Two-Phase Semantic Role Labeling based on Support Vector Machines

Kyung-Mi Park and **Young-Sook Hwang** and **Hae-Chang Rim**
Department of Computer Science & Engineering, Korea University
5-ka, Anam-dong, SEOUL, 136-701, KOREA
{kmpark, yshwang, rim}@nlp.korea.ac.kr

Abstract

In this study, we try to apply SVMs to the semantic role labeling task, which is one of the multiclass problems. As a result, we propose a two-phase semantic role labeling model which consists of the identification phase and the classification phase. We first identify semantic arguments, and then assign semantic roles to the identified semantic arguments. By taking the two-phase approach, we can alleviate the unbalanced class distribution problem, and select the features appropriate for each task.

1 Introduction

A semantic role in a language is a semantic relationship between a syntactic constituent and a predicate. The shared task of CoNLL-2004 relates to recognize semantic roles in English (X. Carreras, 2004). Given a sentence, the task is to analyze a proposition expressed by a target verb of a sentence. Especially, for each target verb, all constituents in a sentence which fill semantic roles of the verb have to be recognized. This task is based only on partial parsing information, avoiding use of a full parser and external lexico-semantic knowledge base. According to previous results of the CoNLL shared task, the POS tagged, chunked, clause identified, and named-entity recognized sentences are given as an input (Figure 1).

SVM is a well-known machine learning algorithm with high generalization performance in high dimensional feature spaces (H. Yamada, 2003). Also, learning with combination of multiple features is possible by virtue of polynomial kernel functions. However, since it is a binary classifier, we are often confronted with the unbalanced class distribution problem in a multiclass classification task. The larger the number of classes, the more severe the problem is. The semantic role labeling can be formulated as a multiclass classification problem. If we try to apply SVMs in the semantic role labeling problem, we have to find a method of resolving the unbalanced class distribution problem.

Conceptually, semantic role labeling can be divided into two subtasks: the identification task which finds the boundary of semantic arguments in a given sentence, and the classificiation task which determines the semantic role of the argument. This provides us a hint of using SVMs with less severe unbalanced class distribution. In this paper, we present a two-phase semantic role labeling method which consists of an identification phase and a classification phase. By taking two phase model based on SVMs, we can alleviate the unbalanced class distribution problem. That is, since we find only the boundary of an argument in the identification phase, the number of classes is decreased into two (*ARG, NON-ARG*) or three (*B-ARG, I-ARG, O*). Therefore, we have to build only one or three SVM classifiers. We can alleviate the unbalanced class distribution problem by decreasing the number of negative examples, which is much larger than the number of positive exampels without two-phase modeling. In the classification phase, we classify only the identified argument into a proper semantic role. This enables us to reduce the computational cost by ignoring the non-argument constitutents.

Since features for identifying arguments are different from features for classifying a role, we need to determine different feature sets appropriate for the tasks. For identification, we focus on the features to detect the dependency between a constituent and a predicate because the arguments are dependent on the predicate. For semantic role labeling, we consider both the syntactic and the semantic information such as the sentential form of the target predicate, the head of a constituent, and so on. In the following sections, we will explain the two phase semantic role labeling method in detail and show some experimental results.

1	2	3	4	5	6	7	8
Under	IN	B-PP	(S*	O	-	*	(AM-LOC*
the	DT	B-NP	*	O	-	*	*
existing	VBG	I-NP	*	O	exist	(V*V)	*
contract	NN	I-NP	*	O	-	(A1*A1)	*AM-LOC)
,	,	O	*	O	-	*	*
Rockwell	NNP	B-NP	(S*	B-ORG	-	*	*
said	VBD	B-VP	*S)	O	-	*	*
,	,	O	*	O	-	*	*
it	PRP	B-NP	*	O	-	*	(A0*A0)
has	VBZ	B-VP	*	O	-	*	*
already	RB	I-VP	*	O	-	*	(AM-TMP*AM-TMP)
delivered	VBN	I-VP	*	O	deliver	*	(V*V)
793	CD	B-NP	*	O	-	*	(A1*
of	IN	B-PP	*	O	-	*	*
the	DT	B-NP	*	O	-	*	*
shipsets	NNS	I-NP	*	O	-	*	*A1)
to	TO	B-PP	*	O	-	*	*
Boeing	NNP	B-NP	*	B-MISC	-	*	(A2*A2)
.	.	O	*S)	O	-	*	*

Figure 1: An example of semantic role labeling. The columns contain: the word, its POS, its chunk type, clause boundary, its named-entity tag, the target predicate, the result of semantic role labeling in the target predicate *exist*, and *deliver*.

2 Two Phase Semantic Role Labeling based on SVMs

We regard the semantic role labeling as a classification problem of a syntactic constituent. However, a syntactic constituent can be a chunk, or a clause. Therefore, we have to identify the boundaries of semantic arguments before we assign roles to the arguments.

2.1 Semantic Argument Identification

This phase is the step of finding the boundary of semantic arguments. A sequence of chunks or a subclause in the immediate clause of a predicate can be a semantic argument of the predicate. A chunk or a subclause of the predicate becomes a unit of the constituent of an argument. The chunks within the subclause are ignored.

For identifying the semantic arguments of a target predicate, it is necessary to find the dependency relation between each constituent and a predicate. Identifying a dependency relation is important for identifying a subject/object relation (S. Buchholz, 2002) and also for identifying the semantic arguments of a target predicate. Therefore, the features for finding dependency relations are implicitly represented in the feature set for the identification task.

For implementing the method based on the SVMs, we represent a constituent of an argument with B/I/O notation, and assign one of the following classes to each constituent: *B-ARG* class representing the beginning of semantic argument, *I-ARG* class representing a part of a semantic argument, or *O* class indicating that the constituent does not belong to the semantic arguments.

Because we decide the unit of a constituent as a chunk or a subclause, words except the predicate in the target phrase [1] do not belong to constituent. Therefore, these words have to be handled independently. In the training data, we often observed that the beginning of semantic arguments starts from the word right after the predicate. For the agreement with the chunk boundary, we regard the word following a predicate as the beginning word of a new chunk. Namely, when the beginning of chunk tag is *I*, we change *I* to *B*. Also, the words located in front of the predicate in the target phrase are post-processed by 4 hand-crafted rules [2] and 211 automated rules [3] based on frequency in the training data.

In order to restrict the search space in terms of the constituents, we use the clause boundaries. The left search boundary for identifying the semantic argument is set to the left boundary of the second upper clause, and the right search boundary is set to the right boundary of the immediate clause.

2.1.1 Features for Identifying Semantic Argument

For this phase, we use 29 features for representing syntactic and semantic information related to constituent and predicate. Table1 shows a set of features employed. The features can be described as follows:

- **position**: This is a binary feature identifying whether the constituent is before (-1) or after (1) the predicate in the immediate clause. The feature value

[1] The chunk containing a predicate is referred to as *target phrase*.

[2] For example, if a word in target phrase is *n't, not or Not*, and POS tag of the word is *RB* and the distance between the word and the predicate is less than 4, then the semantic role is *AM-NEG*.

[3] For example, if a word in target phrase is *already*, and POS tag of the word is *RB*, then the semantic role is *AM-TMP*.

Features		examples
predicate-candidate (intervening features)	position	-2, -1, 1
	distance	0, 1, 2, …
	# of VP, NP, SBAR	0, 1, 2, …
	# of POS [CC], [,], [:]	0, 1, 2, …
	POS [“] & POS [”]	-1, 0, 1
	path	VP-PP-NP, …
predicate itself & context	headword, headword’s POS, chunk type beginning word’s POS context-1: headword, headword’s POS, chunk type	MD, TO, VBZ, …
candidate itself & context	headword, headword’s POS, chunk type context-2: headword, headword’s POS, chunk type context-1: headword, headword’s POS, chunk type context+1: headword, headword’s POS, chunk type	

Table 1: Features for Identifying a semantic argument

Under	the existing	contract	,	Rockwell said	,	it	has already deliverd	793	of	the shipsets	to	Boeing
C	C	C	C	C	C	C	P	C	C	C	C	C
B-ARG	I-ARG	I-ARG	O	O	O	B-ARG	P	B-ARG	I-ARG	I-ARG	O	B-ARG
ARG						ARG	P	ARG				ARG
AM-LOC						A0	P	A1				A2

Figure 2: Two-phase semantic role labeling procedure using the example sentence presented in Figure 1. (P means the target phrase containing the predicate *deliver*, and C means the constituent such as a chunk (e.g. *Under*) or a subclause (e.g. *Rockwell said*))

(-2) means that the constituent is out of the immediate clause.

- **distance**: The distance is measured by the number of chunks between the predicate and the constituent.
- **# of VP, NP, SBAR**: These are numeric features representing the number of the specific chunk types between the predicate and the constituent.
- **# of POS [CC], [,], [:]**: These are numeric features representing the number of the specific POS types between the predicate and the constituent.
- **POS [“] & POS [”]**: This is used as a feature representing the difference between # of POS[“] and # of POS[”] counted in the range from the predicate to the constituent. In Table 1, the feature value (-1) means that # of POS[”] is larger than # of POS[“]. The feature value (1) conversly means that # of POS[“] is larger than # of POS[”]. The featue value (0) means that # of POS[“] is equal to # of POS[”].
- **path**: This is the syntactic path from the predicate to the constituent, and is a symbolic feature comprising all the elements (chunk or subclause) between the predicate and the constituent.
- **beginning word’s POS**: In the target phrase, these values appear only with VPs and represent the POS of the syntactic head (*MD, TO, VB, VBD, VBG, VBN, VBP, VBZ*). This represents the property of the target phrase, for example, the feature value *TO* indicates that the target phrase is to-infinitive.
- **context**: These are information for the predicate itself, the left context of the predicate, the constituent itself, and the left and right context of the constituent. In Table 1, - means the left context, and + means the right context. In case that the constituent is the subclause, the chunk type of the constituent is set to the first chunk type of the subclause.

2.2 Semantic Role Assignment

In this phase, we assign appropriate semantic roles to the identified semantic arguments. For learning SVM classifiers, we consider not all semantic roles, but only 18 semantic roles based on frequency in the training data (Table 2). The (*AM-MOD, AM-NEG*) are post-processed by hand-crafted rules. As we decrease the number of SVM classifiers to be learned in the training data, the training cost of classifiers can be reduced. Furthermore, we can alleviate the unbalanced class distribution problem by ex-

semantic role
A0, A1, A2, A3, A4, R-A0, R-A1, R-A2, C-A1
AM-TMP, AM-ADV, AM-MNR, AM-LOC, AM-DIS
AM-PNC, AM-CAU, AM-DIR, AM-EXT

Table 2: 18 semantic roles

cluding the infrequent classes.

2.2.1 Features for Assigning Semantic Role

This phase also uses all features applied in the semantic argument identification phase, except for # of POS [:] and POS["] & POS["]. In addition, we use the following feature.

- **voice**: This is a binary feature identifying whether the target phrase is active or passive.

In Figure 2, we show two-phase semantic role labeling procedure using the example sentence in Figure 1.

3 Experiments

For experiments, we utilized the SVM light package (T. Joachims, 2002). In both the semantic argument identification and the semantic role assignment phase, we used a polynomial kernel (*degree 2*) with the one-vs-rest classification method. Table 3 shows the experimental results on the test set and Table 4 shows the experimental results on the development set. Table 4 also shows the performance of each phase.

For improving the performance, we try to select the discrminative features for each subtask. Especially, since the performance of the identification phase is critical to the total performance, we concentrate on improving the identification performance. Our system obtains a F-measure of 74.08 in the identification phase, as presenteded in Table 4. For the argument classification task, the our system obtains a classification accuracy (A) of 85.45.

4 Conclusion

In this paper, we present a method of two phase semantic role labeling based on the support vector machines. We found that SVM is useful to incorporate the heterogeneous features for the semantic role labeling. Also, by applying the two phase model, we can alleviate the unbalanced class distribution problem caused by the the negative examples. Experimental results show that our system obtains a F-measure of 63.99 on the test set and 65.78 on the development set.

	Precision	Recall	$F_{\beta=1}$
Overall	65.63%	62.43%	63.99
A0	78.24%	74.60%	76.38
A1	65.83%	66.46%	66.14
A2	49.84%	43.70%	46.57
A3	56.04%	34.00%	42.32
A4	62.86%	44.00%	51.76
A5	0.00%	0.00%	0.00
AM-ADV	45.18%	44.30%	44.74
AM-CAU	36.67%	22.45%	27.85
AM-DIR	20.00%	20.00%	20.00
AM-DIS	56.62%	58.22%	57.41
AM-EXT	61.54%	57.14%	59.26
AM-LOC	26.01%	31.14%	28.34
AM-MNR	43.54%	35.69%	39.22
AM-MOD	97.46%	91.10%	94.17
AM-NEG	94.92%	88.19%	91.43
AM-PNC	40.00%	28.24%	33.10
AM-PRD	0.00%	0.00%	0.00
AM-TMP	51.83%	45.38%	48.39
R-A0	80.49%	83.02%	81.73
R-A1	75.00%	51.43%	61.02
R-A2	100.00%	33.33%	50.00
R-A3	0.00%	0.00%	0.00
R-AM-LOC	0.00%	0.00%	0.00
R-AM-MNR	0.00%	0.00%	0.00
R-AM-PNC	0.00%	0.00%	0.00
R-AM-TMP	0.00%	0.00%	0.00
V	96.66%	96.66%	96.66

Table 3: Results on the test set: closed challenge

	Precision	Recall	$F_{\beta=1}$	A
Overall	67.27%	64.36%	65.78	-
identification	75.96%	72.30%	74.08	-
classification	-	-	-	85.45

Table 4: Results on the development set: closed challenge. (A means accuracy.)

References

X. Carreras and L. Marquez. 2004. *Introduction to the CoNLL-2004 Shared Task: Semantic Role Labeling*. CoNLL.

H. Yamada and Y. Matsumoto. 2003. *Statistical Dependency Analysis with Support Vector Machines*. IWPT03.

S. Buchholz 2002. *Memory-Based Grammatical Relation Finding*. PhD. thesis, Tilburg University.

T. Joachims 2002. *SVM Light available at http:// svmlight.joachims.org/*.