The Role of Implicit Argumentation in Nominal SRL

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

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

Adam Meyers
Dept. of Computer Science
New York University
meyers@cs.nyu.edu

Abstract

Nominals frequently surface without overtly expressed arguments. In order to measure the potential benefit of nominal SRL for downstream processes, such nominals must be accounted for. In this paper, we show that a state-of-the-art nominal SRL system with an overall argument F_1 of 0.76 suffers a performance loss of more than 9% when nominals with implicit arguments are included in the evaluation. We then develop a system that takes implicit argumentation into account, improving overall performance by nearly 5%. Our results indicate that the degree of implicit argumentation varies widely across nominals, making automated detection of implicit argumentation an important step for nominal SRL.

1 Introduction

In the past few years, a number of studies have focused on verbal semantic role labeling (SRL). Driven by annotation resources such as FrameNet (Baker et al., 1998) and PropBank (Palmer et al., 2005), many systems developed in these studies have achieved argument F_1 scores near 80% in large-scale evaluations such as the one reported by Carreras and Màrquez (2005).

More recently, the automatic identification of nominal argument structure has received increased attention due to the release of the NomBank corpus (Meyers, 2007a). NomBank annotates predicating nouns in the same way that PropBank annotates predicating verbs. Consider the following example of the verbal predicate *distribute* from the PropBank corpus:

(1) 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 contains a similar instance of the deverbal nominalization *distribution*:

(2) Searle will give [$_{Arg0}$ pharmacists] [$_{Arg1}$ brochures] [$_{Arg1}$ on the use of prescription drugs] for [$_{Predicate}$ distribution] [$_{Location}$ in their stores].

This instance demonstrates the annotation of split arguments (Arg1) and modifying adjuncts (Location), which are also annotated in PropBank. In cases where a nominal has a verbal counterpart, the interpretation of argument positions Arg0-Arg5 is consistent between the two corpora.

In addition to deverbal (i.e., event-based) nominalizations, NomBank annotates a wide variety of nouns that are not derived from verbs and do not denote events. An example is given below of the partitive noun *percent*:

(3) 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.

Since NomBank's release, a number of studies have applied verbal SRL techniques to the task of nominal SRL. For example, Liu and Ng (2007) reported an argument F_1 of 0.7283. Although this result is encouraging, it does not take into account nominals that surface without overt arguments. Consider the following example:

(4) The [$_{Predicate}$ distribution] represents [$_{NP}$ available cash flow] [$_{PP}$ from the partnership] [$_{PP}$ between Aug. 1 and Oct. 31].

As in (2), *distribution* in (4) has a noun phrase and multiple prepositional phrases in its environment, but not one of these constituents is an argument to *distribution* in (4); rather, any arguments are implicitly supplied by the surrounding discourse. As described by Meyers (2007a), instances such as (2) are called "markable" because they contain overt arguments, and instances such as (4) are called "unmarkable" because they do not. In the NomBank corpus, only markable instances have been annotated.

Previous evaluations (e.g., those by Jiang and Ng (2006) and Liu and Ng (2007)) have been based on markable instances, which constitute 57% of all instances of nominals from the NomBank lexicon. In order to use nominal SRL systems for downstream processing, it is important to develop and evaluate techniques that can handle markable as well as unmarkable nominal instances. To address this issue, we investigate the role of implicit argumentation for nominal SRL. This is, in part, inspired by the recent CoNLL Shared Task (Surdeanu et al., 2008), which was the first evaluation of syntactic and semantic dependency parsing to include unmarkable nominals. In this paper, we extend this task to constituent parsing with techniques and evaluations that focus specifically on implicit argumentation in nominals.

We first present our NomBank SRL system, which improves the best reported argument F_1 score in the markable-only evaluation from 0.7283 to 0.7630 using a single-stage classification approach. We show that this system, when applied to all nominal instances, achieves an argument F_1 score of only 0.6895, a loss of more than 9%. We then present a model of implicit argumentation that reduces this loss by 46%, resulting in an F_1 score of 0.7235 on the more complete evaluation task. In our analyses, we find that SRL performance varies widely among specific classes of nominals, suggesting interesting directions for future work.

2 Related work

Nominal SRL 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. Nominal SRL focuses primarily on relations that hold between nominalizations and their arguments, whereas the SemEval task focuses on a range of semantic relations, many of which are not applicable to nominal argument structure.

Early work in identifying the argument structure of deverbal nominalizations was primarily rule-based, using rule sets to associate syntactic constituents with semantic roles (Dahl et al., 1987; Hull and Gomez, 1996; Meyers et al., 1998). 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 position of the modifier when linked to a verb.

FrameNet and NomBank have facilitated machine learning approaches to nominal argument structure. Gildea and Jurafsky (2002) presented an early FrameNet-based SRL system that targeted both verbal and nominal predicates. Jiang and Ng (2006) and Liu and Ng (2007) have tested the hypothesis that methodologies and representations used in PropBank SRL (Pradhan et al., 2005) can be ported to the task of NomBank SRL. These studies report argument F_1 scores of 0.6914 and 0.7283, respectively. Both studies also investigated the use of features specific to the task of NomBank SRL, but observed only marginal performance gains.

NomBank argument structure has also been used in the recent CoNLL Shared Task on Joint Parsing of Syntactic and Semantic Dependencies (Surdeanu et al., 2008). In this task, systems were required to identify syntactic dependencies, verbal and nominal predicates, and semantic dependencies (i.e., arguments) for the predicates. For nominals, the best semantic F_1 score was 0.7664 (Surdeanu et al., 2008); however this score is not directly comparable to the NomBank SRL results of Liu and Ng (2007) or the results in this paper due to a focus on different aspects of the problem (see the end of section 5.2 for details).

3 NomBank SRL

Given a nominal predicate, an SRL system attempts to assign surrounding spans of text to one of 23 classes representing core arguments, adjunct arguments, and the *null* or non-argument. Similarly to

verbal SRL, this task is traditionally formulated as a two-stage classification problem over nodes in the syntactic parse tree of the sentence containing the predicate.[1] In the first stage, each parse tree node is assigned a binary label indicating whether or not it is an argument. In the second stage, argument nodes are assigned one of the 22 non-null argument types. Spans of text subsumed by labeled parse tree nodes constitute arguments of the predication.

3.1 An improved NomBank SRL baseline

To investigate the effects of implicit argumentation, we first developed a system based on previous markable-only approaches. Our system follows many of the traditions above, but differs in the following ways. First, we replace the standard two-stage pipeline with a single-stage logistic regression model[2] that predicts arguments directly. Second, we model incorporated arguments (i.e., predicates that are also arguments) with a simple maximum likelihood model that predicts the most likely argument label for a predicate based on counts from the training data. Third, we use the following heuristics to resolve argument conflicts: (1) If two arguments overlap, the one with the higher probability is kept. (2) If two non-overlapping arguments are of the same type, the one with the higher probability is kept unless the two nodes are siblings, in which case both are kept. Heuristic (2) accounts for split argument constructions.

Our NomBank SRL system uses features that are selected with a greedy forward search strategy similar to the one used by Jiang and Ng (2006). The top half of Table 2 (next page) lists the selected argument features.[3] We extracted training nodes from sections 2-21 of NomBank, used section 24 for development and section 23 for testing. All parse trees were generated by Charniak's re-ranking syntactic parser (Charniak and Johnson, 2005). Following the evaluation methodology used by Jiang and Ng (2006) and Liu and Ng (2007), we obtained sig-

	Dev. F_1	Testing F_1
Jiang and Ng (2006)	0.6677	0.6914
Liu and Ng (2007)	-	0.7283
This paper	0.7454	**0.7630**

Table 1: Markable-only NomBank SRL results for argument prediction using automatically generated parse trees. The f-measure statistics were calculated by aggregating predictions across all classes. "-" indicates that the result was not reported.

	Markable-only	All-token	% loss
P	0.7955	0.6577	-17.32
R	0.7330	0.7247	-1.13
F_1	0.7630	0.6895	-9.63

Table 3: Comparison of the markable-only and all-token evaluations of the baseline argument model.

nificantly better results, as shown in Table 1 above.[4]

3.2 The effect of implicit nominal arguments

The presence of implicit nominal arguments presents challenges that are not taken into account by the evaluation described above. To assess the impact of implicit arguments, we evaluated our Nom-Bank SRL system over each token in the testing section. The system attempts argument identification for all singular and plural nouns that have at least one annotated instance in the training portion of the NomBank corpus (morphological variations included).

Table 3 gives a comparison of the results from the markable-only and all-token evaluations. As can be seen, assuming that all known nouns take overt arguments results in a significant performance loss. This loss is due primarily to a drop in precision caused by false positive argument predictions made for nominals with implicit arguments.

4 Accounting for implicit arguments in nominal SRL

A natural solution to the problem described above is to first distinguish nominals that bear overt arguments from those that do not. We treat this

---

[1]The syntactic parse can be based on ground-truth annotation or derived automatically, depending on the evaluation.

[2]We use LibLinear (Fan et al., 2008).

[3]For features requiring the identification of support verbs, we use the annotations provided in NomBank. Preliminary experiments show a small loss when using automatic support verb identification.

[4]As noted by Carreras and Màrquez (2005), the discrepancy between the development and testing results is likely due to poorer syntactic parsing performance on the development section.

	#	Description	N	S
Argument features	1	12 **&** parse tree path from n to $pred$		
	2	Position of n relative to $pred$ **&** parse tree path from n to $pred$	*	
	3	First word subsumed by n		
	4	12 **&** position of n relative to $pred$		
	5	12 **&** 14		
	6	Head word of n's parent		*
	7	Last word subsumed n		
	8	n's syntactic category **&** length of parse tree path from n to $pred$		
	9	First word of n's right sibling	*	*
	10	Production rule that expands the parent of $pred$		
	11	Head word of the right-most NP in n if n is a PP	*	
	12	Stem of $pred$		
	13	Parse tree path from n to the lowest common ancestor of n and $pred$		
	14	Head word of n		
	15	12 **&** n's syntactic category		
	16	Production rule that expands n's parent	*	*
	17	Parse tree path from n to the nearest support verb	*	
	18	Last part of speech (POS) subsumed by n	*	
	19	Production rule that expands n's left sibling		*
	20	Head word of n, if the parent of n is a PP		
	21	The POS of the head word of the right-most NP under n if n is a PP		
	...	Features 22-31 are available upon request	0	3
Nominal features	1	n's ancestor subcategorization frames (ASF) (see section 4)	*	
	2	n's word		
	3	Syntactic category of n's right sibling		
	4	Parse tree paths from n to each support verb	*	
	5	Last word of n's left sibling	*	*
	6	Parse tree path from n to previous nominal, with lexicalized source (see section 4)	*	
	7	Last word of n's right sibling	*	
	8	Production rule that expands n's left sibling	*	*
	9	Syntactic category of n		*
	10	PropBank markability score (see section 4)	*	
	11	Parse tree path from n to previous nominal, with lexicalized source and destination	*	
	12	Whether or not n is followed by PP	*	
	13	Parse tree path from n to previous nominal, with lexicalized destination	*	
	14	Head word of n's parent		*
	15	Whether or not n surfaces before a passive verb	*	*
	16	First word of n's left sibling	*	
	17	Parse tree path from n to closest support verb, with lexicalized destination	*	
	18	Whether or not n is a head	*	
	19	Head word of n's right sibling		
	20	Production rule that expands n's parent	*	*
	21	Parse tree paths from n to all support verbs, with lexicalized destinations	*	
	22	First word of n's right sibling	*	*
	23	Head word of n's left sibling	*	
	24	If n is followed by a PP, the head of that PP's object	*	
	25	Parse tree path from n to previous nominal	*	
	26	Token distance from n to previous nominal	*	
	27	Production rule that expands n's grandparent	*	

Table 2: Features, sorted by gain in selection algorithm. **&** denotes concatenation. The last two columns indicate (N)ew features (not used in Liu and Ng (2007)) and features (S)hared by the argument and nominal models.

as a binary classification task over token nodes. Once a nominal has been identified as bearing overt arguments, it is processed with the argument identification model developed in the previous section. To classify nominals, we use the features shown in the bottom half of Table 2, which were selected with the same algorithm used for the argument classification model. As shown by Table 2, the sets of features selected for argument and nominal classification are quite different, and many of the features used for nominal classification have not been previously used. Below, we briefly explain a few of these features.

Ancestor subcategorization frames (ASF) As shown in Table 2, the most informative feature is ASF. For a given token t, ASF is actually a set of sub-features, one for each parse tree node above t. Each sub-feature is indexed (i.e., named) by its distance from t. The value of an ASF sub-feature is the production rule that expands the corresponding node in the tree. An ASF feature with two sub-features is depicted below for the token "sale":

VP: $ASF_2 = VP \rightarrow V, NP$

V (made)   NP: $ASF_1 = NP \rightarrow Det, N$

Det (a)   N (sale)
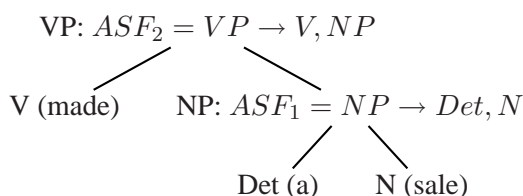
Parse tree path lexicalization A lexicalized parse tree path is one in which surface tokens from the beginning or end of the path are included in the path. This is a finer-grained version of the traditional parse tree path that captures the joint behavior of the path and the tokens it connects. For example, in the tree above, the path from "sale" to "made" with a lexicalized source and destination would be $sale : N \uparrow NP \uparrow VP \downarrow V : made$. Lexicalization increases sparsity; however, it is often preferred by the feature selection algorithm, as shown in the bottom half of Table 2.

PropBank markability score This feature is the probability that the context (± 5 words) of a deverbal nominal is generated by a unigram language model trained over the PropBank argument words for the corresponding verb. Entities are normalized

	Precision	Recall	F_1
Baseline	0.5555	0.9784	0.7086
MLE	0.6902	0.8903	0.7776
LibLinear	0.8989	0.8927	0.8958

Table 4: Evaluation results for identifying nominals with explicit arguments.

to their entity type using BBN's IdentiFinder, and adverbs are normalized to their related adjective using the *ADJADV* dictionary provided by NomBank. The normalization of adverbs is motivated by the fact that adverbial modifiers of verbs typically have a corresponding adjectival modifier for deverbal nominals.

5 Evaluation results

Our evaluation methodology reflects a practical scenario in which the nominal SRL system must process each token in a sentence. The system cannot safely assume that each token bears overt arguments; rather, this decision must be made automatically. In section 5.1, we present results for the automatic identification of nominals with overt arguments. Then, in section 5.2, we present results for the combined task in which nominal classification is followed by argument identification.

5.1 Nominal classification

Following standard practice, we train the nominal classifier over NomBank sections 2-21 using LibLinear and automatically generated syntactic parse trees. The prediction threshold is set to the value that maximizes the nominal F_1 score on development section (24), and the resulting model is tested over section 23. For comparison, we implemented the following simple classifiers.

Baseline nominal classifier Classifies a token as overtly bearing arguments if it is a singular or plural noun that is markable in the training data. As shown in Table 4, this classifier achieves nearly perfect recall.[5]

MLE nominal classifier Operates similarly to

---

[5]Recall is less than 100% due to (1) part-of-speech errors from the syntactic parser and (2) nominals that were not annotated in the training data but exist in the testing data.
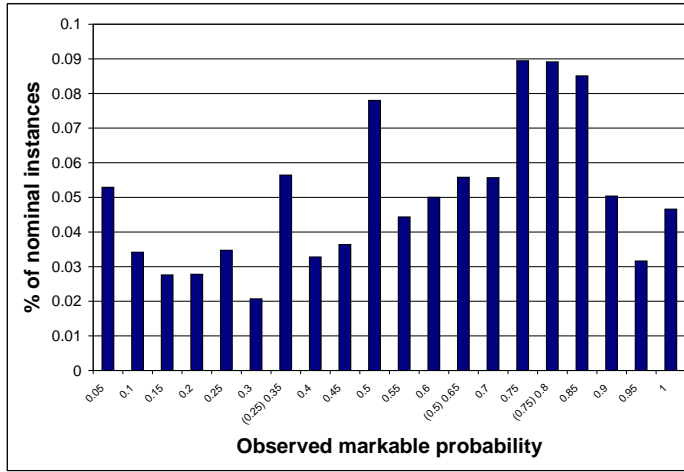
(a) Distribution of nominals. Each interval on the x-axis denotes a set of nominals that are markable between $(x-5)\%$ and $x\%$ of the time in the training data. The y-axis denotes the percentage of all nominal instances in TreeBank that is occupied by nominals in the interval. Quartiles are marked below the intervals. For example, quartile 0.25 indicates that one quarter of all nominal instances are markable 35% of the time or less.
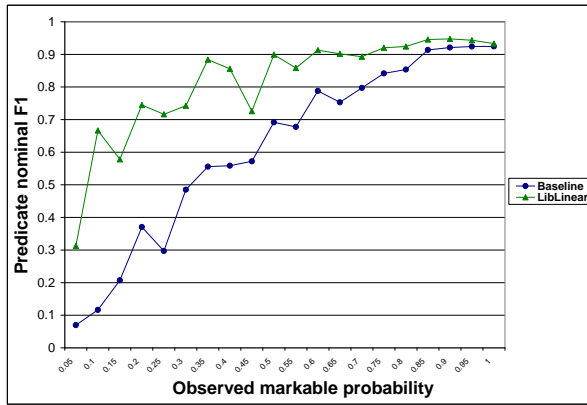
(b) Nominal classification performance with respect to the distribution in Figure 1a. The y-axis denotes the combined F_1 for nominals in the interval.
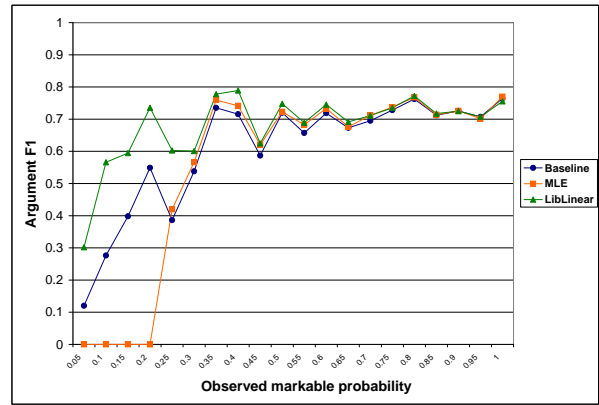
(c) All-token argument classification performance with respect to the distribution in Figure 1a. The y-axis denotes the combined F_1 for nominals in the interval.

Figure 1: Evaluation results with respect to the distribution of nominals in TreeBank.

the baseline classifier, but also produces a score for the classification. The value of the score is equal to the probability that the nominal bears overt arguments, as observed in the training data. A prediction threshold is imposed on this score as determined by the development data ($t = 0.23$). As shown by Table 4, this exchanges recall for precision and leads to a significant increase in the overall F_1 score.

The last row in Table 4 shows the results for the LibLinear nominal classifier, which significantly outperforms the others, achieving balanced precision and recall scores near 0.9. In addition, it is

able to recover from part-of-speech errors because it does not filter out non-noun instances; rather, it combines part-of-speech information with other lexical and syntactic features to classify nominals.

Interesting observations can be made by grouping nominals according to the probability with which they are markable in the corpus. Figure 1a gives the overall distribution of markable nominals in the training data. As shown, 50% of nominal instances are markable only 65% of the time or less, making nominal classification an important first step. Using this view of the data, Figure 1b presents the overall F_1 scores for the baseline and LibLinear nominal

classifiers.[6] As expected, gains in nominal classification diminish as nominals become more overtly associated with arguments. Furthermore, nominals that are rarely markable (i.e., those in interval 0.05) remain problematic due to a lack of positive training instances and the unbalanced nature of the classification task.

5.2 Combined nominal-argument classification

We now turn to the task of combined nominal-argument classification. In this task, systems must first identify nominals that bear overt arguments. We evaluated three configurations based on the nominal classifiers from the previous section. Each configuration uses the argument classification model from section 3.

As shown in Table 3, overall argument classification F_1 suffers a loss of more than 9% under the assumption that all known nouns bear overt arguments. This corresponds precisely to using the baseline nominal classifier in the combined nominal-argument task. The MLE nominal classifier is able to reduce this loss by 25% to an F_1 of 0.7080. The LibLinear nominal classifier reduces this loss by 46%, resulting in an overall argument classification F_1 of 0.7235. This improvement is the direct result of filtering out nominal instances that do not bear overt arguments.

Similarly to the nominal evaluation, we can view argument classification performance with respect to the probability that a nominal bears overt arguments. This is shown in Figure 1c for the three configurations. The configuration using the MLE nominal classifier obtains an argument F_1 of zero for nominals below its prediction threshold. Compared to the baseline nominal classifier, the LibLinear classifier achieves argument classification gains as large as 150.94% (interval 0.05), with an average gain of 52.87% for intervals 0.05 to 0.4. As with nominal classification, argument classification gains diminish for nominals that express arguments more overtly - we observe an average gain of only 2.15% for intervals 0.45 to 1.00. One possible explanation for this is that the argument prediction model has substantially more training data for the nominals in intervals 0.45 to 1.00. Thus, even if the nom-

---

[6]Baseline and MLE are identical above the MLE threshold.

	Nominals		
	Deverbal	Deverbal-like	Other
Baseline	0.7975	0.6789	0.6757
MLE	0.8298	0.7332	0.7486
LibLinear	0.9261	0.8826	0.8905
	Arguments		
Baseline	0.7059	0.6738	0.7454
MLE	0.7206	0.6641	0.7675
LibLinear	0.7282	0.7178	0.7847

Table 5: Nominal and argument F_1 scores for deverbal, deverbal-like, and other nominals in the all-token evaluation.

inal classifier makes a false positive prediction in the 0.45 to 1.00 interval range, the argument model may correctly avoid labeling any arguments.

As noted in section 2, these results are not directly comparable to the results of the recent CoNLL Shared Task (Surdeanu et al., 2008). This is due to the fact that the semantic labeled F_1 in the Shared Task combines predicate and argument predictions into a single score. The same combined F_1 score for our best two-stage nominal SRL system (logistic regression nominal and argument models) is 0.7806; however, this result is not precisely comparable because we do not identify the predicate role set as required by the CoNLL Shared Task.

5.3 NomLex-based analysis of results

As demonstrated in section 1, NomBank annotates many classes of deverbal and non-deverbal nominals, which have been categorized on syntactic and semantic bases in NomLex-PLUS (Meyers, 2007b). To help understand what types of nominals are particularly affected by implicit argumentation, we further analyzed performance with respect to these classes.

Figure 2a shows the distribution of nominals across classes defined by the NomLex resource. As shown in Figure 2b, many of the most frequent classes exhibit significant gains. For example, the classification of partitive nominals (13% of all nominal instances) with the LibLinear classifier results in gains of 55.45% and 33.72% over the baseline and MLE classifiers, respectively. For the 5 most common classes, which constitute 82% of all nominals instances, we observe average gains of 27.47% and 19.30% over the baseline and MLE classifiers,
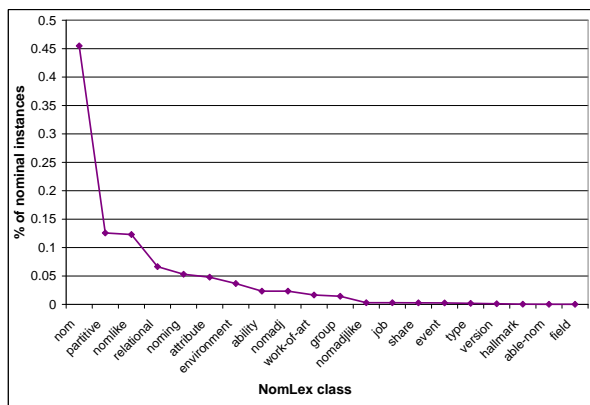
(a) Distribution of nominals across the NomLex classes. The y-axis denotes the percentage of all nominal instances that is occupied by nominals in the class.
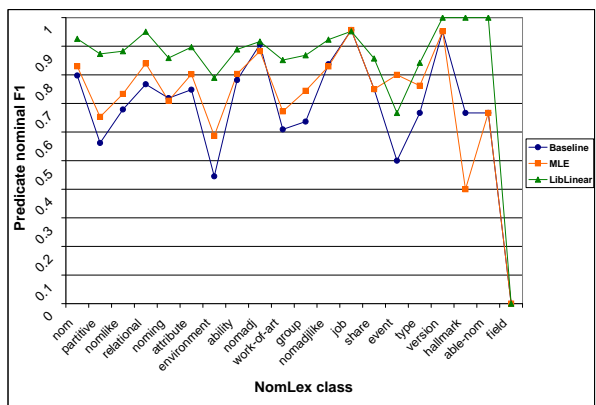
(b) Nominal classification performance with respect to the NomLex classes in Figure 2a. The y-axis denotes the combined F_1 for nominals in the class.

Figure 2: Evaluation results with respect to NomLex classes.

respectively.

Table 5 separates nominal and argument classification results into sets of deverbal (NomLex class *nom*), deverbal-like (NomLex class *nom-like*), and all other nominalizations. A deverbal-like nominal is closely related to some verb, although not morphologically. For example, the noun *accolade* shares argument interpretation with *award*, but the two are not morphologically related. As shown by Table 5, nominal classification tends to be easier - and argument classification harder - for deverbals when compared to other types of nominals. The difference in argument F_1 between deverbal/deverbal-like nominals and the others is due primarily to relational nominals, which are relatively easy to classify (Figure 2b); additionally, relational nominals exhibit a high rate of argument incorporation, which is easily handled by the maximum-likelihood model described in section 3.1.

6 Conclusions and future work

The application of nominal SRL to practical NLP problems requires a system that is able to accurately process each token it encounters. Previously, it was unclear whether the models proposed by Jiang and Ng (2006) and Liu and Ng (2007) would operate effectively in such an environment. The systems described by Surdeanu et al. (2008) are designed with this environment in mind, but their evaluation did not focus on the issue of implicit argumentation. These two problems motivate the work presented in this paper.

Our contribution is three-fold. First, we improve upon previous nominal SRL results using a single-stage classifier with additional new features. Second, we show that this model suffers a substantial performance degradation when evaluated over nominals with implicit arguments. Finally, we identify a set of features - many of them new - that can be used to reliably detect nominals with explicit arguments, thus significantly increasing the performance of the nominal SRL system.

Our results also suggest interesting directions for future work. As described in section 5.2, many nominals do not have enough labeled training data to produce accurate argument models. The generalization procedures developed by Gordon and Swanson (2007) for PropBank SRL and Padó et al. (2008) for NomBank SRL might alleviate this problem. Additionally, instead of ignoring nominals with implicit arguments, we would prefer to identify the implicit arguments using information contained in the surrounding discourse. Such inferences would help connect entities and events across sentences, providing a fuller interpretation of the text.

References

Collin Baker, Charles Fillmore, and John Lowe. 1998. The Berkeley FrameNet project. In Christian Boitet and Pete Whitelock, editors, *Proceedings of the Thirty-Sixth Annual Meeting of the Association for Computational Linguistics and Seventeenth International Conference on Computational Linguistics*, pages 86–90, San Francisco, California. Morgan Kaufmann Publishers.

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

Eugene Charniak and Mark Johnson. 2005. Coarse-to-fine n-best parsing and maxent discriminative reranking. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*.

Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. 2008. Liblinear: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874.

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*.

A. Gordon and R. Swanson. 2007. Generalizing semantic role annotations across syntactically similar verbs. In *Proceedings of ACL*, pages 192–199.

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*.

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.

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*, pages 208–215, Prague, Czech Republic, June. Association for Computational Linguistics.

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.

Sebastian Padó, Marco Pennacchiotti, and Caroline Sporleder. 2008. Semantic role assignment for event nominalisations by leveraging verbal data. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 665–672, Manchester, UK, August. Coling 2008 Organizing Committee.

Martha Palmer, Daniel Gildea, and Paul Kingsbury. 2005. The proposition bank: An annotated corpus of semantic roles. *Computational Linguistics*, 31(1):71–106.

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

Mihai Surdeanu, Richard Johansson, Adam Meyers, Lluís Màrquez, and Joakim Nivre. 2008. The CoNLL 2008 shared task on joint parsing of syntactic and semantic dependencies. In *CoNLL 2008: Proceedings of the Twelfth Conference on Computational Natural Language Learning*, pages 159–177, Manchester, England, August. Coling 2008 Organizing Committee.