Using Deep Learning to Identify ARG1 in the NomBank Dataset

Anoushka Gupta
NYU Courant
ag8733@nyu.edu

Arunima Mitra
NYU Courant
am13018@nyu.edu

Abstract

This paper aims to identify the ARG1 in nominal predicates of the ***[1] Nombank*** Dataset using Deep Learning Algorithms. Identifying ARG1 is part of a broader problem of semantic role labelling, which is the process of annotating predicate-argument structures in sentences with semantic roles. We have treated the identification of ARG1 as a token classification problem. We experiment with various token and sentence-level features to predict whether a token is an ARG1 or not. Our paper focuses on analysing the impact on the prediction of ARG1 when the model is given knowledge of all the possible labels compared to when it is given knowledge of the predicate (PRED) label alone. Our best-performing model achieved an F1 score of *94.26 %* when given knowledge of all labels, compared to an F1 score of *91.08 %* when given knowledge of only the PRED label.

1 Introduction

Nombank (Meyers et al.)
is a data annotation project at New York University that aims to mark the noun arguments in the Wall Street Journal of the Penn Treebank. Each sentence in the Nombank dataset consists of a nominal predicate which is a noun. The predicate usually describes the properties of the subject or the object. The identification of arguments is a very important task of Semantic Role Labelling. This identification helps the machines understand the context and meaning of words and phrases in sentences. For example, for the sentence *"The South Korean government had been projecting a 5 % consumer price increase for the entire year."* The predicate is '5%' and the predicate refers to 'consumer' which is the ARG1.
Our developer community has often applied feature engineering and canonical machine learning algorithms for understanding the semantics of long texts. Our motivation for using deep learning algorithms is to prevent any human intervention or supervision in semantic labelling tasks. Semantic role labelling by manual annotation is very expensive and time-consuming. Hence there is a need to build an automatic pipeline that is able to do this task quickly and with good accuracy. This identification task is treated as a binary token classification problem. Thus, each sentence can have zero or more ARG1 labels. In this paper, we discuss the different models we have implemented and compare them using evaluation metrics like Precision, Recall, and F1 score.

2 Related Work

Harris(1968) makes the *"distributional hypothesis"*, claiming that "the meaning of entities and the meaning of grammatical relations among them, is related to the restriction of combinations of these entities relative to other entities". **[2]** ***Hearst and Marti*** developed a parser to extract grammatical structures from unrestricted text. They classify nouns according to the predicates they occur with. ***[1] Nombank*** is a databank which was created to annotate argument structure in nouns similar to what PropBank does for verbs. ***[3] Jiang and Ng*** treated the Nombank Semantic Role Labelling problem as a classification problem and studied adapting features which were useful in PropBank based SRL systems.

More recent studies have also used the power of deep learning algorithms to perform semantic role labelling. **[4]** ***He et (2017)*** employs a bidirectional long short term memory model (BiLSTM) which takes only the text as input features and doesn't use syntactic information. **[5]** ***Zhou and Xu (2015)*** treat SRL as a BIO tagging problem and use deep RNNs because past information is built up through the recurrent layer when the system takes each word of the sentence. **[4]** ***He, Lee, et. al [2017]*** extended this research by simplifying the input and output layers and using a highway connection. **[6]** ***BERT*** (Bidirectional Encoder Representations from Transformers), a state of the art model, has been claimed efficient for around eleven NLP tasks.

[7] ***ALBERT: A Lite BERT*** is an extension of BERT which introduces two parameter reduction techniques to lower memory consumption and increase the training speed of BERT.

3 Dataset

We refer to the NomBank dataset to perform our semantic role labelling task. NomBank provides argument structure for over 5000 common nouns in the Penn Treebank corpus. Our training set includes 84169 sentences, or 58993 ARG1 tokens. The development set contains 3234 sentences and a total of 2343 ARG1 labels. Our test set has 5381 sentences with 3805 ARG1 labels. Each token occupies a line in the dataset, the line including the word itself, the index of the sentence, the index of the token in that sentence, the POS tag and the NG-BIO tag. In addition, some words will have a semantic role label present at the end of each line. The labels include - "PRED" for predicate, "ARG1" and "SUPPORT". Other labels are omitted from this dataset. Figure 1 shows an example of a sentence from our development dataset.

```
The      DT      B-NP    0       0
consensus        NN      I-NP    1       0
view     NN      I-NP    2       0
expects VBZ      O       3       0
a        DT      O       4       0
0.4      CD      B-NP    5       0
%        NN      I-NP    6       0       PRED    PARTITIVE-QUANT
increase         NN      B-NP    7       0       SUPPORT
in       IN      B-PP    8       0
the      DT      B-NP    9       0
September        NNP     B-NP    10      0
CPI      NNP     I-NP    11      0       ARG1
after    IN      B-PP    12      0
a        DT      B-NP    13      0
flat     JJ      I-NP    14      0
reading NN       I-NP    15      0
in       IN      B-PP    16      0
August   NNP     B-NP    17      0
.        .       O       18      0
```

Figure 1: Example from the development dataset

4 Methodology

Our work focuses on utilising well defined features for each deep neural network model to successfully predict ARG1 in each sentence. Even though for deep learning models, we wouldn't need a lot of feature engineering, we still focus on some critical features which bring out better results. For each of our machine learning frameworks, we take a subset of the below defined features and test our model on the Nombank dataset -

1) Word Level Features - These consist of trivial features, already provided in each of the Nombank datasets, primarily consisting of the token index, the POS tag, and the NG-BIO tag. We also include the stem of the word.
2) Position Level Features - This set of features focuses on properties of the

3) previous and the next word for each token. We create the following features for each token in the dataset - previous POS tag, previous BIO tag, previous stemmed word, next POS tag, next BIO tag, next stemmed word.
4) Distance Feature - This is a feature which marks the distance of each token in a sentence from the predicate in that sentence. We follow a directional approach where all words appearing before the predicate have negative distances and similarly, all tokens appearing after the predicate word have positive distances. This feature in particular, has given us better results.

The idea of any semantic role of a token is always in reference to a specific predicate, arguments and other labels like support words. Having understood that, we study our models with two levels of training knowledge to the system:

1) Only Predicate Knowledge: We provide the system with knowledge of only the predicate words in each sentence for which the ARG1 is found. Our intuition behind providing minimum knowledge of labels is that in real-world systems, we may not have so many labels to help us predict the ARG1.
2) All levels of knowledge about labels: Here, the system knows which words are the predicates, and which words are the arguments (and which type of argument the word is).

All models are trained with the above two levels of knowledge. We later see in the results section how more knowledge about labels aids our models to predict ARG1 more efficiently.

5 Experiments

5.1 AdaBoost Model- Baseline

AdaBoost is a modelling approach which focuses on converting a number of weak learners to strong learners. The weak learners in AdaBoost are decision trees with a single split, called decision stumps. AdaBoost works by putting more weight on difficult to classify instances and less on those already handled well. This builds a model and gives equal weights to all the data. It then assigns higher weights to those points that were wrongly classified. All the points which have higher weights are given more importance in the next model. The model keeps on training in this way until a low error rate is achieved.

Compared to traditional ML approaches like linear regression, AdaBoost accounts for non-linear relationships which often leads to better performance.
The baseline AdaBoost system makes use of the following features listed in Figure 2.

Current word	Previous-to-previous word POS	Next-to-next word POS
POS tag	Previous-to-previous word BIO	Next-to-next word
BIO tag	Next word	Next-Next-Next word
Stemmed word	Next word stemmed	Next-Next-Next word stemmed
Previous word	Next word POS	Next-Next-Next word POS
Previous word POS	Next word BIO	Next-Next-Next word BIO
Previous word BIO	Next-to-next word	Unigram embedded similarity
Previous-to-previous word	Next-to-next word stemmed	Slash embedded similarity

Figure 2 : Features used in Baseline AdaBoost

5.2 Random Forest Model

Random forest is a supervised machine learning algorithm widely used for its simplicity, flexibility and applications in classification problems. It can be adapted very easily to perform multiple tasks on large datasets. The decision trees are a main component of the random forest model, where we feed the classifier multiple features and the model will randomly choose these features a train itself, build a forest of many such decision trees and then average out the results to predict the values. The idea is to have many uncorrelated decision trees than to have just one.

For our random forest model we have provided the classifier with word level, position level as well as distance-level features for ARG1 identification.

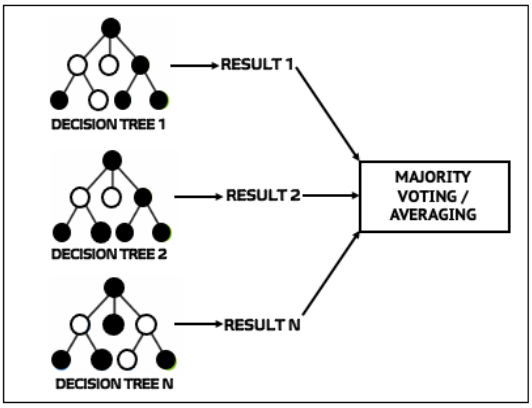

Figure 3 : Random forest classifier

5.3 BERT

We assume our dataset to be raw and completely unprocessed and just pass the word level tags - POS and BIO to our tokenizer. This pipeline makes use of the [6]BERT model. BERT as we know, is a state-of-the-art deep learning language model that has overwhelmed our machine learning people community by giving accurate results in a wide assortment of NLP undertakings including Semantic role labelling, Natural language inference and others. The indispensable headway in BERT is the training of the bidirectional transformer to learn the context of a sentence and relevance of each token, in contrast to the traditional approach of perusing a sentence left-to-right and right-to-left.

We have performed two kinds of experiments with our baseline BERT model - one by giving it word and position level features and another by giving an additional distance feature. We will later see how adding the distance feature helps the model predict ARG1 better.

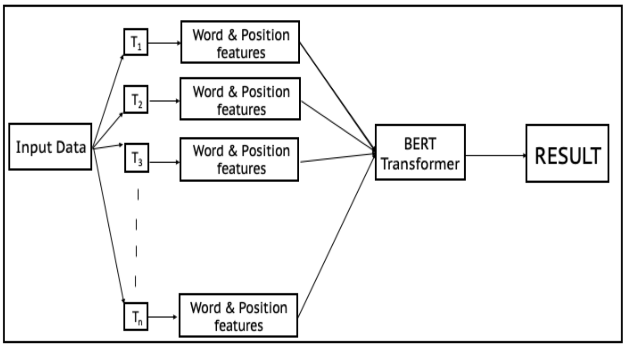

Figure 4 : Data pipeline BERT model

5.4 ALBERT

We extend our experiment to test our dataset against another variation of BERT - ALBERT. [7]ALBERT stands for "A Lite BERT" and it is quite literally a lighter version of BERT, because it takes much less training time than the latter, hence making it an effective variant for scaling to larger model sizes. Albert differs from BERT because it incorporates a factorization technique that trains against a smaller hidden dimension per word (128) and then learns to project it on a larger transformer dimension(1024). ALBERT also reduces the overall number of parameters by sharing its parameters across all the layers and calculating a sentence-order prediction loss.

We leverage ALBERT's fast performance to train and test our models. We observe that our ALBERT model surpasses traditional BERT's performance in accurately identifying ARG1.We have again performed two kinds of

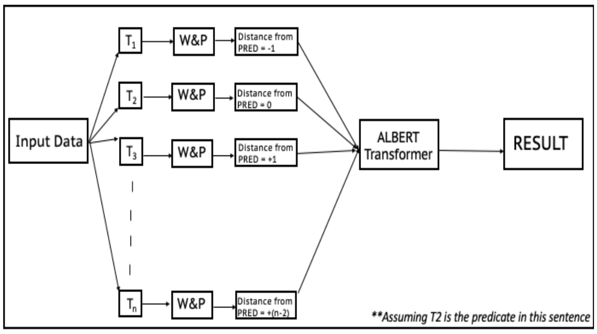

Figure 5 : Data Pipeline - ALBERT Model

experiments with our ALBERT model - one by giving it word and position level features and another by giving an additional distance feature. We will talk more about our results in the next section.

6 Results

The experimental results of our various models can be seen in *Figure 6.*

For a model like the Random Forest Model which was a low performing model we can see that providing the model with knowledge of all labels significantly improved the F1 score from *36.04 %* to *41.94 %.* However, the Baseline model AdaBoost model still outperforms the Random Forest model with all knowledge.

The enhanced BERT model with only knowledge of PRED Label achieves a F1 score of *89.74 %* compared to *83.54 %* when it had only the token and position features.

The ALBERT model sometimes identifies multiple occurrences of the same word as ARG1 multiple times where only one occurrence of the word is an ARG1.

For example, in the following sentence, only the first token 'Sales' is the ARG1 of the predicate '%'. However, our model identifies two ARG1 labels. The second ARG1 label is the word 'sales' occurring later in the sentence.

```
Sales    ARG1
at
general
merchandise
stores
rose
1.7
%
after
declining
0.6
%         PRED  PARTITIVE-QUANT
in
August
,
while
sales    ARG1
of
building
materials
fell
1.8
%
after
rising
1.7
%
.
```

Figure 7 : ALBERT marking more than one occurrence of a word as ARG1.

The best performing model, ALBERT (W+P+D) with knowledge of all labels achieves a F1 score of *94.26 %.* This is a significant improvement from the Baseline AdaBoost system which had a F1 score of *69.5 %.*

7 Conclusion

This paper presented various deep-learning models that can be used to find the ARG1 in the NomBank Dataset. Our best performing model was the ALBERT model with the position, word and distance features. This model had knowledge of all the labels and achieved an F1 score of *94.26 %* on the Nombank Dataset. This system which made use of a pretrained ALBERT model was able to predict the ARG1 label significantly better compared to the Baseline Model.

Model	Knowledge of only PRED label			Knowledge of all labels		
	Precision	Recall	F1 Score	Precision	Recall	F1 Score
Random Forest	55.88	26.59	36.04	53.61	34.44	41.94
BERT (W+P)	81.64	85.43	83.54	88.61	92.72	90.61
BERT (W+P+D)	86.96	92.72	89.74	90.14	92.05	91.08
ALBERT (W+P)	89.4	89.4	89.4	90.27	94.7	92.43
ALBERT (W+P+D)	87.73	94.97	91.08	91.95	96.69	94.26

Figure 6 : Precision, Recall & F1 score of all models

Our experiments showed that adding additional features helped increase the performance of the systems. Giving the model more information about other labels also increased the performance, this result was quite intuitive. However, systems which were given knowledge of only the ARG1 label gave good results. This is a positive result since in the real world we will often not have information of all the labels of a sentence when trying to identify the ARG1 label.

8 Discussion and Future Scope

We can make the following observation from the results. *(Figure 6)*

1. Using state-of-the-art deep learning models like BERT and ALBERT gave significantly better results than using canonical machine learning algorithms. Models like BERT are able to understand the context of words and phrases due to its large parameter size, ability to predict the next sentence, and make use of a masked language model.

2. Giving the machine knowledge of all the arguments improved the performance of all the models. The knowledge of all arguments helped the system train better.

3. The distance feature, which measured the distance of a token from the predicate in the sentence, helped increase the F1 score of both the BERT and ALBERT models. Since predicates and ARG1 are closely related arguments, incorporating this feature made the models more robust.

While we focus on analysing the impact of a two-fold knowledge input, feature engineering could possibly be helpful in improving the performance of models.
We used previous and next-word BIO and POS tags as position features. This feature could be extended to also include the POS and BIO tags of the previous-to-previous word and next-to-next word. The stemmed version of a token is also a possible feature that can be used.

Another promising approach is training the models on datasets that include multiple predicate (PRED) labels in one sentence. Currently, each sentence in the Nombank dataset consists of only one predicate label. For sentences with N predicates, every sentence can be processed N times, where each time the sentence has one predicate label. Using such an input processing pipeline can further highlight the impact of a feature like distance from predicates and can possibly help improve our models.

We have treated the identification of ARG1 of nominal predicates as a binary classification problem where the system predicts if a token is an ARG1 or not. Identification of ARG1 can also be treated as a question-answering problem where for each sentence, the model answers the question, “What are the ARG1 words in this sentence?”. Models like BERT and ALBERT are known to perform well in question-answering tasks, and a similar approach can be adapted for identifying the ARG1 of nominal predicates.

References

[1] 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.

[2] Hearst, Marti A. "Automatic acquisition of hyponyms from large text corpora." *COLING 1992 Volume 2: The 14th International Conference on Computational Linguistics*. 1992.

[3] 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.

[4] 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.

[5] 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, Beijing, China. Association for Computational Linguistics.

[6] 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.

[7] Lan, Zhenzhong, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. "Albert: A lite bert for self-supervised learning of language representations." *arXiv preprint arXiv:1909.11942* (2019).

[8] Shi, Peng, and Jimmy Lin. "Simple bert models for relation extraction and semantic role labeling." *arXiv preprint arXiv:1904.05255* (2019).

[9] Chang Liu and Hwee Tou Ng. 2007. Learning Predictive Structures for Semantic Role Labeling of NomBank. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 208–215, Prague, Czech Republic. Association for Computational Linguistics.

[10] Luheng He, Mike Lewis, and Luke Zettlemoyer. 2015. Question-Answer Driven Semantic Role Labeling: Using Natural Language to Annotate Natural Language. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 643–653, Lisbon, Portugal. Association for Computational Linguistics.

[11] Hai Zhao, Wenliang Chen, and Chunyu Kit. 2009. Semantic Dependency Parsing of NomBank and PropBank: An Efficient Integrated Approach via a Large-scale Feature Selection. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 30–39, Singapore. Association for Computational Linguistics.

[12] Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. 2019. What Does BERT Learn about the Structure of Language?. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3651–3657, Florence, Italy. Association for Computational Linguistics.

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

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

[15] Geoffrey Hinton, Oriol Vinyals, Jeff Dean, et al. Distilling knowledge in a neural network. arXiv preprint arXiv:*1503.02531, 2(7), 2015.*