

# Prediction of Maximal Projection for Semantic Role Labeling

**Weiwei Sun\*, Zhifang Sui**

Institute of Computational Linguistics  
Peking University  
Beijing, 100871, China  
{ws, szf}@pku.edu.cn

**Haifeng Wang**

Toshiba (China) R&D Center  
501, Tower W2, Oriental Plaza  
Beijing, 100738, China  
wanghaifeng@rdc.toshiba.com.cn

## Abstract

In Semantic Role Labeling (SRL), arguments are usually limited in a syntax sub-tree. It is reasonable to label arguments locally in such a sub-tree rather than a whole tree. To identify active region of arguments, this paper models Maximal Projection (MP), which is a concept in D-structure from the projection principle of the Principle and Parameters theory. This paper makes a new definition of MP in S-structure and proposes two methods to predict it: the anchor group approach and the single anchor approach. The anchor group approach achieves an accuracy of 87.75% and the single anchor approach achieves 83.63%. Experimental results also indicate that the prediction of MP improves semantic role labeling.

## 1 Introduction

Semantic Role Labeling (SRL) has gained the interest of many researchers in the last few years. SRL consists of recognizing arguments involved by predicates of a given sentence and labeling their semantic types. As a well defined task of shallow semantic parsing, SRL has a variety of applications in many kinds of NLP tasks.

A variety of approaches has been proposed for the different characteristics of SRL. More recent approaches have involved calibrating features (Gildea and Jurafsky, 2002; Xue and Palmer, 2004;

Pradhan et al., 2005), analyzing the complex input – syntax trees (Moschitti, 2004; Liu and Sarkar, 2007), exploiting the complicated output – the predicate-structure (Toutanova et al., 2005), as well as capturing paradigmatic relations between predicates (Gordon and Swanson, 2007).

In prior SRL methods, role candidates are extracted from a whole syntax tree. Though several pruning algorithms have been raised (Xue and Palmer, 2004), the policies are all in global style. In this paper, a statistical analysis of Penn Prop-Bank indicates that arguments are limited in a local syntax sub-tree rather than a whole one. Prior SRL methods do not take such locality into account and seek roles in a wider area. The neglect of locality of arguments may cause labeling errors such as constituents outside active region of arguments may be falsely recognized as roles.

This paper uses insights from generative linguistics to guide the solution of locality of arguments. In particular, Maximal Projection (MP) which dominates<sup>1</sup> active region of arguments according to the projection principle of principle and parameters. Two methods, the anchor group approach and the single anchor approach, are proposed to find the active sub-tree which is rooted by MP and covers all roles. The solutions put forward in this paper borrow ideas from NP-movement principle in generative linguistics and are in statistical flavor. The anchor group approach achieves an accuracy of 87.75%, and the single anchor approach achieves 83.63%. Though the accuracy is lower, the single anchor approach fits SRL better.

---

This work was partial completed while this author was at Toshiba (China) R&D Center.

© 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.

---

<sup>1</sup>Dominates is a concept in X-bar theory are modeled. Assuming  $\alpha$  and  $\beta$  are two nodes in a syntax tree:  $\alpha$  dominates  $\beta$  means  $\alpha$  is ancestor of  $\beta$ .

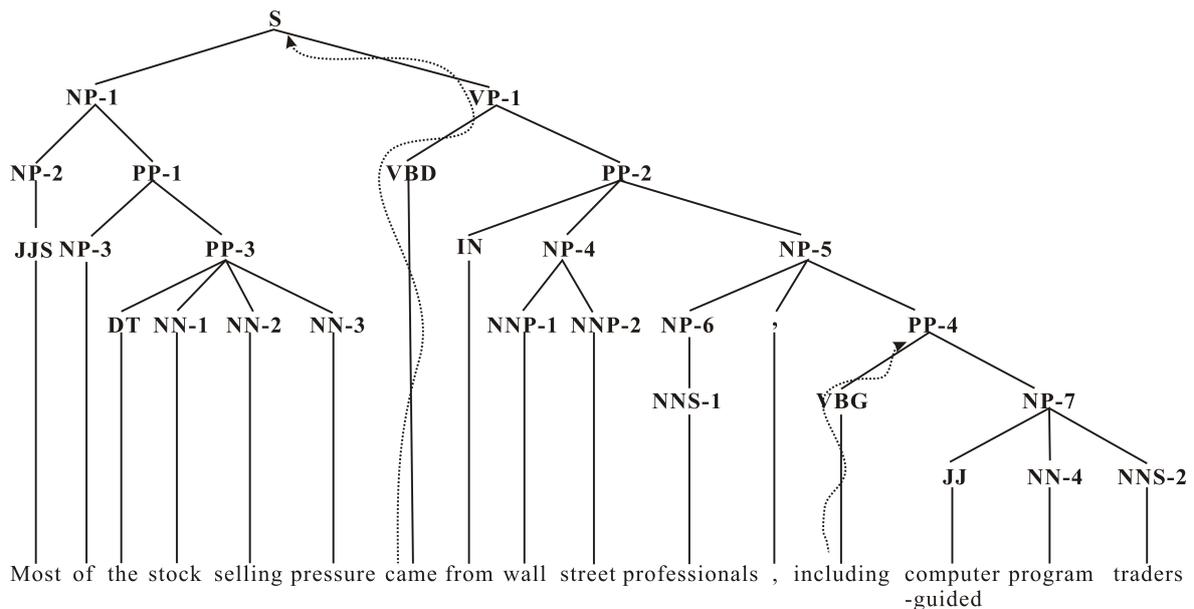


Figure 1: A sentence from WSJ test corpus of CoNLL-2005 shared task

## 2 Maximal Projection and Its Government of Arguments

### 2.1 Maximal Projection

Principle and parameters theory is a framework of generative grammar. X-bar theory, as a module of principle and parameters, restricts context-free phrase structure rules as follows:

1. a phrase always contains a head of the same type, i.e. NPs Ns, VPs Vs, PPs Ps, etc.
2.  $XP(X'') \rightarrow \text{specifier } X'$
3.  $X' \rightarrow X \text{ complement(s)}$

These structural properties are conventionally represented as shown in figure 2.

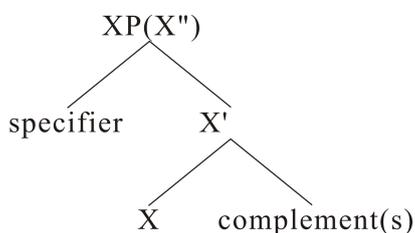


Figure 2: X-bar structure

X is the head of the phrase XP. X' and  $XP(X'')$  are called projections of X. The head is also called the zero projection. X-bar structure is integrated with the properties of lexical items via the **Projection Principle** of principle and parameters. This

principle is summed up as the properties of lexical information project onto the syntax of the sentence. For instance:

- Sue likes Picasso
- \*Sue likes

The subcategorization frame of the lexical item *like* [<sub>S</sub>, NP] ensures that the verb is followed by an NP and the second sentence is of ungrammatical form.

**Maximal Projection** (MP) is the constituent which is projected to the highest level of an X-bar structure from lexical entities and is therefore the top node XP of the X-bar structure.

Take figure 1 for instance, S is the MP of the predicate *come*. Though the syntax tree is not in D-structure (deep structure), the S-structure (surface structure) headed by *come* is similar to its genuine D-structure. In a latter part of this section, a specific definition of MP in S-structure will be given for application.

### 2.2 MP Limits Active Region of Arguments

MP holds all lexical properties of heads. In particular, the MP of a predicate holds predicate structure information and the constituents out of its domain cannot occupy argument positions.  $\theta$ -theory and government are two modules of principle and parameters. They both suggest that the possible positions of semantic roles are in the sub-tree rooted by MP.

Concerning assignment of semantic roles to constituents,  $\theta$ -theory suggests that semantic roles are assigned by predicates to their sisters (Chomsky, 1986). Furthermore, in a X-bar theory, complements are assigned semantic roles by the predicate and specifiers get roles from the V'. In both situations the process of roles assignment is in sisterhood condition and limited in the sub-structure which is dominated by the MP. Only constituents under MP can get semantic roles. **The Case Assignment Principle** also points out: Case is assigned under government (Chomsky, 1981). Take figure 1 for instance, only *NP-1* and *PP-2* can get semantic roles of the head *come*.

From generative linguists' point, MP limits subtree of arguments. Therefore, finding the MP is equivalent to finding the active region of predicate structure.

### 2.3 Definition of MP in S-structure

Though a clear enough definition of MP in D-structure has been previously illustrated, it is still necessary to define a specific one in S-structure for application, especially for automatic parsing which are not exactly correct. This paper defines **MP in S-structure** (hereinafter denote MP for short) as following: for every predicate  $p$  in the syntax tree  $T$ , there exists one and only one MP  $mp$  s.t.

1.  $mp$  dominates all arguments of  $p$ ;
2. all descendent nodes of  $mp$  don't satisfy the former condition.

Due to its different characteristics from arguments, adjunct-like arguments are excluded from the set of arguments in generative grammar and many other linguistic theories. For this reason, this paper does not take them into account.

For gold syntax tree, there exists a one-to-one mapping between arguments and nodes of syntax trees, whereas automatic syntactic parsing contains no such mapping. This paper do not take arguments which cannot get corresponding constituents into account to reduce the influence of automatic parsing error.

Take the sentence of figure 1 to illustrate our definition of MP:  $S$  is MP of *come* since *NP-1* and *PP-2* are arguments of it. There is no node mapping to the argument *Wall Street professionals* in the parsing tree. Instead of covering argument's fragments, we simply take it *PP-4* as MP.

### 2.4 Using MP Information in SRL

The boundaries of a predicate structure are two word positions of the sentence. It is difficult to model these two words. On the contrary, MP, as one ancestor of predicate, has a clear-cut meaning and is ideal for modeling. In this paper, the policy to predict MP rather than two word positions is carried out to deal with locality of arguments.

Automatic prediction of MP can be viewed as a preprocessing especially a pruning preprocessing for SRL. Given a sentence and its parsing, SRL systems can take seeking the active sub-tree rooted by MP as the first step. Then SRL systems can work on the shrunk syntax tree, and follow-up labeling processes can be in a various form. Most of previous SRL methods still work without special processing. Take figure 1 for example: when labeling *include*, as the MP is *PP-4*, just *NP-7* will be extracted as argument candidate.

## 3 Analysis of Locality of Arguments

Principle and parameters suggests that MP bounds arguments. Additionally, a statistical analysis shows that possible positions of arguments are limited in a narrow region of syntax tree. An opposite experiment also shows that MP information is useful for SRL.

### 3.1 Data and Baseline System

In this paper, CoNLL-2005 SRL shared task data (Carreras and Màrquez, 2005) is used as corpus. The data consists of the Wall Street Journal (WSJ) part of the Penn TreeBank with information on predicate argument structures extracted from the PropBank corpus. In addition, the test set of the shared task includes three sections of the Brown corpus. Statistical analysis is based on section 02-21 of WSJ. Experiments are conducted on WSJ and Brown corpus. As defined by the shared task, section 02-21 of PropBank are used for training models while section 23 and Brown corpus are used for test. In terms of syntax information, we use Charniak parser for POS tagging and full parsing.

A majority of prior SRL approaches formulate the SRL problem as a multi-class classification problem. Generally speaking, these SRL approaches use a two-stage architecture: i) argument identification; ii) argument classification, to solve the task as a derivation of Gildea and Jurafsky's pioneer work (Gildea and Jurafsky, 2002). UIUC

	Precision	Recall	$F_{\beta=1}$
Arg0	86.28%	87.01%	86.64
Arg1	79.37%	75.06%	77.15
Arg2	69.48%	62.97%	66.07
Arg3	69.01%	56.65%	62.22
Arg4	72.64%	75.49%	74.04

Table 1: SRL performance of UIUC SRLer

Semantic Role Labeler<sup>2</sup> (UIUC SRLer) is a state-of-the-art SRL system that based on the champion system of CoNLL-2005 shared task (Carreras and Màrquez, 2005). It is utilized as a baseline system in this paper. The system participated in CoNLL-2005 is based on several syntactic parsing results. However, experiments of this paper just use the best parsing result from Charniak parser. Parameters for training SRL models are the same as described in (Koomen, 2005).

### 3.2 Active Region of Arguments

According to a statistical analysis, the average depth from a target predicate to the root of a syntax tree is 5.03, and the average depth from a predicate to MP is just 3.12. This means about 40% of ancestors of a predicate do not dominate arguments directly. In addition, the quantity of leaves in syntax tree is another measure to analyze the domain. On average, a syntax tree covers 28.51 leaves, and MP dominates only 18.19. Roughly speaking, only about 60% of words are valid for semantic roles. Statistics of corpora leads to the following conclusion: arguments which are assigned semantic roles are in a local region of a whole syntax tree.

### 3.3 Typical Errors Caused by Neglect of Locality of Arguments

The neglect of the locality of arguments in prior SRL methods shows that it may cause errors. Some constituents outside active region of arguments may be falsely labeled as roles especially for those being arguments of other predicates. A statistical analysis shows 20.62% of falsely labeled arguments are constituents out of MP domain in labeling results of UIUC SRLer. Take figure 1 for instance, UIUC SRLer makes a mistake when labeling *NP-1* which is Arg1 of the predicate *come* for the target *include*; it labels Arg0 to *NP*. In fact, the active region of *include* is the sub-tree rooted

<sup>2</sup><http://l2r.cs.uiuc.edu/cogcomp/srl-demo.php>

	Precision	Recall	$F_{\beta=1}$
Arg0	91.84%	89.98%	90.90
Arg1	81.73%	75.93%	78.72
Arg2	69.86%	63.06%	66.29
Arg3	71.13%	58.38%	64.13
Arg4	73.08%	74.51%	73.79

Table 2: SRL performance of UIUC SRLer using information of gold MP

by *PP-4*. Since *NP-1* is an argument of another predicate, some static properties of *NP-1* make it confusing as an argument.

### 3.4 SRL under Gold MP

If MP has been found before labeling semantic roles, the set of role candidates will be shrunk, and the capability to identify semantic roles may be improved. An opposite experiment verifies this idea. In the first experiment, UIUC SRLer is re-trained as a baseline. For comparison, during the second experiment, syntax sub-trees dominated by gold MP are used as syntactic information. Both training and test data are preprocessed with gold MP information. That is to say we use pruned data for training, and test is conducted on pruned syntax sub-trees.

Table 1 and 2 show that except for Arg4, all arguments get improved labeling performance, especially Arg0. Since arguments except for Arg0 are realized as objects on the heel of predicate in most case, the information of MP is not so useful for them as Arg0. The experiment suggests that high performance prediction of MP can improve SRL.

## 4 Prediction of MP

Conforming to government and  $\theta$ -theory, MP is not too difficult to predict in D-structure. Unfortunately, sentences being looked at are in their surface form and region of arguments has been expanded. Simple rules alone are not adequate for finding MP owing to a variety of movement between D-structure and S-structure. This paper designs two data driven algorithms based on movement principles for prediction of MP.

### 4.1 NP-movement and Prediction of MP

#### 4.1.1 NP-movement in Principle and Parameters

The relationship between D-structure and S-structure is movement: S-structure equals D-

structure plus movement. NP-movement principle in principle and parameters indicates that noun phrases only move from A-positions (argument position) which have been assigned roles to A-positions which have not, leaving an NP-trace. On account of  $\theta$ -theory and government, A-positions are nodes m-commanded<sup>3</sup> by predicates in D-structure. In NP-movement, arguments move to positions which are C-commanded<sup>4</sup> by target predicate and m-commanded by other predicates. Broadly speaking, A-positions are C-commanded by predicates after NP-movement. The key of the well-known pruning algorithm raised in (Xue and Palmer, 2004) is extracting sisters of ancestors as role candidates. Those candidate nodes are all C-commanders of a predicate. NP-movement can give an explanation why the algorithm works.

#### 4.1.2 Definition of Argument Anchor

To capture the characteristics of A-positions, we make definition of A-anchor as following. For every predicate  $p$  in the syntax tree  $T$ , denote  $\mathcal{A}$  the set of C-commanders of  $p$ :

- a **left-A-anchor** satisfies:
  1. left-A-anchor belongs to  $\mathcal{A}$ ;
  2. left-A-anchor is a noun phrase (including NNS, NNP, etc.) or simple declarative clause (S);
  3. left-A-anchor is on the left hand of  $p$ .
- a **right-A-anchor** satisfies:
  1. right-A-anchor belongs to  $\mathcal{A}$ ;
  2. right-A-anchor is a noun phrase (including NNS, NNP, etc.);
  3. right-A-anchor is on the right hand of  $p$ .

Take figure 1 for example, *NP-1*, *NP-4* and *NP-6* are left-A-anchors of *include*, and no right-A-anchor. There is a close link between A-position and the A-anchor that we defined, since A-anchors occupy A-positions.

#### 4.1.3 Anchor Model for Prediction of MP

Parents of A-anchors and first branching ancestor of the predicate can cover 96.25% of MP and the number of those ancestors is 2.78 times of the

<sup>3</sup>M-command is a concept in X-bar syntax. Assuming  $\alpha$  and  $\beta$  are two nodes in a syntax tree:  $\alpha$  m-commands  $\beta$  means  $\alpha$  C-commands  $\beta$  and the MP of  $\alpha$  dominates  $\beta$

<sup>4</sup>C-command is a concept in X-bar theory. Assuming  $\alpha$  and  $\beta$  are two nodes in a syntax tree:  $\alpha$  C-commands  $\beta$  means every parent of  $\alpha$  is ancestor of  $\beta$ .

number of MP. The number of all ancestors is 6.65 times. The data suggests that taking only these kinds of ancestors as MP candidates can shrink the candidate set with a relatively small loss.

#### 4.2 Anchor Group Approach

MP is one ancestor of a predicate. An natural approach to predict MP is searching the set of all ancestors. This idea encounters the difficulty that there are too many ancestors. In order to reduce the noise brought by non-anchors' parents, the anchor group approach prunes away useless ancestors which are neither parents of A-anchors nor first branching node upon predicate from MP candidate set. Then the algorithm scores all candidates and chooses the MP in argmax flavor. Formally, we denote the set of MP candidates  $\mathcal{C}$  and the score function  $S(\cdot)$ .

$$\hat{mp} = \arg \max_{c \in \mathcal{C}} S(mp|c)$$

Probability function is chosen as score function in this paper. In estimating of the probability  $P(MP|\mathcal{C})$ , log-linear model is used. This model is often called maximum entropy model in research of NLP. Let the set  $\{1, -1\}$  denotes whether a constituent is MP and  $\Phi(c, \{-1, 1\}) \in \mathbb{R}^s$  denotes a feature map from a constituent and the possible class to the vector space  $\mathbb{R}^s$ . Formally, the model of our system is defined as:

$$\hat{mp} = \arg \max_{c \in \mathcal{C}} \frac{e^{\langle \Phi(c, 1), \omega \rangle}}{e^{\langle \Phi(c, 1), \omega \rangle} + e^{\langle \Phi(c, 0), \omega \rangle}}$$

The algorithm is also described in pseudo code as following.

Ancestor Algorithm:

- 1: collect parents of anchors and the first branching ancestor, denote them set  $\mathcal{C}$
- 2: for every  $c \in \mathcal{C}$
- 3: calculate  $P(mp|c)$
- 4: return  $\hat{c}$  that gets the maximal  $P(mp|c)$

#### 4.2.1 Features

We use some features to represent various aspects of the syntactic structure as well as lexical information. The features are listed as follows:

**Path** The path features are similar to the path feature which is designed by (Gildea and Jurafsky, 2002). A path is a sequential collection of phrase tags. There are two kinds of path features here: one is from target predicate through to the candidate; the other is from the candidate to the root of the syntax tree. For *include* in the sentence of figure 1, the first kind of path of *PP-2* is *VBG+PP+NP+PP* and the second is *PP+VP+S*.

**C-commander Thread** As well as path features, C-commander threads are other features which reflect aspects of the syntactic structures. C-commander thread features are sequential containers of constituents which C-command the target predicate. We design three kinds of C-commander threads: 1) down thread collects C-commanders from the anchor to the target predicate; 2) up thread collects C-commanders from the anchor to the left/right most C-commander; 3) full thread collects all C-commanders in the left/right direction from the target predicate. Direction is dependent on the type of the anchor - left or right anchor.

Considering the grammatical characteristics of phrase, we make an equivalence between such phrase types:

- JJ, JJR, JJS, ADJP
- NN, NNP, NNS, NNPS, NAC, NX, NP

Besides the equivalent constituents, we discard these types of phrases:

- MD, RB, RBS, RBR, ADVP

For *include* in figure 1, the up thread of *NP-4* is *VBG+, +NP+NP*; the down thread is *NP+IN+VBD+NP*; the full thread is *VBG+, +NP+NP+IN+VBD+NP*.

The phrase type of candidate is an important feature for prediction

**Candidate** of MP. We also select the rank number of the current candidate and the number of all candidates as features. For the former example, the two features for *PP-2* are 2 and 3, since *NP-4* is the second left-A-anchor and there are three A-anchors of *include*.

**Anchor** Features of anchor include the head word of the anchor, the boundary words and their POS, and the number of the words in the anchor. Those features are clues of judgment of whether the anchor's position is an A-position.

**Forward predicate** For the former example, the forward predicate of *NP-4* is *come*. The features include the predicate itself, the Levin class and the SCF of the predicate.

**predicate** Features of predicate include lemma, Levin class, POS and SCF of the predicate.

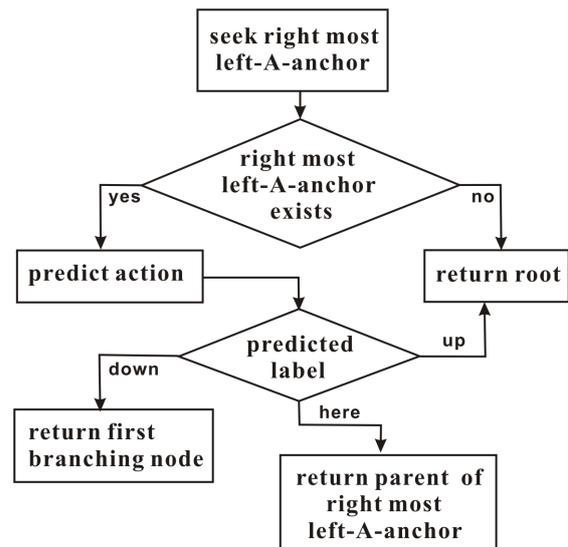


Figure 3: Flow diagram of the single anchor approach

**Formal Subject** An anchor may be formal subject. Take *It is easy to say the specialist is not doing his job* for example, the formal subject will be recognized as anchor of *do*. We use a heuristic rule to extract this feature: if the first NP C-commander of the anchor is "it" and the left word of predicate is "to", the value of this feature is 1; otherwise 0.

**The Maximal Length of C-commanders** Constituent which consists of many words may be a barrier between the predicate and an A-position. For the former example, if the target predicate is *include*, this feature of *NP-1* is 2, since the largest constituent *NP-4* is made up of two words.

### 4.3 Single Anchor Approach

Among all A-anchors, the right most left-A-anchor such as *NP-6* of *include* in figure 1 is the most important one for MP prediction. The parent of this kind of left-A-anchor is the MP of the predicate, obtaining a high probability of 84.59%. The single anchor approach is designed based on right most left-A-anchor. The key of this approach is an action prediction that when right most left-A-anchor is found, the algorithm predicts next action to return which node of syntax tree as MP. There is a label set of three types for learning – *here*, *up*, *down*. After action is predicted, several simple rules are executed as post process of this prediction: i) if there is no left-A-anchor, return the root of the whole syntax tree as MP; ii) if the predicted label is *here*, return the parent of right most left-A-anchor; iii) if the predicted label is *down*, return

Corpus	Prediction Accuracy	
	Action	MP
WSJ	–	87.75%
Brown	–	88.84%

Table 3: Accuracy of the anchor group approach

	Precision	Recall	$F_{\beta=1}$
Arg0	86.23%	87.90%	87.06
Arg1	80.21%	74.79%	77.41
Arg2	70.09%	62.70%	66.19
Arg3	71.74%	57.23%	63.67
Arg4	74.76%	75.49%	75.12

Table 5: SRL performance of UIUC SRLer using information of predicted MP; the anchor group approach; WSJ test corpus

the first branching node upon the predicate; iv) if the predicted label is *up*, return the root. The action prediction also uses maximum entropy model. Figure 3 is the flow diagram of the single anchor approach. Features for this approach are similar to the former method. Features of the verb which is between the anchor and the predicate are added, including the verb itself and the Levin class of that verb.

## 5 Experiments and Results

Experiment data and toolkit have been illustrated in section 3. Maxent<sup>5</sup>, a maximum entropy modeling toolkit, is used as a classifier in the experiments of MP prediction.

### 5.1 Experiments of Prediction of MP

The results are reported for both the anchor group approach and the single anchor approach. Table 3 summaries the accuracy results of MP prediction for the anchor group approach; table 4 summaries results of both action prediction and MP prediction for the single anchor approach. Both the anchor group approach and the single anchor approach have better prediction performance in Brown test set, though the models are trained on WSJ corpus. These results illustrate that anchor approaches which are based on suitable linguistic theories have robust performance and overcome limitations of training corpus.

Corpus	Prediction Accuracy	
	Action	MP
WSJ	88.45%	83.63%
Brown	90.10%	85.70%

Table 4: Accuracy of the single anchor approach

	Precision	Recall	$F_{\beta=1}$
Arg0	87.03%	87.59%	87.31
Arg1	80.24%	74.77%	77.41
Arg2	70.35%	63.06%	66.51
Arg3	71.43%	57.80%	63.90
Arg4	73.33%	75.49%	74.40

Table 6: SRL performance of UIUC SRLer using information of predicted MP; the single anchor approach; WSJ test corpus

### 5.2 Experiments of SRL Using MP Prediction

Like the experiments in the end of section 3, we perform similar experiments under predicted MP. Both training and test corpus make use of predicted MP information. It is an empirical tactic that predicted information of maximal projection, instead of gold information, is chosen for a training set. Experiments suggest predicted information is better. Table 5 is SRL performance using the anchor group approach to predict MP; Table 6 is SRL performance using the single anchor approach.

Compared with table 1 on page 4, table 5 and table 6 both indicate the predicted MP can help to label semantic roles. However, there is an interesting phenomenon. Even though the anchor group approach achieves a higher performance of MP, the single anchor approach is more helpful to SRL. 18.56% of falsely labeled arguments are out of MP domain using the single anchor approach to predict MP, compared to 20.62% of the baseline system.

In order to test robustness of the contribution of MP prediction to SRL, another opposite experiment is performed using the test set from Brown corpus. Table 7 is the SRL performance of UIUC SRLer on Brown test set. Table 8 is the corresponding performance using MP information predicted by the single anchor approach. Comparison between table 7 and table 8 indicates the approach of MP prediction proposed in this paper adapts to other genres of corpora.

Capability of labeling Arg0 gets significant improvement. Subject selection rule, a part of the-

<sup>5</sup>[http://homepages.inf.ed.ac.uk/s0450736/maxent\\_toolkit.html](http://homepages.inf.ed.ac.uk/s0450736/maxent_toolkit.html)

	Precision	Recall	$F_{\beta=1}$
Arg0	82.88%	85.51%	84.17
Arg1	66.30%	63.17%	64.70
Arg2	50.00%	45.58%	47.69
Arg3	0.00%	0.00%	0.00
Arg4	60.00%	20.00%	30.00

Table 7: SRL performance of UIUC SRLer; Brown test corpus

matic hierarchy theory, states that the argument that the highest role (i.e. proto-agent, Arg0 in PropBank) is the subject. This means that Arg0 is usually realized as a constituent preceding a predicate and has a long distance from the predicate. As a solution of finding active region of arguments, MP prediction is helpful to shrink the searching range of arguments preceding the predicate. From this point, we give a rough explanation why experiment results for Arg0 are better.

## 6 Conclusion

Inspired by the locality phenomenon that arguments are usually limited in a syntax sub-tree, this paper proposed to label semantic roles locally in the active region arguments dominated by maximal projection, which is a concept in D-structure from the projection principle of the principle and parameters theory. Statistical analysis showed that MP information was helpful to avoid errors in SRL, such as falsely recognizing constituents outside active region as arguments. To adapt the projection concept to label semantic roles, this paper defined MP in S-structure and proposed two methods to predict MP, namely the anchor group approach and the single anchor approach. Both approaches were based on NP-movement principle of principle and parameters. Experimental results indicated that our MP prediction methods improved SRL.

## Acknowledgements

The work is supported by the National Natural Science Foundation of China under Grants No. 60503071, 863 the National High Technology Research and Development Program of China under Grants No.2006AA01Z144, 973 Natural Basic Research Program of China under Grants No.2004CB318102.

	Precision	Recall	$F_{\beta=1}$
Arg0	83.85%	86.22%	85.02
Arg1	66.67%	63.02%	64.79
Arg2	50.38%	44.90%	47.48
Arg3	0.00%	0.00%	0.00
Arg4	60.00%	20.00%	30.00

Table 8: SRL performance of UIUC SRLer using information of predicted MP; the single anchor approach; Brown test corpus

## References

- Carreras, Xavier and Lluís Màrquez. 2005. Introduction to the CoNLL-2005 shared task: semantic role labeling. In *Proceedings of Conference on Natural Language Learning*.
- Chomsky, Noam. 1981. *Lectures on Government and Binding*. Foris Publications, Dordrecht.
- Chomsky, Noam. 1986. *Barriers*. MIT Press, Barriers.
- Gildea, Daniel and Daniel Jurafsky. 2002. Automatic Labeling of Semantic Roles. *Computational Linguistics*, 28(3):245–288.
- Gordon, Andrew and Reid Swanson. 2007. Generalizing Semantic Role Annotations Across Syntactically Similar Verbs. In *Proceedings of Conference on Association for Computational Linguistics*.
- Koomen, Peter, Vasina Punyakanok, Dan Roth and Wen-tau Yih. 2005. Generalized Inference with Multiple Semantic Role Labeling Systems. In *Proceedings of Conference on Natural Language Learning*.
- Liu, Yudong and Anoop Sarkar. 2004. Experimental Evaluation of LTAG-Based Features for Semantic Role Labeling. In *Proceedings of Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*.
- Mocshitti, Alessandro. 2004. A Study on Convolution Kernels for Shallow Semantic Parsing. In *Proceedings of Conference on Association for Computational Linguistics*.
- Pradhan, Sameer, Kadri Hacioglu, Valerie Krugler, Wayne Ward, James Martin and Daniel Jurafsky. 2005. Support Vector Learning for Semantic Argument Classification. In *Proceedings of Conference on Association for Computational Linguistics*.
- Toutanova, Kristina, Aria Haghighi and Christopher Manning. 2005. Joint Learning Improves Semantic Role Labeling. In *Proceedings of Conference on Association for Computational Linguistics*.
- Xue, Nianwen and Martha Palmer. 2004. Calibrating Features for Semantic Role Labeling. In *Proceedings of Empirical Methods in Natural Language Processing*.