Understanding the Semantic Intent of Domain-Specific Natural Language Query

Juan Xu, Qi Zhang, Xuanjing Huang
Fudan University
School of Computer Science
{11210240066,qz,xjhuang}@fudan.edu.cn

Abstract

Queries asked on search engines nowadays increasingly fall in full natural language, which refer to Natural Language queries (NL queries). Parsing that kind of queries for the purpose of understanding user's query intent is an essential factor to search engine. To this end, a hierarchical structure is introduced to represent the semantic intent of NL query and then we focus on the problem of mapping NL queries to the corresponding semantic intents. We propose a parsing method by conducting two steps as follows: (1) predicting semantic tags for a given input query; (2) building an intent representation for the query using the sequence of semantic tags based on a structured SVM classification model. Experimental results on a manually labeled corpus show that our method achieved a sufficiently high result in term of precision and F1.

1 Introduction

Nowadays, with the development of voice search, queries asked on search engines often fall in full natural language, which refer to NL queries. For example, for the purpose of looking for a restaurant, it is natural for us to ask “find the best Italian restaurant near seattle washington” rather than “Italian restaurant seattle wa”. This means that voice search users are liable to express their intent in natural language which differs significantly from Web users. In addition to the developing of voice search, the increasing use of smartphones with voice assistant further boosts the number of such natural language queries.

This increasing amount of natural language queries brings a big challenge to search engines. Many search engines today, generally speaking, are based on matching keywords against structured information from relational database. Consider the query “find the best Italian restaurant near seattle washington”. Without understanding the semantic intent, it may retrieval some unsatisfying results that merely contain all these keywords, which are not really search terms (e.g. restaurant). But when the search engine understands the intent of this query is to “find restaurant”, and also knows the meanings of individual constituents (i.e the“restaurant” is head search terms, “the best Italian ” and “near seattle washington” are modifier), then it would be able to route the query to a specialized search module (in this case restaurant search) and return the most relevant and essential answers rather than results that merely contain all these keywords.

In no small part, the success of such approach relies on robust understanding of query intent. Most previous works in this area focus on query tagging problem, i.e., assigning semantic labels to query terms (Li et al., 2009; Manshadi and Li, 2009; Sarkas et al., 2010; Bendersky et al., 2011). Indeed, with the label information, a search engine is able to provide users with more relevant results. But previous works have not considered the issue for understanding the semantic intent of NL queries and their methods are not suitable for interpreting the semantic intent of this kind of complex queries. In this work, in order to enable search engines to understand natural language query, we focus on the problem of mapping NL queries from a particular search engine like Google maps, Bing maps etc, to their semantic intents representation. A key contribution of this work is that we formally define a hierarchical structure to represent the semantic intents. As an example, consider a query about finding a local business near some location such as:

Example : *find the best Italian restaurant near space needle seattle washington*

This query has four constituents: the Business that the user is looking for (restaurant), the Neighborhood (space needle seattle washington), the condition and cuisine type that the user specified (the best Italian) and the term that helps user to ask (find). To understand the semantic intent of the query, the model should not only be able to recognize the four constituents but also needs to understand the structure of each constituent. Therefore we are looking for a model that is able to generate the semantic intent representation for this query as shown in Figure 1.

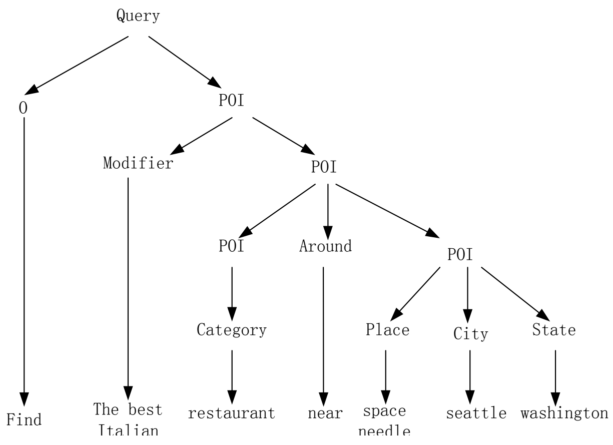


Figure 1: A simple grammar tree for local businesses search.

Generating the hierarchical structure of queries can be beneficial to information retrieval. Knowing the semantic role of each query constituent, we can reformulate the query into a structured form or reweight different query constituents for structured data retrieval (Kim et al., 2009; Paparizos et al., 2009; Gollapudi et al., 2011; Kim and Croft, 2012). Alternatively, the knowledge of the structure of the constituents helps route the query to a specialized search module.

The second contribution of this work is that we define a grammar which is isomorphic to a Context Free Grammar (CFG), and also present an approach which can automatically generate the semantic intent representation for NL queries by considering it as a structure classification problem.

The rest of this paper are organized as follows. Section 2 details the related work. In Section 3 we demonstrate the hierarchical structured representation of queries and introduce our grammar for it. Then, in Section 4, we propose our method for parsing the queries. Section 5 presents the experimental results. We draw the conclusions from our work in Section 6.

2 Related Works

2.1 Query Intent Understanding

To capture the underlying information need encoded within diverse user queries, considerable works have been conducted from many aspects. Most previous works in this area focus on query intent classification as one aspect, i.e., automatically mapping queries into semantic classes or patterns (Li et al., 2008; Arguello et al., 2009; Duan et al., 2012). There is another aspect to study the problem, query tagging, i.e., assigning semantic labels to query terms (Li et al., 2009; Sarkas et al., 2010; Bendersky et al., 2011). For example, in "Restaurant Rochester Chinese MN", the word "Restaurant" should be tagged as *Business*. In particular, Li et al. leverage clickthrough data and a database to automatically derive training data for learning a CRF-based tagger. Manshadi and Li develop a hybrid, generative grammar model for a similar task. Sarkas et al. consider an annotation as a mapping of a query to a table of structured data and attributes of this table, while Bendersky et al. mark up queries with annotations such as part-of-speech tags, capitalization, and segmentation.

There are relatively little published work on understanding the semantic intent of natural language query. Manshadi and Li (2009) and Li (2010) consider the semantic structure of queries. In particular, Li (2010) defines the semantic structure of noun-phrase queries as intent heads (attributes) coupled with some number of intent modifiers (attribute values), e.g., the query [alice in wonderland 2010 cast] is comprised of an intent head *cast* and two intent modifiers *alice in wonderland* and *2010*. Our approach differs from the earlier work in that we investigate the natural language query intent understanding problem, and build a hierarchical representation for it.

2.2 Semantic Parsing

For the purpose of enabling search engines to understand user's query intent, we present an approach to parse NL queries to the corresponding semantic intents, which is similar to the task in semantic parsing (Kate et al., 2005). We parse the queries to a hierarchical structure consisting of all query terms. While the task of semantic parsing is mapping NL input to its interpretation expressed in well-defined formal *meaning representation*(MR) language. For example, *"How many states does the Colorado river run through?"*, the output of semantic parsing is *"count(state(traverse(river(const(colorado))))"*. Although it can express the meanings of NL inputs, it is not suitable for search engines to use, which, generally speaking, are based on matching keywords against documents.

3 Query Intent Representation and Grammar for NL Query

The task we defined is mapping the natural language query (NL query) to the corresponding intent representation(IR), which is specifically for a particular search scenario like Google maps and Bing maps etc, but can generalize well to other search scenarios by redefining the grammar.

3.1 Query Intent Representation

In this work, we propose to use a tree structure to represent the semantic intent of a query. Consider the instance in section 1, the NL query "find the best Italian restaurant near space needle seattle washington" consists of 10 words. The IR is a hierarchical tree structure, as shown in Figure 1.

There are 9 different non-terminal symbols in the tree, of which Query is the start symbol. And it contains 11 rules to formulate the tree structure. As shown in the Figure 1, it clearly depicts that the query has 2 constituents and also depicts the structure of each constituent. After having that tree structure, search engines can easily understand the semantic intent of the query as we discussed in introduction.

3.2 Grammar for NL Query

We have manually designed a grammar for the purpose of automatic generating the hierarchical structure of queries. As mentioned above, we focus on a particular search scenario (i.e map search domain). Based on analysis of NL queries from that domain, we observe that most queries carry an underlying structure. Therefore a set of CFG rules were written for the map search scenario. Below are some sample rules from those CFG rules:

$Query --> O\ POI\ Around\ POI$
$POI --> Place\ IN\ City\ State$
$POI --> Num\ Road\ IN\ City\ State$
$POI --> Road$
$POI --> Category\ IN\ City$
$Query --> Modifier\ Transition\ From\ POI\ TO\ POI$

And we also define a set of semantic tags for that kind of queries which indicate the semantic role of each query constituent. More formally we define a Context-Free Grammar (CFG) for NL query as a 4-tuple G=(N, T, S, R) where

- N is a set of non-terminals;
- T is a set of terminals;
- $S \in N$ is a special non-terminal called start symbol,
- R is a set of rules $\{A \rightarrow \beta\}$ where A is non-terminal and β is a string of symbols from the infinite set of strings of $(N \cup T)^*$.

The sequence of $\Longrightarrow$ used to derive w from S is called a derivation of w. Here $\Longrightarrow^*$ is defined as *the reflexive transitive* closure of $\Longrightarrow$. We can then formally define the language $L(G)$ generated by the grammar G as the set of strings composed of terminal symbols that can be drawn from the designated start symbol S.

$L = \{w | w$ is in $T*$ and $S \Longrightarrow^* w\}$

Given the above definitions, parsing a string w means to find all (if any) the derivations of w from S.

Our grammar composes a constraint ordering on queries. And it is reasonable, because the query we investigated is full natural language query. While there are some NL queries whose words sequences are arbitrary, which is not in our consideration.

4 Methodology

To produce the semantic intent representation for an NL query from map search domain, we need to extract the basic semantic query constituents and build the semantic intent representation with them according to the query. Summarily, the input of our parsing task is a NL query and the output is the intent representation. The proposed method for this problem consists of two phases:

Semantic category	Meaning	Example
O	The useless information for search	where is
M	The modifier which indicates user's personal interests	the best and cheapest
F	The information of predicting the start address in direction search	from
Q	The information of predicting the end address in direction search	to
C	The city	New York
S	The state	North Carolina(Abbr. NC)
P	The place	White House
T	The town	Forbes
A	The predicted information of the area range of a point	nearby
L	The POI category	restaurant
I	The predicted information of a point within an area	in
D	The district	Brooklyn
N	The number of an address	Room 606
R	The road	Church St
G	The country	USA
W	The transition	via subway

Table 1: Illustration for each semantic category.

1. Predicting a sequence of semantic tags for a given input sentence.
2. Building an intent representation with the sequence of semantic tags

We describe a method using CRFs and structured support vector machine (SSVMs) in the following subsection for the first step and the second step, respectively.

4.1 Semantic Tagging with CRFs

Let $w_1 w_2 ... w_N$ be a NL sentence in which N is the number of words and w_i is i^{th} word. Assuming that a chunk tag for a sequence of words $w_i ... w_j (1 \leq i \leq j \leq N)$ is t_i, the lexical semantic prediction problem is to determine a lexical semantic tag s_i for a sequence of words $w_i ... w_j (1 \leq i \leq j \leq N)$. In the meantime, semantic tag s_i is structural and consists of two parts:
1) **Boundary Category**: BC = {B, I}. Here B/I means that current word is at the beginning/in the middle of a semantic chunk.
2) **Semantic Category**: SC = {O, M, F, Q, C, S, P, T, A, L, I, D, N, R, G, W}. This is used to denote the class of the semantic name. The meaning of each semantic class name represented is illustrated in Table 1.

The conditional random fields CRFs (Lafferty et al., 2001) have shown empirical successes in label sequence labelling problem. Therefore, we exploit the use of CRFs to our semantic tagging problem in which a feature set for this task is presented in Table 2.

4.2 Generating Intent Representation with Structural SVMs

This subsection first gives some background about the SSVMs for structured prediction, and then we focus on how to use SSVMs to our intent representation learning problem.

4.2.1 Structural SVMs

Suppose a training set of input-output structure pairs $S = \{(x_1, y_1), \ldots, (x_n, y_n)\} \in (X \times Y)^n$ is given. Structured classification is the problem of predicting y from x in the case where y has a meaningful internal structure. Elements $y \in Y$ may be, for instance, sequences, strings, labeled trees, lattices, or graphs.

The approach we pursue is to learn a discriminant function $F : X \times Y \rightarrow R$ over S. For a specific given input x, we can derive a prediction by maximizing F over the response variable. Hence, the general form of our hypotheses f is

$$f(x; w) = \text{argmax}_{y \in Y} F(x; y; w) \qquad (1)$$

where w denotes a parameter vector.

As the principle of the maximum-margin presented in (Varpnik, 1995), in the structured classification problem, Tsochantaridis (2004) proposed several maximum-margin optimization problems

Feature index	Definition
1	The current word and the preceding word
2	The current word and the following word
3	The current word
4	The knowledge of Pre-1 word in Geo-Knowledge Database
5	The knowledge of Pre-2 word in Geo-Knowledge Database
6	The knowledge of Post-1 word in Geo-Knowledge Database
7	The knowledge of Post-2 word in Geo-Knowledge Database
8	The knowledge of current word in Geo-Knowledge Database

Table 2: Features for CRFs Model.

$\delta\psi_i(y) \equiv \psi(x_i, y_i) - \psi(x_i, y)$. The soft-margin criterion was proposed in order to allow errors in the training set, by introducing slack variables.

$$\mathrm{SVM}_1 : \underbrace{\min}_{w} \frac{1}{2}\|w\|^2 + \frac{C}{n}\sum_{i=1}^{n}\xi_i \quad s.t.\ \forall i, \xi_i \geq 0 \tag{2}$$

$$\forall i, \forall y \in Y \backslash y_i : \langle w, \delta\psi_i(y)\rangle > 1 - \xi_i \tag{3}$$

Alternatively, using a quadratic term $\frac{C}{2n}\sum_i \xi_i^2$ to penalize margin violations, SVM_2 would be obtained. Here $C > 0$ is a constant that control the tradeoff between training error minimization and margin maximization.

To deal with problems in which |Y| is very large, such as Natural Language parsing, Tsochantaridis (2004) proposed two approaches that generalize the formulation SVM_1 and SVM_2 to the cases of arbitrary loss function. We use SVM_1 to introduce that two approaches and they are also work for SVM_2. The first approach is to rescale the slack variables according to the loss incurred in each of the linear constraints.

$$\mathrm{SVM}_1^{\triangle_s} : \underbrace{\min}_{w} \frac{1}{2}\|w\|^2 + \frac{C}{n}\sum_{i=1}^{n}\xi_i \quad s.t.\ \forall i, \xi_i \geq 0 \tag{4}$$

$$\forall i, \forall y \in Y \backslash y_i : \langle w, \delta\psi_i(y)\rangle \geq \frac{1 - \xi_i}{\triangle(y_i, y)} \tag{5}$$

The second approach to include loss function is to rescale the margin as a special case of the Hamming loss. The margin constraints in this setting take the following form:

$$\forall i, \forall y \in Y \backslash y_i : \langle w, \delta\psi_i(y)\rangle \geq \triangle(y_i, y) - \xi_i \tag{6}$$

This set of constraints yields an optimization problem, namely $\mathrm{SVM}_1^{\triangle^m}$.

The algorithm to solve the maximum-margin problem in structured learning problem is presented in detail in (Tsochantaridis et al., 2004). And it can be applied to all SVM formulations mentioned above. The only difference between them is the cost function.

Following the successes of the Structural SVMs algorithm to structured prediction, we exploit the use of SSVM to our parsing task. As discussed in (Tsochantaridis et al., 2004), the major problem to apply the Structural SVMs is to implement the feature mapping $\psi(x, y)$, the loss function $\triangle(y_i, y)$, as well as the maximization algorithm. In the following section, we apply a Structural SVMs to the problem of semantic intent learning in which the mapping function, the maximization algorithm, and the loss function are introduced.

4.2.2 Feature mapping

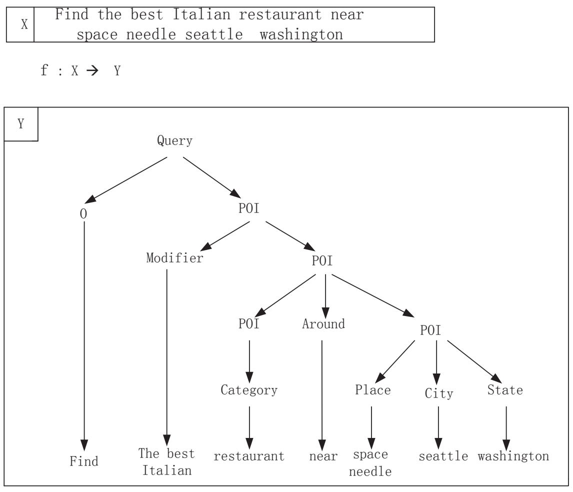
For our task, we can choose a mapping function to get a model that is isomorphic to a probabilistic grammar in which each rule within the grammar is defined by our own based on the application area. Each node in a parse tree y for an NL query x corresponds to a grammar rule g_j, which in turn has a score w_j.

All valid parse trees y for an NL query x are scored by the sum of the w_j of their nodes, and the feature mapping $\psi(x, y)$ is a histogram vector counting how often each grammar rule g_j occurs in the tree y. The example shown in Figure 2 clearly depicts the way features are mapped from a tree structure intent representation of an NL query.

4.2.3 Loss function

Typically, the correctness of a predicted parse tree is measured by its F1 score (see e.g. (Johnson, 1998)), the harmonic mean of precision of recall as calculated based on the overlap of nodes between the trees. In this work, we follow this loss function and introduce the standard zero-one classification loss as a baseline measure method.

Let z and z_i be two parse tree outputs and $|z|$ and $|z_i|$ be the number of brackets in z and z_i, re-



$$\Psi(x,y) = \begin{bmatrix} 1:\text{Query}\rightarrow\text{O POI} \\ 1:\text{O}\rightarrow\text{Find} \\ 1:\text{POI}\rightarrow\text{Modifier POI} \\ 1:\text{POI}\rightarrow\text{category} \\ 0:\text{POI}\rightarrow\text{Place City} \\ \dots \\ 1:\text{State}\rightarrow\text{washington} \end{bmatrix}$$

Figure 2: Example of feature mapping using tree representation.

spectively. Let n be the number of common brackets in the two trees. The loss function between z_i and z is computed as bellow.

$$F - loss(z_i, z) = 1 - \frac{2\times n}{|z| + |z_i|} \quad (7)$$

$$\text{zero} - \text{one}(z_i, z) = \begin{cases} 1 & \text{if } z_i \neq z \\ 0 & \text{otherwise} \end{cases} \quad (8)$$

4.2.4 Maximization algorithm

Note that the learning function can be efficiently computed by finding the structure $y\in Y$ that maximizes $F(x;y;w) = \langle w, \delta\psi_i(y)\rangle$ via a maximization algorithm. To this end, we use a modified version of the CKY parser of Mark Johnson[1] and incorporated it into our algorithm.

4.2.5 SSVM learning algorithm

Algorithm 1 shows our generation of SSVM learning for the semantic intent representation learning problem. The algorithm can apply to all SVM formulations discussed in section 4.2.1. The only difference is in the way the cost function gets set up in step 3 and the other three optional cost function are:

$\text{SVM}_2^{\triangle^s} : H(y) \equiv (1 - \langle\delta\psi_i(y), w\rangle)\sqrt{\triangle(y_i, y)}$

[1]http://www.cog.brown.edu/∼mj/Software.htm.

Algorithm 1 Algorithm of SSVM learning for query parsing

```
Input: $I = (x_i; y_i), i = 1, 2, \dots, l$ in which $x_i$ is
    the NL query's semantic tags sequence and $y_i$
    is the corresponding tree structure.
Output: SSVM model
 1: repeat
 2:   for $i = 1$ to $l$ do
 3:     set up cost function based on the corre-
        sponding optimization problem;
        $\text{SVM}_1^{\triangle^s} : H(y) \equiv (1 - \langle\delta\psi_i(y), w\rangle)\triangle(y_i, y)$
 4:     compute $\hat{y} = \text{argmax}_{y\in Y} H(y)$
 5:     compute $\xi_i = \max\{0, \max_{y\in S_i} H(y)\}$
 6:     if $H(\hat{y}) > \xi_i + \epsilon$ then
 7:       $S_i \leftarrow S_i \bigcup \{\hat{y}\}$
 8:       solving optimization with SVM;
 9:     end if
10:   end for
11: until no $S_i$ has changed during iteration
```

$\text{SVM}_1^{\triangle^m} : H(y) \equiv (\triangle(y_i, y) - \langle\delta\psi_i(y), w\rangle)$
$\text{SVM}_2^{\triangle^m} : H(y) \equiv (\sqrt{\triangle(y_i, y)} - \langle\delta\psi_i(y), w\rangle)$
The feature mapping $\psi(x, y)$, the loss function $\triangle(y_i, y)$, as well as the maximization in step 6 were implemented as mentioned in above. A working set S_i is maintained for each training example (x_i, y_i) to keep track of the selected constraints which define the current relaxation. The algorithm stops, if no constraint is violated by more than ϵ and then we get a SSVM model. Note that the SVM optimization problems from iteration to iteration differ only by a single constraint. We therefore restart the SVM optimizer from the current solution, which really reduces the runtime.

5 Experiments

5.1 Corpus

To facilitate the study we need benchmark corpus with ground-truth semantic intent representations in map search area. Since there is no such kind of corpus publicly available, we constructed a corpus MSItent from answers.yahoo.com, which have a large number of queries submitted by users. The MSItent corpus contains 1200 NL queries. Since those queries were crawled from an open question domain which contains many noises, 670 NL queries were finally chosen and manually labeled.

Two annotators labeled the corpus independently. The annotators first tagged each query with a set of semantic tags, and then build it's corresponding semantic intent representation tree based on a given grammar, which consists of 17 non-terminal and 269 productions, defined specifically for our task. In order to keep the reliability of annotations, another annotator was asked to check the corpus and determine the conflicts. Finally we got a corpus includes 670 NL queries and their corresponding semantic intent representation. Table 3 shows the statistic on our corpus MSItent.

Statistic	Numbers
No.of. Examples	670
Avg. NL sentence length	10.11
No. of non-terminals	17
No. of productions	227

Table 3: Statistics on MSItent corpus. The average length of an NL query in the corpus is 10.11 words. This indicates that MSItent is the hard corpus.

5.2 Experiments Configurations

We use the standard 10-fold cross validation test for evaluating the method. NL test queries were first tagged with a sequence of semantic tags, and those sequences were used to build trees, which has been detailed in section 3, via a structured support vector machine. [2] We evaluate the accuracy of tagging an NL query to a sequence of semantic tags by computing the total number of correct semantic tags in comparison with the gold-standard. For this purpose, CRF model[3] obtains a high result, with 91.73% accuracy.

Our main focus is to evaluate the proposed method in parsing NL queries to IR, we measure the number of test queries that produced complete IR, and the number of those IR that were correct. For our task, a IR is correct if it exactly matches the correct representation, we use the evaluation method for semantic parsing problem presented in (Kate et al., 2005) as the formula be:

$$\text{Precision:} = \frac{\#correct\ IR}{\#completed\ IR}$$

$$\text{Recall:} = \frac{\#correct\ IR}{\#queries}$$

$$\text{F1:} = \frac{2Precision{\cdot}Recall}{Precision + Recall}$$

[2]http://svmlight.joachims.org/

[3]We use CRF++ toolkit, http://crfpp.sourceforge.net/

5.3 Results

In this section we show that with high tagging accuracy as mentioned above, our proposed method for parsing NL queries to IR is effective. To this end, we conducted two sets of experiment. The first experiment was to show the performance of our proposed method in terms of precision, recall and F-score measurements. The second was to investigate the effect of other kernel functions in our learning algorithm.

1. *Performance of our method.* The results are given in Table 4, which shows recall, precision and F1 for test set. The first line shows the performance for PCFG model as trained on our MSIntent corpus by Johnson's implementation. The following two lines show the $\text{SVM}_2^{\triangle^m}$ and $\text{SVM}_2^{\triangle^s}$ with zero-one loss, while the rest lines give the results for the F1-loss. All results are for $C = 39$ and $\epsilon = 0.1$. All values of C between 0.1 to 100 gave comparable results. Table 4 indicates that our method achieved high accuracy results. $\text{SVM}_2^{\triangle^m}$ using the F-loss gives better F1, outperforming the PCFG substantially. We conjecture that we can achieve further gains by incorporating more complex features into the grammar, which would be impossible or at least awkward to use in a PCFG model.

2. *The effect of kernel functions in our learning algorithm.* As seen in Table 5, it shows the training and testing results for various kernel functions including linear kernel, polynomial kernel, and RBF kernel. The regularization parameter C and the criterion parameter ϵ are set to the same values as that in the first experiment. From the results, $\text{SVM}_2^{\triangle^s}$ with polynomial kernel obtains 83.86% recall, 89% precision and 86.43% F1, which is the best result. But we observe that our proposed method can perform well with different kernel functions without significantly difference.

In addition, when performing SSVM on the test set, we might obtain some 'NULL' outputs since the grammar generated by SSVM could not derive this sentence, but generally we obtained high recall. Summarily, Table 5 depicts that the proposed method using different parameters achieved high performance in term of precision.

6 Conclusions

This paper presented a new facet to investigate the semantic intent of NL queries, which maps NL queries to the corresponding semantic intents.

Parameter	Test Recall	Test Precision	Test F1
PCFG	79.10	89.83	84.12
0/1-loss(SVM$_2^{\triangle^m}$)	83.43	88.05	85.78
0/1-loss(SVM$_2^{\triangle^s}$)	83.26	88.57	85.47
F-loss(SVM$_2^{\triangle^m}$)	**83.42**	**88.72**	**85.97**
F-loss(SVM$_2^{\triangle^s}$)	83.01	88.39	85.60

Table 4: Results for parsing NL queries to IR on MSIntent corpus using cross-validation test.

Parameter	Training Accuracy	Test Recall	Test Precision	Test F1
linear+F-loss($\triangle_m$)	92.37	83.42	88.72	85.97
polynomial(d=2)+ F-loss($\triangle_m$)	91.66	82.98	88.23	85.13
polynomial(d=2)+ F-loss($\triangle_s$)	**91.98**	**83.86**	**89.00**	**86.43**
RBF+F-loss($\triangle_m$)	92.00	83.11	88.22	85.17
RBF+F-loss($\triangle_s$)	92.34	82.67	87.96	84.92

Table 5: Experiment results for parsing NL queries to IR on MSIntent corpus using SSVMs with various kernel functions.

We also proposed a method of using a hierarchical structure to represent the semantic intent of N-L query, and then presented an automatic method to learn the semantic intent representation with the corpus of NL queries and their semantic intent representation in tree structure.

Experimental results with a manually labeled corpus have demonstrated that our method achieves a very good performance in term of precision and F1-scores. We thus can confidently conclude that the structured support vector models are suitable to the problem of our semantic intent learning problem. We also provide a semantic tagging tool with a very high accuracy by using a CRF model that can be beneficially used as preprocessing for the semantic intent learning problem.

The main drawback with our approach is that we strict the ordering of NL queries. Note that although strict ordering constraints such as those imposed by CFG is appropriate for modeling query structure in our task, it might be helpful to somehow ignore the ordering. We leave this for future work. Another interesting and practically useful problem that we have left for future work is to extend our method to a version of SVM semi-supervised learning. Having such a capability, we are able to automatically learn the semantic intent of NL queries by processing labeled and unlabeled data.

7 Acknowledgments

The authors wish to thank the anonymous reviewers for their helpful comments. This work was partially funded by National Natural Science Foundation of China (61003092, 61073069), National Major Science and Technology Special Project of China (2014ZX03006005), Shanghai Municipal Science and Technology Commission (No.12511504500) and "Chen Guang" project supported by Shanghai Municipal Education Commission and Shanghai Education Development Foundation(11CG05).

References

Jaime Arguello, Fernando Diaz, Jamie Callan, and Jean-Francois Crespo. 2009. Sources of evidence for vertical selection. In *Proceedings of the 32nd international ACM SIGIR conference on Research and* *development in information retrieval*, pages 315–322. ACM.

Michael Bendersky, W Bruce Croft, and David A Smith. 2011. Joint annotation of search queries. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, volume 1, pages 102–111.

Huizhong Duan, Emre Kiciman, and ChengXiang Zhai. 2012. Click patterns: an empirical representation of complex query intents. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, pages 1035–1044. ACM.

Sreenivas Gollapudi, Samuel Ieong, Alexandros Ntoulas, and Stelios Paparizos. 2011. Efficient query rewrite for structured web queries. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, pages 2417–2420. ACM.

Mark Johnson. 1998. Pcfg models of linguistic tree representations. *Computational Linguistics*, 24(4):613–632.

Rohit J Kate, Yuk Wah Wong, and Raymond J Mooney. 2005. Learning to transform natural to formal languages. In *Proceedings of the National Conference on Artificial Intelligence*, volume 20, page 1062. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999.

Jin Young Kim and W Bruce Croft. 2012. A field relevance model for structured document retrieval. In *Advances in Information Retrieval*, pages 97–108. Springer.

Jinyoung Kim, Xiaobing Xue, and W Bruce Croft. 2009. A probabilistic retrieval model for semistructured data. In *Advances in Information Retrieval*, pages 228–239. Springer.

John Lafferty, Andrew McCallum, and Fernando CN Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data.

Xiao Li, Ye-Yi Wang, and Alex Acero. 2008. Learning query intent from regularized click graphs. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 339–346. ACM.

Xiao Li, Ye-Yi Wang, and Alex Acero. 2009. Extracting structured information from user queries with semi-supervised conditional random fields. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 572–579. ACM.

Xiao Li. 2010. Understanding the semantic structure of noun phrase queries. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1337–1345. Association for Computational Linguistics.

Mehdi Manshadi and Xiao Li. 2009. Semantic tagging of web search queries. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2*, pages 861–869. Association for Computational Linguistics.

Stelios Paparizos, Alexandros Ntoulas, John Shafer, and Rakesh Agrawal. 2009. Answering web queries using structured data sources. In *Proceedings of the 35th SIGMOD international conference on Management of data*, pages 1127–1130. ACM.

Nikos Sarkas, Stelios Paparizos, and Panayiotis Tsaparas. 2010. Structured annotations of web queries. In *Proceedings of the 2010 international conference on Management of data*, pages 771–782. ACM.

Ioannis Tsochantaridis, Thomas Hofmann, Thorsten Joachims, and Yasemin Altun. 2004. Support vector machine learning for interdependent and structured output spaces. In *Proceedings of the twenty-first international conference on Machine learning*, page 104. ACM.

V Varpnik. 1995. The nature of statistical learning theory.