Semi-supervised Prediction of ARG1 of Partitive Nouns in The NomBank

Sohail Pandurang Hodarkar
sph8686@nyu.edu

Euijae Kim
ek3955@nyu.edu

December 22, 2022

Abstract

This paper introduces a method to annotate the ARG1 of a partitive noun using a semi-supervised learning approach. A well-trained classifier to predict the ARG1 of a partitive noun based on a significant amount of data, in general, remarkably increase the accuracy of a test result. Thus, the simplest approach to achieve this experiment would be to make use of sophisticated supervised learning models. However, a major caveat of this is the requirement of manually annotated data. This is clearly infeasible, owing to the extremely voluminous nature of textual information. A workaround is to incorporate some form of Active Learning, which is far less-stringent with the requirement of pre-labeled data. In this paper, we briefly touch upon the rationale behind active learning and proceed to utilize it for our task of ARG1 identification.

1 Introduction

The NomBank [6] is an annotation project at New York University that provides us with insight into argument structure of nominal predicates in the Wall Street Journal corpus, at large. For the course of this study, we will be limiting ourselves to one of the most commonly encountered types of nominal predicates: partitives. Our experiment was concerned with a very specific task- predict the ARG1s of these partitive nouns. We modelled the problem of learning the noun-argument relations, as binary classification task. While researching the various possible systems that can be used to solve similar problems, we decided to address another issue of such voluminous datasets- availability of pre-annotated data. This was addressed by utilizing an active learning-inspired framework. It is a ubiquitous fact that the quality of features chosen is directly proportional to the efficacy of the learning system. Therefore, it was of paramount importance that we cherry-picked these crucial components. Considering that NomBank has a limited set of predefined features (POS tag, BIO tag, position of token in the sentence, and sentence number) we focused on generating a vast range of additional features. These included the introduction of "dummy" tokens, POS Tags, and BIO Tags to capture the information pertaining to a sentence break, additional distance-based features, word embedding-based features to factor for the context in which tokens appear, and some path-based heuristics. A careful analysis of the effect of the addition of these features

was carried out, on the route to obtaining a final system. The predictions made, was then compared with the available annotated data, thereby allowing us to quantitatively evaluate the performance of our system, by computing metrics such as accuracy, precision, recall, and f-measure. The results of the overall, and intermediate systems, have been extensively later in the paper.

2 Related Work

2.1 Semantic Role Labeling

Semantic Role Labeling (SRL) has been an active area of research in Natural Language Processing, that deals with detecting the predicate-argument structures in sentences. While a number of systems have been developed to incorporate computation in identifying these relations, most of them have been dealing with verbs as predicates [1]. However, it has also been observed that nouns appear with closely related words, which, in some sense, can be considered as its argument. The exercise of annotating these noun-argument structures has been extensively carried out at New York University [5]. Previously, efforts have been made to utilize machine learning techniques for the purpose of performing semantic role labeling on the NomBank [3].

2.2 Active Learning

The central idea is to build an efficient classification system, while using substantially lesser manually-annotated data. It is an upcoming semi-supervised learning approach, which relies on an annotator being able to provide the system with the labels of requested samples [4]. These samples can be picked in broadly three manners:

Pool-based Selective Sampling: Here, we label a small subset of the data to begin with. The classifier is trained on this, and the requested samples are those which are not-so-confidently predicted. This has been seen to have applied to text classification tasks, in the recent past.

Stream-based sampling : System is provided with a stream of unlabeled data. A decision on whether to accept the sample in its unlabeled form or query the annotator for its label is made.

Membership query synthesis: The system generates synthetic samples by itself and queries the annotator for their labels.

3 Our Methodology

3.1 Description of the Data

For the purpose of this project, we utilized the **NomBank** corpus provided to us as part of the course. The data consisted of pre-split, pre-cleaned training (partitive_group_nombank.clean.train), development (partitive_group_nombank.clean.dev),

and test (partitive_group_group_nombank.clean.test) files. Table 1 shows the distribution of ARG1s in the training, development, and test data. From this exploratory analysis, it is clear that the dataset is extremely skewed towards the class of Non-ARG1s.

Dataset	**ARG1s**	**Non-ARG1s**	**Total**	**ARG1s : Non-ARG1s**
Training	9,979	311,754	321,733	0.032
Development	372	11,620	11,992	0.032
Test	606	18,376	18,982	0.031

Table 1: Distribution of ARG1s in the Data

3.2 Feature Engineering

In order to build an effective classification system, it is paramount that we pick the right set of cleverly-engineered features. However, we were keen on analyzing the impact of each of these features. To this end, we decided to build our system incrementally, while adding a new set of features each time. In order to understand the effect of each of these feature classes, we shall now categorize them.

Baseline Features: Our first working system was built using a fairly rudimentary set of features. They are elucidated below.

Token	POS Tag
BIO Tag	Stemmed Token
Relative Position of the Token	Capitalization
Distance from Predicate	Previous Token
Previous POS Tag	Previous BIO Tag
Previous Stemmed Token	Previous of Previous Token
Previous of Previous POS Tag	Previous of Previous BIO Tag
Previous of Previous Stemmed Token	Next Token
Next POS Tag	Next BIO Tag
Next Stemmed Token	Next to Next Token
Next to Next POS Tag	Next to Next BIO Tag
Next to Next Stemmed Token	Predicate Class

Table 2: Baseline Features

Modified Tags: Instead of omitting the previous (forward) features when closer to the beginning (end) of the sentence, we opted to include "dummy" features to indicate sentence breaks. As we shall see later, these did have a induce a slight improvement in the overall performance of the system. These features have been described below.

Feature	Description
Begin_Sentence	Token before the first token in a sentence
Begin_Sentence_One	Token before Begin_Sentence
Begin_Sent	POS Tag of Begin_Sentence
Begin_Sent_One	POS Tag of Begin_Sentence_One
Begin_Sen	BIO Tag of Begin_Sentence
Begin_Sen_One	BIO Tag of Begin_Sentence_One
End_Sentence	Token after the last token in a sentence
End_Sentence_One	Token after End_Sentence
End_Sent	POS Tag of End_Sentence
End_Sent_One	POS Tag of End_Sentence_One
End_Sen	BIO Tag of End_Sentence
End_Sen_One	BIO Tag of End_Sentence_One

Table 3: Modified Tags

Additional Distance Features: These were included to account for the proximity of the token under consideration, to important parts of speech. A list of these features is given below.

Absolute Distance from Closest Noun
Absolute Distance from Closest Conjunction
Absolute Distance from Closest Preposition

Table 4: Additional Distance Features

Embedding Features: It is intuitive to think of ARG1s to be appearing in similar contexts as other ARG1s. This similarity can be captured using the cosine similarity between the embedding vector of a token and the average embedding vector of all ARG1s in the training corpus. The resultant features are listed below.

Backward Trigram Similarity
Backward Bigram Similarity
Unigram Similarity
Forward Trigram Similarity
Forward Bigram Similarity

Table 5: Embedding Features

Slash Embedding Features: These features are very structurally similar to the previous class, except that they capture the context of a token (or a sequence of tokens) by excluding it (them) from the embedding. The features from this class, which were introduced at a later stage have been listed below.

Slash Backward Trigram Similarity
Slash Backward Bigram Similarity
Slash Unigram Similarity
Slash Forward Trigram Similarity
Slash Forward Bigram Similarity

Table 6: Slash Embedding Features

Chunk-based Path Features: It is important to note that this is in no way, an accurate representation of the path from a parser. Instead, it is an approximation based on the sequence of BIO Tags in moving from the current token to the predicate. The features used, are listed below.

Path Direction
Chunk-based Path with Collapsed Tags

Table 7: Chunk-based Path Features

3.3 Algorithmic Details

Our algorithm relies on the inherent pool-based selective sampling variant of active learning. We began with a pool of labeled data, that was roughly equivalent to 10% of the entire training data set. An AdaBoost classifier [2] was then trained on this pool. At the end of the training phase, we picked a fixed number of samples that the built classifier was not "confident" about. These were manually labeled and added to the pool of initially labeled data. The classifier was then re-trained on the newly updated pool. This process was continued for a fixed number of iterations.

The question of assessing the "confidence" of a classifier was reduced to analyzing the predicted probabilities of the sample belonging to either class. If these probabilities belonged to a pre-defined interval (0.35 to 0.65), the sample was included in a set of "not-confidently predicted" samples. The fixed number of samples for updating the pool were then picked from this set.

3.4 Evaluation Metrics

An important aspect of assessing the performance of our system was to identify appropriate evaluation metrics. Perhaps the most obvious choice was to consider accuracy. However, with the extreme imbalance in the ARG1 distribution repeatedly highlighted, that idea was quickly discarded. We then settled on a combination of three widely-used metrics which serve as good all-around representative measures for the performance of a system - precision, recall, and f-measure. From here on, we treat ARG1s as the positive class, and, naturally, non-ARG1s as the negative class. We refer to the correctly-predicted positive classes as True Positives (TP), incorrectly-predicted positive classes as False Positives (FP), correctly-predicted negative classes as True Negatives (TN),

and incorrectly-predicted negative classes as False Negatives (FN). With that understanding, the chosen performance measures are defined as follows:

$$Precision = \frac{TP}{TP+FP} \quad Recall = \frac{TP}{TP+FN} \quad F-Measure = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

4 Experimental Results and Observations

Having trained the system on the provided training set, fine-tuned the model to the given development set, and finally made predictions on the unseen test set, a summary of the high-level results obtained by our AdaBoost-inspired, pool-based active learning system is given in Table 8.

Metric	**Result**
Precision	72.82 %
Recall	43.10 %
F-measure	54.15 %

Table 8: High-Level Summary of Experimental Results

As described earlier, we began with a baseline set of features and kept adding to it those additional features which we thought would contribute to boosting the performance of the system. While some features did aid the system in making better predictions, others showed little to no improvement. Amongst those which had a positive impact, some had a more significant role to play than others. A more detailed, and quantified, analysis of the effect of adding certain features has been provided in Table 9.

System	**Precision**	**Recall**	**F-Measure**
S1 = Baseline	43.88 %	25.32 %	32.11 %
S2 = S1 + Modified Tags	44.56 %	25.49 %	32.43 %
S3 = S2 + Additional Distance Features	46.91 %	27.52 %	34.69 %
S4 = S3 + Embedding Features	72.32 %	35.41 %	47.54 %
S5 = S4 + Slash Embedding Features	71.46 %	41.05 %	52.15 %
Final = S5 + Chunk-based Path Features	72.82 %	43.10 %	54.15 %

Table 9: Quantified Analysis of the Impact of Features

It can be observed from the above table, that the addition of embedding-based features was responsible for a considerable improvement in the system's overall performance. While slash embeddings were shown to impact the recall and f-measure positively, a marginal drop in the precision was observed on their inclusion. From a broader viewpoint, the introduction of additional distance features was seen to have a slight improvement on the system's ability to make predictions.

5 Conclusions

Our work, although focusing on the specific task of detecting ARG1s of partitive nouns in the NomBank, attempts to address the much larger issue of sourcing large-scale labeled data for the purpose of building classification systems. While we did limit ourselves to a renowned classifier i.e. AdaBoost, we spent most of our time trying to mine crucial information from the data and incorporate it as features in some way or the other. One key takeaway is that despite achieving decently high precision, it is the recall that lets the system down. In other words, it appears as though the system is largely underestimating the positive class. While the results may not have turned out to be extraordinarily impressive, we believe that our approach of chipping away from traditional supervised learning and transitioning towards a technique that is less reliant on annotated data is an effective way of building systems for similar expansive datasets.

6 Future Work

A semi-supervised learning paradigm can be thought of as one having several intricacies. Any form of experimentation with these intricacies is sure to serve as a template for potential research in the time to come. An attempt to incorporate sophisticated algorithms (designed to handle the severe imbalance in data), would be a good starting point. Considering that we explored the pool-based selective sampling variant of active learning, it would be interesting to examine the efficacy of a stream-based classification approach given the extremely skewed nature of the chosen corpus. Moreover, semi-supervised learning can be viewed as a stepping stone toward extending the ARG1 detection system to other unannotated corpora, such as the popular Brown Corpus.

References

[1] Xavier Carreras and Lluís Màrquez. Introduction to the CoNLL-2005 shared task: Semantic role labeling. In *Proceedings of the Ninth Conference on Computational Natural Language Learning (CoNLL-2005)*, pages 152–164, Ann Arbor, Michigan, June 2005. Association for Computational Linguistics.

[2] Xavier Carreras, Lluís Màrquez, and Lluís Padró. Named entity extraction using AdaBoost. In *COLING-02: The 6th Conference on Natural Language Learning 2002 (CoNLL-2002)*, 2002.

[3] Zheng Ping Jiang and Hwee Tou Ng. Semantic role labeling of NomBank: A maximum entropy approach. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 138–145, Sydney, Australia, July 2006. Association for Computational Linguistics.

[4] François Laviolette, Mario Marchand, and Sara Shanian. Selective sampling for classification. In *Advances in Artificial Intelligence*, pages 191–202. Springer Berlin Heidelberg, 2008.

[5] Adam Meyers, Ruth Reeves, Catherine Macleod, Rachel Szekely, Veronika Zielinska, Brian Young, and Ralph Grishman. Annotating noun argument structure for NomBank. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04)*, Lisbon, Portugal, May 2004. European Language Resources Association (ELRA).

[6] Adam Meyers, Ruth Reeves, Catherine Macleod, Rachel Szekely, Veronika Zielinska, Brian Young, and Ralph Grishman. The NomBank project: An interim report. In *Proceedings of the Workshop Frontiers in Corpus Annotation at HLT-NAACL 2004*, pages 24–31, Boston, Massachusetts, USA, May 2 - May 7 2004. Association for Computational Linguistics.