Parsing Arguments of Nominalizations in English and Chinese*

Sameer Pradhan, Honglin Sun, Wayne Ward, James H. Martin and Daniel Jurafsky

Center for Spoken Language Research,
University of Colorado, Boulder, CO 80303

Abstract

In this paper, we use a machine learning framework for semantic argument parsing, and apply it to the task of parsing arguments of eventive nominalizations in the FrameNet database. We create a baseline system using a subset of features introduced by Gildea and Jurafsky (2002), which are directly applicable to nominal predicates. We then investigate new features which are designed to capture the novelties in nominal argument structure and show a significant performance improvement using these new features. We also investigate the parsing performance of nominalizations in Chinese and compare the salience of the features for the two languages.

1 Introduction

The field of NLP had seen a resurgence of research in shallow semantic analysis. The bulk of this recent work views semantic analysis as a tagging, or labeling problem, and has applied various supervised machine learning techniques to it (Gildea and Jurafsky (2000, 2002); Gildea and Palmer (2002); Surdeanu et al. (2003); Hacioglu and Ward (2003); Thompson et al. (2003); Pradhan et al. (2003)). Note that, while all of these systems are limited to the analysis of verbal predicates, many underlying semantic relations are expressed via nouns, adjectives, and prepositions. This paper presents a preliminary investigation into the semantic parsing of eventive nominalizations (Grimshaw, 1990) in English and Chinese.

*This research was partially supported by the ARDA AQUAINT program via contract OCG4423B and by the NSF via grant IS-9978025

2 Semantic Annotation and Corpora

For our experiments, we use the FrameNet database (Baker et al., 1998) which contains *frame*-specific semantic annotation of a number of predicates in English. Predicates are grouped by the semantic frame that they instantiate, depending on the sense of their usage, and their arguments assume one of the frame elements or roles specific to that frame. The predicate can be a verb, noun, adjective, prepositional phrase, etc. FrameNet contains about 500 different frame types and about 700 distinct frame elements. Following example illustrates the general idea. Here, the predicate "complain" instantiates a "Statement" frame once as a nominal predicate and once as a verbal predicate.

Did [$_{Speaker}$ she] make an official [$_{Predicate:nominal}$ complaint] [$_{Addressee}$ to you] [$_{Topic}$ about the attack.]

[$_{Message}$"Justice has not been done"] [$_{Speaker}$ he] [$_{Predicate:verbal}$ complained.]

Furthermore, nominal predicates in FrameNet include ultra-nominals (Barker and Dowty, 1992), nominals and nominalizations. For the purposes of this study, a human analyst went through the nominal predicates in FrameNet and selected those that were identified as nominalizations in NOMLEX (Macleod et al., 1998). Out of those, the analyst then selected ones that were eventive nominalizations.

These data comprise 7,333 annotated sentences, with 11,284 roles. There are 105 frames with about 190 distinct frame role[1] types. A stratified sampling over predicates was performed to select 80% of this data for training, 10% for development and another 10% for testing.

For the Chinese semantic parsing experiments, we selected 22 nominalizations from the Penn Chinese Treebank and tagged all the sentences

[1]We will use the terms *role* and *arguments* interchangeably

containing these predicates with PropBank (Kingsbury and Palmer, 2002) style arguments – ARG0, ARG1, etc. These consisted of 630 sentences. These are then split into two parts: 503 (80%) for training and 127 (20%) for testing.

3 Baseline System

The primary assumption in our system is that a semantic argument aligns with some syntactic constituent. The goal is to identify and label constituents in a syntactic tree that represent valid semantic arguments of a given predicate.

3.1 Features We created a baseline system using a subset of the features introduced by Gildea and Jurafsky that are directly applicable to nominal predicates. Most of the features are extracted from the syntactic parse of a sentence. We used the Charniak parser (Chaniak, 2001) to parse the sentences in order to perform feature extraction. The features are listed below:

Predicate – The predicate itself is used as a feature.

Path – The syntactic path through the parse tree from the parse constituent being classified to the predicate.

Constituent type – This is the syntactic category (NP, PP, S, etc.) of the constituent corresponding to the semantic argument.

Position – This is a binary feature identifying whether the constituent is before or after the predicate.

Head word – The syntactic head of the constituent.

3.2 Classifier and Implementation We formulate the parsing problem as a multi-class classification problem and use a Support Vector Machine (SVM) classifier in the ONE *vs* ALL (OVA) formalism, which involves training n classifiers for a n-class problem – including the NULL class. We use TinySVM[2] along with YamCha[3] (Kudo and Matsumoto (2000, 2001)) as the SVM training and test software.

3.3 Performance We evaluate our system on three tasks: i) *Argument Identification*: Identifying parse constituents that represent arguments of a given predicate, ii) *Argument Classification*: Labeling the constituents that are known to represent arguments with the most likely roles, and iii) *Argument Identification and Classification*: Finding constituents that represent arguments of a predicate, and labeling them with the most likely roles. The baseline performance on the three tasks is shown in Table 1.

Task	P (%)	R (%)	$F_{\beta=1}$	A (%)
Id.	81.7	65.7	72.8	
Classification	-	-	-	70.9
Id. + Classification	65.7	42.1	51.4	

Table 1: Baseline performance on all three tasks.

4 New Features

To improve the baseline performance we investigated additional features that would provide useful information in identifying arguments of nominalizations. Following is a description of each feature along with an intuitive justification. Some of these features are not instantiated for a particular constituent. In those cases, the respective feature values are set to "UNK".

1. Frame – The frame instantiated by the particular sense of the predicate in a sentence. This is an oracle feature.

2. Selected words/POS in constituent – Nominal predicates tend to assign arguments, most commonly through postnominal of-complements, possessive prenominal modifiers, etc. We added the values of the first and last word in the constituent as two separate features. Another two features represent the part of speech of these words.

3. Ordinal constituent position – Arguments of nouns tend to be located closer to the predicate than those for verbs. This feature captures the ordinal position of a particular constituent to the left or right of the predicate on a left or right tree traversal, eg., first PP from the predicate, second NP from the predicate, etc. This feature along with the position will encode the before/after information for the constituent.

4. Constituent tree distance – Another way of quantifying the position of the constituent is to identify its index in the list of constituents that are encountered during linear traversal of the tree from the predicate to the constituent.

5. Intervening verb features – Support verbs play an important role in realizing the arguments of nominal predicates. We use three classes of intervening verbs: i) verbs of being – ones with part of speech AUX, ii) light verbs – a small set of known light verbs eg., make, take, have, etc., and iii) other verbs – with part of speech VBx. We added three features for each: i) a binary feature indicating the presence of the verb in between the predicate and the constituent ii) the actual word as a feature, and iii) the path through the tree from the constituent to the verb, as the subject of intervening verbs sometimes tend to be arguments of nominalizations.

[2] http://cl.aist-nara.ac.jp/~talus-Au/software/TinySVM/

[3] http://cl.aist-nara.ac.jp/~taku-Au/software/yamcha/

The following example could explain the intuition behind this feature:

[$_{Speaker}$ Leapor] *makes* general [$_{Predicate}$ assertions] [$_{Topic}$ about marriage]

6. Predicate NP expansion rule – This is the noun equivalent of the verb sub-categorization feature used by Gildea and Jurafsky (2002). This is the expansion rule instantiated by the parser, for the lowermost NP in the tree, encompassing the predicate. This would tend to cluster NPs with a similar internal structure and would thus help finding argumentive modifiers.

7. Noun head of prepositional phrase constituents – Instead of using the standard head word rule for prepositional phrases, we use the head word of the first NP inside the PP as the head of the PP and replace the constituent type PP with PP-<preposition>.

8. Constituent sibling features – These are six features representing the constituent type, head word and part of speech of the head word of the left and right siblings of the constituent in consideration. These are used to capture arguments represented by the modifiers of nominalizations.

9. Partial-path from constituent to predicate – This is the path from the constituent to the lowest common parent of the constituent and the predicate. This is used to generalize the path statistics.

10. Is predicate plural – A binary feature indicating whether the predicate is singular or plural as they tend to have different argument selection properties.

11. Genitives in constituent – This is a binary feature which is true if there is a genitive word (one with the part of speech POS, PRP, PRP$ or WP$) in the constituent, as these tend to be subject/object markers for nominal arguments. Following example helps clarify this notion:

[$_{Speaker}$ Burma 's] [$_{Phenomenon}$ oil] [$_{Predicate}$ search] hits virgin forests

12. Constituent parent features – Same as the sibling features, except that that these are extracted from the constituent's parent.

13. Verb dominating predicate – The head word of the first VP ancestor of the predicate.

14. Named Entities in Constituent – As in Surdeanu et al. (2003), this is represented as seven binary features extracted after tagging the sentence with BBN's IdentiFinder (Bikel et al., 1999) named entity tagger.

5 Feature Analysis and Best System Performance

5.1 English For the task of *argument identification*, features 2, 3, 4, 5 (the verb itself, path to light-verb and presence of a light verb), 6, 7, 9, 10 an 13 contributed positively to the performance. Interestingly the Frame feature degrades performance significantly. We trained a new classifier using all the features that contributed positively to the performance and the $F_{\beta=1}$ score increased from the baseline of 72.8% to 76.3% ($\chi^2; p < 0.05$).

For the task of *argument classification*, adding the Frame feature to the baseline features, provided the most significant improvement, increasing the classification accuracy from 70.9% to 79.0% ($\chi^2; p < 0.05$). All other features added one-by-one to the baseline did not bring any significant improvement to the baseline. All the features together produced a classification accuracy of 80.9%. Since the Frame feature is an oracle, we were interested in finding out what all the other features combined contributed. We ran an experiment with all features, except Frame, added to the baseline, and this produced an accuracy of 73.1%, which however, is not a statistically significant improvement over the baseline of 70.9%.

For the task of *argument identification and classification*, features 8 and 11 (right sibling head word part of speech) hurt performance. We trained a classifier using all the features that contributed positively to the performance and the resulting system had an improved $F_{\beta=1}$ score of 56.5% compared to the baseline of 51.4% ($\chi^2; p < 0.05$).

We found that a significant subset of features that contribute marginally to the classification performance, hurt the identification task. Therefore, we decided to perform a two-step process in which we use the set of features that gave optimum performance for the argument identification task and identify all likely argument nodes. Then, for those nodes, we use all the available features and classify them into one of the possible classes. This "two-pass" system performs slightly better than the "one-pass" mentioned earlier. Again, we performed the second pass of classification with and without the Frame feature.

Table 2 shows the improved performance numbers.

5.2 Chinese For the Chinese task, we use the one-pass algorithm as used for English. A baseline system was created using the same features as used for English (Section 3). We evaluate this system on just the combined task of argument identification and classification. The baseline performance is shown in Table 3.

Task	P (%)	R (%)	$F_{\beta=1}$	A (%)
Id.	83.8	70.0	76.3	
Classification (w/o Frame)	-	-	-	73.1
Classification (with Frame)	-	-	-	80.9
Id. + Classification (one-pass, w/o Frame)	69.4	47.6	56.5	
Id. + Classification (two-pass, w/o Frame)	62.2	53.1	57.3	
Id. + Classification (two-pass, with Frame)	69.4	59.2	63.9	

Table 2: Best performance on all three tasks.

To improve the system's performance over the baseline, we added all the features discussed in Section 4, except features Frame – as the data was labeled in a PropBank fashion, there are no frames involved as in FrameNet; Plurals and Genitives – as they do not exist in Chinese, and Named Entities – owing to the unavailability of a Chinese Named Entity tagger. We found that of these features, 2, 3, 4, 6, 7 and 13 hurt the performance when added to the baseline, but the other features helped to some degree, although not significantly. The improved performance is shown in Table 3

Task	P (%)	R (%)	$F_{\beta=1}$
Baseline	86.2	32.2	46.9
Baseline + more features	83.9	44.1	57.8

Table 3: Parsing performance for Chinese.

6 Conclusion

These preliminary results tend to indicate that the task of parsing arguments of nominalized predicates in English is more difficult than that of verb predicates. For a training set of 5,000 sentences annotated with verb arguments, the $F_{\beta=1}$ on a standard test set reported by (Pradhan et al., 2003a) was 71.9%, as opposed to that of 57.3% using approximately 7,500 sentences training data, a super-set of the features, a similar training algorithm, but a different corpus. Although the comparison is not exact, it gives some insight.

An interesting linguistic phenomenon was observed which explains part of the reason why recall for Chinese argument parsing is so low. In Chinese, arguments which are internal to the NP which encompasses the nominalized predicate, tend to be multi-word, and are not associated with any node in the parse tree. These violates our basic assumption of the arguments aligning with parse tree constituents, and are guaranteed to be missed. In the case of English however, these tend to be single word arguments which are represented by a leaf in the parse tree and stand a chance of getting classified correctly.

As noted earlier, the bulk of past research in this area has focused on verbal predicates. Two notable exceptions to this include the work of (Hull and Gomez, 1996) on identifying the semantic arguments to nominalizations, and the work of (Lapata, 2002) on interpreting the relation between the head of a nominalized compound and its modifier noun. Unfortunately, meaningful comparisons to these efforts are difficult due to differing evaluation metrics.

References

[Baker et al.1998] Collin F. Baker, Charles J. Fillmore, and John B. Lowe. 1998. The Berkeley FrameNet project. In *COLING/ACL-98*, pages 86–90, Montreal.

[Barker and Dowty1992] Chris Barker and David Dowty. 1992. Non-verbal thematic proto-roles. In *NELS-23, Amy Schafer, ed., GSLA, Amherst*, pages 49–62.

[Bikel et al.1999] Daniel M. Bikel, Richard Schwartz, and Ralph M. Weischedel. 1999. An algorithm that learns what's in a name. *Machine Learning*, 34:211–231.

[Chaniak2001] Eugene Chaniak. 2001. Immediate-head parsing for language models. In *ACL*, Toulouse, France.

[Gildea and Jurafsky2000] Daniel Gildea and Daniel Jurafsky. 2000. Automatic labeling of semantic roles. In *ACL*, pages 512–520, Hong Kong, October.

[Gildea and Jurafsky2002] Daniel Gildea and Daniel Jurafsky. 2002. Automatic labeling of semantic roles. *Computational Linguistics*, 28(3):245–288.

[Gildea and Palmer2002] Daniel Gildea and Martha Palmer. 2002. The necessity of syntactic parsing for predicate argument recognition. In *ACL*, PA.

[Grimshaw1990] Jane Grimshaw. 1990. *Argument Structure*. The MIT Press, US.

[Hacioglu and Ward2003] Kadri Hacioglu and Wayne Ward. 2003. Target word detection and semantic role chunking using support vector machines. In *HLT*, Edmonton, Canada.

[Hull and Gomez1996] Richard D. Hull and Fernando Gomez. 1996. Semantic interpretation of nominalizations. In *AAAI Conference*, Oregon, pages 1062–1068.

[Kingsbury and Palmer2002] Paul Kingsbury and Martha Palmer. 2002. From Treebank to PropBank. In *LREC-2002*, Las Palmas, Canary Islands, Spain.

[Kudo and Matsumoto2000] Taku Kudo and Yuji Matsumoto. 2000. Use of support vector learning for chunk identification. In *CoNLL-2000*, pages 142–144.

[Kudo and Matsumoto2001] Taku Kudo and Yuji Matsumoto. 2001. Chunking with support vector machines. In *NAACL*.

[Lapata2002] Maria Lapata. 2002. The disambiguation of nominalizations. *Computational Linguistics*, 28(3):357–388.

[Macleod et al.1998] C. Macleod, R. Grishman, A. Meyers, L. Barrett, and R. Reeves. 1998. Nomlex: A lexicon of nominalizations.

[Pradhan et al.2003a] Sameer Pradhan, Kadri Hacioglu, Valerie Krugler, Wayne Ward, James Martin, and Dan Jurafsky. 2003a. Support vector learning for semantic argument classification. Technical Report TR-CSLR-2003-3, CSLR, Boulder, Colorado.

[Pradhan et al.2003] Sameer Pradhan, Kadri Hacioglu, Wayne Ward, James Martin, and Dan Jurafsky. 2003. Semantic role parsing: Adding semantic structure to unstructured text. In *ICDM*, Melbourne, Florida.

[Surdeanu et al.2003] Mihai Surdeanu, Sanda Harabagiu, John Williams, and Paul Aarseth. 2003. Using predicate-argument structures for information extraction. In *ACL*, Sapporo, Japan.

[Thompson et al.2003] Cynthia A. Thompson, Roger Levy, and Christopher D. Manning. 2003. A generative model for semantic role labeling. In *ECML*.