with previous works.

Table 3. Experimental results with GEOQUERY corpus. We set the k-best parameter using in Algorithm 1 equal to 5, and the number of iterations is 15.

Methods	Precision	Recall	F-measure
KRIP (2006)	93.3%	71.7%	81.01
WASP(2006)	87.2%	74.77%	80.5
λ−WASP(2007)	92.0%	86.6%	89.2
CCG(2007)	95.5%	83.20%	88.9
λ-PA-structured	96.20%	88.1%	91.9

Using the $\lambda - SCFG$ representation also improves the accuracy in comparison with MRL representation. This indicates why our method shows improved results in comparison with WASP in the GEOQUERY domain. Table 3 also shows that we can obtain an improvement on the state of the art results by using a structured prediction methods, difference from the method of (W+R) [21] and (CCG) [24]. The advantage of our methods is the combination of using rich features with a loss function through a synchronous grammar.

In order to illustrate the accuracy of the PA-structured learning on semantic parsing, we have estimated the proposed models with respect to the size of training data. In addition, we used the same measure method in evaluation showing in Figure 5 and 6 for the two corpora. It means a logical form output is correct if it is the same with the gold-standard output.

Figure 5 and 6 showed the F-measure of the PA-structured with the size of training data (i.e 10, 20, ...). It showed that in both corpora, F-measure values are increasing with respect to the size of training data. It is also showed that there is a room to improve our results on these corpora. The F-measure of using 792 sentence for training is approximately to 0.75, while it is 0.91 by using the F-Measure method in [21]. We check these logical form outputs and found that there are several outputs having the same structure with the gold-standard but its order is different. This suggests that using a reordering method to correct the order of logical forms output may lead to an improvement. In addition, Figure 6 showed that the system's performance of using 640 sentences for training are similar to that of using 672 sentences for training on the GEO880 corpus. This suggested us that using an active learning mechanism to select suitable training examples for our framework seems promising to explore in the future work.

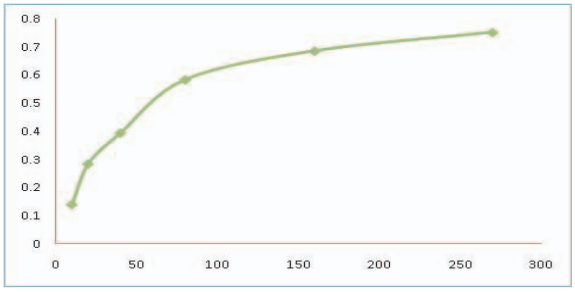

Figure 5. F-measure of the PA-structured on the CLANG corpus. F-measure values are with the size of training data (10, 20, 40, 80, 160, 270)

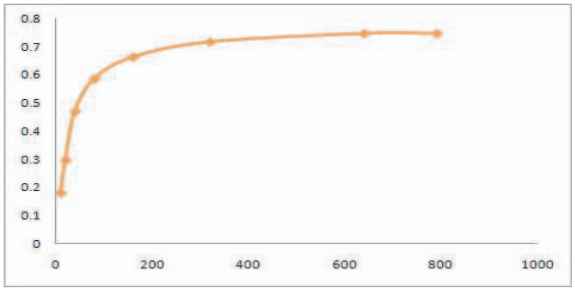

Figure 6. F-measure of the PA-structured on the GEO880 corpus. F-measure values are with the size of training data (10, 20, 40, 80,..,640, 792)

4 Conclusions

We have proposed a novel online structured prediction algorithm for semantic parsing using synchronous grammars for both meaning

representation and λ-calculus representation. We introduce an online learning algorithm with the use of hidden features to effectively estimate the semantic parsing models with a discriminative learning method. The advantages of our method are that it can incorporate a lot of features to a fine property model using a loss function to estimate the weights corresponding to features. In addition, our online learning algorithm is simple and easy to implement.

Experimental results on the two standard corpora show that the proposed models achieve the best result in CLANG corpus in term of F-measure, and attain a significant improvement on the state of the art results in the Prolog logical form corpus(GEO880).

There are some open directions for our methods. First, syntactic and semantic information of source language could be used in order to improve the current performance. In addition, in a real application there are some cases in which one natural language sentence can be mapped to several semantic representation. In order to deal with such

kinds of data, we would like to extend our current frameworks using multi-instance learning models. In future work, we also plan to use other kinds of the synchronous parsing framework such as CCG and TAG, in order to handle long-distance dependencies that occur in open-domain text. We also plan to exploit the proposed models on the more larger corpus such as presented in [10]

References

[1] Aho and Ullman. *The Theory of Parsing*. Prentice Hall, Englewood Cliffs, NJ, 1972.

[2] J. Allen. *Natural Language Understanding (2nd Edition)*. Mento Park, Benjaming/Cumming, 1995.

[3] B. Taskar, D. Klein, M. Collins, D. Koller, and C.D. Manning, 2004. Max-Margin Parsing. *In proceedings of EMNLP, 2004*.

[4] K. Crammer et al. 2006. Online Passive-Aggressive Algorithms. *Journal of Machine Learning Research 7* (2006) 551.585.

[5] M. Collins. 2002. Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms. *In Proceedings of EMNLP*.

[6] M. Collins and T. Koo. 2005. Discriminative reranking for natural language parsing. *Computational Linguistics*, 31(1):25.69.

[7] Y. Censor and S.A. Zenios. 1997. Parallel optimization: theory, algorithms, and applications. Oxford University Press, 1997.

[8] DeNero and Klein. Tailoring word alignments to syntactic machine translation. In *In Proceedings of ACL*, 2007.

[9] R. Ge and R. Mooney. A statistical semantic parser that integrates syntax and semantics. In *In Proceedings of CONLL*, 2005.

[10] Y. He and S. Young. Semantic processing using the hidden vector state models, 2005.

[11] T. J. I. Tsochantaridis, T. Hofmann and Y. Altun. Support vector machine learning for interdependent and structured output spaces. In *In Proceedings ICML 2004*, 2004.

[12] R. J. Kate and R. J. Mooney. Learning to transform natural to formal languages. In *In Proceedings of AAAI*, pages 825–830, 2005.

[13] R. J. Kate and R. J. Mooney. Using string kernels for learning semantic parsers. In *In Proceedings of the COLING/ACL 2006*, 2006.

[14] R. McDonald, K. Crammer, and F. Pereira. 2005. Online large-margin training of dependency parsers. *In Proceedings of ACL*.

[15] Le-Minh Nguyen, Akira Shimazu, and Xuan-Hieu Phan. Semantic parsing with structured svm ensemble classification models. In *In Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions*, 2006.

[16] Le-Minh Nguyen, Akira Shimazu, Xuan-Hieu Phan, Thai-Phuong Nguyen. A multilingual dependency analysis system using online passive-aggressive learning. In *Shared Task paper Proceedings of EMNLP-CONLL 2007*, 2006.

[17] Percy Liang, Alexandre Bouchard-Coîe, Dan Klein, Benjamin Taskar. An end-to-end discriminative approach to machine translation. In *An End-to-End Discriminative Approach to Machine Translation*, 2006.

[18] Peter F. Brown et al. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–312, June 1993.

[19] L.R. Tang. 2003. Integrating Top-down and Bottom-up Approaches in Inductive Logic Programming: Applications in Natural Language Processing and Relation Data Mining. Ph.D. Dissertation, University of Texas, Austin, TX, 2003.

[20] Wong and Mooney. Learning for semantic parsing with statistical machine translation. In *In Proceedings of (HLT/NAACL-2006)*, 2006.

[21] Wong and Mooney. Learning synchronous grammars for semantic parsing with lambda calculus. In *In Proceedings of ACL-07*, 2006.

[22] J. Zelle and R. Mooney. Learning to parse database queries using inductive logic programming. In *In Proceedings AAAI-96, 1050-1055*, 1996.

[23] L. Zettlemoyer and M. Collins. Learning to map sentences to logical form: Structured classification with probabilistic categorical grammars. In *In Proceedings of UAI 2005*, pages 825–830, 2005.

[24] L. Zettlemoyer and M. Collins. Online learning of relaxed ccg grammars for parsing to logical form. In *In Proceedings of the EMNLP-CoNLL-07*, 2007.