Finding ARG1 of Partitive Nouns with Pre-trained Embedding Methods

Tanran Zheng* **Wentao Chen*** **Xiangyuan Wang*** **Xinru Xu***
Department of Computer Science
New York University
251 Mercer St, New York, NY
tz408@nyu.edu wc2215@nyu.edu xw2497@nyu.edu xx2085@nyu.edu

Abstract

NomBank is a project created by New York University to annotate the argument structures for common nouns in the Penn Treebank II corpus. Partitive task, which is one of the NomBank-based automatic Semantic Role Labeling (SRL) tasks, focuses on dealing with the partitive nouns. It aims to find a specific set of noun arguments that generally describes partialness relationship such as "part of", "multiples of", "quantity of". In this paper, we propose to solve this task as an end-to-end problem and investigate multiple deep embedding-based methods with pre-trained language models (PLM). For the % task, we achieved 92.48 in F1 score. For the total partitive group task, our approach achieved 79.00 in F1 score.

1 Introduction

Semantic roles are the various roles that a noun phrase may play in a sentence, which have been a core linguistic concept since their modern formulation being introduced in 1960s (Fillmore et al., 1968; Gruber, 1965). Semantic roles represent the participants in an action or relationship captured by a semantic frame. For example, in the sentences "*John broke the window*" and "*The window was broken by John*", "*John*" in both sentences are in an agent relation with the verb "*break*", and "*the window*" in both sentences are in a patient relation. Identifying and analyzing the semantic roles in the sentence can help the language model to understand the meaning of a sentence (Jurafsky and Martin, 2009). Semantic role labeling (SRL) consists of the detection and classification of semantic roles, which is crucial for language processing tasks such as speech recognition, part-of-speech tagging, and parsing.

NomBank[1] (Meyers et al., 2004) is an annotation project that has a deep relationship with the PropBank (Palmer et al., 2005a) corpus, which itself is based on the Penn Treebank (PTB) II corpus . Contrary to PropBank's effort and focus on verbs, the goal of NomBank is to provide argument structure for instances of common nouns in the PTB corpus. Partitive nouns, one of its studied argument-taking nouns, is the focus here.

The partitive noun represents a specific set of noun arguments that generally describes partialness relationship such as "part of", "multiples of", "quantity of". In this paper, we are interest in the **% task** and **partitive task**, i.e. finding the ARG1 of the "%" sign or (more generally) a given partitive noun. This is one of the semantic role labeling (SRL) tasks which is rather approachable when compared with other SRL tasks.

The task could be described as follows. Given a sentence, the PTB annotations for the tokens (e.g. POS, BIO tags), other parsing information, and the partitive noun, we aim to find one target argument of the partitive noun such that the ARG1 would form the partialness relationships mentioned above. Fig 1 shows one example of the training set.

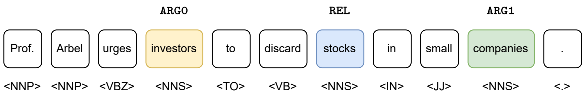

Figure 1: The partitive noun "stocks" takes "investors" as ARG0, and "companies" as ARG1.

The example here shows a special type of partitive relation, the *SHARE* type. The "stocks" here is, in the sense of financial assets, a *part/share* of the "companies", thus exhibiting a partialness relationship. We then could determine "companies" to be the ARG1 of the partitive noun "stocks". Apart from the *SHARE* type, there are other types of partitive relations which are categorized by the fine-grained semantic properties, e.g. the *QUANT* (quantity) type "sales/ARG1 went down 10.4 %/REL"

*Equal contribution.

[1] See the Dataset section for more about NomBank.

(the % task), the *GROUP* type "rescue/ARG1 committee/REL".

2 Related Work

Current methods that could be used to approach this task include the linguistic and embedding-based ones. The linguistic approach would involve manually designing features following some heuristics such as semantic classes or parsing tree paths or simple ones. Jiang and NG (Jiang and Ng, 2006) proposed a maximum entropy (Berger et al., 1996) approach to solve the NomBank-based SRL task, in which they treat the task as a classification problem that consists of argument identification and argument classification. The Opennlp maxent[2], an implementation of Maximum Entropy (MaxEnt) modeling, is used as the classification tool.

Embedding-based approach on the other hand, relaxes the need for feature engineering. Word embeddings, especially the contextualized ones, have been proven to contain certain linguistic properties such as POS tags (Liu et al., 2019). With these knowledge at hand, one saves the time in designing features and could instead focus on designing models that could better utilize the embeddings or on improving data quality. Recent literature (He et al., 2017; Strubell et al., 2018) on SRL tasks that involve embedding-based approach would stage the SRL task as an end-to-end token-wise tagging problem, where the model would behave like a classifier and produce the tag for each token from the input sequence.

Among the embedding-based approaches that achieve high scores in the general NLP tasks, pre-trained language models (PLM) are the most popular nowadays. BERT (Devlin et al., 2018), which stands for Bidirectional Encoder Representations from Transformers, is a model that aims to pre-train deep bidirectional representations from the unlabeled text by jointly tuning the left and right contexts in all layers. The pre-training process involves two tasks, next-sentence prediction (NSP) and masked-language modeling (MLM). The MLM task is a fill-in-the-blank or reading comprehension task, where the model is trained to use the context words surrounding a randomly chosen masked token (denoted by a special token [MASK]) to predict the appropriate word in the masked position. To mitigate the influence introduced by the special token, the special token would be further replaced by other normal tokens with a probability during pre-training. The pre-train/fine-tune paradigm adopted by BERT has proven to be effective in various NLP tasks and has continued to achieve state-of-the-art results with more recent pre-trained models. In this work, we choose the pre-trained BERT as our work basis to design different approaches to solve the problem.

As one of the embedding-based approaches, prompt learning is a new paradigm raised in recent years. Unlike traditional supervised learning, which trains a model to take in an input $\mathbf{x}$ and predict an output $\mathbf{y}$ as $P(\mathbf{y}|\mathbf{x})$. To use prompt learning based models to perform prediction tasks, the original input $\mathbf{x}$ is modified using a template into a textual string prompt $\mathbf{x}'$ that has some unfilled slots, and then the language model is used to probabilistically fill the unfilled information to obtain a final string $\hat{\mathbf{x}}$ from which the final output $\mathbf{y}$ can be derived (Liu et al., 2021). Prompt learning based models have been applied to various related NLP tasks in previous work, including but not limited to Natural Language Inference (NLI) (Schick and Schütze, 2020a), text classification (Yin et al., 2019; Schick and Schütze, 2020b), relation extraction (Chen et al., 2021; Han et al., 2021), sequence tagging(Cui et al., 2021; Ben-David et al., 2021), and semantic parsing (Shin et al., 2021; Berant and Liang, 2014). With the impressive records in these NLP tasks, we are interested to see its potential in SRL tasks.

3 Data

NomBank project (Meyers et al., 2004) is a databank that annotates the argument structure of nouns in the Penn Treebank II corpus. The Penn Treebank II corpus, which features a million words of Wall Street Journal material annotated in Treebank II style, is designed to allow the extraction of simple predicate-argument structure (Marcus, Mitchell P. et al., 1995). Along with the PropBank from the University of Pennsylvania, the NomBank project aims at creating better tools for the automatic analysis of text (Meyers et al., 2004), and makes it possible to develop automatic SRL systems that analyze the argument structures of noun predicates (Jiang and Ng, 2006).

Compared to the PropBank, NomBank framework is a modified version of PropBank (Palmer et al., 2005b) to fit nominalizations. It uses the frames from PropBank when possible for nouns

[2] https://opennlp.apache.org/

related to verbs, and moreover, creates classes of noun arguments which do not correspond to any verb. It also deals with noun-specific phenomena, such as support constructions and transparent nouns. For each instance of a common noun in the Penn Treebank that is accompanied by one of its arguments (ARG0, ARG1, ARG2, ARG3, ARG4) or is a nominalization and is accompanied by one of the allowable types of adjuncts, NomBank annotates a proposition and a subset of the features {REL, SUPPORT, ARG0, ARG1, ARG2, ARG3, ARG4, ARGM} (Meyers et al., 2004). In this project, we focus on `ARG1`, which is the patients of the partitive noun.

We use the `partitive_group_nombank` dataset and the `%_nombank` dataset provided by the NomBank project. The first dataset includes the whole second dataset in that the second only focuses on the case where the partitive noun is the "%" sign and the *QUANT* type of partitive relationship.

4 Methodology

We present here a series of approaches that involve the use of pre-trained language models. We believe with the extensive pre-training, the language model would possess to some extent the semantic knowledge such that it would be able to determine the partitive relations. The trail of experiment starts from fine-tuning the pre-trained BERT-base model to be a binary token classifier. The model would take each token of the input sentence and determine if it is the `ARG1`.

In addition to treating the partitive task as a per-token classification task, we also experiment with prompt learning, re-framing the task into a reading comprehension task which the BERT model has been extensively trained during the pre-training stage with the MLM task. This approach has its advantage of being able to perform well under few-shot settings where the training data is scarce.

We will introduce the approaches in detail in the following sections.

4.1 Baseline: MaxEnt model

As introduced in Section 2, the baseline is a maximum entropy model that serves as a multi-class token classifier that operates on each token of the input sentence. In order to make a good estimation on the probability distribution of the training data, meaningful features need to be extracted so that the model could have a clear understanding of each of the output classes. Some of the features include the part-of-speech tag, the BIO tag of the chunking task, the word stem, the surrounding tokens (up to three tokens each side) of the `REL/SUPPORT/PRED` token, the unigram/bigram/trigram embedding generated by spacy[3]'s `tok2vec` embedding model, the n-gram embedding similarities, the token distances. The features are generated on a per-token basis and are divided by tabs, therefore each line would correspond to a token and an empty line indicates the sentence break.

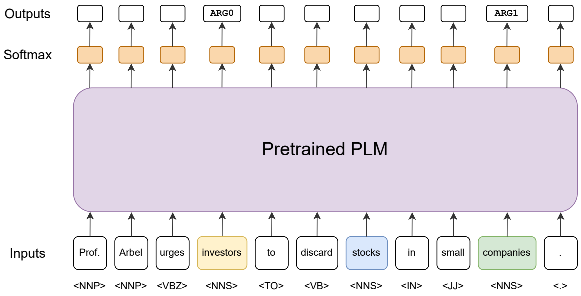

Figure 2: BERT + MLP classifier

4.2 BERT + MLP classifier

Instead of manually designing features on the per-token, per-ngram granularity, this approach would only involve tokenizing the input sentence and feeding into the PLM. Then for each input token, the model would generate a hidden state representation. This representation would then go through an added MLP layer that serves as a classifier that would predict whether the token is an `ARG1` or not. The architecture is illustrated in Figure 2.

As we are running the BERT-MLP architecture as a per-token classifier, there is naturally a consequence of over-generating the `ARG1`, i.e. the model would predict more than one token to be the `ARG1` in one sentence. This is an unwanted behavior that's been introduced by the way we design the classifier. To solve this, we have experimented with different strategies: 1) using the first encountered `ARG1`; 2) using the `ARG1` that is closest to the `PRED`; 3) using the one with the largest logits value among the `ARG1` candidates. Experiment on toy dataset showed that the last strategy achieves the best result, therefore we stuck with the largest logits `ARG1`.

[3] https://spacy.io/

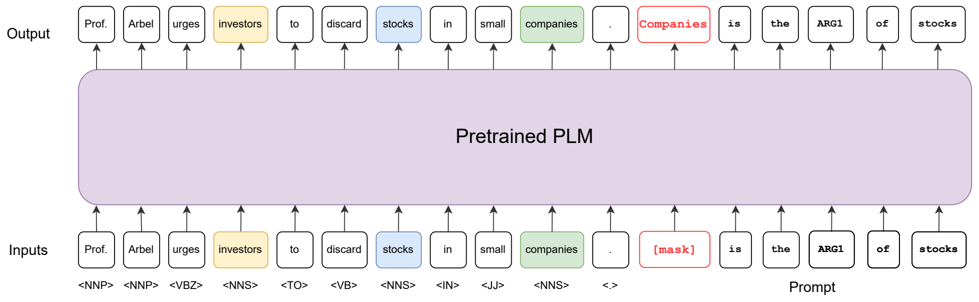

Figure 3: The architecture of prompt learning model.

4.3 BERT + statistical features + MLP classifier

The NomBank dataset includes a number of additional statistical features such as the part-of-speech tags, the BIO tag of the chunking task. We were interested in their effectiveness in combination with the PLM embedding method. On the basis of the previous architecture, we concatenate the statistical features with the hidden state of each token, then input into the MLP for classification.

4.4 BERT with prompt learning

The prompt learning method is different from all the previous methods in that rather than the previous classification-based approaches, it treats the partitive task as a reading comprehension task, which naturally conforms to the MLM pre-training task. Therefore there's no need for fine-tuning, nor do we have to add an additional MLP layer to act as the classifier. During experiment, we tried different ways of phrasing the prompt template, here list some of them:

For an input text a, $P(a)$ denotes the templated input,

1. $P(a) = a$, ___ is the patient of the word %.
2. $P(a) = a$, ___ has a partialness relationship with %.
3. $P(a) = a$, the patient of % is at position ___.

Since there's no ARG1 in the English vocabulary, we cannot directly use ARG1 in our template. So we rephrase the question with natural language, emphasizing the partitive relationship between the target token and the given partitive noun (the % sign of the % task). For the first two templates, we would train the model to output the actual token itself which occurred in the initial input a. For the third template, the to-be-filled blank is the position of the ARG1 token in the input a, so that the answer word would only be chosen from digits rather than the whole vocabulary table. The architecture is illustrated in Figure 3.

5 Experiments

5.1 Dataset

For the % task, the %_nombank dataset is split into training, validation and test set. There are 2,176 sentences in the training set, 83 sentences in the validation set, and 150 sentences in the test dataset. Each line in the datasets represents a token in one sentence and an empty line indicates a sentence break. The line contains the token itself, the part-of-speech tag, the chunking BIO tag, and two indices indicating the position of the token in the sentence and the position of the containing sentence in the document. At the end of each line is the semantic role label. There are three labels in the % task, the ARG1, the PRED that normally labels the % sign, and the SUPPORT. If a token is not labeled with one of these three, then its semantic role label is omitted. Here is one sample sentence from the training set. In this example, the PARTITIVE-QUANT indicates the type of the partitive relation.

```
That        DT   B-NP  0 73
compares    VBZ  O     1 73 SUPPORT
with        IN   B-PP  2 73 SUPPORT
3.5         CD   B-NP  3 73
%           NN   I-NP  4 73 PRED PARTITIVE-QUANT
butterfat   NN   B-NP  5 73
for         IN   B-PP  6 73
whole       JJ   B-NP  7 73
milk        NN   I-NP  8 73 ARG1
.           .    O     9 73
```

Figure 4: Sample sentence from the training set

The partitive_group_nombank training set contains 10,834 sentences, the validation set 393 sentences and 660 sentences in the test set. In ad-

Model	Precision	Recall	F1 Score
MaxEnt	88.24	60.00	71.43
BERT + MLP	91.86	91.86	91.86
BERT + stat. features + MLP	91.73	90.35	91.04
Prompt Learning	92.65	92.30	**92.48**

Table 1: Results on the % dataset

dition to the data format of the %_nombank dataset, the semantic role labels contain more types, including the ARG0 and ARG2 labels. What's more, there may be more than one ARG1 in one sentence in the partitive dataset. We have to process the partitive dataset into the same format as the % dataset so that we can use the same model architecture.

5.2 Experiment Details

For the BERT related approaches, we use PyTorch and the Huggingface Transformers library to implement the models. All models including the prompt learning version are trained on a single NVIDIA Tesla T4 GPU with 16 GB VRAM on Google Colab. The models are fine-tuned/trained for 10 epochs each with a learning rate of 1e-5. All approaches are trained with a batch size of 32.

5.3 Evaluation Metrics

Same as the metrics used by the baseline MaxEnt model, we use precision, recall and F1-score to evaluate our models. Let *TP* represent the number of true positive samples, *FP* the number of false positives, and *FN* the number of false negatives, we could calculate the precision, recall and F1-score with

$$precision = \frac{TP}{TP + FP}$$
$$recall = \frac{TP}{TP + FN}$$
$$F1 = 2 \cdot \frac{precision \cdot recall}{precision + recall}$$

6 Results

For the % task, because in each sentence there is only one target ARG1, the task is rather simple and we could experiment with multiple approaches. But for the partitive task, because there could be more than one ARG1 label in one sentence, we really could not use the original prompt template in this task. Therefore we only experiment with the two fine-tuning based approaches in the partitive task.

6.1 % Task

The result is shown in Table 1. As can be seen from the results, the PLM based approaches generally perform better than the machine learning based MaxEnt model. We conclude that it is due to the benefit of the extensive pre-training BERT has gone through, thanks to which the model has acquired some level of semantic knowledge of the language such that it is able to understand the partitive relation involving the % sign. Among the PLM based approaches, the prompt learning one achieves though by a small margin a relative better result. It is also worth noting that the explicit introduction of part-of-speech tag and the chunking BIO tag features do not help much in improving the classifier's performance.

6.2 Partitive Task

The result is shown in Table 2. Same as in the % task, the PLM based approaches perform better than the MaxEnt method, but this time the gap is closer. The added features also have a rather negative influence on the model's performance. Compared with the simpler % task, the more complex partitive task with its expanded semantic role label set has been a major disadvantage on the models' performance. The uncertainty of the number of ARG1 in a sentence is also a challenge on the model's ability.

7 Conclusion

In this project we have discussed several approaches on finding the ARG1 of partitive nouns (including the % sign) of the NomBank dataset. We are glad to see the effective results the pre-trained BERT achieved in this specific SRL task, indicating the pre-trained model has acquired to some extent the semantic knowledge on the partitive relation.

It is worth noting that by simply concatenating the pre-trained hidden state with the part-of-speech tags and chunking BIO tags, the performance of the classifier did not receive an intuitive improvement. Further experiments would involve designing ar-

Model	Precision	Recall	F1 Score
MaxEnt	77.66	46.70	58.32
BERT + MLP	79.55	78.45	**79.00**
BERT + stat. features + MLP	79.82	77.34	78.57

Table 2: Results on the partitive dataset

chitectures combining the statistical features more effectively into the current pre-train/fine-tuning procedure.

To further investigate the NomBank-based automatic semantic role labeling tasks, several works could be done in future experiment. First, we would like to experiment with more diverse prompt templates, which is also known as *prompt template engineering*. We can manually design prompt templates or design algorithms to search for the prompt templates for this specific task that may improve the performance of our prompt learning model. Second, we could also dive deeper into the PLM that we used in our approaches, where we can try different PLMs such as SpanBERT or other BERT variants. Last but not least, we would like to research the probability to combine the linguistic approach and deep embedding-based approach. As an initial thought, we can design a "voting" system that takes the outputs of the MaxEnt model and those of the embedding-based model, and then makes the predictions by evaluating these outputs from both sides. This is a task worthwhile in trying as the linguistic approaches (e.g. MaxEnt model we mentioned) contains human's understanding of the language and the embedding-based approaches, on the other hand, encodes implicit patterns that are learnt from massive language data.

References

Eyal Ben-David, Nadav Oved, and Roi Reichart. 2021. PADA: A prompt-based autoregressive approach for adaptation to unseen domains. *CoRR*, abs/2102.12206.

Jonathan Berant and Percy Liang. 2014. Semantic parsing via paraphrasing. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1415–1425, Baltimore, Maryland. Association for Computational Linguistics.

Adam L. Berger, Stephen A. Della Pietra, and Vincent J. Della Pietra. 1996. A maximum entropy approach to natural language processing. *Computational Linguistics*, 22(1):39–71.

Xiang Chen, Ningyu Zhang, Xin Xie, Shumin Deng, Yunzhi Yao, Chuanqi Tan, Fei Huang, Luo Si, and Huajun Chen. 2021. Knowprompt: Knowledge-aware prompt-tuning with synergistic optimization for relation extraction. *CoRR*, abs/2104.07650.

Leyang Cui, Yu Wu, Jian Liu, Sen Yang, and Yue Zhang. 2021. Template-based named entity recognition using BART. *CoRR*, abs/2106.01760.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Charles J. Fillmore, Paul Kiparsky, James D. McCawley, Emmon W. Bach, and Robert Thomas Harms. 1968. *Universals in linguistic theory*. Holt, Rinehart and Winston.

Jeffrey Steven Gruber. 1965. *Studies in lexical relations*. Ph.D. thesis, Massachusetts Institute of Technology.

Xu Han, Weilin Zhao, Ning Ding, Zhiyuan Liu, and Maosong Sun. 2021. PTR: prompt tuning with rules for text classification. *CoRR*, abs/2105.11259.

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, Vancouver, Canada. Association for Computational Linguistics.

Zheng Ping Jiang and Hwee Tou Ng. 2006. 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. Association for Computational Linguistics.

Dan Jurafsky and James H. Martin. 2009. *Speech and language processing : an introduction to natural language processing, computational linguistics, and speech recognition*, 2nd ed.. edition. Prentice Hall series in artificial intelligence. Pearson Prentice Hall, Upper Saddle River, N.J.

Nelson F. Liu, Matt Gardner, Yonatan Belinkov, Matthew E. Peters, and Noah A. Smith. 2019. Linguistic knowledge and transferability of contextual representations. 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 1073–1094, Minneapolis, Minnesota. Association for Computational Linguistics.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2021. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *CoRR*, abs/2107.13586.

Marcus, Mitchell P., Santorini, Beatrice, and Mary Ann Marcinkiewicz. 1995. Treebank-2.

Adam Meyers, Ruth Reeves, Catherine Macleod, Rachel Szekely, Veronika Zielinska, Brian Young, and Ralph Grishman. 2004. 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. Association for Computational Linguistics.

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

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

Timo Schick and Hinrich Schütze. 2020a. Exploiting cloze questions for few-shot text classification and natural language inference. *CoRR*, abs/2001.07676.

Timo Schick and Hinrich Schütze. 2020b. Exploiting cloze questions for few-shot text classification and natural language inference. *CoRR*, abs/2001.07676.

Richard Shin, Christopher H. Lin, Sam Thomson, Charles Chen, Subhro Roy, Emmanouil Antonios Platanios, Adam Pauls, Dan Klein, Jason Eisner, and Benjamin Van Durme. 2021. Constrained language models yield few-shot semantic parsers. *CoRR*, abs/2104.08768.

Emma Strubell, Patrick Verga, Daniel Andor, David Weiss, and Andrew McCallum. 2018. Linguistically-informed self-attention for semantic role labeling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5027–5038, Brussels, Belgium. Association for Computational Linguistics.

Wenpeng Yin, Jamaal Hay, and Dan Roth. 2019. Benchmarking zero-shot text classification: Datasets, evaluation and entailment approach. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3914–3923, Hong Kong, China. Association for Computational Linguistics.