

# Semantic Dependency Parsing using N-best Semantic Role Sequences and Roleset Information

Joo-Young Lee, Han-Cheol Cho, and Hae-Chang Rim

Natural Language Processing Lab.

Korea University

Seoul, South Korea

{jylee, hccho, rim}@nlp.korea.ac.kr

## Abstract

In this paper, we describe a syntactic and semantic dependency parsing system submitted to the shared task of CoNLL 2008. The proposed system consists of five modules: syntactic dependency parser, predicate identifier, local semantic role labeler, global role sequence candidate generator, and role sequence selector. The syntactic dependency parser is based on Malt Parser and the sequence candidate generator is based on CKY style algorithm. The remaining three modules are implemented by using maximum entropy classifiers. The proposed system achieves 76.90 of labeled F1 for the overall task, 84.82 of labeled attachment, and 68.71 of labeled F1 on the WSJ+Brown test set.

## 1 Introduction

In the framework of the CoNLL08 shared task (Surdeanu et al., 2008), a system takes POS tagged sentences as input and produces sentences parsed for syntactic and semantic dependencies as output. A syntactic dependency is represented by an ID of head word and a dependency relation between the head word and its modifier in a sentence. A Semantic dependency is represented by predicate rolesets and semantic arguments for each predicate.

The task combines two sub-tasks: syntactic dependency parsing and semantic role labeling. Among the sub-tasks, we mainly focus on the semantic role labeling task. Compared to previous

CoNLL 2004 and 2005 shared tasks (Carreras and Màrquez, 2004; Carreras and Màrquez, 2005) and other semantic role labeling research, major differences of our semantic role labeling task are 1) considering nominal predicates and 2) identifying roleset of predicates. Based on our observation that verbal predicate and nominal predicate have different characteristics, we decide to build different classification models for each predicate types. The models use same features but, their statistical parameters are different. In this paper, maximum entropy<sup>1</sup> is used as the classification model, but any other classification models such as Naive Bayes, SVM, etc. also can be used. To identify roleset, we investigate a roleset match scoring method which evaluate how likely a roleset is matched with the given predicate.

## 2 System Description

The proposed system sequentially performs syntactic dependency parsing, predicate identification, local semantic role classification, global sequence generation, and roleset information based selection.

### 2.1 Syntactic Dependency Parsing

In the proposed system, Malt Parser (Nivre et al., 2007) is adopted as the syntactic dependency parser. Although the training and test set of CoNLL08 use non-projective dependency grammar, we decide to use projective parsing algorithm, Nivre arc-standard, and projective/non-projective conversion functions that Malt Parser provides. The reason is that non-projective parsing shows worse performance than projective parsing with conversion in our preliminary experiment.

<sup>1</sup>We use Zhang Le's MaxEnt toolkit, [http://homepages.inf.ed.ac.uk/s0450736/maxent\\_toolkit.html](http://homepages.inf.ed.ac.uk/s0450736/maxent_toolkit.html)

© 2008. Licensed under the *Creative Commons Attribution-Noncommercial-Share Alike 3.0 Unported* license (<http://creativecommons.org/licenses/by-nc-sa/3.0/>). Some rights reserved.

We projectize the non-projective training sentences in the training set to generate projective sentences. And then, the parser is trained with the transformed sentences. Finally, the parsing result is converted into non-projective structure by using a function of Malt Parser.

## 2.2 Predicate Identification

Unlike previous semantic role labeling task (Carreras and Màrquez, 2004; Carreras and Màrquez, 2005), predicates of sentences are not provided with input in the CoNLL08. It means that a system needs to identify which words in a sentence are predicates.

We limit predicate candidates to the words that exist in the frameset list of Propbank and Nombank. Propbank and Nombank provide lists of about 3,100 verbal predicates and about 4,400 nominal predicates. After dependency parsing, words which are located in the frameset list are selected as predicate candidates. The predicate identifier determines if a candidate is a predicate or not. The identifier is implemented by using two maximum entropy models, the one is for verbal predicates and the other is for nominal predicates. The following features are used for predicate identification:

---

### Common Features For Predicate Identification

---

- Lemma of Previous Word
- Lemma of Current Word
- Lemma of Next Word
- POS of Previous Word
- POS of Current Word
- POS of Next Word
- Dependency Label of Previous Word
- Dependency Label of Current Word
- Dependency Label of Next Word

---

### Additional Features for Verbal Predicate

---

- Lemma + POS of Current Word
- Trigram Lemma of Previous, Current, and Next Word

---

### Additional Features for Nominal Predicate

---

- Lemma of Head of Current Word
  - POS of Head of Current Word
  - Dependency Label of Head of Current Word
- 

Verbal predicate identifier shows 87.91 of F1 and nominal predicate identifier shows 81.58 of F1.

Through a brief error analysis, we found that main bottle neck for verbal predicate is auxiliary verb be and have.

## 2.3 Local Semantic Role Labeling

Prediate identification is followed by argument labeling. For the given predicate, the system first eliminates inappropriate argument candidates. The argument identification uses different strategies for verbs, nouns, and other predicates.

The argument classifier extracts features and labels semantic roles. None is used to indicate that a word is not a semantic argument. The classifier also uses different maximum entropy models for verbs, nouns, and other predicates

### 2.3.1 Argument Candidate Identification

As mentioned by Pradhan et al. (2004), argument identification poses a significant bottleneck to improving performance of Semantic Role Labeling system. We tried an algorithm motivated from Hacioglu (2004) which defined a *tree-structured family membership* of a predicate to identify more probable argument candidates and prune the others. However, we find that it works for verb and other predicate type, but does not work properly for noun predicate type. The main reason is due to the characteristics of arguments of noun predicates. First of all, a noun predicate can be an argument for itself, whereas a verb predicate cannot be. Secondly, dependency relation paths from a noun predicate to its arguments are usually shorter than a verb predicate. Although some dependency relation paths are long, they actually involve non-informative relations like IN, MD, or TO. Finally, major long distance relation paths could be identified by several path patterns acquired from the corpus.

Based on the above analysis, we specify a new argument identification strategy for nominal predicate type. The argument identifier regards a predicate and its nearest neighbors - its parent and children - as argument candidates. However, if the POS tag of a nearest neighbor is IN, MD, or TO, it will be ignored and the next nearest candidates will be used. Moreover, several patterns (three consecutive nouns, adjective and two consecutive nouns, two nouns combined with conjunction, and etc.) are applied to find long distance argument candidates.

### 2.3.2 Argument Classification

For argument classification, various features have been used. Primarily, we tested a set of features suggested by Hacioglu (2004). The voice of the predicate, left and right words, its POS tag for a predicate, and lexical clues for adjunctive arguments also have been tested. Based on the type of predicate (i.e. verb predicate, noun predicate, and other predicate) three classification models are trained by using maximum entropy with the following same features:

---

#### Features for Argument Classification

---

- Dependent Relation Type
  - Family Membership
  - Position
  - Lemma of Head Word
  - POS of Head Word
  - Path
  - POS Pattern of Predicate's Children
  - Relation Pattern of Predicate's Children
  - POS Pattern of Predicate's Siblings
  - Relation Pattern of Predicate's Siblings
  - POS of candidate
  - Lemma of Left Word of Candidate
  - POS of Left Word of Candidate
  - Lemma of Right Word of Candidate
  - POS of Right Word of Candidate
- 

The classifier produces a list of possible semantic roles and its probabilities for each word in the given sentence.

### 2.4 Global Semantic Role Sequence Generation

For local semantic role labeling, we assume that semantic roles of words are independent of each other. Toutanova et al. (2005) and Surdeanu et al. (2007) show that global constraint and optimization are important in semantic role labeling. We use CKY-based dynamic programming strategy, similar to Surdeanu et al. (2007), to verify whether role sequences satisfy global constraint and generate candidates of global semantic role sequences.

In this paper, we just use one constraint: no duplicate arguments are allowed for verbal predicates. For verbal predicates, CKY module builds a list of all kinds of combinations of semantic roles augmented with their probabilities. While building the list of semantic role sequences, it removes the

sequences that violate the global constraint. The output of CKY module is the list of semantic role sequences satisfying the global constraint.

### 2.5 Global Sequence Selection using Roleset Information

Finally, we need to select the most likely semantic role sequence. In addition, we need to identify a roleset for a predicate. We perform these tasks by finding a role sequence and roleset maximizing a score on the following formula:

$$\alpha \cdot c + \beta \cdot rf + \gamma \cdot mc \quad (1)$$

where,  $c$ ,  $rf$ ,  $mc$  are role sequence score, relative frequency of roleset, and matching score with roleset respectively.  $\alpha, \beta, \gamma$  are tuning parameters of each factor and decided empirically by using development set. In this paper, we set  $\alpha, \beta, \gamma$  to 0.5, 0.3, 0.2, respectively.

The role sequence score is calculated in the global semantic role sequence generation explained in Section 2.4. The relative frequency of a roleset means how many times the roleset occurred in the training set compared to the total occurrence of the predicate. It can be easily estimated by MLE.

The remaining problem is how to calculate the matching score. We use maximum entropy models as binary classifiers which output *match* and *not-match* and use probability of *match* as matching score. The features used for the roleset matching classifiers are based on following intuitions:

- If core roles (e.g., A0, A2, etc) defined in a roleset occur in a given role sequence, it seems to be the right roleset for the role sequence.
- If matched core roles are close to or have dependency relations with a predicate, it seems to be the right roleset.
- If a roleset has a particle and the predicate of a sentence also has that particle, it seems to be the right roleset. For example, the lemma of `predicate` node for the roleset `cut.05` in frameset file "cut.xml.gz" is `cut.back`, so the particle of `cut.05` is `back`. If the predicate of a sentence also has particle 'back', it seems to be the right roleset.
- If `example` node of a roleset in frameset file has a functional word for certain core role that

also exists in a given sentence, it seems to be the right roleset. For example, example node is defined as follows<sup>2</sup>:

```
<roleset id="cut.09" ...>
  <example>
    <text>
      As the building's new owner,
      Chase will have its work cut
      out for it.
    </text>
    <arg n="1">its work</arg>
    <rel>cut out</rel>
    <arg n="2" f="for">it</arg>
  </example>
</roleset>
```

Here, semantic role A2 has functional word *for*. If a given role sequence has A2 and its word is 'for', then this role sequence probably matches that roleset.

Based on these intuitions, we use following features for roleset matching:

- **Core Role Matching Count** The number of core roles exist in both roleset definition and given role sequence
- **Distance of Matched Core Role** Distance between predicate and core role which exists in both roleset and given role sequence. We use number of word and dependency path length as a distance
- **Indication for Same Particle** It becomes *yes* if given predicate and roleset have same particle. (otherwise *no*)
- **Indication for Same Functional Word** It becomes *yes* if one of core argument is same to the functional word of roleset. (otherwise *no*)

To train the roleset match classifiers, we extract semantic role sequence and its roleset from training data as a positive example. And then, we generate negative examples by changing its roleset to other roleset of that predicate. For example, the above sentence in `<text>` node<sup>3</sup> becomes a positive example for `cut.09` and negative examples for other roleset such as `cut.01`, `cut.02`, etc.

<sup>2</sup>Some nodes are omitted to simplify the definition of example.

<sup>3</sup>Of cause, we assume that this sentence exist in training corpus. So, we will extract it from corpus, not from frameset file.

	WSJ+Brown	WSJ	Brown
LM	76.90	77.96	68.34
LA	84.82	85.69	77.83
LF	68.71	69.95	58.63

Table 1: System performance. LM, LA, LF means macro labeled F1 for the overall task, labeled attachment for syntactic dependencies, and labeled F1 for semantic dependencies, respectively

Labeled Prec.	Labeled Rec.	Labeled F1
88.68	73.89	80.28

Table 2: Performance of Local Semantic Role Labeler on WSJ test set. Gold parsing result, correct predicates, and correct rolesets are used.

### 3 Experimental Result

We have tested our system with the test set and obtained official results as shown in Table 1. We have also experimented on each module and obtained promising results.

We have tried to find the upper bound of the local semantic role labeling module. Table 2 shows the performance when gold syntactic parsing result, correct predicates, and correct rolesets are given. Comparing to phrase structure parser based semantic role labelings such as Pradhan et al. (2005) and Toutanova et al. (2005), our local semantic role labeler needs to enhance the performance. We will try to add some lexical features or chunk features in future works.

Next, we have analyzed the effect of roleset based selector. Table 3 shows the effect of matching score and relative frequency which are the weighted factor of selection described in section 2.5. Here, baseline means that it selects a role sequence which has the highest score in CKY module and roleset is chosen randomly. The results show that roleset matching score and relative frequency of roleset are effective to choose the correct role sequence and identify roleset.

### 4 Conclusion

In this paper, we have described a syntactic and semantic dependency parsing system with five different modules. Each module is developed with maximum entropy classifiers based on different predicate types. In particular, dependency relation compression method and extracted path patterns are used to improve the performance in the argu-

	Prec.	Rec.	F1
Baseline ( <i>c</i> )	69.34	58.42	63.41
+ <i>mc</i>	71.40	60.20	65.32
+ <i>rf</i>	75.94	63.98	69.45
+ <i>mc, rf</i>	76.46	64.45	69.95

Table 3: Semantic scores of global sequence selection in WSJ test set. *mc, rf* means matching score and relative frequency, respectively

ment candidate identification. The roleset matching method is devised to select the most appropriate role sequence and to identify the correct roleset.

However, the current features for roleset matching seem to be not enough and other useful features are expected to be found in the future work. There is also a room for improving the method to integrate the role sequence score, matching score, and the relative frequency.

## References

- Joakim Nivre, Jens Nilsson, Johan Hall, Atanas Chanev, Gülsen Eryigit, Sandra Kübler, Svetoslav Marinov, Erwin Marsi. 2007. MaltParser: A Language-Independent System for Data-Driven Dependency Parsing. *Natural Language Engineering*, 13(2):95–135.
- Kadri Hacioglu. 2008. Semantic role labeling using dependency trees. In *COLING '04: Proceedings of Proceedings of the 20th international conference on Computational Linguistics*. Morristown, NJ, USA.
- Kristina Toutanova, Aria Haghighi, and Christopher D. Manning. 2005. Joint learning improves semantic role labeling. In *ACL '05: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*. Morristown, NJ, USA.
- Mihai Surdeanu and Richard Johansson and Adam Meyers and Lluís Màrquez and Joakim Nivre. 2008. The CoNLL-2008 Shared Task on Joint Parsing of Syntactic and Semantic Dependencies. In *Proceedings of the 12th Conference on Computational Natural Language Learning (CoNLL-2008)*.
- Mihai Surdeanu, Lluís Màrquez, Xavier Carreras, and Pere Comas. 2007. Combination Strategies for Semantic Role Labeling. *The Journal of Artificial Intelligence Research*, 29:105–151.
- Sameer Pradhan, Kadri Hacioglu, Valerie Krugler, Wayne Ward, James H. Martin, and Daniel Jurafsky. 2005. Support Vector Learning for Semantic Argument Classification. *Machine Learning*. 60:11–39.
- Sameer Pradhan, Wayne Ward, Kadri Hacioglu, James Martin, and Dan Jurafsky. 2004. Shallow Semantic Parsing Using Support Vector Machines. In *Proceedings of the Human Language Technology Conference/North American chapter of the Association of Computational Linguistics (HLT/NAACL)*. Boston, MA, USA.
- Xavier Carreras and Lluís Màrquez. 2005. Introduction to the CoNLL-2005 Shared Task: Semantic Role Labeling. In *Proceedings of CoNLL-2005*.
- Xavier Carreras and Lluís Màrquez. 2004. Introduction to the CoNLL-2004 Shared Task: Semantic Role Labeling. In *Proceedings of CoNLL-2004*.