Beyond NomBank: A Study of Implicit Arguments for Nominal Predicates

Matthew Gerber and **Joyce Y. Chai**
Department of Computer Science
Michigan State University
East Lansing, Michigan, USA
{gerberm2,jchai}@cse.msu.edu

Abstract

Despite its substantial coverage, NomBank does not account for all within-sentence arguments and ignores extra-sentential arguments altogether. These arguments, which we call *implicit*, are important to semantic processing, and their recovery could potentially benefit many NLP applications. We present a study of implicit arguments for a select group of frequent nominal predicates. We show that implicit arguments are pervasive for these predicates, adding 65% to the coverage of NomBank. We demonstrate the feasibility of recovering implicit arguments with a supervised classification model. Our results and analyses provide a baseline for future work on this emerging task.

1 Introduction

Verbal and nominal semantic role labeling (SRL) have been studied independently of each other (Carreras and Màrquez, 2005; Gerber et al., 2009) as well as jointly (Surdeanu et al., 2008; Hajič et al., 2009). These studies have demonstrated the maturity of SRL within an evaluation setting that restricts the argument search space to the sentence containing the predicate of interest. However, as shown by the following example from the Penn TreeBank (Marcus et al., 1993), this restriction excludes extra-sentential arguments:

(1) [arg_0 The two companies] [$pred$ **produce**] [arg_1 market pulp, containerboard and white paper]. The goods could be manufactured closer to customers, saving [$pred$ **shipping**] costs.

The first sentence in Example 1 includes the PropBank (Kingsbury et al., 2002) analysis of the verbal predicate *produce*, where arg_0 is the agentive producer and arg_1 is the produced entity. The second sentence contains an instance of the nominal predicate *shipping* that is not associated with arguments in NomBank (Meyers, 2007).

From the sentences in Example 1, the reader can infer that *The two companies* refers to the agents (arg_0) of the *shipping* predicate. The reader can also infer that *market pulp, containerboard and white paper* refers to the shipped entities (arg_1 of *shipping*).[1] These extra-sentential arguments have not been annotated for the *shipping* predicate and cannot be identified by a system that restricts the argument search space to the sentence containing the predicate. NomBank also ignores many within-sentence arguments. This is shown in the second sentence of Example 1, where *The goods* can be interpreted as the arg_1 of *shipping*. These examples demonstrate the presence of arguments that are not included in NomBank and cannot easily be identified by systems trained on the resource. We refer to these arguments as *implicit*.

This paper presents our study of implicit arguments for nominal predicates. We began our study by annotating implicit arguments for a select group of predicates. For these predicates, we found that implicit arguments add 65% to the existing role coverage of NomBank.[2] This increase has implications for tasks (e.g., question answering, information extraction, and summarization) that benefit from semantic analysis. Using our annotations, we constructed a feature-based model for automatic implicit argument identification that unifies standard verbal and nominal SRL. Our results indicate a 59% relative (15-point absolute) gain in F_1 over an informed baseline. Our analyses highlight strengths and weaknesses of the approach, providing insights for future work on this emerging task.

[1] In PropBank and NomBank, the interpretation of each role (e.g., arg_0) is specific to a predicate sense.

[2] Role coverage indicates the percentage of roles filled.

Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, pages 1583–1592,
Uppsala, Sweden, 11-16 July 2010.

In the following section, we review related research, which is historically sparse but recently gaining traction. We present our annotation effort in Section 3, and follow with our implicit argument identification model in Section 4. In Section 5, we describe the evaluation setting and present our experimental results. We analyze these results in Section 6 and conclude in Section 7.

2 Related work

Palmer et al. (1986) made one of the earliest attempts to automatically recover extra-sentential arguments. Their approach used a fine-grained domain model to assess the compatibility of candidate arguments and the slots needing to be filled.

A phenomenon similar to the implicit argument has been studied in the context of Japanese anaphora resolution, where a missing case-marked constituent is viewed as a zero-anaphoric expression whose antecedent is treated as the implicit argument of the predicate of interest. This behavior has been annotated manually by Iida et al. (2007), and researchers have applied standard SRL techniques to this corpus, resulting in systems that are able to identify missing case-marked expressions in the surrounding discourse (Imamura et al., 2009). Sasano et al. (2004) conducted similar work with Japanese indirect anaphora. The authors used automatically derived nominal case frames to identify antecedents. However, as noted by Iida et al., grammatical cases do not stand in a one-to-one relationship with semantic roles in Japanese (the same is true for English).

Fillmore and Baker (2001) provided a detailed case study of implicit arguments (termed *null instantiations* in that work), but did not provide concrete methods to account for them automatically. Previously, we demonstrated the importance of filtering out nominal predicates that take no local arguments (Gerber et al., 2009); however, this work did not address the identification of implicit arguments. Burchardt et al. (2005) suggested approaches to implicit argument identification based on observed coreference patterns; however, the authors did not implement and evaluate such methods. We draw insights from all three of these studies. We show that the identification of implicit arguments for nominal predicates leads to fuller semantic interpretations when compared to traditional SRL methods. Furthermore, motivated by Burchardt et al., our model uses a quantitative analysis of naturally occurring coreference patterns to aid implicit argument identification.

Most recently, Ruppenhofer et al. (2009) conducted SemEval Task 10, "Linking Events and Their Participants in Discourse", which evaluated implicit argument identification systems over a common test set. The task organizers annotated implicit arguments across entire passages, resulting in data that cover many distinct predicates, each associated with a small number of annotated instances. In contrast, our study focused on a select group of nominal predicates, each associated with a large number of annotated instances.

3 Data annotation and analysis

3.1 Data annotation

Implicit arguments have not been annotated within the Penn TreeBank, which is the textual and syntactic basis for NomBank. Thus, to facilitate our study, we annotated implicit arguments for instances of nominal predicates within the standard training, development, and testing sections of the TreeBank. We limited our attention to nominal predicates with unambiguous role sets (i.e., senses) that are derived from verbal role sets. We then ranked this set of predicates using two pieces of information: (1) the average difference between the number of roles expressed in nominal form (in NomBank) versus verbal form (in PropBank) and (2) the frequency of the nominal form in the corpus. We assumed that the former gives an indication as to how many implicit roles an instance of the nominal predicate might have. The product of (1) and (2) thus indicates the potential prevalence of implicit arguments for a predicate. To focus our study, we ranked the predicates in NomBank according to this product and selected the top ten, shown in Table 1.

We annotated implicit arguments document-by-document, selecting all singular and plural nouns derived from the predicates in Table 1. For each missing argument position of each predicate instance, we inspected the local discourse for a suitable implicit argument. We limited our attention to the current sentence as well as all preceding sentences in the document, annotating all mentions of an implicit argument within this window.

In the remainder of this paper, we will use $iarg_n$ to refer to an implicit argument position n. We will use arg_n to refer to an argument provided by PropBank or NomBank. We will use p to mark

		Pre-annotation			Post-annotation	
			Role average			
Predicate	#	Role coverage (%)	Noun	Verb	Role coverage (%)	Noun role average
price	217	42.4	1.7	1.7	55.3	2.2
sale	185	24.3	1.2	2.0	42.0	2.1
investor	160	35.0	1.1	2.0	54.6	1.6
fund	109	8.7	0.4	2.0	21.6	0.9
loss	104	33.2	1.3	2.0	46.9	1.9
plan	102	30.9	1.2	1.8	49.3	2.0
investment	102	15.7	0.5	2.0	33.3	1.0
cost	101	26.2	1.1	2.3	47.5	1.9
bid	88	26.9	0.8	2.2	72.0	2.2
loan	85	22.4	1.1	2.5	41.2	2.1
Overall	1,253	28.0	1.1	2.0	46.2	1.8

Table 1: Predicates targeted for annotation. The second column gives the number of predicate instances annotated. *Pre-annotation* numbers only include NomBank annotations, whereas *Post-annotation* numbers include NomBank and implicit argument annotations. *Role coverage* indicates the percentage of roles filled. *Role average* indicates how many roles, on average, are filled for an instance of a predicate's noun form or verb form within the TreeBank. Verbal role averages were computed using PropBank.

predicate instances. Below, we give an example annotation for an instance of the *investment* predicate:

(2) [$iarg_0$ Participants] will be able to transfer [$iarg_1$ money] to [$iarg_2$ other investment funds]. The [p **investment**] choices are limited to [$iarg_2$ a stock fund and a money-market fund].

NomBank does not associate this instance of *investment* with any arguments; however, we were able to identify the investor ($iarg_0$), the thing invested ($iarg_1$), and two mentions of the thing invested in ($iarg_2$).

Our data set was also independently annotated by an undergraduate linguistics student. For each missing argument position, the student was asked to identify the closest acceptable implicit argument within the current and preceding sentences. The argument position was left unfilled if no acceptable constituent could be found. For a missing argument position, the student's annotation agreed with our own if both identified the same constituent or both left the position unfilled. Analysis indicated an agreement of 67% using Cohen's kappa coefficient (Cohen, 1960).

3.2 Annotation analysis

Role coverage for a predicate instance is equal to the number of filled roles divided by the number of roles in the predicate's lexicon entry. Role coverage for the marked predicate in Example 2 is 0/3 for NomBank-only arguments and 3/3 when the annotated implicit arguments are also considered. Returning to Table 1, the third column gives role coverage percentages for NomBank-only arguments. The sixth column gives role coverage percentages when both NomBank arguments and the annotated implicit arguments are considered. Overall, the addition of implicit arguments created a 65% relative (18-point absolute) gain in role coverage across the 1,253 predicate instances that we annotated.

The predicates in Table 1 are typically associated with fewer arguments on average than their corresponding verbal predicates. When considering NomBank-only arguments, this difference (compare columns four and five) varies from zero (for *price*) to a factor of five (for *fund*). When implicit arguments are included in the comparison, these differences are reduced and many nominal predicates express approximately the same number of arguments on average as their verbal counterparts (compare the fifth and seventh columns).

In addition to role coverage and average count, we examined the location of implicit arguments. Figure 1 shows that approximately 56% of the implicit arguments in our data can be resolved within the sentence containing the predicate. The remaining implicit arguments require up to forty-six sen-

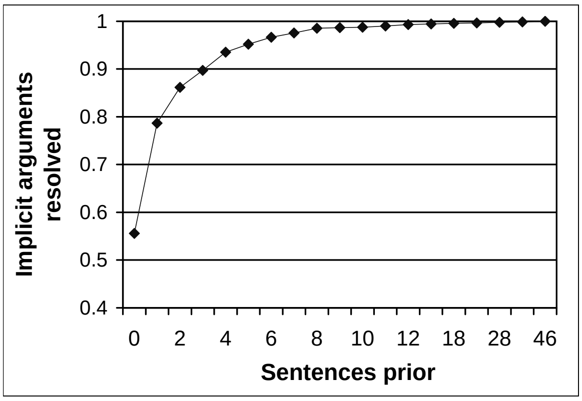

Figure 1: Location of implicit arguments. For missing argument positions with an implicit filler, the y-axis indicates the likelihood of the filler being found at least once in the previous x sentences.

tences for resolution; however, a vast majority of these can be resolved within the previous few sentences. Section 6 discusses implications of this skewed distribution.

4 Implicit argument identification

4.1 Model formulation

In our study, we assumed that each sentence in a document had been analyzed for PropBank and NomBank predicate-argument structure. NomBank includes a lexicon listing the possible argument positions for a predicate, allowing us to identify missing argument positions with a simple lookup. Given a nominal predicate instance p with a missing argument position $iarg_n$, the task is to search the surrounding discourse for a constituent c that fills $iarg_n$. Our model conducts this search over all constituents annotated by either PropBank or NomBank with non-adjunct labels.

A candidate constituent c will often form a coreference chain with other constituents in the discourse. Consider the following abridged sentences, which are adjacent in their Penn TreeBank document:

(3) [Mexico] desperately needs investment.

(4) Conservative Japanese investors are put off by [Mexico's] investment regulations.

(5) Japan is the fourth largest investor in [c Mexico], with 5% of the total [p **investments**].

NomBank does not associate the labeled instance of *investment* with any arguments, but it is clear from the surrounding discourse that constituent c (referring to Mexico) is the thing being invested in (the $iarg_2$). When determining whether c is the $iarg_2$ of *investment*, one can draw evidence from other mentions in c's coreference chain. Example 3 states that Mexico needs investment. Example 4 states that Mexico regulates investment. These propositions, which can be derived via traditional SRL analyses, should increase our confidence that c is the $iarg_2$ of *investment* in Example 5.

Thus, the unit of classification for a candidate constituent c is the three-tuple $\langle p, iarg_n, c' \rangle$, where c' is a coreference chain comprising c and its coreferent constituents.[3] We defined a binary classification function $Pr(+| \langle p, iarg_n, c' \rangle)$ that predicts the probability that the entity referred to by c fills the missing argument position $iarg_n$ of predicate instance p. In the remainder of this paper, we will refer to c as the *primary filler*, differentiating it from other mentions in the coreference chain c'. In the following section, we present the feature set used to represent each three-tuple within the classification function.

4.2 Model features

Starting with a wide range of features, we performed floating forward feature selection (Pudil et al., 1994) over held-out development data comprising implicit argument annotations from section 24 of the Penn TreeBank. As part of the feature selection process, we conducted a grid search for the best per-class cost within LibLinear's logistic regression solver (Fan et al., 2008). This was done to reduce the negative effects of data imbalance, which is severe even when selecting candidates from the current and previous few sentences. Table 2 shows the selected features, which are quite different from those used in our previous work to identify traditional semantic arguments (Gerber et al., 2009).[4] Below, we give further explanations for some of the features.

Feature 1 models the semantic role relationship between each mention in c' and the missing argument position $iarg_n$. To reduce data sparsity, this feature generalizes predicates and argument positions to their VerbNet (Kipper, 2005) classes and

[3]We used OpenNLP for coreference identification: http://opennlp.sourceforge.net

[4]We have omitted many of the lowest-ranked features. Descriptions of these features can be obtained by contacting the authors.

#	Feature value description
1*	For every f, the VerbNet class/role of p_f/arg_f concatenated with the class/role of p/$iarg_n$.
2*	Average pointwise mutual information between $\langle p, iarg_n \rangle$ and any $\langle p_f, arg_f \rangle$.
3	Percentage of all f that are definite noun phrases.
4	Minimum absolute sentence distance from any f to p.
5*	Minimum pointwise mutual information between $\langle p, iarg_n \rangle$ and any $\langle p_f, arg_f \rangle$.
6	Frequency of the nominal form of p within the document that contains it.
7	Nominal form of p concatenated with $iarg_n$.
8	Nominal form of p concatenated with the sorted integer argument indexes from all arg_n of p.
9	Number of mentions in c'.
10*	Head word of p's right sibling node.
11	For every f, the synset (Fellbaum, 1998) for the head of f concatenated with p and $iarg_n$.
12	Part of speech of the head of p's parent node.
13	Average absolute sentence distance from any f to p.
14*	Discourse relation whose two discourse units cover c (the primary filler) and p.
15	Number of left siblings of p.
16	Whether p is the head of its parent node.
17	Number of right siblings of p.

Table 2: Features for determining whether c fills $iarg_n$ of predicate p. For each mention f (denoting a $f$iller) in the coreference chain c', we define p_f and arg_f to be the predicate and argument position of f. Features are sorted in descending order of feature selection gain. Unless otherwise noted, all predicates were normalized to their verbal form and all argument positions (e.g., arg_n and $iarg_n$) were interpreted as labels instead of word content. Features marked with an asterisk are explained in Section 4.2.

semantic roles using SemLink.[5] For explanation purposes, consider again Example 1, where we are trying to fill the $iarg_0$ of *shipping*. Let c' contain a single mention, *The two companies*, which is the arg_0 of *produce*. As described in Table 2, feature 1 is instantiated with a value of *create.agent-send.agent*, where *create* and *send* are the VerbNet classes that contain *produce* and *ship*, respectively. In the conversion to LibLinear's instance representation, this instantiation is converted into a single binary feature *create.agent-send.agent* whose value is one. Features 1 and 11 are instantiated once for each mention in c', allowing the model to consider information from multiple mentions of the same entity.

Features 2 and 5 are inspired by the work of Chambers and Jurafsky (2008), who investigated unsupervised learning of narrative event sequences using pointwise mutual information (PMI) between syntactic positions. We used a similar PMI score, but defined it with respect to semantic arguments instead of syntactic dependencies. Thus, the values for features 2 and 5 are computed as follows (the notation is explained in the caption for Table 2):

$$pmi(\langle p, iarg_n \rangle, \langle p_f, arg_f \rangle) = \\ log \frac{P_{coref}(\langle p, iarg_n \rangle, \langle p_f, arg_f \rangle)}{P_{coref}(\langle p, iarg_n \rangle, *) P_{coref}(\langle p_f, arg_f \rangle, *)} \quad (6)$$

To compute Equation 6, we first labeled a subset of the Gigaword corpus (Graff, 2003) using the verbal SRL system of Punyakanok et al. (2008) and the nominal SRL system of Gerber et al. (2009). We then identified coreferent pairs of arguments using OpenNLP. Suppose the resulting data has N coreferential pairs of argument positions. Also suppose that M of these pairs comprise $\langle p, arg_n \rangle$ and $\langle p_f, arg_f \rangle$. The numerator in Equation 6 is defined as $\frac{M}{N}$. Each term in the denominator is obtained similarly, except that M is computed as the total number of coreference pairs comprising an argument position (e.g., $\langle p, arg_n \rangle$) and *any* other argument position. Like Chambers and Jurafsky, we also used the discounting method suggested by Pantel and Ravichandran (2004) for low-frequency observations. The PMI score is somewhat noisy due to imperfect output, but it provides information that is useful for classification.

[5]http://verbs.colorado.edu/semlink

Feature 10 does not depend on c' and is specific to each predicate. Consider the following example:

(7) Statistics Canada reported that its [arg_1 industrial-product] [p **price**] index dropped 2% in September.

The "[p price] index" collocation is rarely associated with an arg_0 in NomBank or with an $iarg_0$ in our annotations (both argument positions denote the seller). Feature 10 accounts for this type of behavior by encoding the syntactic head of p's right sibling. The value of feature 10 for Example 7 is *price:index*. Contrast this with the following:

(8) [$iarg_0$ The company] is trying to prevent further [p **price**] drops.

The value of feature 10 for Example 8 is *price:drop*. This feature captures an important distinction between the two uses of *price*: the former rarely takes an $iarg_0$, whereas the latter often does. Features 12 and 15-17 account for predicate-specific behaviors in a similar manner.

Feature 14 identifies the discourse relation (if any) that holds between the candidate constituent c and the filled predicate p. Consider the following example:

(9) [$iarg_0$ SFE Technologies] reported a net loss of $889,000 on sales of $23.4 million.

(10) That compared with an operating [p **loss**] of [arg_1 $1.9 million] on sales of $27.4 million in the year-earlier period.

In this case, a *comparison* discourse relation (signaled by the underlined text) holds between the first and sentence sentence. The coherence provided by this relation encourages an inference that identifies the marked $iarg_0$ (the loser). Throughout our study, we used gold-standard discourse relations provided by the Penn Discourse TreeBank (Prasad et al., 2008).

5 Evaluation

We trained the feature-based logistic regression model over 816 annotated predicate instances associated with 650 implicitly filled argument positions (not all predicate instances had implicit arguments). During training, a candidate three-tuple $\langle p, iarg_n, c' \rangle$ was given a positive label if the candidate implicit argument c (the primary filler) was annotated as filling the missing argument position. To factor out errors from standard SRL analyses, the model used gold-standard argument labels provided by PropBank and NomBank. As shown in Figure 1 (Section 3.2), implicit arguments tend to be located in close proximity to the predicate. We found that using all candidate constituents c within the current and previous two sentences worked best on our development data.

We compared our supervised model with the simple baseline heuristic defined below:[6]

> Fill $iarg_n$ for predicate instance p with the nearest constituent in the two-sentence candidate window that fills arg_n for a different instance of p, where all nominal predicates are normalized to their verbal forms.

The normalization allows an existing arg_0 for the verb *invested* to fill an $iarg_0$ for the noun *investment*. We also evaluated an oracle model that made gold-standard predictions for candidates within the two-sentence prediction window.

We evaluated these models using the methodology proposed by Ruppenhofer et al. (2009). For each missing argument position of a predicate instance, the models were required to either (1) identify a single constituent that fills the missing argument position or (2) make no prediction and leave the missing argument position unfilled. We scored predictions using the Dice coefficient, which is defined as follows:

$$\frac{2 * |Predicted \bigcap True|}{|Predicted| + |True|} \tag{11}$$

$Predicted$ is the set of tokens subsumed by the constituent predicted by the model as filling a missing argument position. $True$ is the set of tokens from a single annotated constituent that fills the missing argument position. The model's prediction receives a score equal to the maximum Dice overlap across any one of the annotated fillers. Precision is equal to the summed prediction scores divided by the number of argument positions filled by the model. Recall is equal to the summed prediction scores divided by the number of argument positions filled in our annotated data. Predictions not covering the head of a true filler were assigned a score of zero.

[6] This heuristic outperformed a more complicated heuristic that relied on the PMI score described in section 4.2.

	#	Imp. #	Baseline P	Baseline R	Baseline F_1	Discriminative P	Discriminative R	Discriminative F_1	p	Oracle R	Oracle F_1
sale	64	60	50.0	28.3	36.2	47.2	41.7	**44.2**	0.118	80.0	88.9
price	121	53	24.0	11.3	15.4	36.0	32.6	**34.2**	0.008	88.7	94.0
investor	78	35	33.3	5.7	9.8	36.8	40.0	**38.4**	< 0.001	91.4	95.5
bid	19	26	100.0	19.2	**32.3**	23.8	19.2	21.3	0.280	57.7	73.2
plan	25	20	83.3	25.0	38.5	78.6	55.0	**64.7**	0.060	82.7	89.4
cost	25	17	66.7	23.5	34.8	61.1	64.7	**62.9**	0.024	94.1	97.0
loss	30	12	71.4	41.7	52.6	83.3	83.3	**83.3**	0.020	100.0	100.0
loan	11	9	50.0	11.1	18.2	42.9	33.3	**37.5**	0.277	88.9	94.1
investment	21	8	0.0	0.0	0.0	40.0	25.0	**30.8**	0.182	87.5	93.3
fund	43	6	0.0	0.0	0.0	14.3	16.7	**15.4**	0.576	50.0	66.7
Overall	437	246	48.4	18.3	26.5	44.5	40.4	**42.3**	< 0.001	83.1	90.7

Table 3: Evaluation results. The second column gives the number of predicate instances evaluated. The third column gives the number of ground-truth implicitly filled argument positions for the predicate instances (not all instances had implicit arguments). P, R, and F_1 indicate precision, recall, and F-measure ($\beta = 1$), respectively. p-values denote the bootstrapped significance of the difference in F_1 between the baseline and discriminative models. Oracle precision (not shown) is 100% for all predicates.

Our evaluation data comprised 437 predicate instances associated with 246 implicitly filled argument positions. Table 3 presents the results. Predicates with the highest number of implicit arguments - *sale* and *price* - showed F_1 increases of 8 points and 18.8 points, respectively. Overall, the discriminative model increased F_1 performance 15.8 points (59.6%) over the baseline.

We measured human performance on this task by running our undergraduate assistant's annotations against the evaluation data. Our assistant achieved an overall F_1 score of 58.4% using the same candidate window as the baseline and discriminative models. The difference in F_1 between the discriminative and human results had an exact p-value of less than 0.001. All significance testing was performed using a two-tailed bootstrap method similar to the one described by Efron and Tibshirani (1993).

6 Discussion

6.1 Feature ablation

We conducted an ablation study to measure the contribution of specific feature sets. Table 4 presents the ablation configurations and results. For each configuration, we retrained and retested the discriminative model using the features described. As shown, we observed significant losses when excluding features that relate the semantic roles of mentions in c' to the semantic role of the missing argument position (first configuration). The second configuration tested the effect of using *only* the SRL-based features. This also resulted in significant performance losses, suggesting that the other features contribute useful information. Lastly, we tested the effect of removing discourse relations (feature 14), which are likely to be difficult to extract reliably in a practical setting. As shown, this feature did not have a statistically significant effect on performance and could be excluded in future applications of the model.

Configuration	Percent change (p-value) P	R	F_1
Remove 1,2,5	-35.3 (< 0.01)	-36.1 (< 0.01)	-35.7 (< 0.01)
Use 1,2,5 only	-26.3 (< 0.01)	-11.9 (0.05)	-19.2 (< 0.01)
Remove 14	0.2 (0.95)	1.0 (0.66)	0.7 (0.73)

Table 4: Feature ablation results. The first column lists the feature configurations. All changes are percentages relative to the full-featured discriminative model. p-values for the changes are indicated in parentheses.

6.2 Unclassified true implicit arguments

Of all the errors made by the system, approximately 19% were caused by the system's failure to generate a candidate constituent c that was a correct implicit argument. Without such a candidate, the system stood no chance of identifying a correct implicit argument. Two factors contributed to this type of error, the first being our assumption that implicit arguments are also core (i.e., arg_n) arguments to traditional SRL structures. Approximately 8% of the overall error was due to a failure of this assumption. In many cases, the true implicit argument filled a non-core (i.e., adjunct) role within PropBank or NomBank.

More frequently, however, true implicit arguments were missed because the candidate window was too narrow. This accounts for 12% of the overall error. Oracle recall (second-to-last column in Table 3) indicates the nominals that suffered most from windowing errors. For example, the *sale* predicate was associated with the highest number of true implicit arguments, but only 80% of those could be resolved within the two-sentence candidate window. Empirically, we found that extending the candidate window uniformly for all predicates did not increase performance on the development data. The oracle results suggest that predicate-specific window settings might offer some advantage.

6.3 The *investment* and *fund* predicates

In Section 4.2, we discussed the *price* predicate, which frequently occurs in the "[p price] index" collocation. We observed that this collocation is rarely associated with either an overt arg_0 or an implicit $iarg_0$. Similar observations can be made for the *investment* and *fund* predicates. Although these two predicates are frequent, they are rarely associated with implicit arguments: *investment* takes only eight implicit arguments across its 21 instances, and *fund* takes only six implicit arguments across its 43 instances. This behavior is due in large part to collocations such as "[p investment] banker", "stock [p fund]", and "mutual [p fund]", which use predicate senses that are not eventive. Such collocations also violate our assumption that differences between the PropBank and NomBank argument structure for a predicate are indicative of implicit arguments (see Section 3.1 for this assumption).

Despite their lack of implicit arguments, it is important to account for predicates such as *investment* and *fund* because incorrect prediction of implicit arguments for them can lower precision. This is precisely what happened for the *fund* predicate, where the model incorrectly identified many implicit arguments for "stock [p fund]" and "mutual [p fund]". The left context of *fund* should help the model avoid this type of error; however, our feature selection process did not identify any overall gains from including this information.

6.4 Improvements versus the baseline

The baseline heuristic covers the simple case where identical predicates share arguments in the same position. Thus, it is interesting to examine cases where the baseline heuristic failed but the discriminative model succeeded. Consider the following sentence:

(12) Mr. Rogers recommends that [p **investors**] sell [$iarg_2$ takeover-related stock].

Neither NomBank nor the baseline heuristic associate the marked predicate in Example 12 with any arguments; however, the feature-based model was able to correctly identify the marked $iarg_2$ as the entity being invested in. This inference captured a tendency of investors to sell the things they have invested in.

We conclude our discussion with an example of an extra-sentential implicit argument:

(13) [$iarg_0$ Olivetti] has denied that it violated the rules, asserting that the shipments were properly licensed. However, the legality of these [p **sales**] is still an open question.

As shown in Example 13, the system was able to correctly identify *Olivetti* as the agent in the selling event of the second sentence. This inference involved two key steps. First, the system identified coreferent mentions of *Olivetti* that participated in exporting and supplying events (not shown). Second, the system identified a tendency for exporters and suppliers to also be sellers. Using this knowledge, the system extracted information that could not be extracted by the baseline heuristic or a traditional SRL system.

7 Conclusions and future work

Current SRL approaches limit the search for arguments to the sentence containing the predicate of interest. Many systems take this assumption a step further and restrict the search to the predicate's local syntactic environment; however, predicates and the sentences that contain them rarely exist in isolation. As shown throughout this paper, they are usually embedded in a coherent and semantically rich discourse that must be taken into account. We have presented a preliminary study of implicit arguments for nominal predicates that focused specifically on this problem.

Our contribution is three-fold. First, we have created gold-standard implicit argument annotations for a small set of pervasive nominal predicates.[7] Our analysis shows that these annotations add 65% to the role coverage of NomBank. Second, we have demonstrated the feasibility of recovering implicit arguments for many of the predicates, thus establishing a baseline for future work on this emerging task. Third, our study suggests a few ways in which this research can be moved forward. As shown in Section 6, many errors were caused by the absence of true implicit arguments within the set of candidate constituents. More intelligent windowing strategies in addition to alternate candidate sources might offer some improvement. Although we consistently observed development gains from using automatic coreference resolution, this process creates errors that need to be studied more closely. It will also be important to study implicit argument patterns of non-verbal predicates such as the partitive *percent*. These predicates are among the most frequent in the TreeBank and are likely to require approaches that differ from the ones we pursued.

Finally, any extension of this work is likely to encounter a significant knowledge acquisition bottleneck. Implicit argument annotation is difficult because it requires both argument and coreference identification (the data produced by Ruppenhofer et al. (2009) is similar). Thus, it might be productive to focus future work on (1) the extraction of relevant knowledge from existing resources (e.g., our use of coreference patterns from Gigaword) or (2) semi-supervised learning of implicit argument models from a combination of labeled and unlabeled data.

Acknowledgments

We would like to thank the anonymous reviewers for their helpful questions and comments. We would also like to thank Malcolm Doering for his annotation effort. This work was supported in part by NSF grants IIS-0347548 and IIS-0840538.

[7]Our annotation data can be freely downloaded at http://links.cse.msu.edu:8000/lair/projects/semanticrole.html

References

Aljoscha Burchardt, Anette Frank, and Manfred Pinkal. 2005. Building text meaning representations from contextually related frames - a case study. In *Proceedings of the Sixth International Workshop on Computational Semantics*.

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

Nathanael Chambers and Dan Jurafsky. 2008. Unsupervised learning of narrative event chains. In *Proceedings of the Association for Computational Linguistics*, pages 789–797, Columbus, Ohio, June. Association for Computational Linguistics.

Jacob Cohen. 1960. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20(1):3746.

Bradley Efron and Robert J. Tibshirani. 1993. *An Introduction to the Bootstrap*. Chapman & Hall, New York.

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.

Christiane Fellbaum. 1998. *WordNet: An Electronic Lexical Database (Language, Speech, and Communication)*. The MIT Press, May.

C.J. Fillmore and C.F. Baker. 2001. Frame semantics for text understanding. In *Proceedings of WordNet and Other Lexical Resources Workshop, NAACL*.

Matthew Gerber, Joyce Y. Chai, and Adam Meyers. 2009. The role of implicit argumentation in nominal SRL. In *Proceedings of the North American Chapter of the Association for Computational Linguistics*, pages 146–154, Boulder, Colorado, USA, June.

David Graff. 2003. English Gigaword. Linguistic Data Consortium, Philadelphia.

Jan Hajič, Massimiliano Ciaramita, Richard Johansson, Daisuke Kawahara, Maria Antònia Martí, Lluís Màrquez, Adam Meyers, Joakim Nivre, Sebastian Padó, Jan Štěpánek, Pavel Straňák, Mihai Surdeanu, Nianwen Xue, and Yi Zhang. 2009. The CoNLL-2009 shared task: Syntactic and semantic dependencies in multiple languages. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL 2009): Shared Task*, pages 1–18, Boulder, Colorado, June. Association for Computational Linguistics.

Ryu Iida, Mamoru Komachi, Kentaro Inui, and Yuji Matsumoto. 2007. Annotating a Japanese text corpus with predicate-argument and coreference relations. In *Proceedings of the Linguistic Annotation Workshop in ACL-2007*, page 132139.

Kenji Imamura, Kuniko Saito, and Tomoko Izumi. 2009. Discriminative approach to predicate-argument structure analysis with zero-anaphora resolution. In *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, pages 85–88, Suntec, Singapore, August. Association for Computational Linguistics.

P. Kingsbury, M. Palmer, and M. Marcus. 2002. Adding semantic annotation to the Penn TreeBank. In *Proceedings of the Human Language Technology Conference (HLT'02)*.

Karin Kipper. 2005. *VerbNet: A broad-coverage, comprehensive verb lexicon*. Ph.D. thesis, Department of Computer and Information Science University of Pennsylvania.

Mitchell Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of English: the Penn TreeBank. *Computational Linguistics*, 19:313–330.

Adam Meyers. 2007. Annotation guidelines for NomBank - noun argument structure for PropBank. Technical report, New York University.

Martha S. Palmer, Deborah A. Dahl, Rebecca J. Schiffman, Lynette Hirschman, Marcia Linebarger, and John Dowding. 1986. Recovering implicit information. In *Proceedings of the 24th annual meeting on Association for Computational Linguistics*, pages 10–19, Morristown, NJ, USA. Association for Computational Linguistics.

Patrick Pantel and Deepak Ravichandran. 2004. Automatically labeling semantic classes. In Daniel Marcu Susan Dumais and Salim Roukos, editors, *HLT-NAACL 2004: Main Proceedings*, pages 321–328, Boston, Massachusetts, USA, May 2 - May 7. Association for Computational Linguistics.

Rashmi Prasad, Alan Lee, Nikhil Dinesh, Eleni Miltsakaki, Geraud Campion, Aravind Joshi, and Bonnie Webber. 2008. Penn discourse treebank version 2.0. Linguistic Data Consortium, February.

P. Pudil, J. Novovicova, and J. Kittler. 1994. Floating search methods in feature selection. *Pattern Recognition Letters*, 15:1119–1125.

Vasin Punyakanok, Dan Roth, and Wen-tau Yih. 2008. The importance of syntactic parsing and inference in semantic role labeling. *Comput. Linguist.*, 34(2):257–287.

Josef Ruppenhofer, Caroline Sporleder, Roser Morante, Collin Baker, and Martha Palmer. 2009. Semeval-2010 task 10: Linking events and their participants in discourse. In *Proceedings of the Workshop on Semantic Evaluations: Recent Achievements and Future Directions (SEW-2009)*, pages 106–111, Boulder, Colorado, June. Association for Computational Linguistics.

Ryohei Sasano, Daisuke Kawahara, and Sadao Kurohashi. 2004. Automatic construction of nominal case frames and its application to indirect anaphora resolution. In *Proceedings of Coling 2004*, pages 1201–1207, Geneva, Switzerland, Aug 23–Aug 27. COLING.

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.