Semantic Role Labeling of NomBank: A Maximum Entropy Approach

Zheng Ping Jiang and **Hwee Tou Ng**
Department of Computer Science
National University of Singapore
3 Science Drive 2, Singapore 117543
{jiangzp, nght}@comp.nus.edu.sg

Abstract

This paper describes our attempt at NomBank-based automatic Semantic Role Labeling (SRL). NomBank is a project at New York University to annotate the argument structures for common nouns in the Penn Treebank II corpus. We treat the NomBank SRL task as a classification problem and explore the possibility of adapting features previously shown useful in PropBank-based SRL systems. Various NomBank-specific features are explored. On test section 23, our best system achieves F1 score of 72.73 (69.14) when correct (automatic) syntactic parse trees are used. To our knowledge, this is the first reported automatic NomBank SRL system.

1 Introduction

Automatic Semantic Role Labeling (SRL) systems, made possible by the availability of PropBank (Kingsbury and Palmer, 2003; Palmer et al., 2005), and encouraged by evaluation efforts in (Carreras and Marquez, 2005; Litkowski, 2004), have been shown to accurately determine the argument structure of verb predicates.

A successful PropBank-based SRL system would correctly determine that "Ben Bernanke" is the subject (labeled as ARG0 in PropBank) of predicate "replace", and "Greenspan" is the object (labeled as ARG1):

- Ben Bernanke replaced Greenspan as Fed chair.
- Greenspan was replaced by Ben Bernanke as Fed chair.

The recent release of NomBank (Meyers et al., 2004c; Meyers et al., 2004b), a databank that annotates argument structure for instances of common nouns in the Penn Treebank II corpus, made it possible to develop automatic SRL systems that analyze the argument structures of noun predicates.

Given the following two noun phrases and one sentence, a successful NomBank-based SRL system should label "Ben Bernanke" as the subject (ARG0) and "Greenspan" as the object (ARG1) of the noun predicate "replacement".

- Greenspan's replacement Ben Bernanke
- Ben Bernanke's replacement of Greenspan
- Ben Bernanke was nominated as Greenspan's replacement.

The ability to automatically analyze the argument structures of verb and noun predicates would greatly facilitate NLP tasks like question answering, information extraction, etc.

This paper focuses on our efforts at building an accurate automatic NomBank-based SRL system. We study how techniques used in building PropBank SRL system can be transferred to developing NomBank SRL system. We also make NomBank-specific enhancements to our baseline system. Our implemented SRL system and experiments are based on the September 2005 release of NomBank (NomBank.0.8).

The rest of this paper is organized as follows: Section 2 gives an overview of NomBank, Section 3 introduces the Maximum Entropy classification model, Section 4 introduces our features and feature selection strategy, Section 5 explains the experimental setup and presents the experimental results, Section 6 compares NomBank SRL to

Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing (EMNLP 2006), pages 138–145,
Sydney, July 2006.

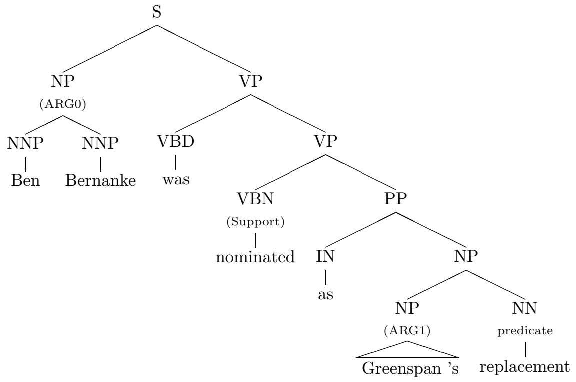

Figure 1: A sample sentence and its parse tree labeled in the style of NomBank

PropBank SRL and discusses possible future research directions.

2 Overview of NomBank

The NomBank (Meyers et al., 2004c; Meyers et al., 2004b) annotation project originated from the NOMLEX (Macleod et al., 1997; Macleod et al., 1998) nominalization lexicon developed under the New York University Proteus Project. NOMLEX lists 1,000 nominalizations and the correspondences between their arguments and the arguments of their verb counterparts. NomBank frames combine various lexical resources (Meyers et al., 2004a), including an extended NOMLEX and PropBank frames, and form the basis for annotating the argument structures of common nouns.

Similar to PropBank, NomBank annotation is made on the Penn TreeBank II (PTB II) corpus. For each common noun in PTB II that takes arguments, its core arguments are labeled with ARG0, ARG1, etc, and modifying arguments are labeled with ARGM-LOC to denote location, ARGM-MNR to denote manner, etc. Annotations are made on PTB II parse tree nodes, and argument boundaries align with the span of parse tree nodes.

A sample sentence and its parse tree labeled in the style of NomBank is shown in Figure 1. For the nominal predicate "replacement", "Ben Bernanke" is labeled as ARG0 and "Greenspan 's" is labeled as ARG1. There is also the special label "Support" on "nominated" which introduces "Ben Bernanke" as an argument of "replacement". The support construct will be explained in detail in Section 4.2.3.

We are not aware of any NomBank-based automatic SRL systems. The work in (Pradhan et al., 2004) experimented with an automatic SRL system developed using a relatively small set of manually selected nominalizations from FrameNet and Penn Chinese TreeBank. The SRL accuracy of their system is not directly comparable to ours.

3 Model training and testing

We treat the NomBank-based SRL task as a classification problem and divide it into two phases: argument identification and argument classification. During the argument identification phase, each parse tree node is marked as either argument or non-argument. Each node marked as argument is then labeled with a specific class during the argument classification phase. The identification model is a binary classifier , while the classification model is a multi-class classifier.

Opennlp maxent[1], an implementation of Maximum Entropy (ME) modeling, is used as the classification tool. Since its introduction to the Natural Language Processing (NLP) community (Berger et al., 1996), ME-based classifiers have been shown to be effective in various NLP tasks. ME modeling is based on the insight that the best model is consistent with the set of constraints imposed and otherwise as uniform as possible. ME models the probability of label l given input x as in Equation 1. $f_i(l, x)$ is a feature function that maps label l and input x to either 0 or 1, while the summation is over all n feature functions and with λ_i as the weight parameter for each feature function $f_i(l, x)$. Z_x is a normalization factor. In the identification model, label l corresponds to either "argument" or "non-argument", and in the classification model, label l corresponds to one of the specific NomBank argument classes. The classification output is the label l with the highest conditional probability $p(l|x)$.

$$p(l|x) = \frac{exp(\sum_{i=1}^{n} \lambda_i f_i(l, x))}{Z_x} \qquad (1)$$

To train the ME-based identification model, training data is gathered by treating each parse tree node that is an argument as a positive example and the rest as negative examples. Classification training data is generated from argument nodes only.

During testing, the algorithm of enforcing non-overlapping arguments by (Toutanova et al., 2005) is used. The algorithm maximizes the log-probability of the entire NomBank labeled parse

[1] http://maxent.sourceforge.net/

tree. Specifically, assuming we only have two classes "ARG" and "NONE", the log-probability of a NomBank labeled parse tree is defined by Equation 2.

$$Max(T) = max \begin{cases} NONE(T) + \sum(Max(child)) \\ ARG(T) + \sum(NONETree(child)) \end{cases} \tag{2}$$

$Max(T)$ is the maximum log-probability of a tree T, $NONE(T)$ and $ARG(T)$ are respectively the log-probability of assigning label "NONE" and "ARG" by our argument identification model to tree node T, $child$ ranges through each of T's children, and $NONETree(child)$ is the log-probability of each node that is dominated by node $child$ being labeled as "NONE". Details are presented in Algorithm 1.

Algorithm 1 Maximizing the probability of an SRL tree

Input p{syntactic parse tree}
Input m{argument identification model, assigns each constituent in the parse tree log likelihood of being a semantic argument}
Output score{maximum log likelihood of the parse tree p with arguments identified using model m}

MLParse(p, m)
if parse p is a leaf node **then**
  return $max(Score(p, m, ARG), Score(p, m, NONE))$
else
  $MLscore = 0$
  for each node c_i in $Children(p)$ **do**
    $MLscore$ += $MLParse(c_i, m)$
  end for
  $NONEscore = 0$
  for each node c_i in $Children(p)$ **do**
    $NONEscore$ += $NONETree(c_i, m)$
  end for
  return $max(Score(p, m, NONE)$+$MLscore$, $Score(p, m, ARG)$+$NONEscore)$
end if

NONETree(p,m)
$NONEscore = Score(p, m, NONE)$
if parse p is a leaf node **then**
  return $NONEscore$
else
  for each node c_i in $Children(p)$ **do**
    $NONEscore$ += $NONETree(c_i, m)$
  end for
  return $NONEscore$
end if

Subroutine:
$Children(p)$ returns the list of children nodes of p.
$Score(p, m, state)$ returns the log likelihood assigned by model m, for parse p with $state$. $state$ is either ARG or NONE.

NomBank sections 02-21 are used as training data, section 24 and 23 are used as development and test data, respectively.

3.1 Training data preprocessing

Unlike PropBank annotation which does not contain overlapping arguments (in the form of parse tree nodes domination) and does not allow predicates to be dominated by arguments, NomBank annotation in the September 2005 release contains such cases. In NomBank sections 02-21, about 0.6% of the argument nodes dominate some other argument nodes or the predicate. To simplify our task, during training example generation, we ignore arguments that dominate the predicate. We also ignore arguments that are dominated by other arguments, so that when argument domination occurs, only the argument with the largest word span is kept. We do *not* perform similar pruning on the test data.

4 Features and feature selection

4.1 Baseline NomBank SRL features

Table 1 lists the baseline features we adapted from previous PropBank-based SRL systems (Pradhan et al., 2005; Xue and Palmer, 2004). For ease of description, related features are grouped, with a specific individual feature given individual reference name. For example, feature b11FW in the group b11 denotes the first word spanned by the constituent and b13LH denotes the left sister's head word. We also experimented with various feature combinations, inspired by the features used in (Xue and Palmer, 2004). These are listed as features b31 to b34 in Table 1.

Suppose the current constituent under identification or classification is "NP-Ben Bernanke" in Figure 1. The instantiations of the baseline features in Table 1 for this example are presented in Table 2. The symbol "NULL" is used to denote features that fail to instantiate.

4.2 NomBank-specific features

4.2.1 NomBank predicate morphology and class

The "NomBank-morph" dictionary provided by the current NomBank release maps the base form of a noun to various morphological forms. Besides singular-plural noun form mapping, it also maps base nouns to hyphenated and compound nouns. For example, "healthcare" and "medical-care" both map to "care". For NomBank SRL fea-

Baseline Features (Pradhan et al., 2005)	
b1	predicate: stemmed noun
b2	subcat: grammar rule that expands the predicate's parent
b3	phrase type: syntactic category of the constituent
b4	head word: syntactic head of the constituent
b5	path: syntactic path from the constituent to the predicate
b6	position: to the left or right of the predicate
b11	first or last word/POS spanned by the constituent (b11FW, b11LW, b11FP, b11LP)
b12	phrase type of the left or right sister (b12L, b12R)
b13	left or right sister's head word/POS (b13LH, b13LP, b13RH, b13RP)
b14	phrase type of parent
b15	parent's head word or its POS (b15H, b15P)
b16	head word of the constituent if its parent has phrase type PP
b17	head word or POS tag of the rightmost NP node, if the constituent is PP (b17H, b17P)
b18	phrase type appended with the length of path
b19	temporal keyword, e.g., "Monday"
b20	partial path from the constituent to the lowest common ancestor with the predicate
b21	projected path from the constituent to the highest NP dominating the predicate
Baseline Combined Features (Xue and Palmer, 2004)	
b31	b1 & b3
b32	b1 & b4
b33	b1 & b5
b34	b1 & b6

Table 1: Baseline features for NomBank SRL

Baseline Features (Pradhan et al., 2005)	
b1	replacement
b2	NP $\rightarrow$ NP NN
b3	NP
b4	Bernanke
b5	NP↑S↓VP↓VP↓PP↓NP↓NN
b6	left
b11	Ben, Bernanke, NNP, NNP
b12	NULL, VP
b13	NULL, NULL, was, VBD
b14	S
b15	was, VBD
b16	NULL
b17	NULL, NULL
b18	NP-7
b19	NULL
b20	NP↑S
b21	NP↑S↓VP↓VP↓PP↓NP
Baseline Combined Features (Xue and Palmer, 2004)	
b31	replacement & NP
b32	replacement & Bernanke
b33	replacement & NP↑S↓VP↓VP↓PP↓NP↓NN
b34	replacement & left

Table 2: Baseline feature instantiations, assuming the current constituent is "NP-Ben Bernanke" in Figure 1.

Additional Features Based on NomBank	
a1	Nombank morphed noun stem
a2	Nombank nominal class
a3	identical to predicate?
a4	a DEFREL noun?
a5	whether under the noun phrase headed by the predicate
a6	whether the noun phrase headed by the predicate is dominated by a VP node or has neighboring VP nodes
a7	whether there is a verb between the constituent and the predicate
Additional Combined Features	
a11	a1 & a2
a12	a1 & a3
a13	a1 & a5
a14	a3 & a4
a15	a1 & a6
a16	a1 & a7
Additional Features of Neighboring Arguments	
n1	for each argument already classified, b3-b4-b5-b6-r, where r is the argument class, otherwise b3-b4-b5-b6
n2	backoff version of n1, b3-b6-r or b3-b6

Table 3: Additional NomBank-specific features for NomBank SRL

tures, we use this set of more specific mappings to replace the morphological mappings based on WordNet. Specifically, we replace feature b1 in Table 1 with feature a1 in Table 3.

The current NomBank release also contains the "NOMLEX-PLUS" dictionary, which contains the class of nominal predicates according to their origin and the roles they play. For example, "employment" originates from the verb "employ" and is classified as "VERB-NOM", while the nouns "employer" and "employee" are classified as "SUBJECT" and "OBJECT" respectively. Other classes include "ADJ-NOM" for nominalization of adjectives and "NOM-REL" for relational nouns. The class of a nominal predicate is very indicative of the role of its arguments. We would expect a "VERB-NOM" predicate to take both ARG0 and ARG1, while an "OBJECT" predicate to take only ARG0. We incorporated the class of nominal predicates as additional features in our NomBank SRL system. We add feature a2 in Table 3 to use this information.

4.2.2 DEFREL relational noun predicate

About 14% of the argument node instances in NomBank sections 02-21 are identical to their nominal predicate nodes. Most of these nominal predicates are DEFREL relational nouns (Meyers et al., 2004c). Examples of DEFREL relational nouns include "employee", "participant", and "husband", where the nominal predicate itself takes part as an implied argument.

We include in our classification features an indicator of whether the argument coincides with the nominal predicate. We also include a feature testing if the argument is one of the DEFREL nouns we extracted from NomBank training sections 02-21. These two features correspond to a3 and a4 in Table 3.

4.2.3 Support verb

Statistics show that almost 60% of the arguments of nominal predicates occur locally inside the noun phrase headed by the nominal predicate. For the cases where an argument appears outside the local noun phrase, over half of these arguments are introduced by support verbs. Consider our example "Ben Bernanke was nominated as Greenspan's replacement.", the argument "Ben Bernanke" is introduced by the support verb "nominate". The arguments introduced by support verbs can appear syntactically distant from the nominal predicate.

To capture the location of arguments and the existence of support verbs, we add features indicating whether the argument is under the noun phrase headed by the predicate, whether the noun phrase headed by the predicate is dominated by a VP phrase or has neighboring VP phrases, and whether there is a verb between the argument and the predicate. These are represented as features a5, a6, and a7 in Table 3. Feature a7 was also proposed by the system in (Pradhan et al., 2004).

We also experimented with various feature combinations, inspired by the features used in (Xue and Palmer, 2004). These are listed as features a11 to a16 in Table 3.

4.2.4 Neighboring arguments

The research of (Jiang et al., 2005; Toutanova et al., 2005) has shown the importance of capturing information of the global argument frame in order to correctly classify the local argument.

We make use of the features {b3,b4,b5,b6} of the neighboring arguments as defined in Table 1. Arguments are classified from left to right in the textual order they appear. For arguments that are already labeled, we also add their argument class r. Specifically, for each argument to the left of the current argument, we have a feature b3-b4-b5-b6-r. For each argument to the right of the current argument, the feature is defined as b3-b4-b5-b6. We extract features in a window of size 7, centered at the current argument. We also add a backoff version (b3-b6-r or b3-b6) of this specific feature. These additional features are shown as n1 and n2 in Table 3.

Suppose the current constituent under identification or classification is "NP-Ben Bernanke". The instantiations of the additional features in Table 3 are listed in Table 4.

Additional Features based on NomBank	
a1	replacement
a2	VERB-NOM
a3	no
a4	no
a5	no
a6	yes
a7	yes
Additional Combined Features	
a11	replacement & VERB-NOM
a12	replacement & no
a13	replacement & no
a14	no & no
a15	replacement & yes
a16	replacement & yes
Additional Features of Neighboring Arguments	
n1	NP-Greenspan-NP↑NP↓NN-left
n2	NP-left

Table 4: Additional feature instantiations, assuming the current constituent is "NP-Ben Bernanke" in Figure 1.

4.3 Feature selection

Features used by our SRL system are automatically extracted from PTB II parse trees manually labeled in NomBank. Features from Table 1 and Table 3 are selected empirically and incrementally according to their contribution to test accuracy on the development section 24. The feature selection process stops when adding any of the remaining features fails to improve the SRL accuracy on development section 24. We start the selection process with the basic set of features {b1,b2,b3,b4,b5,b6}. The detailed feature selection algorithm is presented in Algorithm 2.

Features for argument identification and argument classification are independently selected. To select the features for argument classification, we assume that all arguments have been correctly identified.

After performing greedy feature selection, the baseline set of features selected for identification is {b1-b6, b11FW, b11LW, b12L, b13RH, b13RP, b14, b15H, b18, b20, b32-b34}, and the baseline

Algorithm 2 Greedy feature selection

Input $F_{candidate}$\{set of all candidate features\}
Output F_{select}\{set of selected features\}
Output M_{select}\{selected model\}

Initialize:
$F_{select} = \{b1, b2, b3, b4, b5, b6\}$
$F_{candidate} = AllFeatures - F_{select}$

$M_{select} = Train(F_{select})$
$E_{select} = Evaluate(M_{select}, DevData)$
loop
  for each feature f_i in $F_{candidate}$ **do**
    $F_i = F_{select} \bigcup f_i$
    $M_i = Train(F_i)$
    $E_i = Evaluate(M_i, DevData)$
  end for
  $E_{max} = Max(E_i)$
  if $E_{max} > E_{select}$ **then**
    $F_{select} = F_{select} \bigcup f_{max}$
    $M_{select} = M_{max}$
    $E_{select} = E_{max}$
    $F_{candidate} = F_{candidate} - f_{max}$
  end if
  if $F_{candidate} == \phi$ or $E_{max} \leq E_{select}$ **then**
    return F_{select}, M_{select}
  end if
end loop

Subroutine:
$Evaluate(Model, Data)$ returns the accuracy score by evaluating Model on Data.
$Train(FeatureSet)$ returns maxent model trained on the given feature set.

set of features selected for classification is {b1-b6, b11, b12, b13LH, b13LP, b13RP, b14, b15, b16, b17P, b20, b31-b34}. Note that features in {b19, b21} are not selected. For the additional features in Table 3, greedy feature selection chose {a1, a5, a6, a11, a12, a14} for the identification model and {a1, a3, a6, a11, a14, a16, n1, n2} for the classification model.

5 Experimental results

5.1 Scores on development section 24

After applying the feature selection algorithm in Section 4.3, the SRL F1 scores on development section 24 are presented in Table 5. We separately present the F1 score for identification-only and classification-only model. We also apply the classification model on the output of the identification phase (which may contain erroneously identified arguments in general) to obtain the combined accuracy. During the identification-only and combined identification and classification testing, the tree log-probability maximization algorithm based on Equation 2 (and its extension to multi-classes) is used. During the classification-only testing, we classify each correctly identified argument using the classification ME model. The "baseline" row lists the F1 scores when only the baseline features are used, and the "additional" row lists the F1 scores when additional features are added to the baseline features.

	identification	classification	combined
baseline	80.32	84.86	69.70
additional	80.55	87.31	70.12

Table 5: NomBank SRL F1 scores on development section 24, based on correct parse trees

	identification	classification	combined
baseline	82.33	85.85	72.20
additional	82.50	87.80	72.73

Table 6: NomBank SRL F1 scores on test section 23, based on correct parse trees

5.2 Testing on section 23

The identification and classification models based on the chosen features in Section 4.3 are then applied to test section 23. The resulting F1 scores are listed in Table 6. Using additional features, the identification-only, classification-only, and combined F1 scores are 82.50, 87.80, and 72.73, respectively.

Performing chi-square test at the level of significance 0.05, we found that the improvement of the classification model using additional features compared to using just the baseline features is statistically significant, while the corresponding improvements due to additional features for the identification model and the combined model are not statistically significant.

The improved classification accuracy due to the use of additional features does not contribute any significant improvement to the combined identification and classification SRL accuracy. This is due to the noisy arguments identified by the inadequate identification model, since the accurate determination of the additional features (such as those of neighboring arguments) depends critically on an accurate identification model.

5.3 Using automatic syntactic parse trees

So far we have assumed the availability of correct syntactic parse trees during model training and testing. We relax this assumption by using the re-ranking parser presented in (Charniak and

Johnson, 2005) to automatically generate the syntactic parse trees for both training and test data.

The F1 scores of our best NomBank SRL system, when applied to automatic syntactic parse trees, are 66.77 for development section 24 and 69.14 for test section 23. These F1 scores are for combined identification and classification, with the use of additional features. Comparing these scores with those in Table 5 and Table 6, the usage of automatic parse trees lowers the F1 accuracy by more than 3%. The decrease in accuracy is expected, due to the noise introduced by automatic syntactic parsing.

6 Discussion and future work

6.1 Comparison of the composition of PropBank and NomBank

Counting the number of annotated predicates, the size of the September 2005 release of NomBank (NomBank.0.8) is about 83% of PropBank release 1. Preliminary consistency tests reported in (Meyers et al., 2004c) shows that NomBank's inter-annotator agreement rate is about 85% for core arguments and lower for adjunct arguments. The inter-annotator agreement for PropBank reported in (Palmer et al., 2005) is above 0.9 in terms of the Kappa statistic (Sidney and Castellan Jr., 1988). While the two agreement measures are not directly comparable, the current NomBank.0.8 release documentation indicates that only 32 of the most frequently occurring nouns in PTB II have been adjudicated.

We believe the smaller size of NomBank.0.8 and the potential noise contained in the current release of the NomBank data may partly explain our lower SRL accuracy on NomBank, especially in the argument identification phase, as compared to the published accuracies of PropBank-based SRL systems.

6.2 Difficulties in NomBank SRL

The argument structure of nominalization phrases is less fixed (i.e., more flexible) than the argument structure of verbs. Consider again the example given in the introduction, we find the following flexibility in forming grammatical NomBank argument structures for "replacement":

- The positions of the arguments are flexible, so that "Greenspan's replacement Ben Bernanke", "Ben Bernanke's replacement of Greenspan" are both grammatical.
- Arguments can be optional, so that "Greenspan's replacement will assume the post soon", "The replacement Ben Bernanke will assume the post soon", and "The replacement will assume the post soon" are all grammatical.

With the verb predicate "replace", except for "Greenspan was replaced", there is no freedom of forming phrases like "Ben Bernanke replaces" or simply "replaces" without supplying the necessary arguments to complete the grammatical structure.

We believe the flexible argument structure of NomBank noun predicates contributes to the lower automatic SRL accuracy as compared to that of the PropBank SRL task.

6.3 Integrating PropBank and NomBank SRL

Work in (Pustejovsky et al., 2005) discussed the possibility of merging various Treebank annotation efforts including PropBank, NomBank, and others. Future work involves studying ways of concurrently producing automatic PropBank and NomBank SRL, and improving the accuracy by exploiting the inter-relationship between verb predicate-argument and noun predicate-argument structures.

Besides the obvious correspondence between a verb and its nominalizations, e.g., "replace" and "replacement", there is also correspondence between verb predicates in PropBank and support verbs in NomBank. Statistics from NomBank sections 02-21 show that 86% of the support verbs in NomBank are also predicate verbs in PropBank. When they coincide, they share 18,250 arguments of which 63% have the same argument class in PropBank and NomBank.

Possible integration approaches include:

- Using PropBank data as augmentation to NomBank training data.
- Using re-ranking techniques (Collins, 2000) to jointly improve PropBank and NomBank SRL accuracy.

7 Conclusion

We have successfully developed a statistical NomBank-based SRL system. Features that were previously shown to be effective in PropBank SRL are carefully selected and adapted for NomBank SRL. We also proposed new features to address the special predicate-argument structure in NomBank data. To our knowledge, we presented the first result in statistical NomBank SRL.

References

Adam Berger, Stephen Della Pietra, and Vincent Della Pietra. 1996. A Maximum Entropy Approach to Natural Language Processing. *Computational Linguistics*.

Xavier Carreras and Lluis Marquez. 2005. Introduction to the CoNLL-2005 Shared Task: Semantic Role Labeling. In *Proceedings of CoNLL-2005*.

Eugene Charniak and Mark Johnson. 2005. Coarse-to-fine n-Best Parsing and MaxEnt Discriminative Reranking. In *Proceedings of ACL-2005*.

Michael Collins. 2000. Discriminative Reranking for Natural Language Parsing. In *Proceedings of ICML 2000*.

Zheng Ping Jiang, Jia Li, and Hwee Tou Ng. 2005. Semantic Argument Classification Exploiting Argument Interdependence. In *Proceedings of IJCAI 2005*.

Paul Kingsbury and Martha Palmer. 2003. PropBank: the Next Level of TreeBank. In *Proceedings of Treebanks and Lexical Theories*.

Kenneth C. Litkowski. 2004. SENSEVAL-3 Task: Automatic Labeling of Semantic Roles. In *Proceedings of Senseval-3: The Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*.

Catherine Macleod, Adam Meyers, Ralph Grishman, Leslie Barrett, and Ruth Reeves. 1997. Designing a Dictionary of Derived Nominals. In *Proceedings of Recent Advances in Natural Language Processing*.

Catherine Macleod, Ralph Grishman, Adam Meyers, Leslie Barrett, and Ruth Reeves. 1998. NOMLEX: A Lexicon of Nominalizations. In *Proceedings of EURALEX'98*.

Adam Meyers, Ruth Reeves, Catherine Macleod, Rachel Szekely, Veronika Zielinska, and Brian Young. 2004a. The Cross-Breeding of Dictionaries. In *Proceedings of LREC-2004*.

Adam Meyers, Ruth Reeves, Catherine Macleod, Rachel Szekely, Veronika Zielinska, Brian Young, and Ralph Grishman. 2004b. Annotating Noun Argument Structure for NomBank. In *Proceedings of LREC-2004*.

Adam Meyers, Ruth Reeves, Catherine Macleod, Rachel Szekely, Veronika Zielinska, Brian Young, and Ralph Grishman. 2004c. The NomBank Project: An Interim Report. In *Proceedings of HLT-NAACL 2004 Workshop on Frontiers in Corpus Annotation*.

Martha Palmer, Daniel Gildea, and Paul Kingsbury. 2005. The Proposition Bank: An Annotated Corpus of Semantic Roles. *Computational Linguistics*.

Sameer S. Pradhan, Honglin Sun, Wayne Ward, James H. Martin, and Dan Jurafsky. 2004. Parsing Arguments of Nominalizations in English and Chinese. In *Proceedings of HLT/NAACL 2004*.

Sameer Pradhan, Kadri Hacioglu, Valerie Krugler, Wayne Ward, James H. Martin, and Daniel Jurafsky. 2005. Support Vector Learning for Semantic Argument Classification. *Machine Learning*.

James Pustejovsky, Adam Meyers, Martha Palmer, and Massimo Poesio. 2005. Merging PropBank, NomBank, TimeBank, Penn Discourse Treebank and Coreference. In *ACL 2005 Workshop on Frontiers in Corpus Annotations II: Pie in the Sky*.

Siegel Sidney and N. John Castellan Jr. 1988. *Nonparametric Statistics for the Behavioral Sciences*. McGraw-Hill, New York.

Kristina Toutanova, Aria Haghighi, and Christopher D. Manning. 2005. Joint Learning Improves Semantic Role Labeling. In *Proceedings of ACL 2005*.

Nianwen Xue and Martha Palmer. 2004. Calibrating Features for Semantic Role Labeling. In *Proceedings of EMNLP-2004*.