FrameNet-based Semantic Parsing using Maximum Entropy Models

Namhee Kwon
Information Sciences Institute
University of Southern California
4676 Admiralty Way
Marina del Rey, CA 90292
nkwon@isi.edu

Michael Fleischman
Messachusetts Institute of Technology,
77 Massachusetts Ave
Cambridge, MA 02139
mbf@mit.edu

Eduard Hovy
Information Sciences Institute
University of Southern California
4676 Admiralty Way
Marina del Rey, CA 90292
hovy@isi.edu

Abstract

As part of its description of lexico-semantic predicate frames or conceptual structures, the FrameNet project defines a set of semantic roles specific to the core predicate of a sentence. Recently, researchers have tried to automatically produce semantic interpretations of sentences using this information. Building on prior work, we describe a new method to perform such interpretations. We define sentence segmentation first and show how Maximum Entropy re-ranking helps achieve a level of 76.2% F-score (answer among top-five candidates) or 61.5% (correct answer).

1 Introduction

To produce a semantic analysis has long been a goal of Computational Linguistics. To do so, however, requires a representation of the semantics of each predicate. Since each predicate may have a particular collection of semantic roles (agent, theme, etc.) the first priority is to build a collection of predicate senses with their associated role frames. This task is being performed in the FrameNet project based on frame semantics (Fillmore, 1976).

Each frame contains a principal lexical item as the target predicate and associated frame-specific roles, such as offender and buyer, called frame elements. FrameNet I contains 1,462 distinct predicates (927 verbs, 339 nouns, 175 adjectives) in 49,000 annotated sentences with 99,000 annotated frame elements. Given these, it would be interesting to attempt an automatic sentence interpretation.

We build semantic parsing based on FrameNet, treating it as a classification problem. We split the problem into three parts: sentence segmentation, frame element identification for each segment, and semantic role tagging for each frame element. In this paper, we provide a pipeline framework of these three phases, followed by a step of re-ranking from *n*-best lists of every phase for the final output. All classification and re-ranking are performed by Maximum Entropy.

The top-five final outputs provide an F-score of 76.2% for the correct frame element identification and semantic role tagging. The performance of the single best output is 61.5% F-score.

The rest of the paper is organized as follows: we review related work in Section 2, explain Maximum Entropy in Section 3, describe the detailed method in Section 4, show the re-ranking process in Section 5, and conclude in Section 6.

2 Related Work

The first work using FrameNet for semantic parsing was done by Gildea and Jurafsky (G & J, 2002) using conditional probabilistic models. They divide the problem into two sub-tasks: frame element identification and frame element classification. Frame element identification identifies the frame element boundaries in a sentence, and frame element classification classifies each frame element into its appropriate semantic role. The basic assumption is that the frame element (FE) boundaries match the parse constituents, and both identification and classification are then done for each constituent[1].

In addition to the separate two phase model of frame element identification and role classification, they provide an integrated model that exhibits improved performance. They define a frame element group (FEG) as a set of frame element roles present in a particular sentence. By integrating FE identification with role labeling, allowing FEG priors and role labeling decision to affect the determination of next FE identification, they accomplish F-score of 71.9% for FE identification and 62.8% for both of FE identification and role labeling. However, since this integrated approach has an exponential complexity in the number of constituents, they apply a pruning scheme of using only the top *m*

[1] The final output performance measurement is limited to the number of parse constituents matching the frame element boundaries.

hypotheses on the role for each constituent (m = 10).

Fleischman et al.(FKH, 2003) extend G & J's work and achieve better performance in role classification for correct frame element boundaries. Their work improves accuracy from 78.5% to 84.7%. The main reasons for improvement are first the use of Maximum Entropy and second the use of sentence-wide features such as Syntactic patterns and previously identified frame element roles. It is not surprising that there is a dependency between each constituent's role in a sentence and sentence level features reflecting this dependency improve the performance.

In this paper, we extend our previous work (KFH) by adopting sentence level features even for frame element identification.

3 Maximum Entropy

ME models implement the intuition that the best model is the one that is consistent with the set of constraints imposed by the evidence, but otherwise is as uniform as possible (Berger et al. 1996). We model the probability of a class c given a vector of features x according to the ME formulation below:

$$p(c \mid x) = \frac{1}{Z_x} \exp[\sum_{i=0}^{n} \lambda_i f_i(c,x)]$$

Here Z_x is normalization constant, $f_i(c,x)$ is a feature function which maps each class and vector element to a binary value, n is the total number of feature functions, and λ_i is a weight for the feature function. The final classification is just the class with the highest probability given its feature vector and the model.

It is important to note that the feature functions described here are not equivalent to the subset conditional distributions that are used in G & J's model. ME models are log-linear models in which feature functions map specific instances of features and classes to binary values. Thus, ME is not here being used as another way to find weights for an interpolated model. Rather, the ME approach provides an overarching framework in which the full distribution of classes (semantic roles) given features can be modeled.

4 Model

We define the problem into three subsequent processes (see Figure 1): 1) sentence segmentation 2) frame element identification, and 3) semantic role tagging for the identified frame elements. In order to use sentence-wide features for the FE identification, a sentence should have a single non-overlapping constituent sequence instead of all the independent constituents. Sentence segmentation is applied before FE identification for this purpose. For each segment the classification into FE or not is performed in the FE identification phase, and from the FE-tagged constituents the semantic role classification is applied in the role tagging phase.

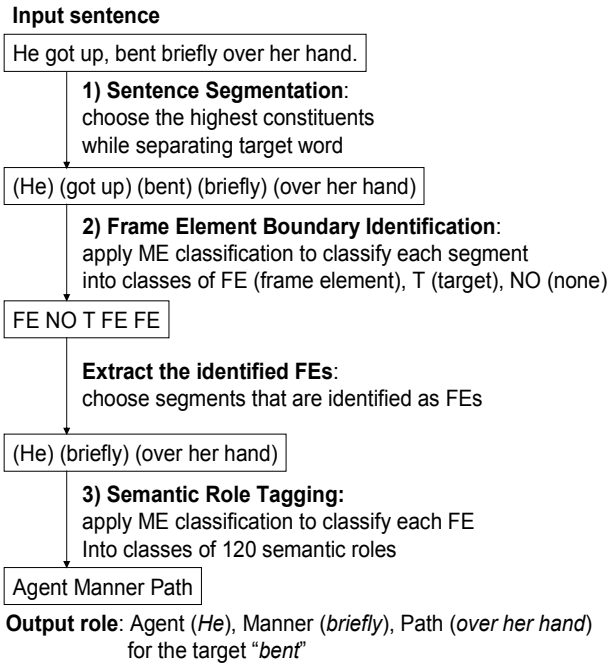

Fig. 1. The sequence of steps on a sample sentence.

4.1 Sentence Segmentation

The advantages of applying sentence segmentation before FE identification are considered in two ways. First we can utilize sentence-wide features, and second the number of constituents as FE candidates is reduced, which reduces the convergence time in training.

We segment a sentence with parse constituents[2]. During training, we split a sentence into true frame elements and the remainder. After choosing frame elements as segments, we choose the highest level constituents in parse tree for other parts, and then make a complete sentence composed of a sequence of constituent segments. During testing, we need to consider all combinations of various level constituents. We know the given target word should be a separate segment because a target word is not a part of other FEs. Since most frame elements tend to be among the higher levels of a parse tree, we decide to use the highest constituents while separating the target word. Figure 2 shows an example of the segmentation for

[2] We use Michael Collins's parser : http://www.cis.upenn.edu/~mcollins/

an actual sentence in FrameNet with the target word "bent".

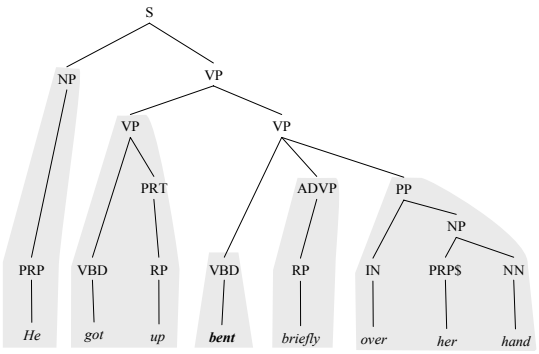

Fig. 2. A sample sentence segmentation: "bent" is a target predicate in a sentence and the shaded constituent represents each segment.

However, this segmentation for testing reduces the FE coverage of constituents, which means our FE classification performance is limited. Table 1 shows the FE coverage and the number of constituents for our development set. The FE coverage of individual constituents (86.36%) means the accuracy of the parser. This limitation and will be discussed in detail in Section 4.4.

Method	Number of constituents	FE coverage (%)
Individual constituents	115,380	86.36
Sentence segmentation	29,688	77.25

Table 1. The number of constituents and FE coverage for development set.

4.2 Frame Element Identification

Frame element identification is executed for the sequence of segments. For the example sentence in Figure 2, "(He) (got up) (bent) (briefly) (over her hand)", there are five segments and each segment has its own feature vector. Maximum Entropy classification into the classes of FE, Target, or None is conducted for each. Since the target predicate is given we don't need to classify a target word into a class, but we do not exclude it from the segments because we want to get benefit of using previous segment's features.

The initial features are adopted from G & J and FKH, and most features are common to both of frame element identification and semantic role classification. The features are:

- **Target predicate (target):** The target predicate, the principal word in a sentence, is the feature that is provided by the user. Although there can be many predicates in a sentence, only one predicate is defined at a time.
- **Target identification (tar):** The target identification is a binary value, indicating whether the given constituent is a target or not. Because we have a target word in a sequence of segments, we provide this information explicitly.
- **Constituent path (path):** From the syntactic parse tree of a sentence, we extract the path from each constituent to the target predicate. The path is represented by the nodes through which one passes while traveling up the tree from the constituent and then down through the governing category to the target word. For example, "over her hand" in a sentence of Figure 2 has a path PP↑VP↓VBD.
- **Phrase Type (pt):** The syntactic phrase type (e.g., NP, PP) of each constituent is also extracted from the parse tree.
- **Syntactic Head (head):** The syntactic head of each constituent is obtained based on Michael Collins's heuristic method[3]. When the head is a proper noun, "proper-noun" substitutes for the real head. The decision if the head is proper noun is done by the part of speech tag in a parse tree.
- **Logical Function (lf):** The logical functions of constituents in a sentence are simplified into three values: *external argument*, *object argument*, *other*. We follow the links in the parse tree from the constituent to the ancestors until we meet either S or VP. If the S is found first, we assign *external argument* to the constituent, and if the VP is found, we assign *object argument.* Otherwise, *other* is assigned. Generally, a grammatical function of *external argument* is a subject, and that of *object argument* is an object. This feature is applied only to constituents whose phrase type is NP.
- **Position (pos):** The position indicates whether a constituent appears before or after the target predicate and whether the constituent has the same parent as the target predicate or not.
- **Voice (voice):** The voice of a sentence (active, passive) is determined by a simple regular expression over the surface form of the sentence.
- **Previous class (c_n):** The class information of the n^{th}-previous constituent (*target*, *frame element*, or *none*) is used to exploit the dependency between constituents. During training, this information is provided by simply

[3] http://www.ai.mit.edu/people/mcollins/papers/heads

looking at the true classes of the frame element occurring *n*-positions before the current element. During testing, hypothesized classes of the *n* elements are used and Viterbi search is performed to find the most probable tag sequence for a sentence.

The combination of these features is used in ME classification as feature sets. The feature sets are optimized by previous work and trial and error experiments. Table 2 shows the lists of feature sets for "briefly" in a sentence of "He got up, bent briefly over her hand". These feature sets contain the previous or next constituent's features, for example, *pt_-1* represents the previous constituent's phrase type and *lf_1* represents the next constituent's logical function.

Feature Set	Example Functions
f(c, target)	*f(c, "bent") = 1*
f(c, target, pt)	*f(c, "bent",ADVP) = 1*
f(c, target, pt, lf)	*f(c, "bent",ADVP,other) = 1*
f(c, pt, pos, voice)	*f(c, ADVP,after_yes,active) = 1*
f(c, pt, lf)	*f(c, ADVP,other) = 1*
f(c, pt_-1, lf_-1)	*f(c, VBD_-1, other_-1) = 1*
f(c, pt_1, lf_1)	*f(c, PP_1, other_1) = 1*
f(c, pt_-1, pos_-1,voice)	*f(c, VBD_-1,t_-1,active) = 1*
f(c, pt_1, pos_1, voice)	*f(c, PP_1,after_yes_1, active) = 1*
f(c, head)	*f(c, "briefly") = 1*
f(c, head, target)	*f(c, "briefly", "bent") = 1*
f(c, path)	*f(c, ADVP↑VP↓VBD) = 1*
f(c, path_-1)	*f(c, VBD_-1) = 1*
f(c, path_1)	*f(c, PP↑VP↓VBD_1) = 1*
f(c, tar)	*f(c, 0) = 1*
f(c, c_-1)	*f(c, "target"_-1) = 1*
f(c, c_-1,c_-2)	*f(c, "target"_-1,"NO FE"_-2) = 1*

Table 2. Feature sets used in ME frame element identification. Example functions of "briefly" from the sample sentence in Fig.2 are shown.

4.3 Semantic Role Classification

The semantic role classification is executed only for the constituents that are classified into FEs in the previous FE identification phase. Maximum Entropy classification is performed to classify each FE into classes of semantic roles.

Most features from the frame element identification in Section 4.2 are still used, and two additional features are applied. The feature sets are in Table 3.

- **Order (order):** The relative position of a frame element in a sentence is given. For example, in the sentence from Figure 2, there are three frame elements, and the element "He" has order 0, while "over her hand" has order 2.
- **Syntactic pattern (pat):** The sentence level syntactic pattern is generated from the parse tree by looking at the phrase type and logical functions of each frame element in a sentence. For example, in the sentence from Figure 2, "He" is an external argument Noun Phrase, "bent" is a target predicate, and "over her hand" is an external argument Prepositional Phrase. Thus, the syntactic pattern associated with the sentence is [NP-ext, target, PP-ext].

Feature Sets	
f(c, target)	*f(r, head)*
f(r, target, pt)	*f(r, head, target)*
f(r, target, pt, lf)	*f(r, head, target, pt)*
f(r, pt, pos, voice)	*f(r, order, syn)*
f(r, pt, pos, voice, target)	*f(r,target, order, syn)*
f(r, r_-1)	*f(r,r_-1,r_-2)*

Table 3. Feature sets used in ME semantic role classification.

4.4 Experiments and Results

Since FrameNet II was published during our research, we continued using FrameNet I (120 semantic role categories). We can, therefore, compare our results with previous research by matching exactly the same data as used in G & J and FKH. We thank Dan Gildea for providing the following data set: training (36,993 sentences / 75,548 frame elements), development (4,000 sentences / 8,167 frame elements), and held our test sets (3,865 sentences / 7,899 frame elements).

We train the ME models using the GIS algorithm (Darroch and Ratcliff, 1972) as implemented in the YASMET ME package (Och, 2002). For testing, we use the YASMET MEtagger (Bender et al. 2003) to perform the Viterbi search for choosing the most probable tag sequence for a sentence using the probabilities from training. Feature weights are smoothed using Gaussian priors with mean 0 (Chen and Rosenfeld, 1999). The standard deviation of this distribution and the number of GIS iterations for training are optimized on development set for each experiment. Table 4 shows the performance for test set. The evaluation is done for individual frame elements.

To segment a sentence before FE identification or role tagging improves the overall performance (from 57.6% to 60.0% in Table 4). Since the segmentation reduces the FE coverage of segments, we conduct the experiment with the manually chosen segmentation to see how much the segmentation helps the performance. Here, we extract segments from the parse tree constituents, so the FE coverage is 86% for test set, which maches the parsing accuracy. Table 5 shows the performance of the frame element identification for

test set: F-score is 77.2% that is much better than 71.7% of our automatic segmentation.

Method	FE identification			FE identification & Role tagging		
	Prec	Rec	F	Prec	Rec	F
G & J separated model	73.6	63.1	67.5	67.0	46.8	55.1
FKH ME model	73.6	67.9	70.6	60.0	55.4	57.6
Our model (segmentation + ME classification)	75.5	68.2	71.7	62.9	56.8	60.0

Table 4. Performance comparison for test set.

Precision	Recall	F-score
82.1	72.9	77.2

Table 5. Result of frame element identification on manual segmentation of test set

5 *n*-best Lists and Re-ranking

As stated, the sentence segmentation improves the performance by using sentence-wide features, but it drops the FE coverage of constituents. In order to determine a good segmentation for a sentence that does not reduce the FE coverage, we perform another experiment by using re-ranking. We obtain all possible segmentations for a given sentence, and conduct frame element identification and semantic role classification for all segmentations. During both phases, we get *n*-best lists with Viterbi search, and finally choose the best output with re-ranking method. Figure 3 shows the overall framework of this task.

5.1 Maximum Entropy Re-ranking

We model the probability of output r given candidates' feature sets $\{x_1 .. x_t\}$ where t is the total number of candidates and x_j is a feature set of the j^{th} candidate according to the following ME formulation:

$$p(r \mid \{x_1..x_t\}) = \frac{1}{Z_x} \exp[\sum_{i=0}^{n} \lambda_i f_i(r, \{x_1..x_t\})]$$

where Z_x is a normalization factor, $f_i(r,\{x_1..x_t\})$ is a feature function which maps each output and all candidates' feature sets to a binary value, n is the total number of feature functions, and λ_i is the weight for a given feature function. The weight λ_i is associated with only each feature function while the weight in the ME classifier is associated with all possible classes as well as feature functions. The final decision is r having the highest probability of $p(r|\{x_1..x_t\})$ from t number of candidates.

As a feature set for each candidate, we use the ME classification probability that is calculated during Viterbi search. These probabilities are conditional probabilities given feature sets and these feature sets depend on the previous output, for example, semantic role tagging is done for the identified FEs in the previous phase. For this reason, the product of these conditional probabilities is used as a feature set.

$$p(r \mid s) = p(seg \mid s) * p(fe \mid seg) * p(r \mid fe)$$

where s is a given sentence, *seg* is a segmentation, *fe* is a frame element identification, and r is the final semantic role tagging. *p(fe|seg)* and *p(r|fe)* are produced from the ME classification but *p(seg|s)* is computed by a heuristic method and a development set optimization experiment. The adopted *p(seg|s)* is composed of *p(each segment's part of speech tag | target's part of speech tag)*, *p(the number of total segments in a sentence | total number of words in a sentence)*, and the average of each segment's *p(head word of FE | target)*.

Two additional feature sets other than *p(r|s)* are applied to get slight improvement for re-ranking performance, which are *average of p(parse tree depth of FE | target)* and *average of p(head word of FE | target)*.

5.2 Experiments and Results

We apply ME re-ranking in YASMET-ME package. We train re-ranking model with development set after obtaining candidate lists for the set. For a simple cross validation, the development set is divided into a sub-training set (3,200 sentences) and a sub-development set (800 sentences) by selecting every fifth sentence. Training for re-ranking is executed with the sub-training set and optimization is done with the sub-development set. The final test is applied to test set.

The possible number of segmentations is different depending on sentences, but the average number of segmentation lists is 15.2[4] for the development set. For these segmentations, we compute *10*-best[5] lists for the FE identification and *10*-best lists for the semantic role classification.

[4] To reduce the number of different segmenations while not dropping the FE coverage, the segmentations having too many segments for a long sentence are excluded.

[5] The experiment showed *10*-best lists outperformed other *n*-best lists where *n* is less than *10*. The bigger number was not tested because of huge number of lists.

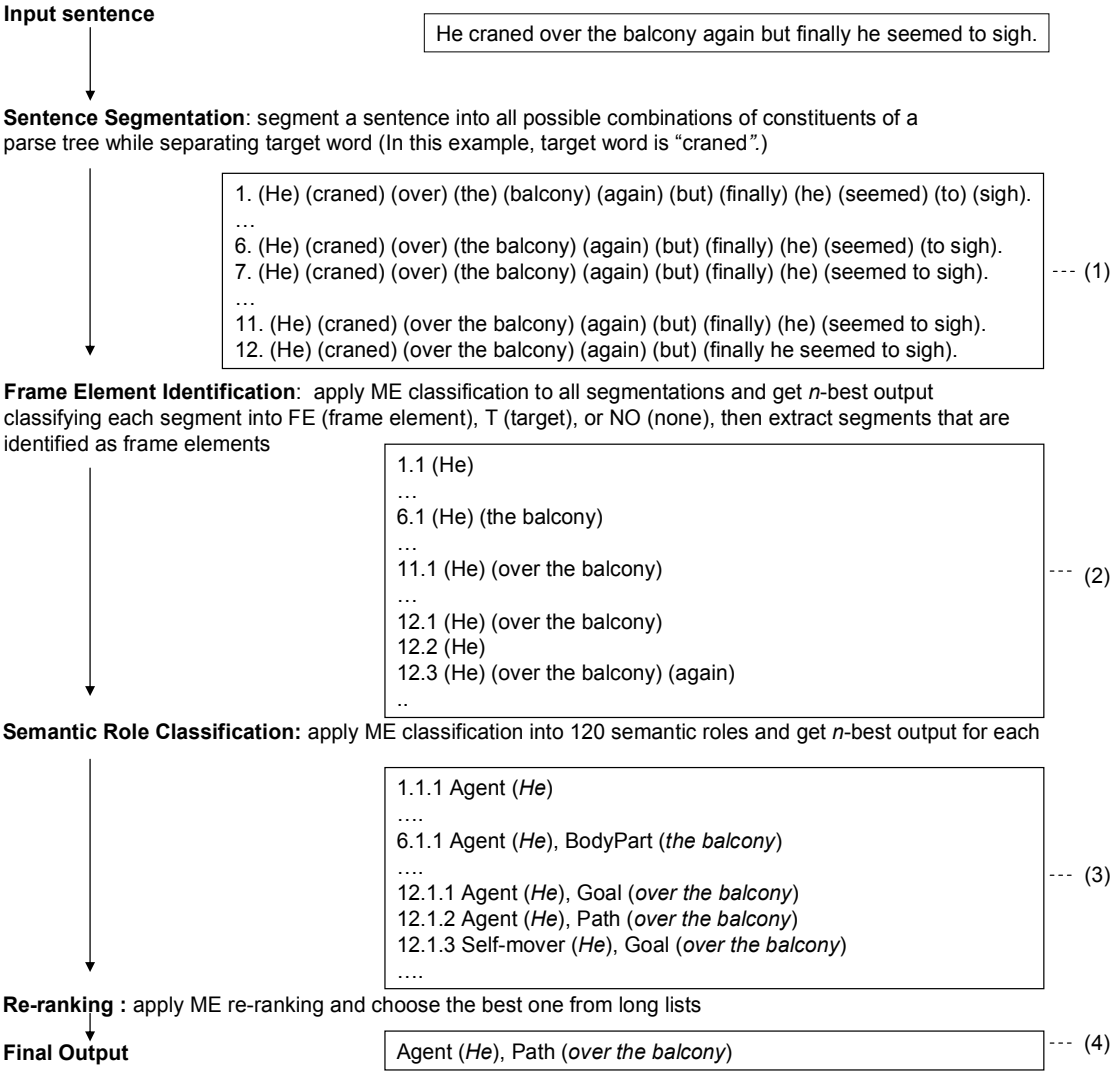

Fig. 3. The framework of the re-ranking method with an actual system output. (1) contains different number of segmentations depending on each sentence, (2) has *mn* number of lists when we obtain *m* possible segmentations in (1) and we get *n*-best FE identifications, (3) has *mnn* number of lists when we get *n*-best role classifications given *mn* lists (4) shows finally chosen output.

Table 6 shows the performance of re-ranking. To evaluate the performance of top-*n*, the best tagging output for a sentence is chosen among *n*-lists and the performance is computed for that list. The top-5 lists show two interesting points: one is that precision is very high, and the other is that F-score including role tagging is not much different from F-score of only FE identification. In other words, there are a few (not 120) confusing roles for a given frame element, and we have many frame elements that are not identified even in *n*-best lists.

Re-rank	FE identification			FE identification & Role tagging		
	Prec	Rec	F	Prec	Rec	F
Top-1	77.4	66.0	71.2	66.7	57.0	61.5
Top-2	81.8	69.2	75.0	75.6	64.0	69.4
Top-5	86.8	72.4	78.0	83.7	69.9	76.2

Table 6. Re-ranking performance for test set

To improve our re-ranker, more features regarding these problems should be added, and a more principled method to obtain the probability of segmenations, *p(seg)* in Sectioin 5.1, needs to be investigated.

Table 7 compares the final output with G & J's best result. Our model is slightly worse than their integrated model, but it supports much further experimentation in segmentation and re-ranking.

Method	FE identification			FE indetification & Role tagging		
	Prec	Rec	F	Prec	Rec	F
G & J integrated model	74.0	70.1	72.0	64.6	61.2	62.9
Our model w/ re-ranking	77.4	66.0	71.2	66.7	57.0	61.5

Table 7. The final output for test set.

6 Conclusion

We describe a pipeline framework to analyze sentences into frame elements and semantic roles based on the FrameNet corpus. The process includes four steps: sentence segmentation, FE identification, role classification, and final re-ranking of the *n*-best outputs.

In future work, we will investigate ways to reduce the gap between the five-best output performance and the single best output. More features should be extracted to improve re-ranking accuracy. Although the segmentation improves the performance, since the final output is dominated by the initial segmentation, we will explore a smart segmentation method, possibly one not even limited to constituents.

In addition to the provided syntactic features, we will apply semantic features using ontology. Finally, the challenge is to apply this type of work to new predicates, ones not yet treated in FrameNet. We are searching for methods to achieve this.

References

O. Bender, K. Macherey, F.J. Och, and H. Ney. 2003. Comparison of Alignment Templates and Maximum Entropy Models for Natural Language Processing. Proc. of *EACL-2003*. Budapest, Hungary.

A. Berger, S. Della Pietra and V. Della Pietra, 1996. A Maximum Entropy Approach to Natural Language Proc. of *Computational Linguistics,* vol. 22, no. 1.

S.F. Chen and R. Rosenfeld. 1999. A Gaussian Prior for Smoothing Maximum Entropy Models. *Technical Report CMUCS-99-108*, Carnegie Mellon University.

M. Collins. 1997. Three Generative, Lexicalized Models for Statistical Parsing. *Proc. of the 35th Annual Meeting of the ACL*. pages 16-23, Madrid, Spain.

J. N. Darroch and D. Ratcliff. 1972. *Generalized Iterative Scaling for Log-Linear Models*. Annals of Mathematical Statistics, 43:1470-1480.

C.Fillmore 1976. Frame Semantics and the Nature of Language. *Annals of the New York Academy of Science Conference on the Origin and Development of Language and Speech,* Volume 280 (pp. 20-32).

M. Fleischman, N. Kwon, and E. Hovy. 2003. Maximum Entropy Models for FrameNet Classification. Proc. of *Empirical Methods in Natural Language Processing conference (EMNLP) 2003*. Sapporo, Japan.

D. Gildea and D. Jurafsky. 2002. Automatic Labeling of Semantic Roles. *Computational Linguistics*, 28(3) 245-288 14.

K. Hacioglu, W. Ward. 2003. Target word detection and semantic role chunking using support vector machines. Proc. of *HLT-NAACL 2003*, Edmonton, Canada.

F.J. Och. 2002. Yet Another Maxent Toolkit: YASMET www-i6.informatik.rwth-aachen.de/ Colleagues/och/.

S. Pradhan, K. Haciolgu, W. Ward, J. Martin, D. Jurafsky. 2003. Semantic Role Parsing: Adding Semantic Structure to Unstructured Text. Proc of *of the International Conference on Data Mining (ICDM-2003),* Melbourne, FL

C. Thompson, R. Levy, and C. Manning. 2003. A Generative Model for FrameNet Semantic Role Labeling. Proc. of *the Fourteenth European Conference on Machine Learning*, Croatia