COM6513 Natural Language Processing (2017/18)
Class Project: Complex Word Identification

Registration Number : 170122078

1 Introduction

The task was to improve a Complex Identification System, in order to predict word that can be considered complex or difficult by non-native speakers, based on annotations collected from native and non-native speakers (Stajner et al., 2018).

Automatically categorising words into complex and non-complex, is considered important for several NLP systems, which have been developed to simplify texts to non-native language learners with low literacy levels and disability problems (Zampieri et al., 2017).

2 Initial System

To begin with a baseline system was provided, which performed complex word identification for English and Spanish. The datasets provided together with the baseline system, where training and development sets for both of the aforementioned languages. The training data for both languages and two features were extracted; the length of the target phrase/word from the sentence divided by the average word length in each language, and the number of target word/words. These features were then used to train a Logistic Regression model. Predictions were made using the same features extracted from the development set for both languages. The predictions were labels, with 0 being for a predicted non-complex word an 1 for a complex word. The accuracy of the system's results were evaluated against the actual target word labels, using the F1 metric. The F1 scores for the system supplied were 0.69 for English and 0.72 for Spanish.

3 Improved System

The following section aims to go through the stages of improving the model, from the initial considerations, to the testing stage to the system in its finalised form.

3.1 Model Selection

As mentioned in Section 2, the baseline system supplied utilised Logistic Regression from scikit learn. From the nature of the task it was concluded that supervised models were required for binary classification. Therefore several models meeting these criteria were considered and few were selected for testing based on their strengths and weaknesses. All the models tested were drawn from the scikit learn python package. The models that were initially considered for the task, suitable for binary classification are:

- Logistic Regression
- Decision Tree Classifiers
- Ensemble Methods (i.e. Extremely Randomised Trees, Random Forest)
- Perceptron and the Multilinear Perceptron

Logistic Regression was the model implemented in the baseline system. Estimations in logistic regression choose parameters that maximize the likelihood of observing the sample values. Some of the limitations of this approach is that it requires features to be independent of each other, it performs better with larger datasets and the model can over-fit easily with larger number of input features (Hyeoyn-Ae, 2013).

Decision Tree Classifiers have the advantage of requiring very little data preparation and can handle both categorical and numerical data. Their disadvantage though lies with low sample sizes and the fact that they are prone to overfitting and are sensitive with features with large variations in numbers (Gupta, 2017). Scikit Learn uses an improved version of the CART algorithm (Learn,

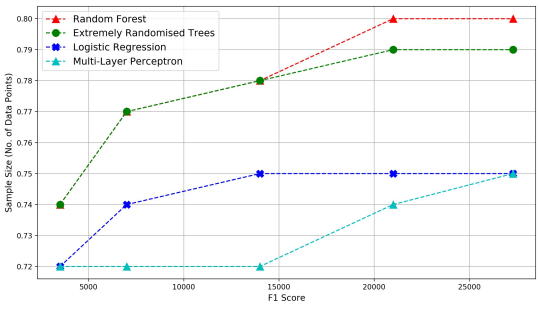
2017), and this algorithm has the limitations that it splits by one variable only, is non-parametric and it may have unstable decision trees (Gupta, 2017).

The ensemble methods of Random Forest and the Extremely Randomised Trees, address some of the issues that appear in the Decision Tree Classifiers. Random Forest in particular is like a collection of Decision Trees (Wang, 2017), and addresses overfitting by creating random subsets of the features and building smaller trees with the subset, and then combining the subtrees. It is therefore slower than a Decision Tree, but for our task where the datasets are small speed should not be a problem. Extremely Randomised Trees are a variant of Random Forest, at each step the entire sample is used and boundaries are picked at random rather than selecting the best one. Assuming feature selection for both is optimal the performance should be similar although the Extremely Randomised Trees should be faster (Geurts et al., 2006).

The Perceptron is a linear model for binary classification, and with a logistic sigmoid activation function will essentially yield the same results as the logistic regression model. Thus its limitations are similar, with the biggest one being that it cannot handle non-linear separable datasets. This problem is addressed by the Multilayer Perceptron.

In order to test which model would be sufficiently trained with the size of the training dataset, a learning process graph was produced with several of the models considered, in Figure 3.1.

Figure 1: Learning progress for different models. F1 Scores for the models tested with different sample sizes from the training data.



Considering the information supplied in this section and also the fact that the Multilayer Perceptron model appears to require more data to stabilise the learning rate; the models brought forward for testing were Logistic Regression, Random Forest Classifiers and Extremely Randomised Trees Classifiers.

3.2 Feature Selection

Prior to proceeding to this feature selection, some research was performed on what makes a word complex. After reading through previous research and making some observations from studying complex words, the following features were selected for testing.

- **Length of word** - This was already included in the base system supplied and after research it was shown that it was a potentially promising feature. From the observations of (Zampieri et al., 2017) the authors observed that complex words were on average longer than non-complex words. After testing this theory on the training data supplied, the same pattern was observed in both languages, with the complex words being on average much longer than non-complex ones.

- **Number of syllables** - The theory behind this feature is that complex words might potentially contain more syllables. To test this theory initially the words were split into syllables, by obtaining the order of consonants and vowels in the word. For example in the word *python*, we have two syllables *py* and *thon*. The orders of consonants and vowels are CV and CCVC. Through this method a syllable extractor was created, which returns the syllable number in a word with an accuracy of 15/20 for English 16/20 for Spanish. After extracting syllables on the training set it was observed that complex words had on average more syllables, averaging at 3 for English and 3.5 for Spanish, whilst non-complex words averaged at 2 for English and 2.6 for Spanish.

- **Probability of word** - The logic behind this feature is that a complex word will be less frequent, or ”rarer”. Therefore having observed fewer times or none at all certain words, can increase the probability that the word is complex in contrast to common words someone would regularly use. Since the datasets provided are small, two corpora were also used to obtain more depth for this

model. The nltk brown corpus for English and the nltk cess_esp for Spanish.After testing this feature on our dataset it was observed that for both languages the average probability for a complex word was significantly lower than that of a non-complex word.

- **N-Grams** - This was a feature that aimed at extracting information about the words surrounding the target word, with the triagrams being : previous word - current word - next word. This aims at providing more context for the models.
- **PoS tags** - Part of Speech tags could have the possibility of affecting if a word is complex. For example if the word is a verb or a noun might have a factor in determining if a word is complex or not.
- **Number of vowels/consonants** - The theory behind this feature is that a word with more vowels or many consonants can have a higher probability of been classed as complex.
- **Continuous consonants** - Similarly to the feature above, if a word that contains a large number of consonants continually then it might increase the probability of classifying it as complex.
- **Current word representation** - In order to pass on the current word as a feature into the model, instead of one hot encoding the word, word were converted as a float numbers beginning with *0.* (In order to avoid dealing with large numbers) and then what followed was the characters representation as integers, depending on their position in the alphabet. For example *cat* would be encoded as 0.3119.

3.3 Packages Used

The packages used in this system where Scikit Learn and NLTK. Spacy was also used during the feature extration process for the PoS taggs for both in English and Spanish, but since it did not improve the accuracy in any of the models and languages it was abandoned, as it also increased significantly training and testing times.

3.4 Selection and Optimisation with the Development Dataset

Combinations of all of the features mentioned in Section 3.2 were tested together with the models in Section 3.1. Table 1 aims to show the scores for the best performing feature combination for all models and in both languages, when tested with the development dataset.

Best scoring models, as seen in Table 1 were the Random Forest Classifier for the English language and the Extremely Randomised Trees for the Spanish language, each with the features identified below. These two models together with the corresponding feature combinations, were then selected to be tested with the test dataset which was supplied at a later date. The final system can be seen by which model, and features it comprises for each language from Table 1, where they are in bold.

As it is apparent from Table 1 the most informative features were the *Length of word*, *Number of syllables* and the *Probability of the Word*. These features improved almost all of the models when used combined together. It was expected that the PoS tags would improve the accuracy at least in some of the models but that was not the case. This result indicates that there is a possibility that the complex words belong to a range of PoS tags, and not one PoS tag can classify more complex words than the others.

Additionally the triagram feature was expected to provide more context for the target words, but failed to do so. This can indicate that possibly the words around the target word do not have any effect on whether the word is complex or not.

4 Evaluating the Improved System on the Test Data

Table 2: F1 metric scores retrieved with the baseline system and the improved system on the Test Dataset

System	Language	F1 Score
Baseline System	English	0.68
	Spanish	0.70
Improved System	English	0.81
	Spanish	0.74

Table 2 illustrates the achieved results of the improved system and the original system supplied. Contrasting the two it is visible that there was a significant increase in accuracy for the English language, and a generous but not that large increase for the Spanish language. The increase in accuracy for both languages was as expected,

Table 1: Description of models tested with the best feature combination for each and their accuracy in both languages. The improved system implemented the models seen in bold, together with the bolded parameters and features for the languages specified.

Model	Parameters	Language	Features	F1 Score
Logistic Regression	class weight = balanced penalty = l1 regularisation	English	Length of word Probability of word Current word	0.75
	default	Spanish	Length of word Number of syllables Probability of word	0.73
Random Forest	**random_state = 124 others = default**	**English**	**Length of word Number of syllables Probability of word Current word**	**0.80**
	random_state = 124 others = default	Spanish	Length of word Number of syllables Probability of word Current word Vowel sequence	0.73
Extremely Randomised Trees	criterion = "entropy" random_state = 124 others = default	English	Length of word Continuous consonant Number of syllables Probability of word Current word	0.79
	criterion = "entropy" random_state = 124245 others = default	**Spanish**	**Length of word Number of syllables Probability of word**	**0.74**

as the models had more context for the complex words than the original baseline system and therefore yielded better results.

5 Future Work

One potential feature that could improve this task, at least for the English language, would be analysing the morphemes of words. According to (Nordquist, 2018) a complex word in English, can be considered one which is made up from two or more morphemes. Additionally a more accurate tool for syllables would be expected to yield better results. Unfortunately I was not successful in loading the polyglot package, which splits a word to its syllables.

6 Conclusion

By using Random Forest Classifier and the Extremely Randomised Trees models together with a combination of features, the F1 score when using the Test dataset improved from 0.69 to 0.81 for the English language and from 0.70 to 0.74 for the Spanish language. The most informative features as it resulted from the testing stage in the development phase where the *Length of the Word*, *Word Probability* and the *Number of Syllables*. The system accuracy could be improved by using a morphological tool to split the target word into morphemes and supplying it in the model as features.

Github Link to Code

https://github.com/GChrysostomou/NLP-Class-Project.git

References

PIerre Geurts, Damien Ernst, and Louis Whenkel. 2006. Extremely randomized trees. In *Mach Learn*. Springer Science. https://orbi.uliege.be/bitstream/2268/9357/1/geurts-mlj-advance.pdf0.

Bhumika Gupta. 2017. Analysis of various decision tree algorithms for classification in data mining .

Park Hyeoyn-Ae. 2013. Introduction to logistic regression: From basic concepts to interpretation with par-

ticular attention to nursing domain. *J Korean Acad Nurs* 43(2):154–164.

Scikit Learn. 2017. Decision trees. http://scikit-learn.org/stable/modules/tree.html.

Richard Nordquist. 2018. Complex words in english. https://www.thoughtco.com/what-is-complex-word-1689889.

Sanja Stajner, Chris Biemann, Shervin Malmasi, Gustavo Paetzold, Lucia Specia, Anas Tack, Seid Muhie Yimam, and Marcos Zampieri. 2018. Complex word identification (cwi) shared task 2018. https://sites.google.com/view/cwisharedtask2018/.

Clem Wang. 2017. What is the difference between random forest and decision trees. https://www.quora.com/What-is-the-difference-between-random-forest-and-decision-trees.

Marcos Zampieri, Shervin Malmasi, Gustavo Paetzold, and Lucia Specia. 2017. Complex word identification: Challenges in data annotation and system performance. In *Proceedings of the 4th Workshop on Natural Language Processing Techniques for Educational Applications*. Association for Computational Linguistics, pages 59–61. http://www.aclweb.org/anthology/W17-5910.