Feature Analysis for Predicate Argument Identification using Random Forests

Jeremy Lu

Abstract

There are a variety of characteristics used to identify ARG1's in the predicate-argument relationships, such as N-gram, predicate, path, and embedding features. Among these, it is unclear which ones are most important when a machine learning model is identifying ARG1's. This paper utilizes feature and permutation importance in a binary classification random forest, to assess a multitude of features and determine their impacts. The most important features in the random forest were distance of word to predicate, word to predicate embedding distance, and the word itself.

1 Introduction

Semantic role labelling is an area of NLP that aims to have machines understand the roles of words in sentences. Based on this, the role of each word or phrase in the full sentence can be identified. As an example, in the sentence:

"The position of the United States, which once contributed 25% of the budget, is that nothing has changed."

The word "budget" represents the entity described by the predicate of "%". This can easily be identified and interpreted by individuals familiar with the English language. However, this task is not trivial for a machine. In this example, "budget" is referred to as the ARG1 of the predicate, which is "%". Multiple ML approaches have been taken in identifying these ARG1's in a sentence, given a predicate, such as using sklearn's max entropy regression or AdaBoost. For these models, much deliberation is needed in determining which features of the sentence should be used in identifying ARG1's. Furthermore, the relevance of these features is also in question; is it necessary to have embedding features? N-gram features? Predicate information? Using a random forest, these feature importances can be easily attained, giving us a better understanding of what is important for machines to consider when determining ARG1's given percent and partitive predicates.

2 Data

The data used in this project is from the Nombank annotation project. It consists of annotated sentences from the Wall Street Journal, and contains around 1,000,000 words, with around 200,000 nouns. There are 3000 occurences of "%", resulting in 2200, 80, and 150 sentences for training, development, and testing data, respectively. For the partitive data, there are around 15,000 sentences, in a 10,800/396/650 train/development/test set.

Each word in our total corpus has six potential columns: Word, POS tag, Bio tag, Word in Sentence Number, Sentence Token Number, and PRED/Role. An example is shown in the following table:

Word	POS	BIO	WID	SID	Role
The	DT	B-NP	0	9	
August	NNP	B-NP	1	9	
GDP	NNP	I-NP	2	9	ARG1
was	VBD	O	3	9	
up	IN	B-PP	4	9	
2.4	CD	B-NP	5	9	
%	NN	I-NP	6	9	PRED
from	IN	B-PP	7	9	
its	PRP$	B-NP	8	9	
year-earlier	JJ	I-NP	9	9	
level	NN	I-NP	10	9	
.	.	O	11	9	

Table 1: Example sentence in NomBank Corpus

We use our training data to build a corpus to build a vocab on and also to train our model. The development set is solely for hyperparameter tuning, evaluating the F1-score to select the best model.

Finally, precision, recall, and F1-score are all evaluated on the final model, and feature and permutation importances are gathered based on this.

3 Building the Random Forest

3.1 Feature Selection Process

For bot the percent and partitive model, each word in the data set comes with a total of 20 features, listed below:

Feature	Description
Word	Word if it is in vocab
Bio	Word's Bio tag
POS	Word's POS
TBWord	Word 2 positions before
TBBio	Bio tag 2 positions before
TBPOS	POS 2 positions before
OBWord	Word 1 position before
OBBio	Bio tag 1 position before
OBBPOS	POS 1 position before
OAWord	Word 1 position after
OAPOS	POS 1 position after
OABio	Bio tag 1 position after
TAWord	Word 2 positions after
TABio	Bio tag 2 positions after
TAPOS	POS 2 positions after
Pred Word	Predicate Word
Pred Bio	Predicate Bio tag
Pred POS	Predicate POS
Embedding Distance	Word to Pred Embedding Dist
Distance to Predicate	Number of words from Pred

Table 2: Description of Features in Models

The model accounts for the word, POS, and Bio tag, along with these attributes from two words before and after (NA if they do not exist). Each word is represented as it's unique ID mapping in the vocab, which consists of all words (non case-sensitive) that appear at least 10 times in the the training corpus. All other words are labelled as outside of vocabulary, or OOV.

Predicate features were also included as a feature in past models. For this project, we choose to include the Predicate word, Bio tag, and POS, as our features. The intuition is that certain words may be more likely to be the ARG1 a given specific Predicate word. However, for the percent model, this was less useful, as all the predicates were "%".

We also captured the relationship between the word and predicate by using Distance to Predicate. This is simply the difference of the word number (order of words in the sentence) of the given word to the predicate. Given an example sentence from our training corpus "But about 25 % of the insiders COMMA according to SEC figures COMMA file their reports late." we have that the Distance to Predicate for the words "But" and "SEC" are -3 and 7, respectively ("%" has word number 3, so "But" is 0-3 and "SEC" is 10-3).

Finally, word embedding features gave a notable boost to the recall of the AdaBoost model featured in the lecture talk (Meyers, 2022). Thus it makes sense to also implement embedding features in our model, especially to assess its importance when compared to other features. Word embeddings allow us to express words as a set of numbers in a vector space, with similar words being closer together in the embedding space. As a result, we use the feature of Embedding Distance, which is normalized and measures the distance of a given word to the predicate in the embedding space. The logic behind this decision is that certain types of words in the embedding space may be more likely to be ARG1's for a given predicate, and to quantify this we use the embedding distance. For the vectors themselves, we use `gensim`'s pre-trained vectors, called `glove-wiki-gigaword-300`, which have been trained through Wikipedia. More details can be found at https://nlp.stanford.edu/projects/glove/.

3.2 Hyperparameter Tuning

To find the optimal parameters for our random forest, a grid search was conducted by building random forests on the training data and then evaluating the performance of the F1 score on the development data set. Both the percent and partitive models considered `max_depth` values of 5, 10, 15, 20, 30, and 40 along with `n_estimators` values of 50, 100, 250, and 500. max_depth determines the depth of each decision tree, and `n_estimators` determines the number of total trees in the random forest. After conducting a grid search, the optimal parameters for the percent and partitive models were `n_estimators = 500` for both and `max_depth = 20` and `max_depth = 30`, respectively.

The other parameters for the random forests were left as defaults. Those defaults can be viewed on sklearn, at https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html.

4 Model Evaluation

4.1 Percent F1 Score

After training the percent model and running it to make predictions on the test data, we find that the model has a precision, recall and F1 score of 92.47%, 57.33%, and 70.78%, respectively. To get a gauge of how good this is, we compare this to the aforementioned AdaBoost model, which has N-gram, path, predicate, and embedding features (Meyers, 2022):

Model	Precision	Recall	F1-score
AdaBoost	83.33%	51.72%	63.83 %
Random Forest	92.47%	57.33%	70.78%

Table 3: Comparison of Accuracy Measures for Percent Models

Looking at the table, we can see that the Random Forest outperforms the AdaBoost model in all three measures of accuracy. Notably, the recall which is quite low relative to precision has jumped over 10% to 57.44%. The precision has also increased to 92.47%, showing that the Random Forest is extremely confident and accurate for words that it does label as the ARG1. Overall, this model is an improvement from past models, and will make the results from looking at feature importances relevant.

4.2 Partitive F1 Score

After training the partitive model and running it to make predictions on the test data, we find that the model has a precision, recall, and F1 score of 93.34%, 46.53%, and 62.10%, respectively. Again, we will compare the AdaBoost model for ARG1 identification given a partitive predicate (Meyers, 2022).

Model	Precision	Recall	F1-score
AdaBoost	87.08%	48.92%	62.65%
Random Forest	93.34%	46.53%	62.10%

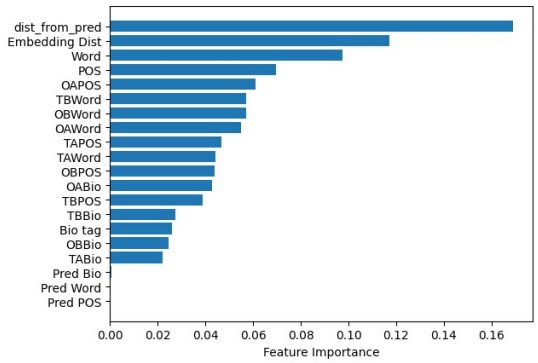
Table 4: Comparison of Accuracy Measures for Partitive Models

Based on this table, we can see that the Random Forest has an increase of just over 6% in precision. This indicates that for words that are labelled as ARG1 by the machine, the Random Forest has a slightly better accuracy than AdaBoost. However, the recall is slightly lower compared to AdaBoost. This means that the Random Forest did not classify enough of the actual ARG1's in the test set as ARG1, compared to the AdaBoost. Due to the lower recall, the F1 score is also slightly lower. Overall, the Random Forest for the partitive task is comparable to the AdaBoost, which still means that our intepretations for feature importance will be relevant.

4.3 Percent Feature Importances

Sklearn's feature importance in random forests works by measuring the decrease in node impurity at each split the feature is considered at. This decrease is then averaged across all splits with the feature in the forest, resulting in the final feature importance. Impurity refers to gini impurity, which is essentially a measure of homogeneity. The higher the purity, the better the split is at determining data; so higher decreases in purity mean that the featuer is more important. One of the weaknesses of feature importance is that it is affected by highly correlated variables. For example, if `word`, `POS`, and `Pred Word` were highly correlated, their importances may be diminished as the importance is spread across three features rather than just one.

The feature importances of our percent model are displayed in the following graph:



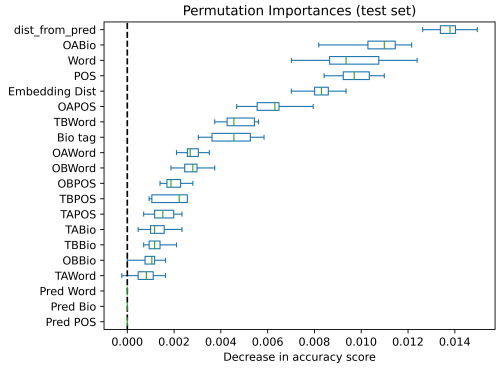
Based on this visualization, we can see that a word's distance from the predicate, it's embedding distance, and the word itself are the three most important features. It makes sense that distance from predicate would be very important, since typically the ARG1 will be close to the predicate; this phenomenon is also more emphasized as the sentence becomes longer. The word itself is also important, which makes sense. Certain words in the training corpus are probably more likely to be ARG1 of a percent predicate, and this is reflected in the model's feature importance. Next, we see that POS is also considered, along with the N-gram statistics, which is information about the words nearby. Fi-

nally, the predicate features are not important at all, which is the case because we only have "%" as the predicate in this model.

One key takeaway from the models presented in class was that the inclusion of embedding features led to an increase in all accuracy metrics, especially for the percent model. This is consistent with the random forest, as the embedding distance is the second most important for a percent model. As hypothesized, the grouping of certain words associated with a predicate (in this case "%") in the embedding vector space might reveal patterns useful for making binary classifications. Using distance, we were are able to find the type of words associated with "%", and this was used frequently by the random forest.

We also look at permutation importance, which measures how much a model relies on a feature when making classifications. This is calculated by randomly shuffling a given feature and then measuring how much the accuracy decreases on a test set. Through shuffling, the feature becomes effectively useless to the model; the more important a feature is, the more the accuracy will decrease. One advantage of the permutation importance is that it reflects both the model and test set characteristics, compared to feature importance which only contains model information.

Permutation importances for the percent Random Forest model are displayed below:

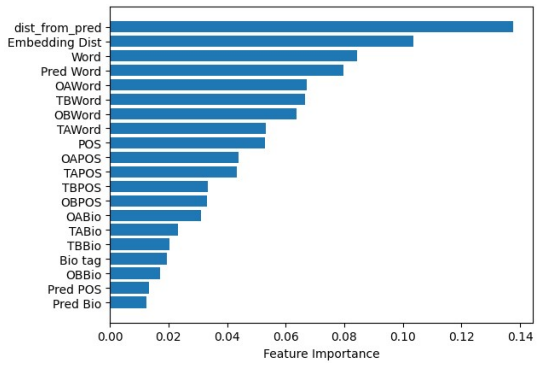


Again, we see that the distance from predicate remains the top feature. Additionally, the word itself, its part of speech, and bio tag are also important, with word and POS remaining in the top 4 just like the feature importances. One point of interest is that embedding distance is only 5th highest. However, this may be a reflection of the test set, and overall it is still a valuable feature to be included in the model. Finally, it appears that the word immediately after has more importance than the other neighboring words, but the reason is unclear.

Overall from both these graphs, we see that the most important features are the distance from predicate, embedding distance, and the features of the word itself (word, POS, Bio tag). This is consistent with the information we know from the previous AdaBoost model, with embedding features greatly helping improve model accuracy (Meyers, 2022). These results from the percent model also suggest that focusing on the relationship between words and the predicate will help to identify the ARG1.

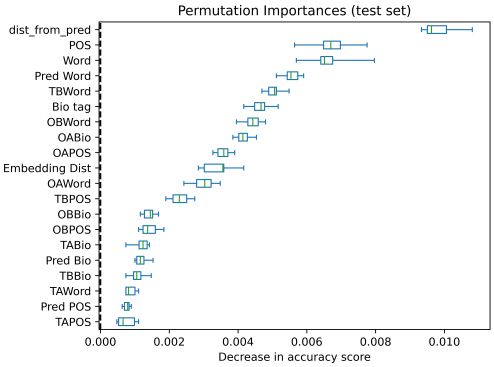
4.4 Partitive Feature Importances

The feature importance for partitive random forest model are displayed in the following graph:



Just like the percent model, distance from predicate and embedding distance are the top two features. This is probably for similar reasons as mentioned – ARG1's are often closer to the predicate and certain words may be more likely to be ARG1 based on the predicate, which can be measured through embedding distance. Because the partitive task differs from the percent one as it has multiple predicate words, the predicate word feature is fourth, much higher than in the percent task. The word itself and neighboring words are also important to the model, much more so than POS and Bio tags. This is different from the percent model, which also viewed words, BIO tags, and POS roughly equal. The partitive model has distinct sections of importance with word, POS, then bio tag. Overall, the embedding distance, distance from predicate, and words (word, predicate and N-gram words) are the most important for the partitive random forest model.

Next, we take a look at the permutation importances for the partitive random forest, which are calculated on the NomBank test set:



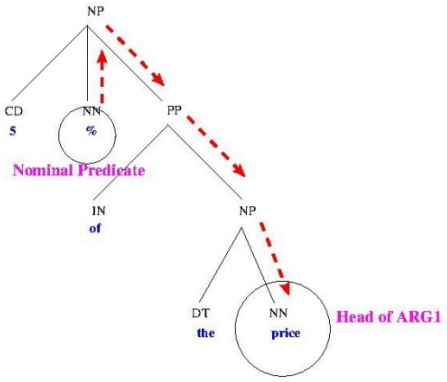
Again, distance from predicate remains the most important feature, just like in the percent model. POS then becomes the most important, followed by the word and the predicate word. Besides the predicate word, these features were also very important for the percent model. Surprisingly, embedding distance drops significantly in importance when moving from the Gini impurity to the permutation method. This could be due to its lessened importance in permutation importance, or that embedding distance is correlated with other features. When embedding distance is shuffled and there is minimal decrease in accuracy, it might be that other features also capture the information contained in embedding distance. These results for permutation importance may be more conclusive if we expanded the test set.

Overall, distance from predicate remains the most important feature in the partitive task. Word, POS, and embedding distance are also important, although the embedding distance drops in the permutation method. There are also clear levels of hierarchy with the N-gram words being more important, followed by POS, and Bio tag. Finally, the predicate word becomes important due to the variety of predicate options in the partitive task.

5 Future Work

5.1 Implementing Path features

One key element that was not included in the four random forest models were path predicates. Since predicates and their ARG1's have common structures in parse trees, features that describe traversal can be used to help improve the model. For example, a typical ARG1 format for a "%" predicate would be "% + Prep + ARG1 object", and this pattern can be used to detect ARG1's. An illustration of this is shown below (Meyers, 2022):



Being able to traverse this tree and find information about elements such as the head noun, conjunctions, etc... may also have information that can be used by the model. In the results of the AdaBoost model, the inclusion of path features greatly increased all accuracy measures for both the percent and partitive tasks (Meyers, 2022). This could also be implemented in the random forest, to compare the importance of these features to embedding, predicate, and other ones currently used and evaluated.

5.2 Miscellaneous features

Past papers at the intersection of predicate argument classification and machine learning have also worked with a variety of other features. These were not featured in the AdaBoost model, but they might improve accuracy as well as give us more insights on how a machine might identify ARG1's. One of these is the active/passive voice of the predicate phrase, which was mentioned in a 2004 paper (Moschitti and Bejan, 2004). While details were not given, the voice of the predicate phrase may help give an idea of what words can be the ARG1. Another feature mentioned in this paper was the governing category, which indicates if a noun phrase is dominated by a sentence or a verb phrase (Moschitti and Bejan, 2004). This is related to the path, but it may also be useful feature derived from a linguistic approach. Finally, in a 2014 paper, whether a potential argument was part of a definite or indefinite NP was included as a feature for machine semantic role labelling (Stern and Dagan, 2014). This essentially means if the noun is described with "the" vs. "a" or 'an" (or certain situations without an article), and could be related to any words being classified as an ARG1.

Overall, voice, governing category, and definite/indefinite NP are all interesting features that could be included in future models, in order to see

how important they are. These features were not included in any of the aforementioned AdaBoost models, so learning more about these features and their importance in properly classify ARG1's can deepen our understanding of predicate-argument relationships and semantic role labelling (Meyers, 2022).

5.3 Model-based changes

In this paper, we have looked at the results of different modeling approaches: AdaBoost and random forests. For future work, we can also explore how other models work, and what features they consider to be most important. Support Vector Machines could be used for binary classification, and sklearn offers the ability to find importances. Logistic regression is another option, but properly interpreting the coefficients is not the same as feature importance, since the coefficients in any regression reflects solely on the training set. Finally, we could also consider a Bayesian approach with a Naive Bayes classifier. Again, sklearn allows for us to easily obtain this model architecture's feature importances as well. Deep learning is another more powerful and advanced approach, but because of its complexity, the results and use of the features is harder to interpret.

Overall, looking at other models and their feature importances can help us gain further understanding of which features are most useful for identifying ARG1's. Comparing the AdaBoost and random forest models to other classification modles such as logistic regression, SVM, or Naive Bayes allows us to compare and contrast feature importances for multiple approaches, giving a more holistic view of what features are best to capture predicate-argument relationships.

6 Conclusion

In this paper, we have explored the task of identifying ARG1's in the predicate-argument relationship, using both percent and partitive predicates. By training random forests and looking at the feature importance, our results show that:

- The random forest is comparable to the AdaBoost model in terms of precision, recall, and F1 score; it is slightly better for the percent model and comparative for the partitive model.
- Features that show the relationship between word and predicate, such as Distance from predicate and embedding distance, were the most important features.
- The word and POS itself were also very important, and for the partitive task, the predicate word was important as well.

All in all, features that aimed to describe the word to predicate relationship from a linguistic approach were most useful to the model. Embedding and predicate features were considered important in our random forests, and it is likely that path features will be very useful as well. Due to the similarities in results between the AdaBoost and random forest, it can be deduced that the mentioned important features will also maintain their relevance in other model architectures.

Acknowledgments

Thank you to Zilang Zeng my TA advisor for providing me with ideas and feedback. He helped me to understand predicate, path, and embedding features. Based on my meetings with him, I was able to implement predicate and embedding features, which greatly helped my model's and insights regarding feature importance.

And thank you to Professor Adam Meyers for his resources on semantic role labelling and identifying ARG1's in the predicate-argument relationship. Additionally, his work on NomBank and previous modeling for the percent and partitive tasks were very helpful in providing a starting point for this project.

References

Adam Meyers. 2022. Nombank semantic role labeling tasks hw 6 and final projects. University Lecture.

Alessandro Moschitti and Cosmin Bejan. 2004. A semantic kernel for predicate argument classification.

Asher Stern and Ido Dagan. 2014. Recognizing implied predicate-argument relationships in textual inference. volume 2, pages 739–744.