Semantic Role Labeling of NomBank Partitives

Adam Meyers
New York University
`meyers@cs.nyu.edu`

Advait Pravin Savant
New York University
`as14634@nyu.edu`

John E. Ortega
Northeastern University
`j.ortega@northeastern.edu`

Abstract

This article is about Semantic Role Labeling for English partitive nouns (*5%/REL of the price/ARG1*; *The price/ARG1 rose 5 percent/REL*) in the NomBank annotated corpus. Several systems are described using traditional and transformer-based machine learning, as well as ensembling. Our highest scoring system achieves an F1 of 91.74% using "gold" parses from the Penn Treebank and 91.12% when using the Berkeley Neural parser. This research includes both classroom and experimental settings for system development.

1 Introduction

Semantic Role Labeling (SRL), provides a way to represent semantic concepts via labeled predicate/argument pairs.

For example, an SRL analysis of the following three sentences include a **patient** relationship between the predicate: *break* and the argument *glass*. The 3rd sentence includes an additional **agent** relationship between the predicate *break* and the argument *John*.

- The glass broke
- The glass was broken
- John broke the glass.

SRL has become popular in linguistics and NLP following Jeffrey Gruber's dissertation (Gruber, 1965). His study focused on arguments of the main verb in a sentence (*They/Agent bought/Predicate a car/Theme*). Since then, the scope of semantic role labeling has widened to cover arguments of both verbs and nonverbs. See, for example, work on PropBank, Nombank, the Penn Discourse Treebank, FrameNet among other projects (Palmer et al., 2005; Meyers et al., 2004b; Fillmore and Atkins, 1998; Miltsakaki et al., 2004). Predicates can be not only verbs, but also nouns, subordinate conjunctions, adverbs or adjectives.

1. Mary's/**ARG0** request/**REL** for a loan/**ARG1** → Mary requested a loan
2. His/**ARG0** promise/**REL** to improve/**ARG2** → He promised to improve
3. She/**ARG0** made/SUPPORT demands/**REL** on her staff/**ARG2** → She demanded that her staff (do something)

Figure 1: Sample Nominalizations

Quant: A set/**REL** of tasks/**ARG1**
Part: 11 components/**REL** to the index/**ARG1**
Meronym: Anthers/**REL** in these plants/**ARG1**
Group: The Airline/**ARG2** creditors/**ARG1** committee/**REL**
Share: Her/**ARG0** portion/**REL** of pie/**ARG1**

Figure 2: Partitive Noun Examples

This article is about a semantic role labeling (SRL) task based on a subset of NomBank (Meyers et al., 2004a,b): the identification of **ARG1** arguments of predicates belonging to the five NomBank classes in Figure 2. Collectively, we call these **partitive nouns**. In PropBank and NomBank (Palmer et al., 2005; Meyers et al., 2004a) propositions consist of a predicate (REL) and one or more aguments (ARG0, ARG1, ARG2, ...) of that predicate. 13K out of the 115K NomBank propositions are partitive examples. There are a total of about 8K different noun predicates in NomBank, including about 500 partitives. This article focuses on automatically identifying arguments of partitive nouns in the NomBank corpus. We discuss several student systems that were created during a graduate class, a baseline system, student-inspired modifications to the baseline and some ensemble systems. All development assumed gold parses. However, we also tested our final systems based on output of the Berkeley Neural parser (Kitaev and Klein, 2018).

2 Partitive Nouns in an SRL Framework

SRL can model how different sentences are nearly equivalent semantically, as illustrated in Fig-

ure 1. Similarly, at the sentence level, the relations **ARG0(eat, clam)** and **ARG1(eat, tourists)** can be realized by the following sentences, all containing forms of *eat, clam,* and tourists:

- The giant clam ate the tourists.
- The tourists were eaten by the giant clam.
- The tourists, who were eaten by the the giant clam, were very wealthy.
- The giant clam, after eating tourists, left town.
- The giant clam wanted to eat the tourists.

The particular labels used (like **ARGO** and **ARG1**) vary between frameworks, although we adopt the nomenclature of PropBank and Nom-Bank.

We mark the nominal predicate with the label **REL** and arguments with argument labels like **ARG1**, placed at the end of the phrase, e.g., in Figure 2, example 1, there is a **PART** relation between the predicate *components* and the argument *the index*. In the example, the **REL** and **ARG1** labels mark the predicate and the argument (via the head words). Equivalently, we can say that the relation: **Part(component,index)** holds. We also ignore the question of how the heads or extents of phrases are chosen. We will follow a convention in which the argument is referred to by its label, i.e., if a relation *ARG0(eat,clam)* holds, we can refer to *the clam* as the **ARG0** of *eat*.

Partitive noun predicates quantify over their **ARG1** so that the relation represents a group of **ARG1**s or a part of an **ARG1**. Subclasses include: **quant**, **part**, **meronym**, **group**, and **share**, as in Figure 2. Partitive nouns have several features in common. Partitive nouns tend to be transparent: this means that the **ARG1** tends to act as a "semantic" head. For example, *a variety of sandwiches* is interpreted as an instance of *sandwiches* and thus edible, rather than an instance of *variety* and thus too abstract to eat. Coordinate conjunctions (CCs) like *and* and *or* are similar in this respect – the conjuncts, not the CCs are semantic heads. Thus the phrase *the sandwich and the apple* is an instance of *foods* (a generalization of the two conjuncts), not the coordinate conjunction *and* (the word that links the other words in the phrase together).

This paper is about several SRL systems for detecting ARG1s of partitive nouns.

3 Previous Work

3.1 Previous SRL Annotation

Some previous work using NomBank SRL (Surdeanu et al., 2008; Hajič et al., 2009) focuses on nominalizations (nouns related to verbs as in Figure 1) as part of a larger task that includes verbs and their arguments). Other work (Jiang and Ng, 2006) uses names of frames for non-nominalizations as features for machine learning. Our approach partitions annotation by classes of nouns that are not related to verb-frames. This approach is similar to FrameNet (Baker et al., 1998). Arguments of nouns sharing the same frame are treated like separate tasks. For example, FrameNet's **contingency** frame is shared by verbs, nouns and adjectives in the following examples:

- Success/OUTCOME may depend/VB on available resources/DETERMINANT
- Success/OUTCOME seems dependant/JJ on available resources/DETERMINANT
- Success/OUTCOME will be a function/NN of the Available resources/DETERMINANT

FrameNet Systems attempt to find sets of predicates and arguments belonging to particular frames. Shared Tasks such as ACE (Doddington et al., 2004) also follow this approach. Frame-like classes in NomBank are noun-centric. Figure 2 provides examples of the five partitive classes. In addition, there are 12 other non-nominalization frame-like classes in the NomBank dataset. (See figure 5 in Appendix A. We leave these to future work.) This paper provides a framework for such investigations.

In the NomBank (Meyers et al., 2004a) project, approximately 115,000 noun predicates were annotated, along with their arguments. The annotation scheme is compatible with the previous PropBank (Palmer et al., 2005) annotation of verbs and their arguments. Most subsequent automatic SRL systems (Jiang and Ng, 2006) treat NomBank propositions in a similar manner as PropBank. They use frames associated with each predicate, but they do not generalize over shared frames. For example, ARG1s of all 8000 distinct partitive predicate lemmas are assumed to form 8000 distinct classes, whereas our approach generalizes ARG1s of these 8000 predicate lemmas to a single class.

3.2 Previous SRL Systems

Most SRL systems fall into two categories. One approach is to develop a feature space from the input text based on linguistic analysis, and then extrapolate patterns via predictive modeling from feature space to output space. In the second approach, a deep learning system is created based on distributed word representations and feature spaces learned from parameterized transformations

Citation	Detail	F1
(Collobert et al., 2011)	CNN	77.92
(Zhou and Xu, 2015)	stacked LSTMs	81.07
(He et al., 2017)	BiLSTM	83.2
(Ouchi et al., 2018)	span-based BiLSTM	87.0

Figure 3: Neural Network SRLs for CONLL 2005 data

(Goodfellow et al., 2016). These systems mainly identify relations between verb predicates and noun arguments. Figure 3 summarizes the results of several neural network systems on the CONLL 2005 (Carreras and Màrquez, 2005) WSJ and Brown datasets.

(Pradhan et al., 2005) and (Xue and Palmer, 2003) report results on PropBank (Kingsbury and Palmer, 2002) data. (Xue and Palmer, 2004) experiment with various linguistic features, using representations of syntactic parse trees for use in SRL (F score 88.51 with gold parses, F score 76.21 with (Collins, 1999) parses). (Pradhan et al., 2005) develop an SRL system, integrating different syntactic views, based on experiments with CCG parsers and Charniak parsers (F score 89.4 with gold parses, F score 79.3 with automatic parses).

(Jiang and Ng, 2006) work with Nombank data. They develop Nombank specific features in view of linguistic considerations, incorporate parse tree based features. A maximum entropy classification model is trained based on the generated features. For every Nombank class, there is a feature which is used denote the class. For every nominal predicate, for corresponding words in the sentence, the class of the predicate is incorporated in the feature space. The class of a nominal predicate is seen to be indicative of the role of its arguments. During testing, in order to enforce non-overlapping arguments, an algorithm maximizes the log probability of the labeled parse tree (F score 72.73 with gold parses, F score 69.14 with automatic parses).

4 Research in a Classroom Environment

We used % and partitive tasks as assignments for three graduate NLP classes (Spring 2012, Spring 2022 and Fall 2023). The % task was assigned as one of several shared-task homework assignment during the first half of the semester. During the second half of the semester the partitive task was a final project task, the basis of a final paper and final presentation. The Fall 2023 version of the task is the basis of this article. Students were permitted to use any methods they wanted for the task. For the partitive task, they were free to use our baseline

system as a starting point (results are in table 1 and the **feature based 1** rows of table 2. Our final results include the incorporation of aspects of student systems, as discussed below.

5 Data and Evaluation Methodology
5.1 Data Format

For each annotated NomBank predicate in a subset of the Penn Treebank 2 WSJ corpus (PTB), we created a representation of the sentence containing that word. The representation includes information from both PTB and NomBank. We created datasets for various classes of NomBank predicates. In this paper, we will focus on systems trained and tested on the subset of NomBank devoted to partitive nouns, leaving NomBank subsets for nouns with other frames for future work.

We represent each sentence containing a partitive noun predicate as a set of tuples, one tuple for each token in the sentence. Each tuple is on one line in the data file, with blank lines between sentences. This format is an extension of the CONLL 2000 format (Tjong Kim Sang and Buchholz, 2000).

We eliminated problematic, but rare types of examples from the data. Since it is rare for a sentence to contain two partitive nouns (less than 1% of all sentences with partitives), we only kept the first instance of the selected predicate type (partitive) in any given sentence. We also eliminate rare cases in which NomBank's tokenization is at odds with the PENN Treebank's tokenization, e.g., the single PTB token **warehouse-club** is divided into two tokens in NomBank: **club** is a **group** noun (a subtype of partitive) and **warehouse** is its ARG1.

Figure 4 represents a sentence containing the word **percent** and its **ARG1 Output**. Each token has the following fields: **WORD**, **part of speech (POS)**, **CHUNK Begin-Inside-Outside (BIO) tag**, **TOKEN number**, **PRED/ARG label** and **FRAME label**.

We assume two versions of each tasks. For the "gold" version, the system starts with parses and POS tags from the Penn Treebank. For the "non-gold" version, we use parses and POS tasks

WORD	POS	BIO	#	FUNC	FRAME
Output	NN	B-NP	0	ARG1	
in	IN	B-PP	1		
August	NNP	B-NP	2		
rose	VBD	O	3	SUP	
5	CD	B-NP	4		
percent	NN	I-NP	5	PRED	QUANT
.	.	O	6		

326 Figure 4: Annotation of *Output in August rose 5 percent.*

from the Berkeley Neural parser (Kitaev and Klein, 2018). Both tasks use chunk tags, generated as per (Tjong Kim Sang and Buchholz, 2000), the predicate noun and its classes from Nombank, and support verbs, as per NomBank. We represent NPs as their head nouns. We represent PP ARG1s as the head of the PP object, e.g., in **5 portions of that food**, the word *food* will be recorded as the ARG1 of the *group* noun *portion*, not the preposition *of*. The "gold" versions test the SRL systems independently of parsing error, where as the non-gold systems provide a realistic testbed for SRL performance from raw text. Both types of systems exist for much previous work in SRL.

We initially assume that predicting the ARG1 may be a function of: properties of the head noun argument; properties of the predicate nominal; properties of the support verb (if present) and properties of paths between key items (token-based paths and parser-based paths). The latter should include paths between the nominal predicate and ARG1 noun, as well as paths between the support verbs and ARG1 nouns. The sample feature set for our baseline system (section 6) takes this approach.

Figure 4 simplifies the SRL task in some respects. The simplification make is easy to test machine learning (M) algorithms and different sets of features, with the goal of predicting which nouns (heads of noun phrases) are ARG1s of predicate partitive nouns. As noted, the gold data makes the task somewhat artificial, presumably resulting in higher scores than would be possible with similar, but automatically obtained features.

5.2 Organization of Datasets

In the Penn Treebank, the % sign is a noun (just like the word "percent"). It is a partitive and it occurs with an ARG1 nearly 3,000 times, making it the most frequent nominal predicate in the corpus. These instances of % are of course part of the approximately 13,000 partitive instances, the most frequent shared frame in NomBank.

For our experiments, we created two datasets, one consisting of % instances and a larger one consisting of partitives. We divided each set into three subcorpora: training data consists of instances from PTB directories 02 to 21; development data comes from directory 24 and test data comes from 23 (a standard data split for this corpus). The training corpus contained 1K partitive ARG1s and 2225 % ARG1s; Dev contained 370 and 87; Test contained 550 and 150.

We would expect most systems to do better on the % subcorpus than the partitive subcorpus, because the former is based on arguments of the same predicate, while the latter is based on a class of predicates with some elements of meaning in common. Thus the students first made systems for the % task and then for the partitive task.

For initial development, we evaluate systems only on gold versions of tasks (manual parses). However, for our final results, we evaluate on both gold and non-gold versions (parser output).

5.3 Evaluation Method

We report precision, recall and f-scores of ARG1 in tables 1 and 2. There is exactly one ARG1 for each sentence, as per the data format. For example, in Figure 4, *Output* is labeled ARG1.For most cases, we choose the head noun of the ARG1, e.g., if the ARG1 phrase is **The big prize**, a correct match requires that the system identify **prize** as the ARG1. For proper noun phrases, e.g., **Exxon Mobil Corp.**, we assume that any of the name words (NNP) is correct (Exon or Mobil or Corp.).

6 Baseline System

We created a baseline system as both a proof of concept and a starting point for student systems. Through error analysis of the baseline system, the class may better understand the task. Furthermore, it is possible that an improved system could be created both by modifying the baseline system based on features from student systems and/or ensembling it with student systems. First we used Sklearn's Adaboost machine learning algorithm (Freund and Schapire, 1995), with features generated for each token in each sentence. The model was generated from our training corpus and it was used to predict the ARG1 (true or false) status of tokens in the development and test corpora. All experimentation with different features was measured against the development set. Prior to submission of this paper, we ran one system on the test corpus, the one that performed best on the development set.

6.1 Baseline Features and their Motivations

In this section, we enumerate the features used in the baseline system. These features are motivated by our linguistic understanding of SRL with noun predicates in general, and partitives in particular. This includes both: information that characterizes typical ARG1s and predicate nouns (the head words, POS classes, BIO tags, nearby words,

327

frame classes of predicate nouns, etc.) and information about how ARG1s are related to the predicates (token distance, tree distance, etc.). The baseline system provides a testing ground for such features.

6.1.1 Simple ARG1 Features

Simple ARG1 features include the **word** (head of ARG1), its **POS** tag, its **BIO** tag, and these same features for the two words following and preceding the head. Following Firth (Firth, 1957) and subsequent NLP research, we are assuming that words are, in part, defined by their context. Thus features of neighboring words are assumed to bear on meaning of the current ARG1.

6.1.2 Embedding-based Features

Word embeddings, generated from large corpora, are also compatible with Firth's principle because each word is defined based on a neural network model of the context of that word. We use pre-trained embeddings from SPACY's pre-trained en_core_web_md tok2vec embeddings which was from a medium sized (43 mb) corpus. For each example ARG1 in the training corpus, we calculate a total of 10 embeddings: two types of embeddings for each of five different n-grams. The five n-grams include: the head ARG1 word itself, and the forward and backward bigrams and trigrams. For example, for the sentence: *The consumer price index/***ARG1** *rose five percent/***REL.**, five n-grams are generated: 1. **consumer price index**, 2. **price index**, 3. **index**, 4. **index rose**, and **index rose five**. For each n-gram, we calculate two embeddings: the normal embedding for that n-gram and and the embedding of all words in the sentence with the n-gram removed. We will refer to the second type of embedding as a **slash** embedding.For each of these 10 types of embeddings, we calculate an average embedding based on the training file. For each ARG1 in the corpus, we find how similar (cosine similarity) the 10 embeddings are to these averages. These similarities correspond to 10 feature values.

6.1.3 Predicate Class Features

The specific partitive class of the noun, as well as any other NomBank class the predicate is labeled with, e.g., **GROUP, MERONYM, PART, QUANT, SHARE, BOOK-CHAPTER, BORDER, CONTAINER, DIVISION, ENVIRONMENT, INSTANCE-OF-SET, NOM, NOMADJ PART-OF-BODY-FURNITURE-ETC, SHARE, WORK-OF-ART**. These labels are taken from the lexical entries of each predicate. The labels for well-defined classes (the classes exemplified in Figures 2 and 5) are used consistently according to the aforementioned specifications. The remaining predicate labels are chosen by the annotators during the creation of NomBank. None of these features are used for the **%** task, since these values would be the same for all instances of **%**.

6.1.4 Path Features

Simple path features are based on token distance: number of tokens between predicate and ARG1, tokens between support and ARG1. Positive and negative values reflect the ARG being before or after the support verb or nominal predicate.

In the initial baseline systems (**Feature based 1 and 2** in Table 2, we approximate paths in the parse tree, based on sequences of BIO tags, collapsing BI* to phrase sequences and including the direction. For example, consider the following sequence with BIO tags from a predicate to its ARG1:

20\B-NP %\I-NP of\B-PP the\B-NP pie\I-NP

The path feature generated from this sequence is:

right_NP_PP_of_NP_NOUN

This characterizes the path from a head noun to the object of a complement of the preposition **of**. Thus different left and right tags characterize paths between predicates and arguments, and between support verbs and arguments. These path features are derived by essentially collapsing sequences of BI* into phrases, e.g., B-NP I-NP I-NP $\rightarrow$ NP. The path features encode simulated paths as *type 1 path features* in order to differentiate them from *type 2 path features*. We will henceforth refer to this method as **Path-heuristic 1**.

For **feature-based system 3**, we use a parse tree for generating paths between words. We developed **Path heuristic 2**, based on observations about sentence structure. This heuristic represents classes of observed grammatical relationships between: the predicate noun and the (head of) ARG1; and also between a support verb and the (head of) ARG1.

Path heuristic 2 labels one of four paths for compatible (Manning et al., 2014) parse trees:

PATH1 From the support verb to a preceding ARG1, we assume $\uparrow$ **VP** $\uparrow$ **S** $\downarrow$ **NP**

PATH2 From the support verb to a following ARG1, we assume $\uparrow$ **VP** $\downarrow$ **NP**

PATH3 From a predicate noun to a preceding ARG1 (with no support verb), we assume ↑ **NP** ↓ **NP**

PATH4 From a predicate noun to a following ARG1. Derive the path from the sequence of BIO tags from the predicate noun to the ARG1 (similar to the path heuristic used in **Feature based systems 1 and 2**). This includes the common path from predicate to ARG1: ↑ **NP** ↓ **PP** ↓ **NP** (**5% of the price.**)

For example, consider the sentence. "The price rose 5 percent." and the parse tree:

```
(S
  (NP (DT The) (NN price))
  (VP (VBD rose)
    (NP (CD five) (NN percent)))))
```

Here, "percent" is the predicate, "rose" is a support verb and "price" is the ARG1. For words preceding the support verb "rose", we can investigate the application of PATH3, which estimates paths between a support verb and a preceding ARG1. Starting with "rose", we go up a verb phrase (↑VP), then down a noun phrase (↓NP). Similarly, we could apply PATH2 in the following tree (for **They increased the price five percent**):

```
(S
  (NP (PRP They)
  (VP (VBD increased)
      (NP (DT the) (NN price))
      (NP (CD five) (NN percent)))))
```

The support verb **increased** is linked to the following ARG1 **the price**, by going up to the VP and then down to the following NP.

Type 1 path features consist of encoding the actual path sub-sequence and ordinal or one-hot encoded representations. For each path feature variable, there are multiple possible path values. Type 2 path features consist of three binary features that check for the presence or absence of specific paths (our first three heuristics).

We implement a system with the above mentioned type 2 path features in combination with other baseline features, path features, and use a parse tree for generating paths. In comparison with the standard baseline system, we see that the incorporation of parse tree based path information and the above mentioned heuristics aids in predictive modeling for nominal SRL.

6.2 One Hot Encoding

The results in table 1 correspond to versions of feature based systems in table 2. As we develop our feature based system, we consider different encoding schemes for our categorical features. Feature based system 2, uses one hot encoding over ordinal encoding used in Baseline system (feature based) 1. In table 2, we suspect the jump in F-score from system 1 to system 2 is associated with the use of better path features by virtue of one hot encoding. The embedding based numeric features are not subject to change based on encoding schemes.

Consider a path feature which has n distinct values corresponding to n paths. Numbering different paths from 0 to (n-1) may not have allowed the model to yield predictive utility (ARG1 vs not-ARG1). The model has to predict the first path as 0, the second path as 1, and so on. By representing the path features in terms of one hot encoding, the model may be able to better use the path features for downstream prediction. We see this empirically, in our F-scores. Imposition of an ordinal relationship between non-ordinal entities like parts of speech tags would imply an ontology which may not exist. Incorporating ordinal information in our model would not be the most appropriate definition of the features since it would entail us putting in additional assumptions not corroborated with linguistic phenomena. Feature based system 3 expands on Feature based system 1. It explicitly uses a parser based system via Path Heuristic 2. It uses one hot encoding rather than ordinal encoding. It also uses the features from previous systems.

6.3 Baseline Results

The baseline system 1 results on the Gold task are discussed here. As illustrated in Table 1, all of the feature types contributed to the results on the development set. Thus removing a type of feature resulted in lower precision and/or recall, optimizing for an F1 score. The development sets for **%** and **partitive** included, respectively, 87 and 372 instances of ARG1s. The training corpus had 2225 and 9987 instances respectively. As per section 6, we only ran a limited set of systems on the test set, including the ALL system from table 1 (50 instances of **%** and 555 instances of **partitives**).

Both embedding features (characterizing ARG1s for these classes in general) and path features (characterizing relations between ARG1 and the predicate and ARG1) contributed to the final result. Both tasks get some positive result, even if we only use

System	Prec	Rec	F1
% Task			
All	**83.33**	**51.72**	**63.83**
N-gram Only	60.0	7.24	26.79
All but Path	65.22	34.48	45.11
All but Embed	84.09	42.53	56.49
All but Basic Embed	88.37	43.68	58.46
All but Slash Embed	84.31	49.43	62.32
Partitive Task			
All	**87.08**	**48.92**	**62.65**
N-gram Only	79.17	25.54	38.62
All but Path	77.6	26.08	39.03
All but Embed	87.75	48.12	62.15
All but Basic Embed	86.05	49.73	63.03
All but Slash Embed	85.32	50.00	63.05

Table 1: Baseline Results on (Gold) Development Sets

the most primitive of N-gram features. Both types of embedding features contributed to the final results as well, indicating that both the head ARG1s (normal embeddings) and words in ARG1 contexts (slash embeddings) are characteristic of ARG1s.

7 Student Systems

Student systems included modified versions of the baseline system, systems with similar architectures and pure transformer-based systems, primarily using versions of BERT (Devlin et al., 2019). In Figure 2, we provide improved results for systems that combine student system innovations with the baseline system. The remainder of this section will be about these systems.

7.1 Baseline System Modifications

Building on the baseline features, student group 16 uses a random forest (Breiman, 2001) in order to predict ARG1s based on the same input features as the baseline. They also do a grid search across models in order to obtain an estimator. For the best performing random forest estimator, they evaluate the relative importance of features. Features representing the distance from the predicate and embedding distances are seen to have a higher predictive importance. The f-measure increases slightly to 64.41 (percent) and 62.72 (partitive).

Student Group 15's system achieved f-measures of 82.91 (% task) and 77.46 (partitive task). They made two significant improvements on the baseline system: they changed the embeddings to one-hot embeddings and they used better path features. They have a set path of features, derived from the

sequence of BIO chunk tags between a given word and the predicate, the support verb in a sentence.

7.2 Transformer Based Systems

Student group 2 used a multi-layer-perceptron classifier with BERT (Devlin et al., 2019) embeddings, achieving 91.86 F-score for the % and 79.2 for partitives. Student group 9 uses DistilBert (Sanh et al., 2019) for fine-tuning, with tokens as features, achieving an 88.6 F-score on the partitive task. They report a 2.6 % F-score improvement using DistilBert vs Bert. Student group 10 uses a BiLSTM (Hochreiter and Schmidhuber, 1997) with similar features as baseline system 1, plus fasttext word embeddings. They report an F-score of 78.54 for the % task.

7.3 Deep Learning Using Linguistic Features

Student group 4 studied the integration of linguistic features with the representational capacity of attention based deep neural networks (Wu et al., 2018) (Sachan et al., 2021). Works such as (Clark et al., 2019) (Tenney et al., 2019) postulate that pre-trained large language models such as BERT implicitly learn to represent syntactic information without any inductive bias. Student group 4 considered that engineered features based on linguistic considerations could be of utility in predictive modeling for our task. This is corroborated by our baseline systems which are based on linguistics oriented feature engineering. They formulate the learning of linguistic features based on POS tags, BIO tags and directed parse tree distances of a word to the predicate as an auxiliary task performed by a separate head. This auxiliary head has a shared base representation with the BERT model which performs the downstream task of predicting semantic roles. Since the base representation is shared, the model is pushed to learn representations which can perform these tasks simultaneously, potentially enable the learning of better representations for the downstream task. This approach was used for **deep learning system 2** in table 2 and achieved higher results than deep learning system 1.

8 Ensembling

Based on principles of multi-view learning and ensemble learning (Sun, 2013; Dong et al., 2020), we develop an ensemble model. Governed by linguistics, an underlying data-generating process produces sentences along with their predicates and corresponding semantic roles as input, output pairs.

For every model, as we make a choice of representation to define the input and output data, we are modeling a view of the data-generating process. The performance of a model is proportional to the potency of a view to faithfully represent this process. It can be of benefit to construct a model which aggregates/processes information obtained from multiple views.

Our ensemble system consists of a purely deep learning component and a purely feature-based component. The deep learning component consists of a BERT-based model (Devlin et al., 2019) that uses distributed word vector representations as input. The feature-based component consists of an Adaboost model(Freund and Schapire, 1995) that uses the features motivated by our baseline system as input. We make predictions on semantic roles based on the aggregation of outputs of a BERT-based model and a feature-based model. We develop a voting scheme for deriving the ensemble output based on a weighted aggregation of the outputs of the two models. We learn the weights adaptively as we train the ensemble model on the data. (Jacobs et al., 1991; Eigen et al., 2014; Bishop, 2006).

9 Results Summary

Table 2 lists the results on the gold test set. Table 3 runs these same systems on the non-gold test set, providing insight to how our systems would perform on a realistic pipeline, beginning with raw text. Appendix B provides additional visual comprehension of the results. Feature based system 1 is our baseline system, with feature engineering done based on linguistic considerations. Feature based system 2 modifies the baseline system to use one hot encoding instead of ordinal encoding for feature generation. It also includes additional features denoting paths between a word in the sentence and the predicate. For example, a measure of the path distance to the predicate is considered based on the numbers of BIO chunks in between the predicate and a concerned word. One hot encoding gives concrete improvements, as initially demonstrated in a student system (Section 6.2). Along with type 1 path features, Feature based system 3 incorporates type 2 path features (section 6.1.4).

Inspired by student work, deep learning system 1 uses the BERT base (Devlin et al., 2019) to provide contextualized word embeddings, and a fine tuned BiLSTM. Positional embeddings are significant in this set up. Concatenating predicate indicator em-

System	Prec	Rec	F1
% Task			
1. Feature based 1	80.78	53.85	64.63
2. Feature based 2	93.59	74.51	82.95
3. Feature based 3	94.47	77.93	85.41
4. Deep learning 1	97.96	87.64	92.52
5. Deep learning 2	98.46	87.82	92.84
6. Ensemble	98.24	92.33	95.19
Partitive Task			
1. Feature based 1	84.58	48.65	61.76
2. Feature based 2	86.20	70.69	77.68
3. Feature based 3	91.64	81.15	86.07
4. Deep learning 1	96.55	82.18	88.79
5. Deep learning 2	97.36	82.41	89.26
6. Ensemble	98.79	85.62	91.74

Table 2: Results on Gold Test Set

System	Prec	Rec	F1
% Task			
1. Feature based 1	76.82	53.32	62.94
2. Feature based 2	86.59	75.76	80.81
3. Feature based 3	88.63	78.69	83.37
4. Deep learning 2	97.05	88.64	92.66
5. Ensemble	98.97	90.93	94.78
Partitive Task			
1. Feature based 1	50.58	71.74	59.33
2. Feature based 2	83.53	69.08	75.62
3. Feature based 3	92.34	76.85	83.89
4. Deep learning 2	95.59	83.40	89.08
5. Ensemble	98.76	84.58	91.12

Table 3: Results on Non-Gold Test Set

beddings with word embeddings after the BERT base forward pass helps the downstream deep learning model. The tasks involve predicting if a word is an ARG1 of a PRED. Considering that each sentence has one ARG1 for a given PRED, the number of words to be classified as not ARG1 would be higher than the number of words to be classified as ARG1. Scaling the loss function for class imbalance appropriately improves the modeling process.

We evaluate linguistic features in a deep learning framework (Wu et al., 2018), deep learning system 2 modifies system 1. As discussed in section 7.3, apart from modeling for the downstream task of predicting ARG1, we take a weighted average of the intermediate BERT presentations, analogous to ELMo (Peters et al., 2018), in order to model for an auxiliary task. The ensemble system combines the methods of the component systems and ultimately

achieved the highest score.

Tables 2 and 3 differ in a predictable pattern: systems run on the gold data are slightly better than systems run on non-gold tasks. Given parsing error is close to 5%, this result is expected.[1]

The gold data, accurately representing the data generating process, helped our models achieve better performance. Variations between gold annotations and non-gold annotations will be contingent on the properties of the parser used. When trained on non-gold data, models can adapt to these variations and learn representations that yield predictive performance, albeit slightly lower as compared to gold data. For predictive performance on a dataset, models need only differentiate between ARG1 labels and non-ARG1 labels in the created feature space. Systems on the partitive task have generally lower F-scores than those on the percent task. Additionally, for non-gold systems, those on the partitive task show a slightly larger relative decrease in F-scores. This is as expected, as the partitive task is more complex than the percent task.

10 Concluding Remarks

We observed that a variety of techniques contributed to our best result: traditional ML features, deep learning with/without linguistic features and multi-view ensembling. A cooperative shared task turned out to be an excellent research model. We demonstrated that partitive nouns, the largest frame-like class of Nombank nouns provide a high-performing testbed for SRL, and possibly related tasks, like event and relation extraction. We intend to make our dataset available for research purposes.[2]

We may explore avenues of further research, including: meta learning (Finn et al., 2017) across noun categories; parse tree path embeddings (Roth and Lapata, 2016); syntax infused models (Sachan et al., 2021); and in-context learning with large language models (Brown et al., 2020).

11 Limitations of Our Approach

Partitives are possibly the simplest NomBank noun class. They usually are accompanied by one argument and rarely occur with two. Their ARG1s have few semantic limitations – they need

to be measurable by the partitive predicate, e.g., *a pound of meat* is OK, but *a pound of sincerity* is semantically ill-formed. We would expect that arguments of other NomBank noun classes (Appendix A) would be more difficult to classify. Systems would achieve lower scores and may require other approaches.

Furthermore, the WSJ corpus represents the financial news register of 1990s English. Amounts in general and partitives in particular may be idiosyncratic to these data. Running our system and analyzing results for other corpora may provide further insight on semantic roles for noun predicates.

References

C. F. Baker, C. J. Fillmore, and J. B. Lowe. 1998. The Berkeley FrameNet Project. In *Coling-ACL98*, pages 86–90.

Christopher M. Bishop. 2006. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, Berlin, Heidelberg.

Leo Breiman. 2001. Random forests. *Machine learning*, 45:5–32.

---

[1] Deep Learning 1 is absent from Table 3 as it does not use any gold features and thus would be the same as in Table 2.

[2] Details will be provided on the NomBank website: `https://nlp.cs.nyu.edu/meyers/NomBank.html`. There may be licensing considerations because NomBank annotates Wall Street Journal text.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. *Preprint*, arXiv:2005.14165.

Xavier Carreras and Lluís Màrquez. 2005. 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. Association for Computational Linguistics.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. 2019. What does BERT look at? an analysis of BERT's attention. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 276–286, Florence, Italy. Association for Computational Linguistics.

M. Collins. 1999. *Head-Driven Statistical Models for Natural Language Parsing*. Ph.D. thesis, University of Pennsylvania.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of machine learning research*, 12(ARTICLE):2493–2537.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

George Doddington, Alexis Mitchell, Mark Przybocki, Lance Ramshaw, Stephanie Strassel, and Ralph Weischedel. 2004. The automatic content extraction (ACE) program – tasks, data, and evaluation. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04)*, Lisbon, Portugal. European Language Resources Association (ELRA).

Xibin Dong, Zhiwen Yu, Wenming Cao, Yifan Shi, and Qianli Ma. 2020. A survey on ensemble learning. *Frontiers of Computer Science*, 14:241–258.

David Eigen, Marc'Aurelio Ranzato, and Ilya Sutskever. 2014. Learning factored representations in a deep mixture of experts. *Preprint*, arXiv:1312.4314.

C. J. Fillmore and B. T. S. Atkins. 1998. FrameNet and Lexicographic Relevance. In *LREC: First International Conference on Language Resources and Evaluation*, pages 417–423.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. *Preprint*, arXiv:1703.03400.

J. Firth. 1957. A synopsis of linguistic theory. In *Studies in Linguistic Analysis*. Blackwell.

Y. Freund and R. E. Schapire. 1995. A decision-theoretic generalization of on-line learning and an application to boosting. *j-LECT-NOTES-COMP-SCI*, 904.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep Learning*. MIT Press. http://www.deeplearningbook.org.

J. S. Gruber. 1965. *Studies in Lexical Relations*. Ph.D. thesis, MIT. Reproduced by Indiana University Linguistics Club, January 1970.

J. Hajič, M. Ciaramita, R. Johansson, D. Kawahara, M. A. Martí, L. Màrquez, A. Meyers, J. Nivre, S. Padó, J. Štěpánek, P. Straňák, M. Surdeanu, N. Xue, and Y. Zhang. 2009. The CoNLL-2009 shared task: Syntactic and semantic dependencies in multiple languages. In *CoNLL-2009*, Boulder, Colorado, USA.

Luheng He, Kenton Lee, Mike Lewis, and Luke Zettlemoyer. 2017. Deep semantic role labeling: What works and what's next. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 473–483.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Robert A. Jacobs, Michael I. Jordan, Steven J. Nowlan, and Geoffrey E. Hinton. 1991. Adaptive mixtures of local experts. *Neural Computation*, 3(1):79–87.

Z. P. Jiang and H. T. Ng. 2006. Semantic role labeling of nombank: A maximum entropy approach. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing (EMNLP 2006)*, Sydney, Australia.

Paul Kingsbury and Martha Palmer. 2002. From treebank to propbank. In *Proceedings of the 3rd International Conference on Language Resources and Evaluation (LREC-2002)*, Las Palmas, Spain. European Language Resources Association (ELRA).

Nikita Kitaev and Dan Klein. 2018. Constituency parsing with a self-attentive encoder. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2676–2686, Melbourne, Australia. Association for Computational Linguistics.

Christopher D Manning, Mihai Surdeanu, John Bauer, Jenny Rose Finkel, Steven J Bethard, and David Mc-Closky. 2014. The stanford corenlp natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60. Association for Computational Linguistics.

A. Meyers, R. Reeves, C. Macleod, R. Szekely, V. Zielinska, B. Young, and R. Grishman. 2004a. Annotating Noun Argument Structure for NomBank. In *Proceedings of LREC-2004*, Lisbon, Portugal.

A. Meyers, R. Reeves, C. Macleod, R. Szekely, V. Zielinska, B. Young, and R. Grishman. 2004b. The NomBank Project: An Interim Report. In *NAACL/HLT 2004 Workshop Frontiers in Corpus Annotation*, Boston.

E. Miltsakaki, A. Joshi, R. Prasad, and B. Webber. 2004. Annotating discourse connectives and their arguments. In *NAACL/HLT 2004 Workshop: Frontiers in Corpus Annotation*, pages 9–16, Boston, Massachusetts, USA. Association for Computational Linguistics.

Hiroki Ouchi, Hiroyuki Shindo, and Yuji Matsumoto. 2018. A span selection model for semantic role labeling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1630–1642, Brussels, Belgium. Association for Computational Linguistics.

M. Palmer, D. Gildea, and P. Kingsbury. 2005. The Proposition Bank: An annotated corpus of semantic roles. *Computational Linguistics*, 31(1):71–106.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Sameer Pradhan, Wayne Ward, Kadri Hacioglu, James Martin, and Daniel Jurafsky. 2005. Semantic role labeling using different syntactic views. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 581–588, Ann Arbor, Michigan. Association for Computational Linguistics.

Michael Roth and Mirella Lapata. 2016. Neural semantic role labeling with dependency path embeddings. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1192–1202, Berlin, Germany. Association for Computational Linguistics.

Devendra Singh Sachan, Yuhao Zhang, Peng Qi, and William Hamilton. 2021. Do syntax trees help pretrained transformers extract information? *Preprint*, arXiv:2008.09084.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Shiliang Sun. 2013. A survey of multi-view machine learning. *Neural computing and applications*, 23:2031–2038.

M. Surdeanu, R. Johansson, A. Meyers, L. Márquez, and J. Nivre. 2008. The CoNLL-2008 Shared Task on Joint Parsing of Syntactic and Semantic Dependencies. In *Proceedings of the CoNLL-2008 Shared Task*, Manchester, GB.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019. Bert rediscovers the classical nlp pipeline. *arXiv preprint arXiv:1905.05950*.

Erik F. Tjong Kim Sang and Sabine Buchholz. 2000. Introduction to the CoNLL-2000 shared task chunking. In *Fourth Conference on Computational Natural Language Learning and the Second Learning Language in Logic Workshop*.

Minghao Wu, Fei Liu, and Trevor Cohn. 2018. Evaluating the utility of hand-crafted features in sequence labelling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2850–2856, Brussels, Belgium. Association for Computational Linguistics.

Nianwen Xue and Martha Palmer. 2003. Annotating the propositions in the Penn Chinese treebank. In *Proceedings of the Second SIGHAN Workshop on Chinese Language Processing*, pages 47–54, Sapporo, Japan. Association for Computational Linguistics.

Nianwen Xue and Martha Palmer. 2004. Calibrating features for semantic role labeling. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 88–94, Barcelona, Spain. Association for Computational Linguistics.

Jie Zhou and Wei Xu. 2015. End-to-end learning of semantic role labeling using recurrent neural networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1127–1137.

Appendix A Other Nombank Classes

ENVIRONMENT: A period/**REL** of industrial consolidation/**ARG1**

ABILITY Their/**ARG0** right/**REL** to talk/**ARG1**

WORK-OF-ART Congress/**ARG0**'s idea/**REL** of reform/**ARG1**

CRISS-CROSS Topics/**REL** of interest/**ARG1**

RELATIONAL (ACTREL) His/**ARG2** math/**ARG1** professor/**ARG0+REL**

RELATIONAL (DEFREL) Mindy Hymowitz's/**ARG1** mother/**ARG0+REL**

ATTRIBUTE The company/**ARG1** 's value of $3 Billion/**ARG2**

JOB the new Treasury/**ARG2** post/**REL** of inspector general/**ARG0**

HALLMARK: the cornerstone/**ARG0+REL** of Phillips' chemicals operations/**ARG1**

VERSION: the House/**ARG0** version/**REL** of the deficit-cutting bill/**ARG1**

TYPE: his brand/**REL** of Christianity/**ARG1**

ISSUE: the subject/**REL** of U.S. direct investment in Japan/**ARG1**

FIELD: the rapidly growing field/**REL** of bio-analytical instrumentation/**ARG1**

EVENT: the drought of 1988/**ARGM-TMP**

Figure 5: More NomBank "Other" Classes.
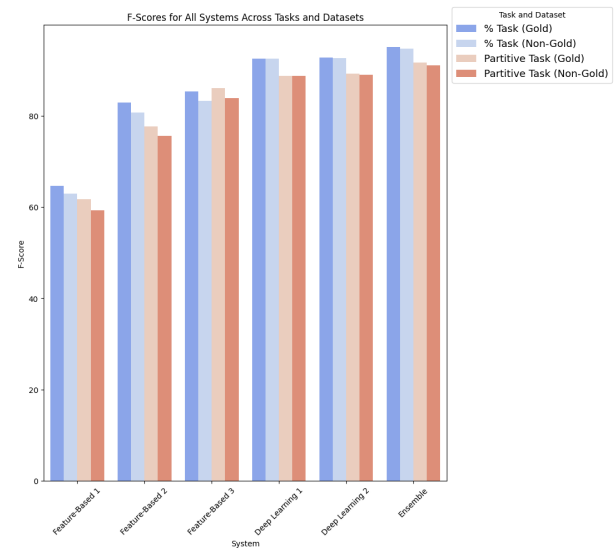
Appendix B Visualizations

Figure 6: Bar Chart showing F-scores for different systems and datasets.
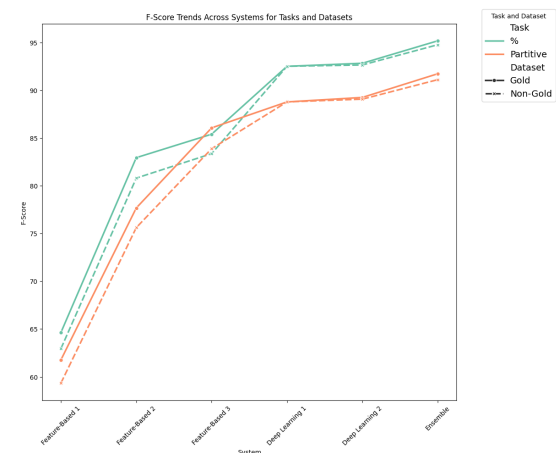

Figure 7: Line Chart displaying F-scores across systems for gold and non-gold datasets.

B.1 Description of figures

Figures 6 and 7 demonstrate our results and complement tables 2 and 3. In figures 8 and 9, the relative importance score for top 15 features (sorted in order of magnitude) is displayed. This score is derived from the Adaboost model (Freund and Schapire, 1995) used for training feature based system 3, with descriptions as elaborated in sections 9, 6.
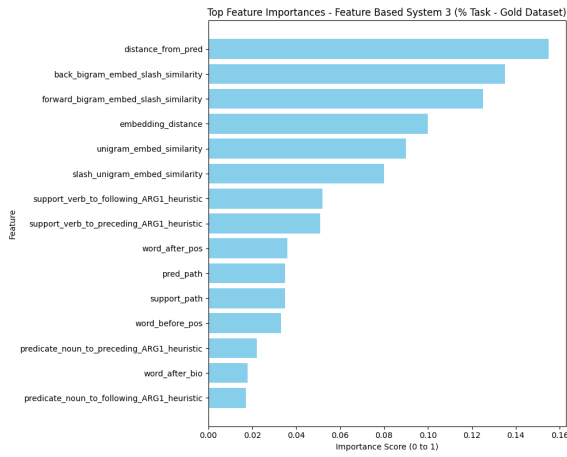

Figure 8: Feature Importance for our best feature-based system on the Percent task.
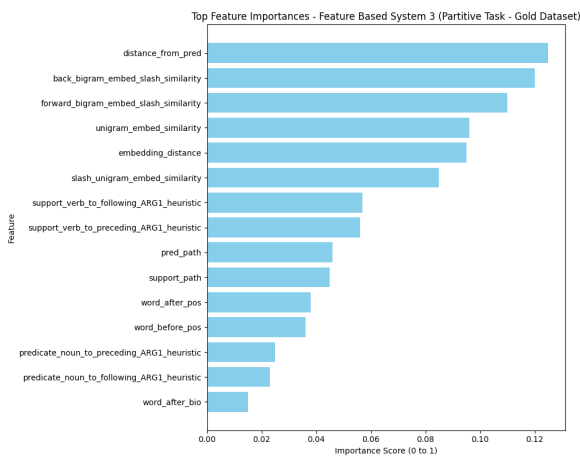

Figure 9: Feature Importance for our best feature-based system on the Partitive task.