BLCUNLP: Corpus Pattern Analysis for Verbs Based on Dependency Chain

Yukun Feng, Qiao Deng and **Dong Yu**†
College of Information Science, Beijing Language and Culture University
No.15 Xueyuan Rd., Beijing, China, 100083. †The corresponding author.
{fengyukun,dengqiao,yudong}@blcu.edu.cn

Abstract

We implemented a syntactic and semantic tagging system for SemEval 2015 Task 15: Corpus Pattern Analysis. For syntactic tagging, we present a Dependency Chain Search Algorithm that is found to be effective at identifying structurally distant subjects and objects. Other syntactic labels are identified using rules defined over dependency parse structures and the output of a verb classification module. Semantic tagging is performed using a simple lexical mapping table combined with post-processing rules written over phrase structure constituent types and named entity information. The final score of our system is 0.530 F1, ranking second in this task.

1 Introduction

Corpus Pattern Analysis (CPA) is an important language analysis technique, which attempts to describe the patterns of word usage in text. In this paper, we present the system we developed for SemEval-2015 Task 15: CPA, Subtask1: CPA parsing. The system operates in two stages: syntactic tagging and semantic tagging. We first search for the syntactic roles of a verb's arguments in a sentence. We use the following tag set for the syntactic roles: "subj" is for subject, "obj" is for object, "iobj" is for indirect object, "advprep" is for adverbial preposition or other adverbial/verbal link, "acomp" is for adverbial or verb complement, and "scomp" is for noun or adjective complement. For example, take a sentence whose core verb is "plan": "Mr Eigen plans to wage his war diplomatically". The correct tagging of syntactic and semantic roles is: Mr [**subj/Human** Eigen] plans [**advprep/LexicalItem** to] [**acomp/Activity** wage] his war diplomatically.

Due to time constraints, we put more effort into improving the accuracy of syntactic tagging. We rely on simpler techniques for semantic tagging. For syntactic tagging, we use Stanford CoreNLP to extract linguistic attributes, deduce dependency chains through dependency relations and to classify verbs. When performing semantic tagging, we use a data driven mapping of words to their most frequent semantic tag in the task's training data in conjunction with a small number of post-processing rules.

2 Our Methods

2.1 System Framework

Our system consists of five modules (Figure 1). The first module is Preprocessing, which generates input files with the correct format for Stanford CoreNLP to extract linguistic attributes.

The second module is Linguistic Attributes. For the syntactic layer, tagged arguments must have direct or indirect dependency relations with the core verb. Dependency relations are thus a critical attribute for correctly selecting tagged units and types. We employ a number of additional linguistic attributes for our tagging rules: parts of speech (POS) provide useful information for syntactic tagging; direct dependency relations and phrase type are helpful in identifying and following a dependency chain. Last, named entity (NE) tags and phrase-structure constituent types contribute to semantic tagging. In general, we extract four categories of attributes from sentences: dependency relations, POS tags, phrase-structure parse, and NE.

The third module is Verb Classification. Even when a verb's dependency relations with related

Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015), pages 325–328,
Denver, Colorado, June 4-5, 2015. ©2015 Association for Computational Linguistics

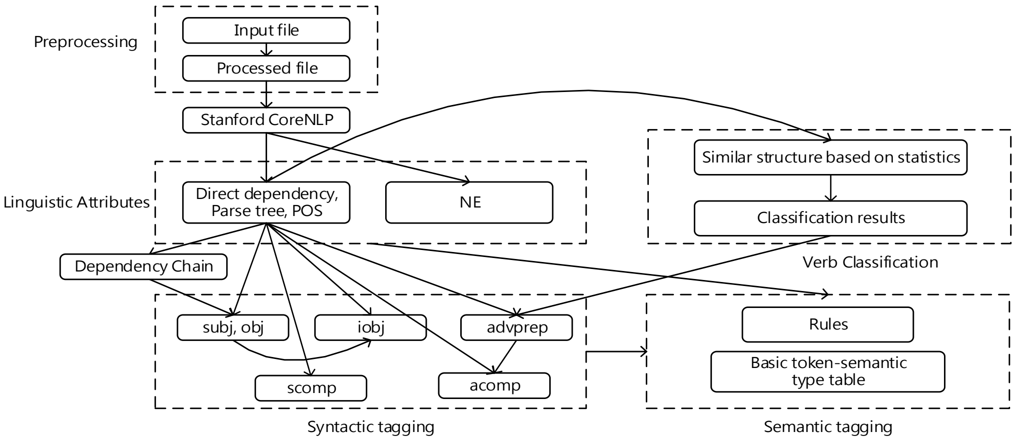


Figure 1. The system has five modules: Preprocessing, Linguistic Attributes, Verb classification, Syntactic tagging, and Semantic tagging. First, it preprocesses input files and extracts 4 attributes: direct dependency, parse tree, POS, and NE. Second, it uses the first three attributes for syntactic tagging, during which indirect dependencies are deduced for "subj" and "obj" relations, and verbs are classified as candidates for "advprep" tagging. Last, our system uses all four attributes and some post-processing rules to do semantic tagging.

prepositions are the same, we find that different verbs have varying degrees of preference for an "advprep" argument. For example, both "abandon" and "account" can be followed by "for", yet only "account" is tagged as "advprep". According to corpus statistics, "account" frequently co-occurs with prepositions. The Verb Classification module is designed to decide whether a verb is strongly related to prepositions, allowing the use of this information in our tagging rules.

The fourth module is the Syntactic Tagging. This module assigns syntactic tags using a set of rules that operate over the annotations provided by the Linguistic Attributes module. When tagging "subj" and "obj" with basic dependency relations, we observed that many of the tagged arguments have no direct dependency relation with the core verb. We handle these arguments by performing a heuristic search for the subj or obj of the nearest ancestor having the missing relation. We find that this is an effective approach.

The last module is Semantic Tagging. The training data provides us with plenty of semantically tagged words, and most of the tagged words have only one corresponding semantic type. We construct a word to semantic tag mapping heuristic based on the most frequent tag for each word in the training set. Semantic tags are related to certain NE tags and phrase-structure constituent types. For instance, person name is normally tagged as "Human", and a place is often tagged as "Location". To capture this, we augment our mapping table with a small number of semantic tagging rules.

2.2 Linguistic Attribute Extraction

We use the Stanford CoreNLP toolkit to get word-to-word dependency relations, phrase-structure parse trees, POS, and NE attributes. Our system rewrites some of the syntactic tags. For example, the CoreNLP tag "nsubj" is replaced by "subj" in train data. Table 1 shows the aggregation of all of the linguistic attributes used by the tagging modules in our system.

Attributes	Description
dependent ID	Sequence number in dependency tree
dependent	Dependent token
phrase type	Phrase type
POS	Part of speech
NE	Named entity type
governor-dependent type	Dependency relation
governor	Governor token
governor ID	Sequence number of the governor

Table 1. Attributes used for syntactic and semantic tagging.

2.3 Verb Classification

Before tagging, we divide verbs into two categories according to the relationship between the verb and its related prepositions, which leads to better "advprep" tagging. Our system gathers corpus statistics that cue the affinity of each verb for the advprep relation. Specifically, we compute how often the verb takes a direct prepositional argument and how often the direct prepositional argument is adjacent to the verb:

$$P(\text{DirectPrep} \mid V) = \frac{\text{cnt}(\text{DirectPrep and } V)}{\text{cnt}(V)}$$

$$P(\text{Adjacent}|\text{DirectPrep}, V) = \frac{\text{cnt}(\text{Adjacent,DirectPrep and } V)}{\text{cnt}(\text{DirectPrep and } V)}$$

Here, cnt(V) is the total number of sentences that contain the verb V, cnt(DirectPrep and V) is the number of sentences where the verb V has a direct prepositional argument, and cnt(Adjacent, DirectPrep and V) counts sentences where the verb not only has a direct dependency relation but is also directly adjacent to the preposition. Take the verb “account” as an example, according to our statistics, P(DirectPrep|V) of “account” is 0.9241, and P(Adjacent|DirectPrep, V) is 0.8425. Therefore, we can tell that “account” is strongly related to prepositions. Through considerable experiments, we set up two threshold values to decide whether one verb is related to certain prepositions. When P(DirectPrep|V)>=0.45 and P(Adjacent|DirectPrep, V)>=0.5, the current verb is considered to be related to prepositions.

2.4 Syntactic Tagging

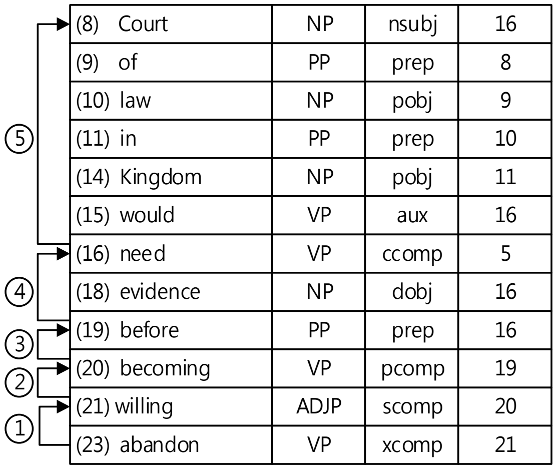
2.4.1 subj and obj

For both subj and obj tagging, we first check whether the verb has any direct subj and obj dependencies. When such dependencies exist, we use them directly to assign the subj or obj tag. If a subj or obj is not contained in the direct dependency relations, we carry out our Dependency Chain Search Algorithm to attempt to find and tag a near-by possibly related subj or obj. Figure 2 illustrates this algorithm for subj relations.

```
goverWordID = GetGoverWordID(verbID);
for goverWordID != TREE_ROOT_NODE
   POS = GetPOSofID(goverWordID);
   if POS == "VP"
         subjID = GetDirectSubjID(goverWordID);
         if subjID != "null"
               Tagging(subjID, "subj");
          break;
   goverWordID = GetGoverWordID(goverWordID);
```

Figure 2. Dependency Chain Search Algorithm.

Figure 3 illustrates the operation of this algorithm. The first column of the table is the dependent word with its id, the second is POS, the third is dependency relation, and the forth is govern id.



(8) Court	NP	nsubj	16
(9) of	PP	prep	8
(10) law	NP	pobj	9
(11) in	PP	prep	10
(14) Kingdom	NP	pobj	11
(15) would	VP	aux	16
(16) need	VP	ccomp	5
(18) evidence	NP	dobj	16
(19) before	PP	prep	16
(20) becoming	VP	pcomp	19
(21) willing	ADJP	scomp	20
(23) abandon	VP	xcomp	21

Figure 3. An example of the Dependency Chain Search algorithm at work. The algorithm traverses five dependency relations to find that “court” is the subject of “abandon”.

2.4.2 iobj

For tokens whose indirect dependency relation with the verb is “iobj”, we tag it directly. To increase coverage, we build a table which contains common double object verbs. If the core verb belongs to this table, we replace the original tag “obj” with “iobj”.

2.4.3 advprep

As for prepositions which have direct dependency relations with core verbs and their POS are “PP”, we check the category of the verb generated by the Verb Classification module. We produce the "advprep" tag only if the verb is heuristically identified as a good candidate for this relation, otherwise we abandon tagging.

2.4.4 acomp

As for tokens whose dependency type with verb is ccomp” or “xcomp”, and if its POS is “VP” or its governor’s POS is “VP”, we tag it with “acomp”. For tokens whose tag is “advprep”, we search downward for a near-by word whose dependency type is “pobj” and then tag it with “acomp”.

2.4.5 scomp

When a token has the dependency type “acomp” within the dependency relations produced by Stanford CoreNLP, it is tagged with “scomp”.

2.5 Semantic Tagging

We extract words and their semantic types from the SemEval2015 training data, and populate a word-to-semantic-type mapping table with the most frequent semantic type for each word. We then apply the following semantic tagging rules:

1) If the phrase type of the current token is "WHNP", we tag it as "Anything", or if the token itself is "who", "whom", then we tag it as "Human".
2) If the phrase type of the current token is "SBAR" or "WHADVP", then we tag its semantic type as "LexicalItem".
3) If the NE type of the current token is "NUMBER", we tag it as "Numerical Value".
4) If the NE type of the current token is "PERSON", we tag it as "Human".
5) Else, we tag it according to the word-to-semantic-type mapping table.

3 Evaluation Results

Our syntactic and semantic tagging results from the official evaluation are shown in Table 2. During the official evaluation, we failed to upload the "undertake" file, which lead to a comparatively lower score on this task.

Verbs	syntactic tagging			semantic tagging		
	P	R	F	P	R	F
operate	.462	.635	.535	.348	.278	.309
apprehend	.749	.634	.687	.669	.403	.503
appreciate	.795	.735	.764	.718	.489	.581
continue	.857	.776	.814	.701	.495	.580
crush	.788	.679	.729	.561	.296	.388
decline	.862	.862	.862	.660	.474	.552
undertake	.000	.000	.000	.000	.000	.000

Table 2. Syntactic and semantic tagging results.

The final overall F-score of our system is 0.53, ranking second on the task, with the baseline system achieving 0.624. This F-score is calculated by averaging the F-scores achieved on syntactic and semantic tagging. On the evaluation data, if we ignore the "undertake" file that we failed to upload, the average F-score of syntactic tagging increases to 0.732, and the combined overall score increases to 0.619. Similar to our work, the baseline methods are also rule based, but we observe that our rules underperform the baseline. We believe this is because we used a simpler rule set that we spent less time refining for the semantic task.

4 Conclusions

In this paper, we propose simple but reliable techniques for syntactic and semantic tagging. These techniques were shown to perform well within SemEval 2015 Task 15: Corpus Pattern Analysis. We find that an effective way to accomplish "subj" and "obj" syntactic tagging is to utilize our simple Dependency Chain Search algorithm. We also incorporated verb classification using simple rules based on corpus statistics to increase syntactic tagging accuracy.

Acknowledgments

We would like to thank the anonymous reviewers for their helpful suggestions and comments. The research work is funded by the Natural Science Foundation of China (No.61300081, 61170162), and the Fundamental Research Funds for the Central Universities in BLCU (No. 15YJ030006).

References

Patrick Hanks. 2004. *Corpus Pattern Analysis.* In EURAALEX Proceedings. Vol. I, pp. 87-98. Lorient, France: Université de Bretagne-Sud.

Marie-Catherine de Marneffe and Christopher D. *Manning, Stanford typed dependencies manual.* 2008.

Bradbury, Jane and El Maarouf, Ismail. 2013. *An empirical classification of verbs based on Semantic Types: the case of the 'poison' verbs"* . In Proceedings of JSSP.

Buchholz, Sabine and Marsi, Erwin. 2006. *CoNLL-X shared task on multilingual dependency parsing.* In Proceedings of CoNLL, New York.

Carreras, Xavier and Marquez, Lluis. 2004. *Introduction to the CoNLL-2004 shared task: Semantic role labeling.* In Proceedings of CoNLL, Boston.

Hanks, Patrick, and Pustejovsky, James. 2005. *A Pattern Dictionary for Natural Language Processing* . In Revue Française de linguistique appliquée, 10:2.

El Maarouf, Ismail and Baisa, Vít. 2013. *Automatic classification of semantic patterns from the Pattern Dictionary of English Verbs.* In Proceedings of JSSP.

Popescu, Octavian. 2012. *Building a Resource of Patterns Using Semantic Types.* In Proceedings of LREC, Istanbul.