Complex Word Identification

Aneeq ur Rehman
Univeristy of Sheffield
aurrehman1@sheffield.ac.uk

Abstract

In this paper, we present the results of a system build for complex word identification for words in Spanish and English. Depending on the features and the model used, the systems implemented predict the word complexity. We initially present a baseline system that uses some fundamental features to predict the word complexity and compare it with an improved system with additional features and models developed. Our results show that the improved system works better compared to the baseline. Finally reverse engineering is done to identify word types on which the models fail to generalize. Furthermore, suggestions are given for improving predictions on such words.

1 Introduction

Complex word identification is a part of lexical and text simplification. The task is to identify complex words in texts using computational methods (Zampieri et al., 2017) and to replace them with the simplest substitute to enhance the readability of text while preserving its meaning. Complex word identification targets a broad range of audience like people with aphasia, children, nonnative speaker, dyslexia (Yimam et al., 2017) or people with various cognitive and reading impairments or low literacy levels. It has been shown that people with dyslexia read and understand better when short and frequent sentences are used. The motivation for such a system is thus targeted towards these people to simplify the text while preserving its meaning. Complex word identification is thus a binary classification problem, where a word is assigned a zero if it simple and one if it is complex.(Palakurthi and Mamidi, 2016)

$$c(w) = \left\{ \begin{array}{ll} 1, & \textit{if}\, w \in C \\ 0, & \textit{if}\, w \in S \end{array} \right\}$$

Here w is the target word, $c(w)$ is a class of targeted word, C is a set of complex words and S is a set of simple words. We present an initial baseline system which uses four features to evaluate a word complexity and test it with different models and then develop an improved system which adds more features and combines predictions from various models. We compare our predictions with the true labels from the gold data set which is manually annoted. Our improved system shows great improvement and the finally some of the predictions where the improved system fails to predict are analyzed and suggestions are offered for future work.

2 System Design and Implementation

This section details about the design of the baseline and the improved system.

2.1 Baseline

The initial baseline system consisted of the features which were as follows:

- **Relative word length and word length:** The Relative word length was calculated by dividing the numbers of characters a word contains divided by average length of words for English and Spanish respectively. For English the average word length is 5.3 and for Spanish the average word length is 6.2. Similarly the word length was also taken as a feature. Longer words tend to be more complex and vice versa.

- **Syllable Count:** The more the number of syllables a word has, the higher the complexity of that word or noun phrase. Two characters of the word were taken at time and if the first character is not in vowels and if the second one is a vowel, the syllable count was incremented by one. If the word ended with the character `e`, the total count was subtracted by one due to linguistic reasons and the count was incremented by one if the word ended with the `le`. Finally if the word did not match any of the above conditions, the count was incremented by one, i.e. a one syllable word.

- **Total Number of Senses:** The word with more senses is more ambiguous compared to a word with fewer senses. Number of senses of a word was obtained using wordnet from nltk package in python. (Palakurthi and Mamidi, 2016)

The system served as a baseline for the initial testing and different models were used after hyper parameter tuning through Grid Search CV and Randomized Search CV.

2.2 Improved System

The improved system consisted of additional features and effort was made to enrich the feature vector with high quality features. In addition to adding additional features after hyper parameter tuning, new models were added and finally a hybrid model was developed using some of the best models and this model greatly improved results.

2.2.1 Feature Addition

Additional features added are as follows:

- **Unigram Probability:** A huge training corpus of 0.8 billion words was taken available from the training monolingual corpus available on github containing training examples from Wikipedia, news articles, journals and the relative frequency of each of the words appearing in this corpus was calculated. This (word, probability) pair, was then stored in dictionaries which were pickled for English and Spanish respectively. Target words from the training set were also included in these two dictionaries.

- **Stop words and noun Phrase checks :** The target word was also checked for being a noun phrase and stop word. Stop words were taken from nltk stop words for English and the stop words for Spanish were taken from the ranks natural language website which has different lists of texts used for various keyword analyzer tools.

- **Vowel and Consonant Count :** The count of vowels and consonants in a target word were also checked by iterating over the word to check for specific counts of these features, each given as lists. If a character was present in these lists, it was counted as a vowel or a consonant and the total count of these features was added up.

- **Syonym and Hypernym count:** In the previous case the overall senses of the word were counted by using the synsets module at word net. Now for each language and for each target word, first a check was made on whether it was a phrase and for each of the words in the noun phrase, the count of the number of synonyms and hypernyms for each word were added up and normalized by dividing by the number of words present in the phrase.

- **BIO and POS Encoding**: Although this feature was tested, it was excluded from testing for most part due to computational requirement needed for one hot vectors and that there was improvement using initial vectors. The POS tags and the position of each of target word in the sentence (`B,O,I`) was found from the spacy module and each of these tags were one hot encoded and the vectors were passed as features to the model. Spacy module also has additional features with each POS tags it uses.

2.2.2 Model Selection and Hybrid Model Development

The baseline and the improved system was initially tested on **Decision Trees (DT)**, **Random Forest (RF)**, **K- nearest neighbours (K- nn)**, **Adaboost Regessor** , **Extra Trees Regressor (ET)**, **Logistic Regression (LR)** and **Support vector machine (SVM)** algorithms. Random Forest, decision trees, Extra Trees, Adaboost Regressors are available at scikit learn and are used because the all of these algorithms require little data preparation (i.e. normalization, blank values removal) and the cost of using the tree is logarithmic to the number of data points (Pedregosa et al., 2011). So this is quite useful to use. The **Hybrid Model (HM)** used here is a combination of the models that worked best combining four models (Decision Trees, Random Forest, K-neighbors, Logistic Regression). It initially relies on predictions from the K- nearest neighbors and then uses the other best models in order of performance determined earlier and predicts on the basis of these models. K-neighbors is a non-linear model used to determine if a data point will be one part of the class or the other and this model is really useful in our scenario. It looks at the neighboring points to classify the label for a data point. Performance for K- nearest neighbors also improves in general when the training data is large as its predictions are based on its neighboring points. Since the training data here for English is about 27,000 points and 13,000 points for Spanish, this model is a suitable choice to use here. For SVR, the performance did not improve significantly and the results are attached on github but testing could not be done on the test data due to run time of SVR.

3 Testing and Experimentation

Testing was initially done on the development data set available for both the baseline and the improved system.

3.1 Development Data Testing

Table 1 below the models training results on the development data. It can be seen that firstly the performance improves significantly for the improved system with additional features. The K-nearest neighbors gives a F1 score of **0.82** for English and Random Forest gives a F1 score of **0.77** for Spanish for the improved system. This is a significant improvement in general and based on these two results, the motivation to use an ensemble hybrid model utilizing predictions from a combination of models stems. (Malmasi et al., 2016)

Model	Baseline	Improved
DT	S:0.72, E:0.72	S:0.76, E:0.80
K-nn	S:0.70, E:0.70	S:0.73, E:**0.82**
RF	S:0.73, E:0.72	S:**0.77**, E:0.80
AB	S:0.71, E:0.72	S:0.74, E:0.74
ET	S:0.73, E:0.72	S:0.74, E:0.79
LR	S:0.69, E:0.72	S:0.71, E:0.72

Table 1: Development Data: Baseline vs Improved System F1 Scores for English and Spanish

The predictions of the baseline and the improved system on the development data were analyzed. The features outlined in Table 2 below are with the baseline mentioned in section 2.1. The table below outlines some of the target words that were present in the predictions from the baseline system which were not there in the improved system. It can be seen that the improved system, predicts better on noun phrases, common words like `President`, `profundidad` and even complex words like `exponen` and `truce` which are as a result of incorporating unigram probabilities, noun phrase checks and additional features in the systems.

Word and True Label	Baseline	Improved
`Britain-based ,0`	✗	✓
`shelling of the city ,1`	✗	✓
`Vasteras ,0`	✗	✓
`revered ,1`	✗	✓
`President ,0`	✗	✓
`truce ,1`	✗	✓
`encontradas ,0`	✗	✓
`profundidad ,0`	✗	✓
`exponen ,1`	✗	✓
`Mariottin ,1`	✗	✓

Table 2: Word Predictions per System
✓= System Predicts correctly
✗= System Fails to Predict correctly

3.2 Test Data

The Test data set was then evaluated using the baseline and the improved systems and the results were as expected. The hybrid model was also tested on the test data set and it can be seen from Table 3 below that a significant improvement in results was obtained , achieving a F1 Score of **0.91** for English and **0.87** for Spanish when models are combined together. In addition to this it can be seen that the K nearest neighbors and Random Forest work really well and serve as first point of evaluation for the hybrid model.

Model	Baseline	Improved
DT	S:0.73, E:0.71	S:0.77, E:0.79
K-nn	S:0.74, E:0.69	S:**0.77**, E:**0.80**
RF	S:0.73, E:0.71	S:**0.77**, E:**0.80**
AB	S:0.71, E:0.71	S:0.74, E:0.77
ET	S:0.73, E:0.71	S:0.74, E:0.76
LR	S:0.70, E:0.71	S:0.70, E:0.72
Hybrid	S: 0.79, E:0.78	S:**0.87**, E:**0.91**

Table 3: Font guide.

3.3 Peformance Analysis

The learning curves on the test data set for different models is shown in Figure 1 and Figure 2 below for English and Spanish respectively for both the baseline and the improved system. In addition to this, the learning curve for the hybrid model was plotted for the improved system. From Figure 1 and Figure 2, it can be seen that some models are better when the others when less training data is available for example in figure 1 for Spanish, the learning curve for baseline at n = 3000, logistic regression achieves a F1 score of **0.72** in parallel with Extra Trees and Ada boost Algorithm and K nearest neighbors achieves a F1 Score of **0.69**. It is also seen in the improved system that when n = 3000 Logistic Regression achieves a F1 Score of **0.75** and K-neighbors achieves a F1 Score of **0.72**. At n= 13750, when full training data, both models achieve a F1 of **0.72**. Similarly in Figure 2 for English, it can be seen that for lower amount of training data provided, the random forest algorithm **(F1:0.71)** performs better than K Neighbors **(F1:0.69)**. For Extra trees performance on the improved system improves slightly but the optimal performance is again with 9000 samples achieving a F1 of **0.75**.

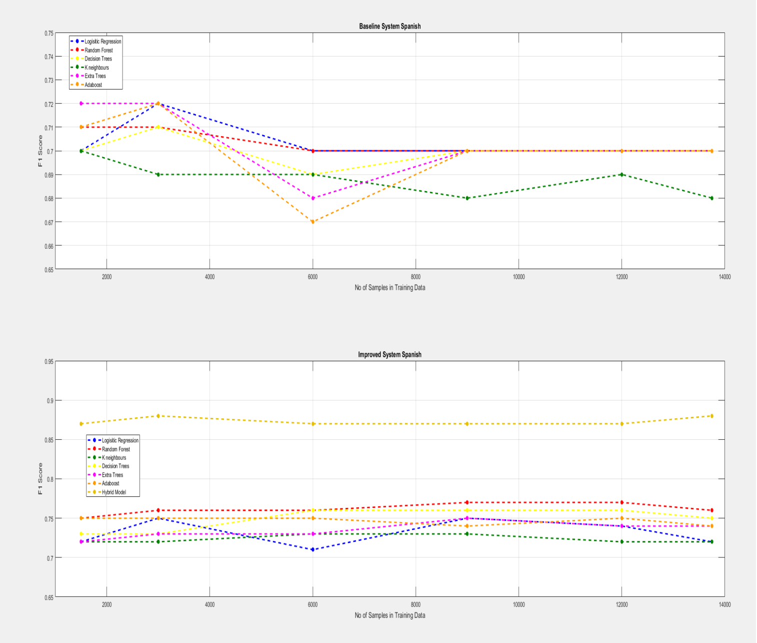

Figure 1: Learning Curve Spanish: Baseline vs Imrproved

Similarly Figure 2 below shows the learning curve for English and it can be seen that for all the models performance generally improves except for adaboost where performance decreases slightly as more training data is available. The K-nearest algorithm achieves a F1 Score of **0.82** whereas the hybrid model achieves a F1 score of **0.91**. On the English data set, it is seen that performance improves as the number of training data samples increases.

3.4 Reverse Engineering

Finally Reverse Engineering was carried out and the predictions of the improved hybrid systems were analyzed. Table 4 below shows some of the predictions where the model failed to generalize. It can be seen that the words where the model fails to predict are very long

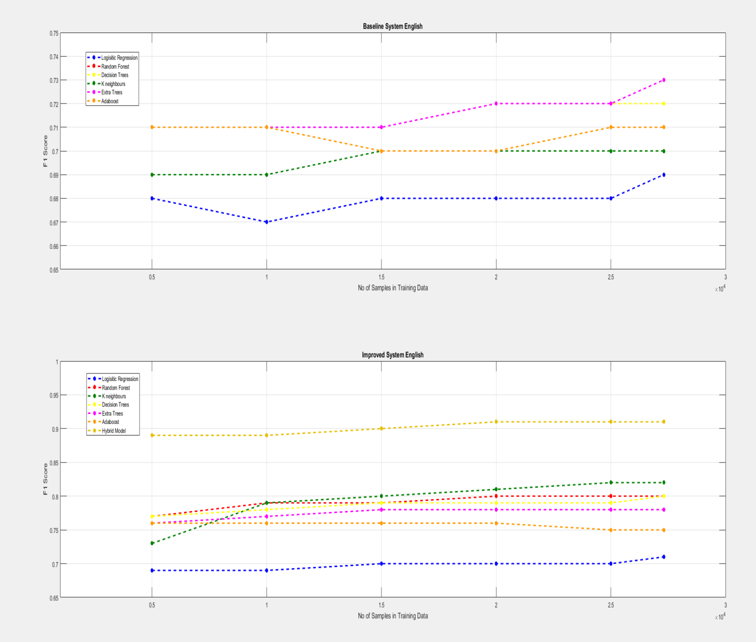

Figure 2: Learning Curve English: Baseline vs Improved

phrases in English and some words which are very context specific, for example Sunni, Shiitte are two sects of muslims and the training on such words can be improved by having texts from various religions, cultures, ethnic origins. For Spanish some words such as regular, tribunal and musical although are the same words as in English, the training can then be provided on models for sets on words which are universal to both English and Spanish. Also words such as Guerra mean war in English, so having a training sets used for machine translation could be useful.

Language Set	Word	Label
English	questioned on suspicion of assisting	1
English	Sunni	0
English	Afghanistan	0
English	Shiite	1
English	marine surveillance ships	0
Spanish	tribunal	1
Spanish	musical	1
Spanish	Guerra	1
Spanish	Longitud	1
Spanish	regular	0
Spanish	Aviacion	1

Table 4: Examples where the Improved System Fails to generalize

3.5 Feature Selection

Finally feature selection was done using different models, results for k-nearest neighbors are shown below in table 5. As seen in figure 3 below the unigram probability has a significant impact on the model performance and set A is the best set of features with substantial improvement in model performance for both Spanish and English. Although set E offers improvement but the change is very small. Set B offers the least improvement but serves as a building block to add more features in the feature set.

Category	Features
A	Syllables,Relative & word length , Senses Unigram Probability
B	Syllables,Relative & word length, Senses
C	Syllables,Relative & word length , Senses Unigram Probability, Consonant & Vowel Sum
D	Syllables,Relative & word length , Senses, Unigram Probability, Consonant & Vowel Sum Noun Phrase, Stop words
E	Syllables,Relative & word length , Senses, Stop words Unigram Probability, Consonant & Vowel Sum ,Noun Phrase, Synonym and Hypernym count

Table 5: Feature Selection

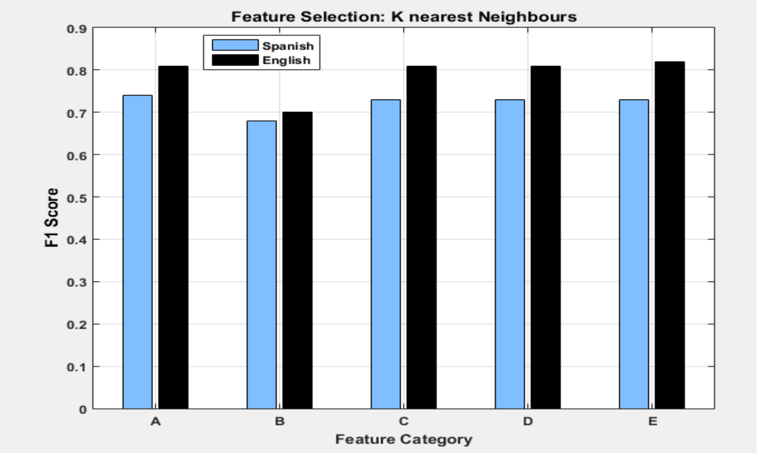

Figure 3: Feature Selection: K-nn

4 Conlusion and Future Work

In this paper, we presented the results of a complex word identification system. We developed an initial baseline with a few rudimentary features and then we tested the system on various models and combinations of models and added additional features. Although our hybrid model results in a promising score, it still has a limitation of its dependency on the true label to make further predictions. Future work can entail developing models where ensemble methods are utilized to combine predictions from various models. Having some corpuses used in machine translation from English to Spanish or vice versa can aid model performance. Furthermore since this is a binary classification problem, implementation of neural networks could result in better models along with additional features such as bigram and trigram probabilities.

References

Shervin Malmasi, Mark Dras, and Marcos Zampieri. 2016. Ltg at semeval-2016 task 11: Complex word identification with classifier ensembles. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 996–1000.

Ashish Palakurthi and Radhika Mamidi. 2016. Iiit at semeval-2016 task 11: Complex word identification using nearest centroid classification. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 1017–1021.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Seid Muhie Yimam, Sanja Štajner, Martin Riedl, and Chris Biemann. 2017. Cwig3g2-complex word identification task across three text genres and two user groups. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, volume 2, pages 401–407.

Marcos Zampieri, Shervin Malmasi, Gustavo Paetzold, and Lucia Specia. 2017. Complex word identification: Challenges in data annotation and system performance. *arXiv preprint arXiv:1710.04989*.