Class-based nominal semantic role labeling: a preliminary investigation

Matt Gerber
Department of Computer Science
Michigan State University
gerberm2@msu.edu

Joyce Chai
Department of Computer Science
Michigan State University
jchai@cse.msu.edu

Abstract

This paper presents a preliminary investigation into the use of NomLex classes for NomBank semantic role labeling (SRL). We hypothesize that modeling each class individually will result in more homogeneous training data and better performance compared to a baseline approach that is not class-based. Our current experimental results, which are based on simple class models, do not show significant gains, but they do reveal interesting characteristics about different classes of nominalizations. We discuss implications of these results for future research in class-based NomBank SRL.

1 Introduction

In recent years, a significant number of studies have focused on the task of semantic role labeling (SRL) of verbal predicate-argument structure. Driven by annotation resources such as PropBank (Kingsbury and Palmer, 2003) and FrameNet (Baker et al., 1998), systems developed in these studies have achieved reasonable performances levels. For example, (Carreras and Marquez, 2005) report an overall f-measure score of 0.7944 for the system developed by (Punyakanok et al., 2005).

More recently, the development and release of the NomBank corpus (Meyers, 2007a) has inspired further investigations into semantic role labeling of nominal predicate-argument structure. NomBank annotates predicating nouns in the same way PropBank annotates predicating verbs. Consider the following example of the verbal predicate *distribute* from the PropBank corpus:

> Freeport-McMoRan Energy Partners will be liquidated and [Arg1 shares of the new company] [Predicate distributed] [Arg2 to the partnership's unitholders].

The NomBank corpus provides a similar attestation of the deverbal nominalization *distribution*, shown below:

> Erbamont will then be liquidated, with [Arg2 any remaining Erbamont holders] receiving a [Predicate distribution] [Arg1 of $37 a share].

PropBank and NomBank contain a semantic frame definition for each predicate in their respective lexicons. A semantic frame definition lists coarse-grained senses for each predicate as well as possible core argument (e.g., Arg0 - ArgN) roles for each sense. The creators of NomBank, to ensure some amount of coherence with the PropBank project, have adapted PropBank frames for use with deverbal nominalizations when possible. For example, in the PropBank frame definition file for *distribute*, Arg1 is interpreted as the item being distributed and Arg2 is the entity to which the distribution is being made. These interpretations are consistent with the interpretations of Arg1 and Arg2 in the NomBank frame definition for *distribution*.

In addition to deverbal (i.e., event-based) nominalizations, NomBank contains entries in its lexicon for a wide variety of nouns that are not derived from verbs and do not denote an event. An example is given below of the noun *percent*:

> Hallwood owns about 11 [Predicate %] [Arg1 of Integra].

In this case, the noun phrase headed by the predicate *%* (i.e., "about 11% of Integra") denotes a fractional part of the argument in position Arg1 (i.e., the entity named by "Integra").

Following the design of PropBank, NomBank permits nouns to take, in addition to core arguments, an unspecified number of adjunct arguments, which further modify the predication. Adjunct arguments may apply to all entries in the lexicon and are not listed in the semantic frame definitions. The following example illustrates the use of the *location* adjunct:

> [Arg0 Cetus] is currently trying to obtain federal regulatory clearance for [Location U.S.] [Predicate distribution].

Our current investigation extends the work of (Jiang and Ng, 2006) and (Liu and Ng, 2007) in automatically identifying NomBank argument structures such as those above. Specifically, we explore the use of NomLex (Macleod et al., 1998) for the partitioning of nominalizations into classes. Members of the same NomLex class, being semantically related, tend to exhibit similar syntactic and semantic realizations of their respective arguments. Conversely, members of different NomLex classes tend to exhibit different syntactic and semantic realizations of their arguments. (Levin, 1993) describes an analogous process at work in verbal predication, where diathesis alternations are constrained by the presence of various semantic components. Because nominalization classes tend to be homogeneous, we hypothesize that a per-class modeling of nominalizations will result in more accurate identification of NomBank semantic roles.

We investigate our hypothesis as follows: in section 2, we discuss previous work in the area of nominalization interpretation. In section 3, we describe a typical approach to NomBank SRL that is used as a baseline in the present study. Sections 4 and 5 discuss NomLex and its role in our NomBank SRL system. Section 6 gives evaluation results and discusses their implications, and section 7 finishes with conclusions and some directions for future work.

2 Related work

Early work in identifying the argument structure of deverbal nominalizations was primarily rule-based. (Dahl et al., 1987), (Hull and Gomez, 1996), and (Meyers et al., 1998) employ rule sets that associate syntactic constituents with semantic roles. (Lapata, 2000) developed a statistical model to classify modifiers of deverbal nouns as underlying subjects or underlying objects, where subject and object denote the grammatical function of the modifier when linked to a verb. Consider two possible interpretations of the phrase "satellite observation" below:

1. The military uses [subject satellite] observation to keep track of enemy troop movements.
2. The stargazers routinely engaged in [object satellite] observation.

In the first, it is the satellites that are being used for observation, while in the second the satellites are being observed.

More recently, the development of the NomBank corpus has supported the work of (Jiang and Ng, 2006) and (Liu and Ng, 2007). Each study tested the hypothesis that machine learning methodologies and representations used in verbal SRL (cf. (Pradhan et al., 2005)) can be ported to the task of NomBank SRL. (Liu and Ng, 2007) reports an overall f-measure score of 0.7283 using automatically generated parse trees, demonstrating that modest performance can be achieved using basic approaches developed in the verbal domain. Both studies also investigated the use of features specific to the task of nominal SRL, but observed only marginal performance gains.

The nominal SRL task of the present study is related to nominal relation interpretation as evaluated in SemEval (Girju et al., 2007). Both tasks identify semantic relations between a head noun and other constituents; however, the tasks focus on different relations. NomBank SRL focuses primarily on relations that hold between deverbal nominalizations and their arguments, whereas SemEval focuses on a range of relations, most of which are not applicable to deverbal nominalizations.

3 NomBank SRL

(Jiang and Ng, 2006) and (Liu and Ng, 2007) exemplify a typical approach to SRL in general and NomBank SRL in particular. Given a predicating nominalization, the goal is to assign surrounding syntactic constituents to one of 23 classes representing core arguments, adjunct arguments, and the *null* or non-argument. Similarly to PropBank SRL (cf. (Pradhan et al., 2005)), this task can be treated as a multi-class classification problem over parse tree nodes. To arrive at the final classification, systems often employ a two-stage approach in which "argument identification" is followed by "argument classification". The identification stage assigns spans of surface text a binary label indicating whether the text is an argument or non-argument. Subsequently, the spans of text identified as arguments are reassigned a label corresponding to the 22 core and adjunct argument types mentioned above, giving the final role labeling.

As with verbal SRL, previous work in NomBank SRL has typically used features derived from a full syntactic parse in addition to shallower, word-based features. Table 1 describes common features from previous work, which we also use in the present study.

4 NomLex

The NomBank distribution includes NomLex-PLUS (Meyers, 2007b), which is the result of automatically expanding the hand-coded NomLex resource (Macleod et al., 1998). NomLex-PLUS (hereinafter referred to simply as "NomLex") encodes rules of association between syntactic constituents and underlying grammatical functions. Additionally, NomLex partitions nominalizations into classes, including *Nom* for deverbal nominalizations such as *distribution* and *Partitive* for nouns such as *%* (both exemplified in section 1). In total, NomLex defines 22 classes, which are summarized in Table 2.

As can be seen, the distribution of entries in NomLex is heavily skewed towards the *Nom* class, which contains deverbal nominalizations. The last column of the table is important because it describes the level of observed ambiguity within the classes. An instances of a nominalization is ambiguous if it falls into multiple NomLex classes. As we will show, classes with a high ambiguity (e.g., *Nom-like*) are problematic for our class-based approach, which is described next.

5 Class-based nominal SRL

As mentioned above, we are interested in the effects of clustering nominalizations by NomLex class membership. Specifically, we expect to observe better NomBank SRL performance by modeling each class independently. To test our hypothesis, we train a separate 23-class logistic regression model[1] for each NomLex class using only unambiguous instances. Each class-specific model uses the baseline feature set (Table 1) and makes argument predictions with a single classification. Empirically, we have found that the single-stage approach outperforms the traditional two-stage approach described in section 3.

5.1 Heuristic post-processing

Our class-based SRL system uses a heuristic post-processing step to produce the final labeling. Two issues are resolved in this step, (1) the labeling of predications and (2) the enforcement of global constraints.

5.1.1 Predicate labeling

NomBank differs from PropBank in that NomBank predicates themselves often assume argument roles through the process of incorporation. This is usually the case when the nominalization is derived from a verb by -er or -ee suffixation, as in the following example:

> Petrolane is the second-largest [Arg1 propane] [Arg0/Predicate distributor] [Modifier-Location in the U.S.].

Most of the features listed in Table 1 have the same value for all NomBank predicates. Thus, instead of applying the logistic regression model to predicate nodes, we label each predicate node with the most likely argument label observed in the training data.

[1] We use the BXR logistic regression software, which is available at http://code.google.com/p/bxr-bayesian-regression

Feature	Description
1	Syntactic category of n.
2	First/last word and part of speech (POS) subsumed by n.
3	Head word of n. In cases where the head of n is not a leaf node, head children of n are traversed until a head leaf is reached.
4	Head word of n, if the parent of n is a PP.
5	The syntactic category, head word, and head POS of n's left and right siblings.
6	The syntactic category, head word, and head POS of n's parent.
7	The head word and head POS of the right-most NP if n is a PP.
8	Parse tree path from n to $pred$, as described in (Gildea and Jurafsky, 2002).
9	Parse tree path from n to the lowest common ancestor of n and $pred$.
10	Stem of $pred$. When the NomBank morphology dictionary (Meyers, 2007b) contains a single entry for $pred$, the stem from that entry is used; otherwise, Porter-stemming is applied.
11	Production rule that expands the parent of $pred$.
12	Surface distance (in tokens) from the span of text subsumed by n to the token subsmed by $pred$. This is positive when n follows $pred$, and negative when n precedes $pred$.
13	Concatenation of 1 and the length of 8, where length is equal to the number of edges along the path.
14	Concatenation of 10 and 1.
15	Concatenation of 10 and 3.
16	Concatenation of 10 and 8.

Table 1: Baseline feature set, where n is the training instance or constituent being classified, and $pred$ is the predicating constituent.

5.1.2 Global constraints

Our system makes argument predictions without taking labels for other nodes into account. As a result, argument labels sometimes violate global labeling constraints, which are enforced using the following heuristics.

No overlapping arguments Overlapping arguments arise when two nodes are labeled as arguments and one node is an ancestor of the other, as shown in Figure 1(a). If each node has the same label we re-score the ancestor node with the average of the two nodes' confidence scores. The descendant node is then reassigned to the *null* class. If the two nodes have different labels, the node with the higher confidence is kept and the other is reassigned to the *null* class. Null re-assignments are made with confidence equal to 1.0.

No duplicate arguments Duplicate arguments

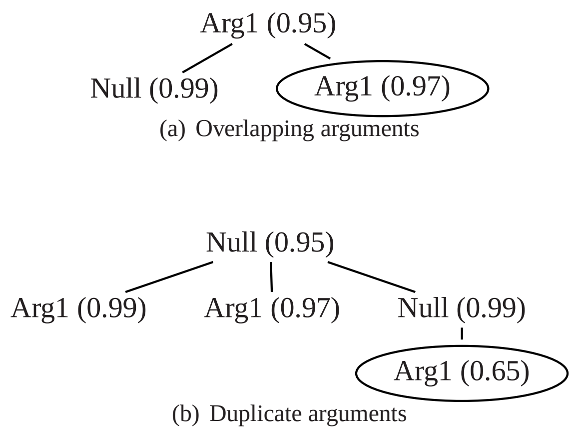

Figure 1: Global constraint violations. Circled nodes are reassigned to the *null* class.

NomLex class	Distinct nominalizations	Frequency in corpus	% ambiguous in corpus
Nom	3933	61797	32.05
Nom-like	1142	27397	68.96
Partitive	509	14156	55.99
Nom-adj	502	3993	50.66
Attribute	416	12729	70.02
Nom-ing	359	4125	7.49
Relational	331	8944	36.79
Work-of-art	187	4709	79.65
Nom-adj-like	137	3287	90.56
Ability	112	7066	67.61
Environment	91	3735	60.40
Group	84	4278	77.20
Hallmark	38	320	80.31
Job	28	1788	85.79
Version	21	889	88.41
Able-nom	18	70	4.28
Type	17	1185	84.47
Event	12	181	0.00
Share	12	3832	98.53
Issue	11	1245	99.91
Criss-cross	7	400	1.00
Field	6	394	96.70
All classes	7973	166520	52.61

Table 2: Distribution of NomLex classes. The second column gives the number of distinct nominalizations per class and the third column gives the frequency of the class within the NomBank corpus. The last column gives the percentage of corpus occurrences that are ambiguous (i.e., the occurrence falls into multiple classes)

arise when two nodes are assigned the same label and one is not an ancestor of the other, as shown in Figure 1(b). If the two nodes are not siblings, the node with the higher confidence score is kept and the other is reassigned to the *null* class. If the nodes are siblings, both are kept. Keeping both sibling nodes accounts for so-called “split” arguments, exemplified below:

> In [Predicate addition], [Arg1 the apple II] [Arg1 was an affordable $1,298].

In this case, no single parse tree node subsumes the Arg1 exactly.

These global constraints replace the tree-likelihood maximization algorithm introduced by (Toutanova et al., 2005), which was also used by (Jiang and Ng, 2006) but not by (Liu and Ng, 2007). As a final step, we reassign a node to the *null* class if it has a confidence score below a threshold t, which is tuned using the development data (WSJ section 24).

When a test nominalization is presented to the system, the NomLex lexicon is consulted to determine the nominalization’s class. For nominalizations that fall into a single class, the corresponding model is applied. A backoff model trained over all classes is applied to nominalizations that fall into multiple classes or are not a member of any class. The backoff model uses the same feature set and post-processing steps used in the class-based models.

6 Preliminary results and analysis

Following standard practice, we draw training nodes from sections 2-21 of the NomBank corpus. We do not train over gold-standard parse trees; rather, all trees are automatically generated using Charniak's syntactic parser.[2] We use all parse tree nodes except those that are ancestors of the predicating node.[3]

We have evaluated the backoff and class-based systems over section 23 of NomBank following the methodology of (Jiang and Ng, 2006) and (Liu and Ng, 2007). Test arguments are spans of text, and a predicted argument node must cover precisely the span of text occupied by the test argument in order to be correct. As with the model training, all features for test instances were extracted from parse trees generated by Charniak's syntactic parser. Table 3 shows per-class and overall results for the backoff and class-based methods. The first set of rows gives evaluation results using unambiguous nominalizations from each class. The last two rows give results from evaluating over all unambiguous nominalizations and all nominalizations, respectively.

A number of observations can be made from from the results in Table 3, and we break them down as follows.

General observations Overall, the class-based approach currently demonstrates negligible gains in comparison to the results reported in (Liu and Ng, 2007). Our results are dominated by the *Nom* class, whose test set is more than five times larger than the second-largest test set. Furthermore, the *Nom* class demonstrates relatively poor performance compared to some of the other large classes. We believe this result is due to the heterogeneous nature of the *Nom* class, which contains all nominalizations that are morphologically related to verbs. Deverbal nominalizations are quite diverse in terms of argument structure, resulting in training data that is difficult to model accurately.

Intra-class regularity As can be seen, some classes perform significantly better than others. As we hypothesized, classes with high f-measure scores typically exhibit a prominent regularity in the realization of their arguments. For example, nominalizations from the *Relational* class ($F1 = 90.94$) have a high degree of incorporation (discussed in section 5.1.1). In this class, 100% of testing nominalizations have an incorporated Arg0, making up 38% of the test arguments for the class. Because of this regularity, the system was able to identify 38% of the *Relational* testing arguments with f-measure equal to 1.0, bringing the overall f-measure up.

We also observe a notable regularity within the *Partitive* class ($F1 = 79.85$), whose members take a single Arg1 86% of the time in the testing set. Compare this to the *Nom* class, whose members take the most common argument (again Arg1) by itself only 15% of the time.

Class-based gains Some classes show relatively small performance gains under the class-based approach. One cause of this result might be that the features used (i.e., those listed in Table 1) are too general to capture class-specific regularities. A key part of our future work in class-based NomBank SRL will be the identification of these regularities and the creation of features that account for them.

Table 3 also shows significant performance losses for the class-based approach on some classes. The most extreme cases of loss (Share and Nom-adj-like) appear to correlate with high class ambiguity (shown in Table 2). As described in section 5, ambiguous nominalizations are not used when training the model for each class. *Share*'s 98% ambiguity implies that only 56 of the 3832 instances of *Share* nominalizations are used as training data, resulting in a model that performs poorly compared to the backoff.

7 Conclusions and future work

We have experimented with a class-based approach to NomBank SRL that takes advantage of the nominalization classes defined by the NomBank lexicon.

[2] Available at ftp://ftp.cs.brown.edu/pub/nlparser

[3] As reported in (Jiang and Ng, 2006), only 0.6% of argument nodes in the training sections overlap with other arguments or the predicating node, and eliminating these nodes simplifies the training and classification process.

Class	Members	Test instances	Backoff F1 (%)	Class-based F1 (%)	Change
Nom	1639	4220	71.17	72.21	1.04
Nom-like	537	798	69.63	71.69	2.06
Relational	219	768	89.38	90.14	0.76
Attribute	173	370	75.81	74.54	-1.27
Nom-ing	225	331	64.97	64.18	-0.79
Partitive	324	317	78.95	79.85	0.90
Nom-adj	312	184	76.42	73.91	-2.51
Ability	61	172	75.08	74.67	-0.41
Environment	51	108	70.43	74.40	3.97
Work-of-art	98	86	65.33	66.67	1.34
Share	2	42	84.93	64.10	-20.83
Group	45	40	76.32	82.19	5.87
Job	11	30	85.71	92.00	6.29
Nom-adj-like	61	28	80.00	74.07	-5.93
Version	8	19	56.25	60.00	3.75
Type	6	15	80.00	86.67	6.67
Hallmark	19	6	80.00	90.91	10.91
Event	11	2	00.00	66.67	66.67
Able-nom	12	1	00.00	00.00	0.00
Field	1	0	N/A	N/A	N/A
Issue	1	0	N/A	N/A	N/A
Criss-cross	0	0	N/A	N/A	N/A
All unambiguous nominalizations	3816	7537	73.55	74.37	0.82
All nominalizations	4704	10410	72.27	72.86	0.59

Table 3: Per-class evaluation results. "Class" denotes the NomLex class of nominalizations being evaluated, "Members" denotes the number of nominalizations exclusively in each class, and "Test instances" denotes the number of test arguments presented to the system. "Backoff F1" and "Class-based F1" denote the performance of the two systems over the presented test instances.

Partitioning nominalizations into classes often results in more homogeneous training data and better performance when compared to a baseline that is not class-based. The evaluation also shows that certain classes of nominalizations are much simpler to model than others due to the regularity with which they express their arguments.

In our future work, we will focus on the definition and evaluation of class-specific feature sets. As part of feature set development, we will be assessing the usefulness of NomLex argument rules, which associate syntactic constituents with grammatical functions.

Another issue that needs to be addressed is nominalization ambiguity with respect to NomLex classes. Instead of backing off to the baseline model in cases where multiple NomLex classes apply, a more reasonable approach would be to first disambiguate the nominalization's NomLex class and then apply a class-based model.

References

Collin Baker, Charles Fillmore, and John Lowe. 1998. The berkeley framenet project. In *Proceedings of the Thirty-Sixth Annual Meeting of the Association for Computational Linguistics*.

Xavier Carreras and Lluis Marquez. 2005. Introduction to the conll-2005 shared task: Semantic role labeling.

Deborah A. Dahl, Martha S. Palmer, and Rebecca J. Passonneau. 1987. Nominalizations in pundit. In *Proceedings of the 25th annual meeting on Association for Computational Linguistics*, pages 131–139, Morristown, NJ, USA. Association for Computational Linguistics.

Daniel Gildea and Daniel Jurafsky. 2002. Automatic labeling of semantic roles. *Computational Linguistics*, 28:245–288.

Roxana Girju, Preslav Nakov, Vivi Nastase, Stan Szpakowicz, Peter Turney, and Deniz Yuret. 2007. Semeval-2007 task 04: Classification of semantic relations between nominals. In *Proceedings of the 4th International Workshop on Semantic Evaluations*.

Richard Hull and Fernando Gomez. 1996. Semantic interpretation of nominalizations. In *Proceedings of AAAI*.

Z. Jiang and H. Ng. 2006. Semantic role labeling of nombank: A maximum entropy approach. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*.

Paul Kingsbury and Martha Palmer. 2003. Propbank: the next level of treebank. In *Proceedings of Treebanks and Lexical Theories*.

Maria Lapata. 2000. The automatic interpretation of nominalizations. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence*, pages 716–721. AAAI Press / The MIT Press.

Beth Levin. 1993. *English verb classes and alternations: A preliminary investigation*. Chicago University Press.

Chang Liu and Hwee Ng. 2007. Learning predictive structures for semantic role labeling of nombank. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*.

Catherine Macleod, Ralph Grishman, Adam Meyers, Leslie Barrett, and Ruth Reeves. 1998. Nomlex: A lexicon of nominalizations. In *Proceedings of the Eighth International Congress of the European Association for Lexicography*.

Adam Meyers, Catherine Macleod, Roman Yangarber, Ralph Grishman, Leslie Barrett, and Ruth Reeves. 1998. Using nomlex to produce nominalization patterns for information extraction. In *Proceedings of the COLING-ACL Workshop on the Computational Treatment of Nominals*.

Adam Meyers. 2007a. Annotation guidelines for nombank - noun argument structure for propbank. Technical report, New York University.

Adam Meyers. 2007b. Those other nombank dictionaries. Technical report, New York University.

Sameer Pradhan, Wayne Ward, and James H. Martin. 2005. Towards robust semantic role labeling. In *Association for Computational Linguistics*.

Vasin Punyakanok, Dan Roth, and Wen tau Yih. 2005. The necessity of syntactic parsing for semantic role labeling. In *International Joint Conference on Artificial Intelligence*.

Kristina Toutanova, Aria Haghighi, and Christopher D. Manning. 2005. Joint learning improves semantic role labeling. In *ACL '05: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 589–596, Morristown, NJ, USA. Association for Computational Linguistics.