1. Problem Formulation

This paper explores various methods for word-sense disambiguation. The methods explored are of a varied nature, including knowledge-driven and data-driven approaches [1]. The methods explored are: Lesk's Algorithim, LeskPOS (an adaptation of Lesk's algorithm that is based on POS tags rather than definitions), the Most Frequent Sense (MFS) approach, a bootstrapping approach using a support vector classifier, and a large-language comprised of a pre-trained RoBERTa with an added fully connected layer and an attention mechanism (R+F+A).

2. Datasets

2.1 SemEval 2013 Shared Task #12

Curated by Navigli et al. [2], this dataset comprises 13 articles spanning various domains. It was used to directly evaluate Lesk's Algorithm and LeskPOS. It also It serves a dual purpose: as a development and test set for both the bootstrapping and R+F+A methods.

2.2 Gigaword Training Corpus

The Gigaword dataset [3] comprises approximately 4 million article titles. It was used as a source of unlabelled data to train the bootstrapping method.

2.3 SemCor

SemCor is a manually labeled dataset that is mapped to WordNet 3.0. Its data is sourced from 352 different texts [4]. It was used as both an unlabelled dataset to train the bootstrapping method, as well as the labeled dataset to train the R+F+A approach.

3. Data Pre-Processing

Across all methods, the data was converted to lower case. Further, the use of lemmatization and stopword removal was analyzed to determine their impact on the performance of the evaluated methods. A correct prediction entails that the model predicts one of a word's senses. Furthermore, multi-word lemmas -like 'United States'- were treated as single entities.

4. Methodology

4.1 Lesk's Algorithim

In Lesk's Algorithm, context words and definitions were tokenized. After preprocessing both the tokens, they were fed into the algorithm. The algorithm works by calculating the overlap between context words and the words in each possible definition of the ambiguous word. The sense with the greatest overlap is presumed to be the sense of the word being disambiguated. The algorithm's effectiveness was evaluated based on its accuracy. Both NLTK's implementation of Lesk's algorithm, and an implementation I made from scratch, were evaluated.

4.2 LeskPOS

LeskPOS enhances Lesk's algorithm by integrating Part-of-Speech (POS) tagging into the disambiguation process. In addition to context-definition overlap, LeskPOS compares the POS of the ambiguous word with the POS of each sense. This is done by mapping the ambiguous word's POS tag to WordNet format and filtering out the senses that have a different POS tag.

4.3 Most Frequent Sense

The MFS method determines the sense of the ambiguous word based on its most common sense.

4.4 Bootstrapping

This is a semi-supervised learning approach. A seed set, consisting of 10 example sentences for each possible sense of each ambiguous word, was generated via OpenAI's GPT 4. The ambiguous words were selected based on their number of lemma senses, and their frequency in the test set. I used a bigram feature extraction approach due to the small amount of labeled data. Further, unigrams did not capture enough context, and trigrams caused overfitting. I chose an SVC as the classifier for its ability to manage non-linear data, and ability to deal with high-dimensional data efficiently.

The SVC was trained on the labeled seed set before being applied to unlabeled data from the Gigaword and SemCor datasets. The datasets and the seed sets were filtered per term to only include relevant sentences (sentences with the ambiguous word). Classifications above a confidence threshold are added to the seed set as labelled data. The classifier is then re-trained on the expanded seed set and applied again to the unlabeled train data. This is done until all training data is utilized, or until a predefined number of iterations are reached. For testing, the trained classifier is applied to a version of the SemEval 2013 Shared Task #12 dataset that is filtered to only include relevant instances.

The bootstrapping method was evaluated on 9 different terms, over various pre-processing decisions and hyper-parameter values. These results are shown on figures 2 to 7 in the appendix. *4.5 R+F+A*
I implemented the pre-trained RoBERTa base model [5], and I augmented it with a fully connected layer and an attention mechanism. This architecture was chosen to leverage RoBERTa's advanced language understanding capabilities derived from its extensive pre-training on diverse language data. The addition of the fully connected layer allows for specific fine-tuning, and it employs parameter sharing- enabling the features to be more generalizable. The attention mechanism and LASSO regularization helps the model extract contextually relevant features from the high-dimensional input data. The results are shown on figure 9 in the appendix.

5. Results and Discussion

5.1 Lesk's Algorithim, LeskPOS, and MFS

As shown on figure 8, the MFS method yielded the highest accuracy on the test set. This suggests that the dataset is biased towards more frequently occurring senses, which is common in English. The lower accuracies of Lesk's algorithm are expected given its inherent limitations: Lesk's approach is very sensitive to the exact wording of definitions, so the absence of a certain word can radically change the results. Moreover, the algorithm determines overlaps only among the glosses in WordNet. These glosses are short and do not provide sufficient vocabulary to relate fine-grained sense distinctions. LeskPOS performed better than Lesk's algorithm, but it also suffers from the same limitations as Lesk's algorithm.

5.2 Bootstrapping

The results vary significantly for each term. This can be attributed to the fact that word senses are not represented equally in the training, and testing data. This can cause words to overfit to certain senses, and hence generalize poorly. The effect of hyperparameter tuning on the results exhibit how sensitive the model is to distributions of word senses. For instance, when the training data size doubles, the accuracy of 'action' drops from ~75% to 0, while 'claim' increases from about 10% to 90% (Figure 2). This indicates that the model tends to overfit the data.

5.3 R+F+A

Like the bootstrapping approach, the results vary significantly (Figure 9). Perfect scores for 'game' and 'year', while an accuracy of 0 for 'law' and 'rule' signals that the model is overfitting to word senses. Further, the increase in accuracy when LASSO is used further indicates that the model may be too complex and may overfit the data.

6. Difficulties faced:

The most challenging part of this assignment was getting the bootstrapping method to not overfit on the seed set. It was also difficult to get the R+F+A to work due to how computationally expensive it is to train.

7. Improvements:

The most important improvement would be to use datasets that have close to uniform representations of word senses for the words. Further, more investigation is needed on optimizing the hyperparameters for the SVC used in the bootstrapping method, and fine-tuning the R+F+A.

8. Sample Output:

```
Word: end
Context: on Thursday , the widely follow trial of steven_j._hayes , who be convict of kill three member
Prediction: end%1:28:00::
Correct Answer: end%1:28:00::


Word: end
Context: as we all know , a few minute before the end of the game ( that their team have already win )
Prediction: end%1:28:00::
Correct Answer: end%1:11:00::
```

Figure 1- An example of the bootstrapping program's output. The first example is a correct classification, and the second is incorrect.

9. Bibliography

[1] Hu, Z., Luo, F., Tan, Y., Zeng, W., & Sui, Z. (2019). WSD-GAN: Word Sense Disambiguation Using Generative Adversarial Networks. Proceedings of the AAAI Conference on Artificial Intelligence, 33(01), 9943-9944. https://doi.org/10.1609/aaai.v33i01.33019943

[2] Roberto Navigli, David Jurgens, and Daniele Vannella. 2013. SemEval-2013 Task 12: Multilingual Word Sense Disambiguation. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 222–231, Atlanta, Georgia, USA. Association for Computational Linguistics.

[3] Graff, D., Kong, J., Chen, K., & Maeda, K. (2003). English gigaword. Linguistic Data Consortium, Philadelphia, 4(1), 34.

[4] Francis, W. N. & Kucera, H. (1979). *Brown Corpus Manual* (). Department of Linguistics, Brown University, Providence, Rhode Island, US.

[5] Y. Liu et al., "RoBERTa: A Robustly Optimized BERT Pretraining Approach," CoRR, vol. abs/1907.11692, 2019, [Online]. Available: http://arxiv.org/abs/1907.11692

(Appendix is on the next page).

10. Appendix

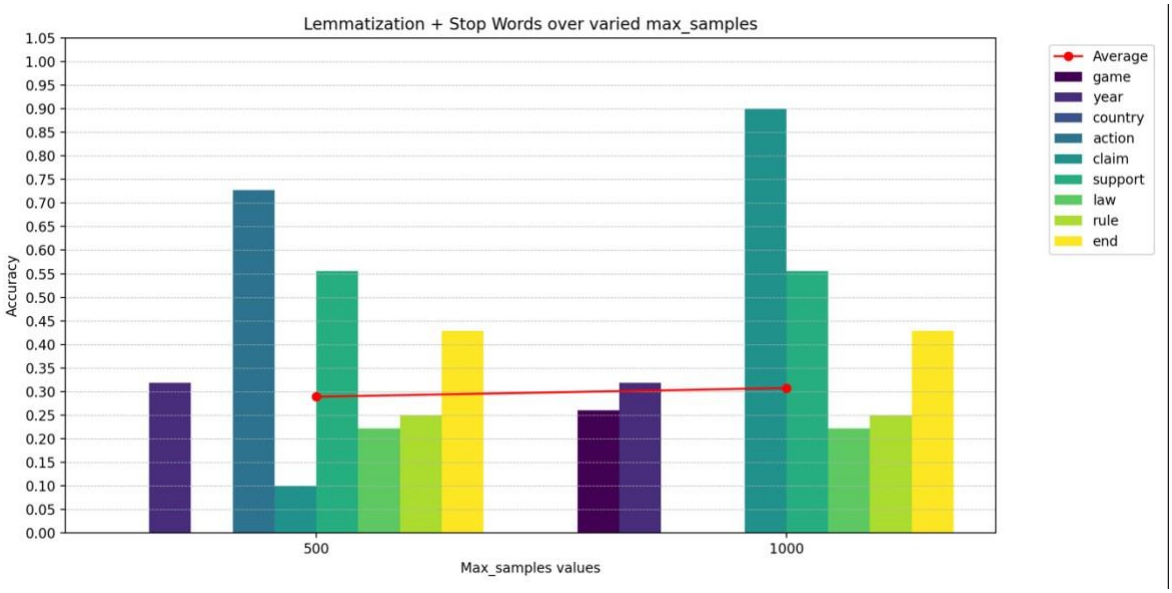

Figure 2- Bootstrapping accuracies vs train data size with data that has been lemmatized and stop words have been removed.

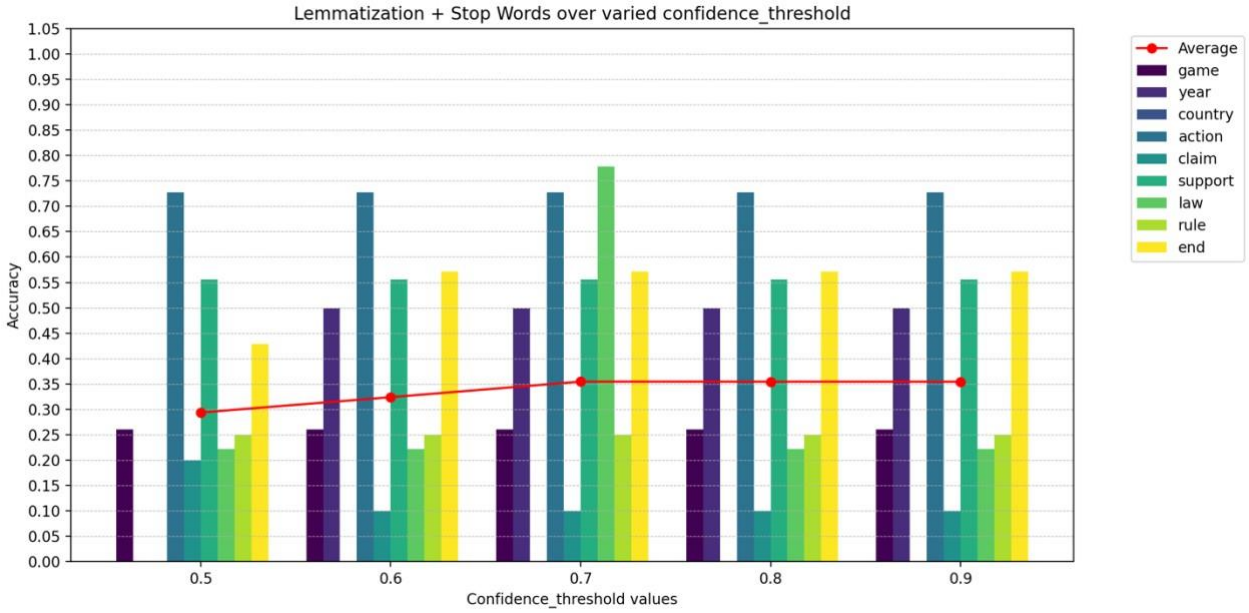

Figure 3- Bootstrapping accuracies vs max iterations with data that has been lemmatized and stop words have been removed

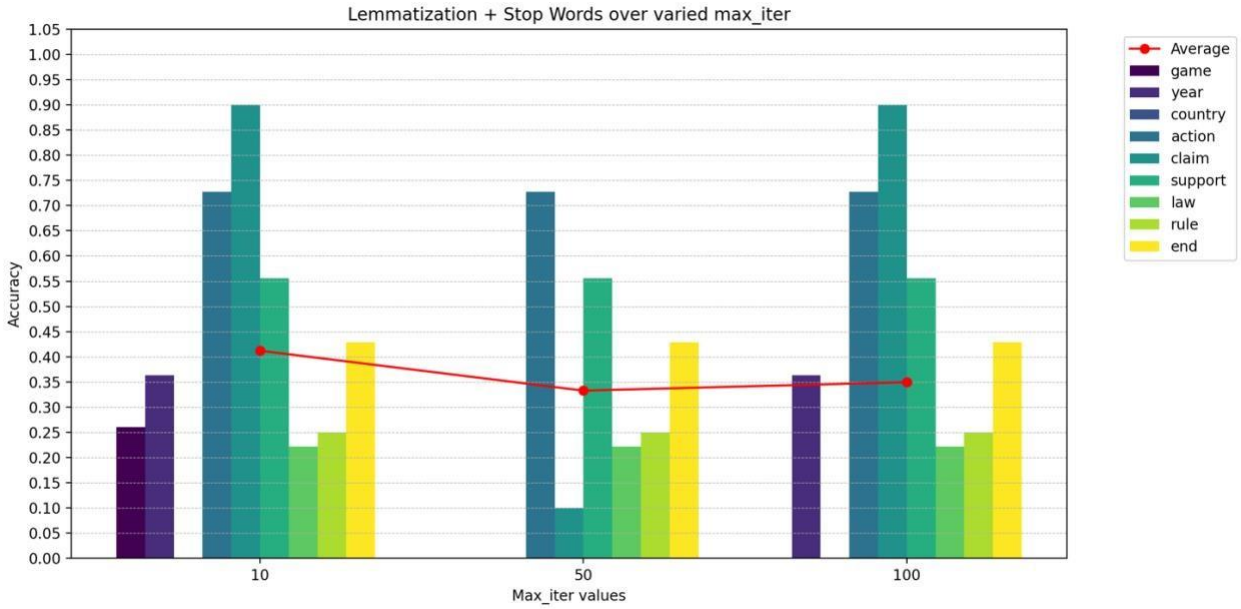

Figure 4- Bootstrapping accuracies vs confidence threshold with data that has been lemmatized and stop words have been removed.

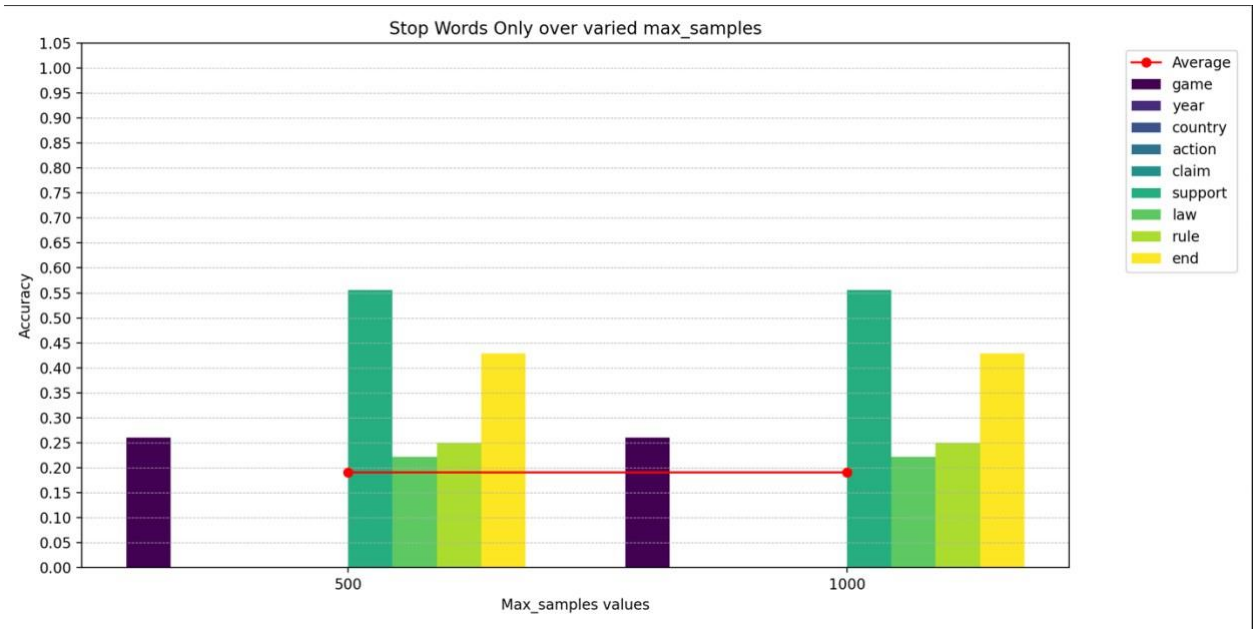

Figure 5- Bootstrapping accuracies vs train data size with data that has been pre-processed by stop-word removal.

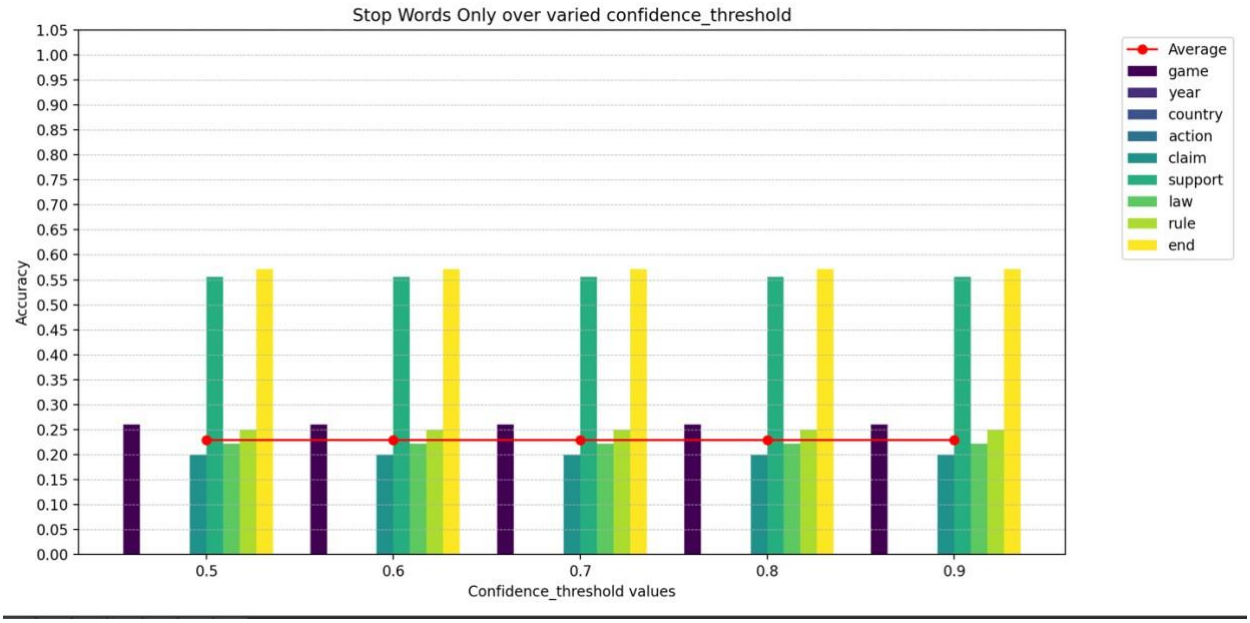

Figure 6- Bootstrapping accuracies vs confidence threshold with data that has been preprocessed by stop-word removal.

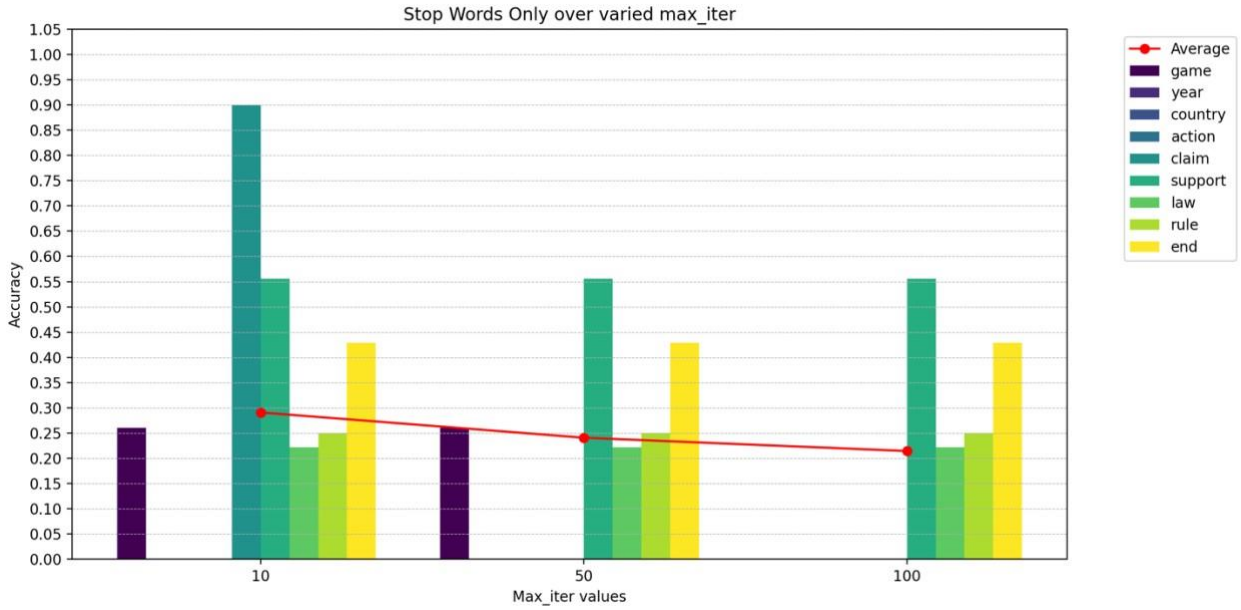

Figure 7- Bootstrapping accuracies vs maximum iterations with data that has been preprocessed by stop-word removal.

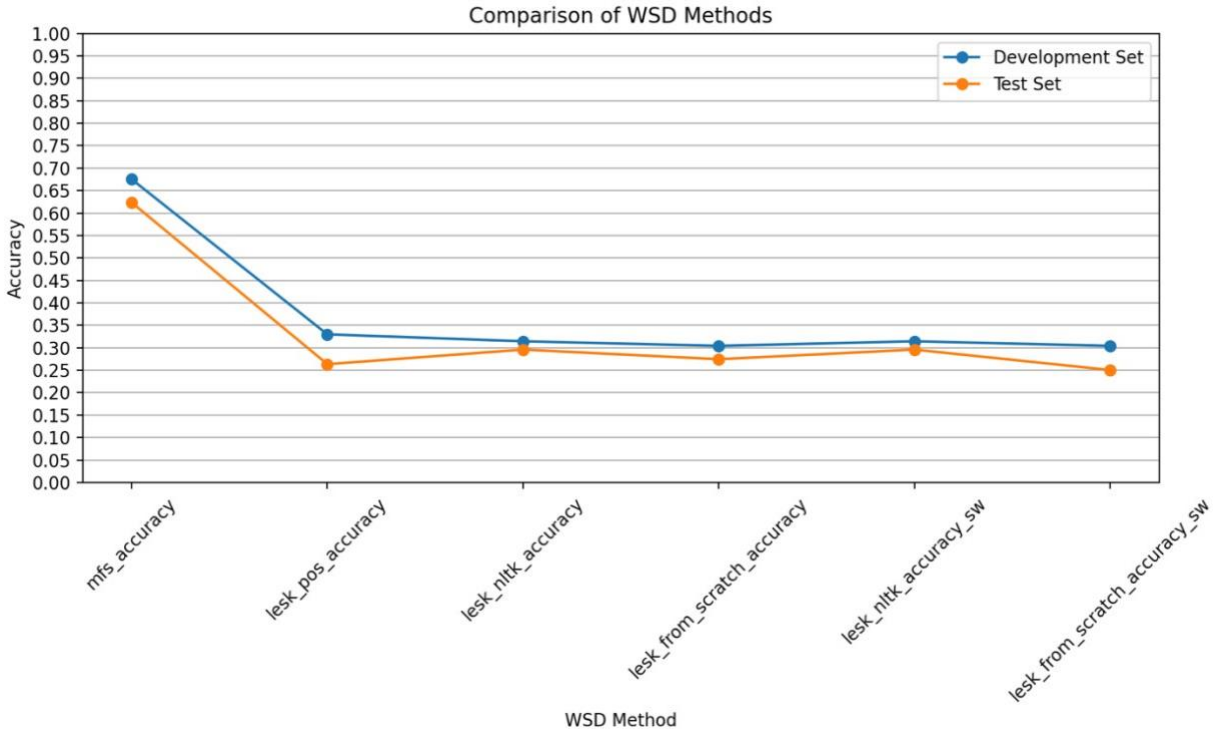

Figure 8- Accuracies of Lesk's algorithm, LeskPOS, and the MFS method.
"sw" indicates stop-word removal of the data. Otherwise, the data has been lemmatized and the stop words have been removed.

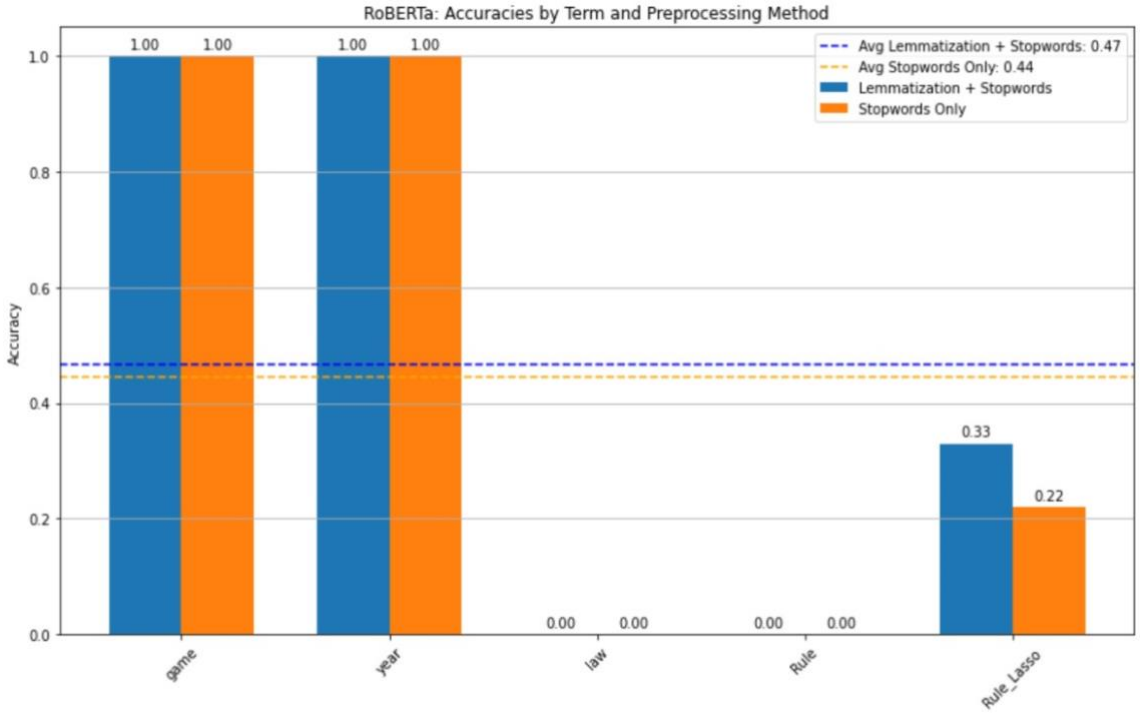

Figure 9- RoBERTa accuracies by term and pre-processing method.