Project 2

Anonymous ACL submission

1 Introduction

The ability to classify simple and complex words correctly is essential in many applications of NLP, including readability assessment and simplification. This study presents several machine learning algorithms that determine their applicability to the correct identification of simple and complex words. Therefore, this model will be compared on the basis performance metrics such as Accuracy, Precision, Recall, and F-score. Our system would take as input a labeled corpus comprising single word annotations of being either simple or complex, along with word frequencies, numbers of syllables, and number of synonyms. The methodology followed is using simple classifiers to more complex machine learning algorithms such as Naive Bayes, Logistic Regression, and finally, Support Vector Classification. The goal is to analyze the performance of various classifiers and to determine which features contribute most effectively to classification accuracy.

2 Approach

For this project a number of classification algorithms were tested in their ability to correctly identify complex and simple words. These algorithms were judged based on the Accuracy, Precision, Recall, and their Fscore.

The Data used in this project was a corpus of words that also contained their label, whether they were simple or complex, noted as a 0 or 1, the number of annotators, a sentence containing the word, and the sentence index. For this program we parsed the corpus and only pulled out the word and its described label.

3 Design and Implementation

In the attempt to classify words as simple and complex there were three different classifications models used. Each more complex than the previous one, the first and simplest model was a model that classified all words as complex. The next classification was based on the length of the word. The last of the simple classifiers was based on the word frequency from a corpus of Google's Ngram Frequencies, and similar to the previous classifier we make a prediction based on the frequency of the word instead of its length. The first of the machine learning classifiers was the Naive Bayes from SkLearn. This classification uses the Naive Bayes algorithm from section background information Naive Bayes Classifiers. The Features used in this classifier were the word length and the word frequency. The second of the machine learning classifiers was the SkLearn Logistic Regression classifier. Similar to the previous classifier we use the word length and word frequency as our features. The final classifier known as MyClassifier uses SkLearn's Support Vector Classification algorithm. The new features added to this classifier was the syllable count and the number of synonyms. This classification model was run with just those two features, but also ran with the features word length and word frequency as well as the two new features.

4 Results

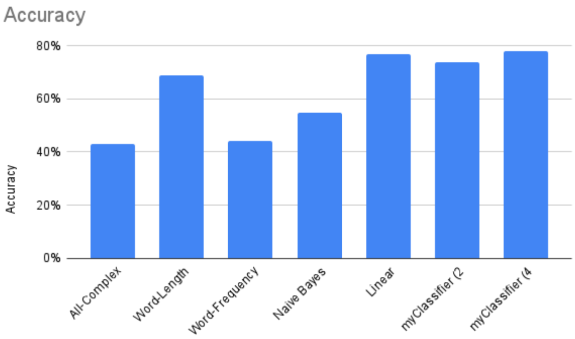

Figure 1: Accuracy of each model

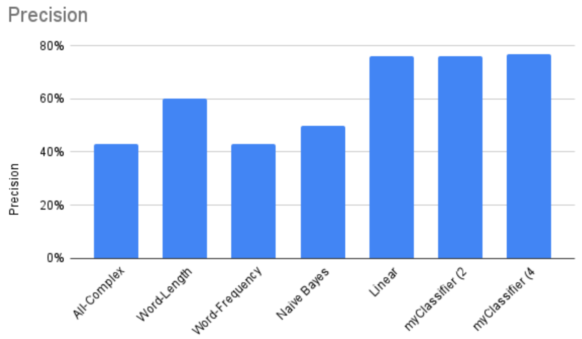

Figure 2: Precision of each model

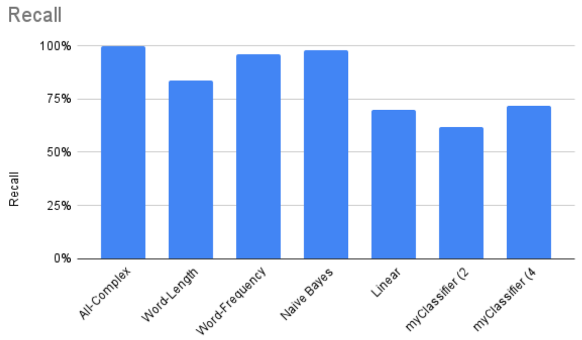

Figure 3: Recall of each model

From looking at the results of each of these classification models it is clear that while labeling all words as complex gives us the highest Recall, it is overall a poor classification model. The reason we get the highest recall is due to it was able to correctly classify all of the complex words due to recall focusing on of the model correctly identifies all of the items in the specific class. It struggled in every other category because of incorrectly classifying all simple words as complex. The word length model required some parameter tuning to maximize the F-score. After testing out all possible lengths from 1 to 17, the length that maximized F-score was found to be 7. The next Classification model was very similar, but for maximizing the F-score no parameters were able to get above 60 percent. The outcomes from words with frequencies of 40 to 8,600 all received an F-score of 60 percent and anything above gave a lower score. Using the Naive Bayes classifier it was unable to beat the word length classification model which was surprising. The reason that this is believed to be is due to the word frequency list not being as helpful as a feature as the word length, so it lowered the total correctness of the algorithm as a whole. The second classification model was the linear regression model which performed the best with only two inputs. This is not surprising because the features have weights, unlike the Naive Bayes classification. The final model tested was myClassifer using the SVC module. When this model was tested with the same features as linear regression, it performed worse, but when increasing the features to allow for syllables and number of synonyms, it performed better. This is due to the SVC classification model being better with nonlinear classification.

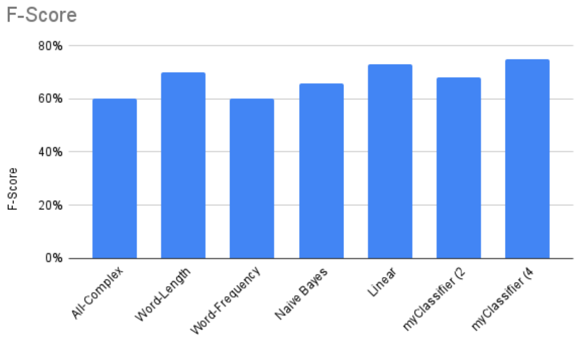

Figure 4: F-score of each model

5 Conclusion

This project demonstrated through extensive experimentation that classification performance greatly depends on the chosen model and features. The simplest classifier, which labeled all words as complex, had a very high recall but was not effective in general, since it could not make any differentiation among word types. Rule-based classifiers using word length and word frequency were somewhat superior but were then found to be lacking compared to machine learning models. Among machine learning classifiers, logistic regression outperformed Naive Bayes, probably because it assigns weights to features instead of assuming independence. Support Vector Classification showed an increase in performance when additional linguistic features were added, such as syllable count and synonym count, which account for the better feature selection and the superiority of the models that are non-linearly related for classification purposes. Future work might be done on deep learning approaches or ensemble methods to increase classification accuracy and robustness.