

# QUERY UNDERSTANDING ENHANCED BY HIERARCHICAL PARSING STRUCTURES

Jingjing Liu<sup>1</sup>, Panupong Pasupat<sup>1</sup>, Yining Wang<sup>2</sup>, Scott Cyphers<sup>1</sup>, and Jim Glass<sup>1</sup>

<sup>1</sup>MIT Computer Science & Artificial Intelligence Laboratory, Cambridge, MA 02139, U.S.A.

<sup>2</sup>IIIS, Tsinghua University, Beijing 100084, China

{jingl, ppasupat, ynwang, cyphers, glass}@csail.mit.edu

## ABSTRACT

Query understanding has been well studied in the areas of information retrieval and spoken language understanding (SLU). There are generally three layers of query understanding: domain classification, user intent detection, and semantic tagging. Classifiers can be applied to domain and intent detection in real systems, and semantic tagging (or slot filling) is commonly defined as a sequence-labeling task -- mapping a sequence of words to a sequence of labels. Various statistical features (e.g.,  $n$ -grams) can be extracted from annotated queries for learning label prediction models; however, linguistic characteristics of queries, such as hierarchical structures and semantic relationships, are usually neglected in the feature extraction process. In this work, we propose an approach that leverages linguistic knowledge encoded in hierarchical parse trees for query understanding. Specifically, for natural language queries, we extract a set of syntactic structural features and semantic dependency features from query parse trees to enhance inference model learning. Experiments on real natural language queries show that augmenting sequence labeling models with linguistic knowledge can improve query understanding performance in various domains.

**Index Terms**— query understanding, semantic tagging, linguistic parsing

## 1. INTRODUCTION

Query understanding has been one of the key issues in language processing, especially for information retrieval or personal assistant systems (e.g., Siri). Given a spoken/textual query from a user, the challenge is to understand the user's intent as well as interpreting the semantic details of the query, in order for the systems to take further actions (e.g., searching for information, continuing a dialogue, or executing commands). There are three key components required by a query understanding engine: (1) domain classification; (2) user intent detection; and (3) semantic tagging. For example, given a query: “play the trailer for the movie starring Johnny Depp and featuring Caribbean pirates,” the domain and the user intent could be identified as “movie” and “play\_trailer”; and the key

semantic concepts (or slots) could be extracted as “<*Actor*> Johnny Depp </*Actor*>” and “<*Plot*> Caribbean pirates </*Plot*>”.

There has been a large body of work on semantic tagging on textual/spoken queries (or utterances) [1][2][3][4][5], such as using conditional random fields (CRF) [6] for label sequence prediction. There have also been some studies on improving query understanding performance, by learning lexicons from the web or user queries [7][8][9], or by jointly modeling domains, intents and slots for SLU [10][11][12]. Commonly used features for learning semantic tagging models include transit features, lexicons,  $n$ -grams, and regular expressions [5][9][13]. However, linguistic characteristics of queries, such as long-distance semantic dependencies and hierarchical syntactic structures, are often neglected in the feature extraction process. Such structural information, if well used, could potentially augment query interpretation, especially for those queries with complex syntactic/semantic structures. Thus, leveraging linguistic knowledge from parsing hierarchies is a promising approach to improving semantic understanding on natural language queries.

Recently, there have been some attempts at fusion methods incorporating linguistic knowledge for semantic parsing and semantic role labeling. For example, [14][15] proposed a hierarchical joint learning model for parsing and named entity recognition based on a CRF-CFG (context-free grammar) parser. [16] proposed a novel method for semi-supervised learning of non-projective log-linear dependency parsers using directly expressed linguistic prior knowledge. And for semantic role labeling, a framework was proposed to incorporate syntactic parsing information such as linguistic and structural constraints into a joint inference model [17]. For query understanding, there has also been a study on using a hybrid context-free grammar for web query tagging, where a domain-dependent CFG was constructed and a discriminative model was used on top of the probabilistic parser to re-rank the  $N$ -best parse trees of queries [18].

In this work, we investigate a query understanding approach that leverages linguistic parsing structures without requiring a domain-dependent grammar. We focus on the type of queries that have complex constituents, as exemplified in Figure 1. This type of query tends to have

long segments (e.g., plot description clauses), which are difficult to capture by general patterns like lexicons or regular expressions. Linguistic parsing knowledge, on the other hand, could play an important role in such cases, where structural information of queries (e.g., sub-clauses and sub-sentences) is represented by hierarchical parse trees. Thus, for such natural language queries with complex semantics, we propose a set of syntactic structural features (e.g., clause-level sub-trees) to make use of the hierarchical information of a query, as well as a set of semantic relational features (e.g., verb-argument relationships, long-distance dependencies) for inference model learning. These linguistic features can be extracted from the parse trees of queries generated by domain-independent parsers. The augmented models can then be deployed to search systems for interpreting users' input -- when a user submits a natural language query to the system, the spoken/textual query (or utterance) will be parsed on the fly by the domain-independent parser, and the additional linguistic features will be taken into account in the inference process for label prediction.

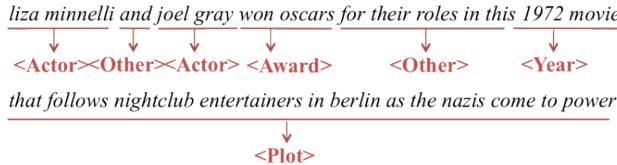


Figure 1: Semantic tags for the query “liza minnelli and joel gray won oscars for their roles in this 1972 movie that follows nightclub entertainers in berlin as the nazis come to power”.

The rest of the paper is organized as follows: in Section 2 we will explain the proposed approach to query tagging by leveraging hierarchical parsing (structural and relational) features. Section 3 describes the experiments and the evaluation conducted on natural language queries collected from real users. Section 4 concludes the paper and points to future work.

## 2. APPROACH

In our proposed approach, we employ semi-Markov CRFs [19] as the query tagging models, which have been extensively studied in the past few years [1][2][3][8]. In this section, we will first explain the models briefly, and then describe the proposed linguistic features that will be leveraged into model learning.

### 2.1 Models

Semi-Markov CRFs [19] model the conditional probability of a segment-based label sequence given the input. More specifically, given the word sequence  $x = (x_1, x_2, \dots, x_M)$ , the goal is to find  $s = (s_1, s_2, \dots, s_N)$ , which denotes a segmentation of the input as well as a classification of all

segments. Each segment is represented by a tuple  $s_j = (u_j, v_j, y_j)$ , where  $u_j$  and  $v_j$  are the start and end indices of the segment, and  $y_j$  is a class label. Segmentation and classification is jointly modeled by:

$$p(s|x) = \frac{1}{Z_\lambda(x)} \exp \left\{ \sum_{j=1}^{N+1} \lambda \cdot f(s_{j-1}, s_j, x) \right\} \quad (1)$$

where  $f(s_{j-1}, s_j, x)$  is a vector of feature functions defined on the segment level. Given labeled queries, we estimate  $\lambda$  in (1) that maximizes the conditional likelihood of training data while regularizing model parameters. The learned model is then used to predict the label sequence  $s$  for a future input  $x$  (i.e., a new query).

### 2.2 Features

Commonly used features for learning semantic tagging models include transit features,  $n$ -grams and regular expressions [1]. Some studies have also employed lexical features with external dictionaries and shallow parsing features such as Part-of-Speech (POS) tags [1][20], which have proved to work well on short phrase-based segments (e.g., named entities). However, for semantically complex queries like the example in Figure 1, lexicons or shallow parsing features are not sufficient for capturing the sentence-level long segments (e.g., “<Plot>”). Thus, to leverage additional linguistic knowledge, we introduce two types of parsing features into query semantic tagging: hierarchical syntactic features and semantic dependency features, which could provide critical information on constituent hierarchies of sentences.

#### 2.2.1 Hierarchical Syntactic Features

The goal of the hierarchical features is to make use of parsing structures such as sub-clauses in parse trees to capture the syntactic information in a query. For example, Figure 2 shows the labeled semantic classes (on the top) and the linguistic parse tree (on the bottom) of an example query. The hierarchical parse tree shows that, an *NP* (noun phrase) specifies the genre of the inquired movie -- “*NP (DT a) (CD 1984) (NN cult) (NN classic)*”, and a chunk of *NN* (singular noun) indicates the actors’ names -- “(*NN emilio*) (*NN estevez*)” and “(*NN harry*) (*NN dean*) (*NN stanton*).”. Also, there is an *SBAR* clause in the parse tree, embracing the description of the movie plot (“*that features prominent drug usages and aliens*”). SBARs often consist of a complementizer and a sub-sentence (e.g., *SBAR* → “*that* + *S*”), which could be generated in several cases: embedded clauses with “*that*,” relative clauses with a pronoun, and dependent clauses with a conjunction. These are all general patterns for such descriptive queries. Intuitively, the sub-tree structures in the parsing results of queries could align naturally with the labeled segments, as exemplified in Figure 2.

a   [Other]
1984   [Release Year]
cult classic   [Genre]
starring   [Other]
emilio estevez   [Actor]
and   [Other]
harry dean stanton   [Actor]
that features   [Other]
prominent drug usages and aliens   [Plot]
(ROOT
(NP
(NP (DT a) (CD 1984) (NN cult) (NN classic))
(PP (VBG starring)
(NP
(NP (NN emilio) (NN estevez)
(CC and)
(NN harry) (NN dean) (NN stanton))
(SBAR
(WHNP (WDT that))
(S
(VP (VBZ features)
(NP (NP ((JJ prominent) (NN drug) (NNS usages))
(CC and)
(NNS aliens))))))

Figure 2: Semantic tags (top) and the linguistic parse tree (bottom) for the query “a 1984 cult classic starring emilio estevez and harry dean stanton that features prominent drug usages and aliens”.

Therefore, we employ a list of structural features to make use of such segment alignments with parse trees. These features will only be fired up when there is an alignment between a labeled segment and a complete subtree.

*Node* feature:

$$f(s_{j-1}, s_j, x) = \delta(N(s_j) = t) \delta(y_j = b) \quad (2)$$

where  $N(s_j)$  is the root of the sub-tree that matches the current segment  $s_j$ .  $t$  could be a parse-tree node such as  $NP$ ,  $VP$  or  $S$ .  $y_j$  is a class label (e.g.,  $b$  could be  $<Plot>$ ,  $<Actor>$ , etc.).

*Node-And-Length* feature:

$$f(s_{j-1}, s_j, x) = \delta(N(s_j) = t) \delta(|s_j| = k) \delta(y_j = b) \quad (3)$$

which is similar to the *Node* feature except that it includes the length of the segment as well ( $k = 1, 2, 3, \dots$ ).

*Node-and-Children* feature:

$$f(s_{j-1}, s_j, x) = \delta(N(s_j) = t) \delta(C(s_j) = n) \delta(y_j = b) \quad (4)$$

where  $C(s_j)$  is the list of the direct children of the parse-tree node  $t$ .  $n$  could be a node sequence such as  $NP-VP$ .

*Node-and-POS* feature:

$$f(s_{j-1}, s_j, x) = \delta(N(s_j) = t) \delta(P(s_j) = n) \delta(y_j = b) \quad (5)$$

where  $P(s_j)$  is the Part-of-Speech sequence of the current segment  $s_j$ .  $n$  could be a POS sequence like  $JJ-NN-NN$ .

*Ancestors-and-Length* feature:

$$f(s_{j-1}, s_j, x) = \delta(A(s_j) = n) \delta(|n| = k) \delta(y_j = b) \quad (6)$$

where  $A(s_j)$  is the path from the root of the whole parse-tree to the root of the sub-tree that matches the current segment  $s_j$ .  $n$  could be a node sequence such as  $Root-S-VP-NP$ . This feature encodes the length of the path ( $|n| = k$ ) as well.

*Node-and-Word-Before* feature:

$$f(s_{j-1}, s_j, x) = \delta(N(s_j) = t) \delta(w_{j-1} = a) \delta(y_j = b) \quad (7)$$

where  $\delta(w_{j-1} = a)$  indicates that the word preceding the current segment  $s_j$  is  $a$ .

*Node-and-Phrase-Before* feature:

$$f(s_{j-1}, s_j, x) = \delta(N(s_j) = t) \delta(ww_{j-1} = a) \delta(y_j = b) \quad (8)$$

where  $\delta(ww_{j-1} = a)$  indicates that the two words preceding the current segment  $s_j$  is  $a$ . Table 1 shows a list of example features extracted from the query parse tree in Figure 2.

Table 1. Examples of features extracted on the segment “prominent drug usages and aliens” (labeled as “ $<PLOT>$ ”) for the query in Figure 2.

Feature	Example
Node	(NP)
Node-and-Length	(NP, 5)
Node-and-Children	(NP, NP-CC-NNS)
Node-and-POS	(NP, JJ-NN-NNS-CC-NNS)
Ancestors-and-Length	(ROOT-NP-SBAR-S-VP-NP, 6)
Node-and-Word-Before	(NP, “features”)
Node-and-Phrase-Before	(NP, “that features”)

## 2.2.2 Semantic Dependency Features

Another type of information that could be leveraged to improve query tagging performance is the semantic dependencies encoded by the parsing process. For example, Figure 3 shows a dependency-based parse tree for the movie query “show me a funny movie starring Johnny Depp and featuring Caribbean pirates.” As shown in the hierarchical representation, the verb “featuring” takes two arguments, “funny movie” ( $arg1$ ) and “Caribbean pirates” ( $arg2$ ). Thus, a relationship (e.g., “featuring (movie,  $<PLOT>$ )”) could be extracted from the parse tree to capture the long-distance semantic dependency between the object (“ $movie$ ”) and its associated attribute (“ $<PLOT>$ ”).

To make use of such dependency information, we employ a list of features that represent both direct and multi-degree (or chain) relationships between query segments. Similarly, these features will only be fired up when a segment occurs as the argument in a semantic relationship.

#### Solo-Dependency feature:

$$f(s_{j-1}, s_j, x) = \delta(R(w) = r) \delta(arg_r(w) = s_j) \delta(y_j = b) \quad (9)$$

where  $R(w) = r$  denotes that the word  $w$  has a dependency relationship  $r$ , and  $arg_r(w) = s_j$  denotes that one of the arguments of  $w$  in the relationship  $r$  is the current segment  $s_j$ .

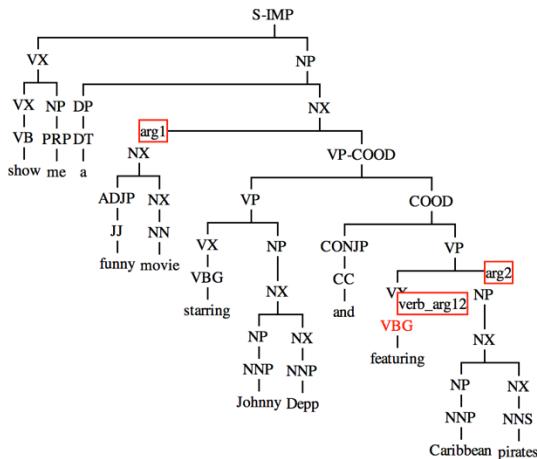


Figure 3: Example of a dependency parse tree for the query “show me a funny movie starring Johnny Depp and featuring Caribbean pirates”.

#### Dual-Dependency feature:

$$f(s_{j-1}, s_j, x) = \delta(R(w) = r) \delta(arg_r(w)_1 = a) \delta(arg_r(w)_2 = s_j) \delta(y_j = b) \quad (10)$$

where  $arg_r(w)_1 = a$  denotes that the head of the first argument of the word  $w$  in the relationship  $r$  is the word  $a$ , and  $arg_r(w)_2 = s_j$  denotes that the second argument is the current segment  $s_j$ .

#### Chain-Dependency:

$$f(s_{j-1}, s_j, x) = \delta(CR(w, s_j) = r) \delta(y_j = b) \quad (11)$$

where  $CR(w, s_j) = r$  denotes that the word  $w$  and the current segment  $s_j$  have a chain dependency relationship  $r$ . A chain dependency could be generated through a list of relationships that share one or more arguments. Table 2 shows some examples of dependency features extracted from the query parse tree in Figure 3.

Table 2. Examples of dependency features extracted for the segment “Caribbean pirates” (labeled as “<PLOT>”) of the query in Figure 3.

Feature	Example
Solo-Dependency	<i>Verb (featuring), arg(&lt;PLOT&gt;)</i>
Dual-Dependency	<i>Verb (featuring), arg1(movie), arg2(&lt;PLOT&gt;)</i>
Chain-Dependency	<i>Show → movie → &lt;PLOT&gt;</i> <i>(a dependency chain was generated from the relationship between “show” and “movie” and that between “movie” and “Caribbean pirates”)</i>

### 3. EXPERIMENTS

To evaluate the proposed approach, we took the movie domain as an example and conducted experiments on real movie queries. To collect natural language queries from real users, we employed the annotation toolkit previously developed in [20], which uses Amazon Mechanical Turk (AMT) [21] as the crowd-sourcing platform. In order to collect semantically complex queries, we designed the query generation HIT (Human Intelligent Task) as a verbal Charade game: in each HIT, the worker was asked to make up a query for searching a selected movie, without mentioning the movie title (as demonstrated in Figure 4).

Instructions	
Please help us write questions for a movie trivia quiz. In each of the 4 questions below:	
1. Choose 1 movie from the 5 given alternatives. You will ask a question about the chosen movie, so choose the one you know the best.	
2. Imagine that you forgot the name of the movie. Ask a question by describing the movie so that someone can remind you what the name was. The question must be at least 15 words long and cannot contain the name of the movie. See the examples below for some ideas.	
Question 1/4	
<input type="radio"/> Made In Heaven <input type="radio"/> A Star is Born <input type="radio"/> Amelie <input type="radio"/> Sunset Boulevard <input checked="" type="radio"/> Scrooge	What movie stars Bill Murray as a cynically selfish TV executive gets haunted by three spirits bearing lessons on Christmas Eve? word count: 21

Figure 4. Screenshot of the movie query generation HIT on AMT. For each question, workers could select one (out of five) movie title that they are familiar with, based on which they will type in a search query without mentioning the movie title.

From the Charade HIT, we collected a total of 9,768 movie queries. Table 3 shows some examples from the query collection. To obtain semantic labels for the collected queries, we put up another labeling HIT, where workers were asked to segment each query and assign semantic classes to the query segments. The class labels include Award, Title, Opinion, Year, Origin, Genre, Director, Plot, Quote, Actor, Soundtrack, Character, and Other. Different from the labeling HITs deployed in the previous study [20] where most segments were short phrases, in this task

workers tend to select long constituents as segments, such as a movie's origin, viewers' opinions, and plot descriptions.

Table 3. Examples of movie queries collected from AMT.

Movie	Query
On the Waterfront	<i>In what film did Marlon Brando play a New York stevedore protesting that he could have been a contender?</i>
Date Night	<i>This movie stars Steve Carell and Tina Fey and a case of mistaken identity turns a bored married couple's attempt at a romantic evening into something thrilling and dangerous.</i>
Get Him to the Greek	<i>What is the name of the 2010 movie in which a record company intern is hired to accompany out-of-control British rock star Aldous Snow to a concert at L.A.'s Greek Theater?</i>
North by Northwest	<i>In what Alfred Hitchcock film does Cary Grant flee from a bi-plane that swoops down on him in a field?</i>

We randomly divided the annotated queries into training (80%) and test (20%) sets. For linguistic feature extraction, we employed two domain-independent parsers, Stanford parser [22] and Enju parser [23]. Hierarchical syntactic features were extracted from the parse trees generated by both parsers. And for relational features, we used the parsing results from the Enju parser, which has a detailed representation of semantic dependencies, such as encoding a sub-tree (instead of a single word) as the argument in a relationship. We took semi-Markov CRFs [19] as the semantic tagging model, and compared the proposed approach with a baseline using  $n$ -grams.

As shown in Table 4, both hierarchical and dependency features achieved an improvement over the baseline. Hierarchical syntactic features from the two parsers obtained an F1-score of 86.28% and 86.16%, respectively. And the combination of dependency features achieved an F1-score of 86.40%, outperforming the baseline of 85.32%.

Table 4. Semantic tagging results with linguistic parsing features. Hierarchical I and II represent the structural features extracted from the Stanford parser and Enju parser, respectively; and All Dependency represents the combination of Solo, Dual and Chain dependency features.

	Recall	Precision	F1-score
<b>Baseline</b>	86.59	84.08	85.32
<b>Hierarchical I</b>	87.53	85.06	86.28
<b>Hierarchical II</b>	87.20	85.14	86.16
<b>Solo Dependency</b>	87.02	85.19	86.09
<b>Dual Dependency</b>	87.42	85.20	86.30
<b>Chain Dependency</b>	87.10	84.80	85.93
<b>All Dependency</b>	87.43	85.39	86.40

To evaluate the proposed approach in another domain, we also conducted the experiments on a set of 9,991 restaurant queries, which was collected in a previous study [20] and contained complex query constituents (e.g., *Amenity*, *Opinion*). The queries were randomly divided into training (80%) and test (20%) sets as well. As shown in Table 5, on the test set, the hierarchical features from the two parsers obtained an F1-score of 84.58% and 84.50%, and the dependency features achieved an F1-score of 84.60%; both outperformed the baseline (82.87%).

Table 5. Semantic tagging results on restaurant queries.

	Recall	Precision	F1-score
<b>Baseline</b>	78.77	87.43	82.87
<b>Hierarchical I</b>	83.32	85.87	84.58
<b>Hierarchical II</b>	83.72	85.28	84.50
<b>All Dependency</b>	83.90	85.31	84.60

## 4. CONCLUSIONS

In this work, we present a query understanding approach that leverages linguistic parsing knowledge by extracting hierarchical syntactic features and semantic dependency features from domain-independent parse trees. Structural information such as sub-clauses and semantic relationships such as chain dependencies are incorporated into inference model learning. Experiments on real natural language queries in multiple domains show that augmenting semantic tagging models with linguistic parsing features can improve understanding performance on semantically complex queries.

For future work, we will explore the direction of hierarchical query understanding beyond one-pass semantic tagging. We plan to deploy the augmented models for query understanding in real systems, and to collect spoken/textual queries from real system users for further evaluation.

## 5. ACKNOWLEDGEMENTS

This research is supported by Quanta Computers, Inc. through the Qmulus project. Thanks to Stephanie Seneff and Victor Zue for helpful discussions.

## 6. REFERENCES

- [1] X. Li. Understanding the Semantic Structure of Noun Phrase Queries. In Proc. of ACL, 2010.
- [2] D. Hakkani-Tr and G. Tur. Tutorial on Spoken Language Understanding. In Proc. of ICASSP, 2007.
- [3] Y. Wang, L. Deng, and A. Acero. Spoken Language Understanding — An Introduction to the Statistical Framework. In IEEE Signal Processing Magazine, 2005.
- [4] S. Hahn, P. Lehnen, C. Raymond, and H. Ney. A Comparison of Various Methods for Concept Tagging for Spoken Language Understanding. In Proc. of LREC, 2008.

- [5] M. Dinarelli, A. Moschitti, and G. Riccardi. Re-Ranking Models For Spoken Language Understanding. In Proc. of EACL, 2009.
- [6] J. Lafferty, A. McCallum, and F. Pereira. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In Proc. of ICML, 2001.
- [7] M. Pasca and B. Van Durme. Weakly-supervised Acquisition of Open-domain Classes and Class Attributes from Web Documents and Query Logs. In Proc. of ACL-HLT, 2008.
- [8] Y. Wang, R. Hoffmann, X. Li, and J. Szymanski. Semi-Supervised Learning of Semantic Classes for Query Understanding – from the Web and for the Web. In Proc. of CIKM, 2009.
- [9] J. Liu, X. Li, A. Acero, and Y. Wang. Lexicon Modeling for Query Understanding. In Proc. of ICASSP, 2011.
- [10] A. Celikyilmaz and D. Hakkani-Tur. A Joint Model for Discovery of Aspects in Utterances. In Proc. of ACL, 2012.
- [11] M. Jeong and G. G. Lee. Triangular-chain Conditional Random Fields. IEEE Transactions on Audio, Speech and Language Processing (IEEE-TASLP), 2008.
- [12] Y-Y. Wang. Strategies for Statistical Spoken Language Understanding with Small Amount of Data – An Empirical Study. In Proc. of Interspeech, 2010.
- [13] J. Liu, S. Cyphers, P. Pasupat, I. McGraw, and J. Glass. A Conversational Movie Search System Based on Conditional Random Fields. In Proc. of Interspeech, 2012.
- [14] J. R. Finkel and C. D. Manning. Joint Parsing and Named Entity Recognition. In Proc. of NAACL, 2009.
- [15] J. R. Finkel and C. D. Manning. Hierarchical Joint Learning: Improving Joint Parsing and Named Entity Recognition with Non-Jointly Labeled Data. In Proc. of ACL, 2010.
- [16] G. Druck, G. Mann, and A. McCallum. Semi-supervised Learning of Dependency Parsers using Generalized Expectation Criteria. In Proc. of ACL, 2009.
- [17] V. Punyakanok, D. Roth, and W. Yih. The Importance of Syntactic Parsing and Inference in Semantic Role Labeling. Computational Linguistics, Vol. 34, No. 2, Pages 257-287, 2008.
- [18] M. Manshadi and X. Li. Semantic Tagging of Web Search Queries. In Proc. of ACL, 2009.
- [19] S. Sarawagi and W. W. Cohen. Semi-Markov Conditional Random Fields for Information Extraction. In Advances in Neural Information Processing Systems (NIPS), 2004.
- [20] J. Liu, P. Pasupat, S. Cyphers, and J. Glass. ASGARD: A Portable Architecture for Multilingual Dialogue Systems. In Proc. of ICASSP, 2013.
- [21] <https://www.mturk.com/mturk/welcome>
- [22] D. Klein and C. D. Manning. Accurate Unlexicalized Parsing. In Proc. of ACL, 2003.
- [23] Y. Miyao, R. Saetre, K. Sagae, T. Matsuzaki, and J. Tsujii. Task-Oriented Evaluation of Syntactic Parsers and Their Representations. In Proc. of ACL-HLT, 2008.