IIIT at SemEval-2016 Task 11: Complex Word Identification using Nearest Centroid Classification

Ashish Palakurthi and **Radhika Mamidi**
Kohli Center on Intelligent Systems
IIIT-Hyderabad, India.
ashish.palakurthi@research.iiit.ac.in
radhika.mamidi@iiit.ac.in

Abstract

This paper describes the system that was submitted to SemEval2016 Task 11: *Complex Word Identification.* It presents a preliminary investigation into exploring word difficulty for non-native English speakers. We developed two systems using Nearest Centroid Classification technique to distinguish complex words from simple words. Optimized over G-score, the presented solution obtained a G-score of 0.67, while the winner achieved a G-score of 0.77 and the average G-score of all the submitted systems in the task was 0.56.

1 Introduction

Lexical Simplification aims at improving the readability and comprehensibility of text by transforming complex text into simple text. Lexical Simplification (Specia et al., 2012; Belder et al., 2010; Horn et al., 2014) is the process of replacing a word in a given context with its simplest substitute to enhance the readability of the text. The process should make sure that while replacing words with other variants, the meaning of the text is preserved. Lexical Simplification (Siddharthan, 2014) is useful to a wide variety of target audience like people with aphasia, children and also non-native speakers. Complex Word Identification (Shardlow 2013; Paetzold 2015) is considered to be the first step in the pipeline of Lexical Simplification. The overall performance of a Lexical Simplification system is thus crucially dependent upon Complex Word Identification. The problem of Complex Word Identification is relatively new in the field of Natural Language Processing. However, a few approaches have been previously proposed for this task. The simplicity score (Bott et al., 2012) of a word is computed by integrating both, frequency and length of a word. They consider a threshold value and simplify words only if the word's frequency is lower than the fixed threshold. Matthew Shardlow (2013) explores the frequency thresholding to differentiate between simple and complex words by experimenting with each threshold value on a particular corpus. However, this approach is not practically convincing. The same author also frames the problem as a machine learning classification problem by designing a few features. We approach the problem at hand on similar lines.

The Complex Word Identification (CWI) task is framed as a binary classification problem. Given a word in a sentence, the task is to predict whether the word is simple or complex. A word is tagged with 0 if it is simple and 1 if the word is found to be complex. [1]

$$c(w) = \begin{cases} 1 & if\ w \in C \\ 0 & if\ w \in S \end{cases}$$

C is the set of complex words, *S* is the set of simple words and *c(w)* represents the class of the word. Here is an example of a sentence taken from the training dataset provided by the organizers.

- ◇ A **frenulum** is a small fold of tissue that secures or **restricts** the **motion** of a mobile organ in the body.

[1] Complex: In the context of this shared task, complex words are the words which are difficult to understand for a non-native English speaker.

Proceedings of SemEval-2016, pages 1017–1021,
San Diego, California, June 16-17, 2016. ©2016 Association for Computational Linguistics

In the above example, the task requires a system to spae the words in bold as complex. The remainder of this paper is structured as follows. In Section 2, we describe our systems and Section 3 discusses experiments and results. We conclude in Section 4.

2 System Description

We use the Nearest Centroid Classification technique (Manning et al., 2008) for the classification of words using Manhattan and Standardised Euclidean distance metrics. This classification method is widely used in Information Retrieval tasks. In this method, each class is represented by the mean of all the training samples belonging to that class in the training data. A new observation is assigned a class label, whose mean is closest to the observation. Given below are the labeled training samples:

$$(\overrightarrow{x_1}, y_1), ..., (\overrightarrow{x_n}, y_n), y_i \in Y$$

$$\overrightarrow{\mu_c} = \frac{1}{|N_c|} \sum_{i \in N_c} \overrightarrow{x_i}$$

$$\hat{y} = arg\ min_{c \in Y} ||\overrightarrow{\mu_c} - \overrightarrow{x_i}||$$

where Y is the set of classes,
$|N_c|$ is the number of samples in class $c \in$ Y,
μ_c is the centroid of all samples belonging to class c,
$\hat{y}$ is the class assigned to the new observation.

We submitted two systems for this shared task. Our first system uses the Manhattan distance metric for the Nearest Centroid Classification. The Manhattan distance function computes the distance to be travelled to get from one data point to another point in a grid-like path. The Manhattan distance between two points is the sum of the differences of their corresponding components. Manhattan distance between two points *A(x,y)* and *B(x,y)* is defined as:

$$d(A, B) \equiv |A_x - B_x| + |A_y - B_y|$$

The distance metric used by System 2 in the training algorithm of the Nearest Centroid Classification was Standardised Euclidean, which is a slight variant of Euclidean distance. Euclidean distance between two points is defined as the sum of the squares of the differences between the corresponding components of the points. The Standardised Euclidean distance between two points *A(x,y)* and *B(x,y)* is defined as:

$$d(A, B) \equiv \frac{\sqrt{(A_x - B_x)^2 + (A_y - B_y)^2}}{V}$$

where, V is the 1-D array of component variances and the numerator is the Euclidean distance between two points *A(x,y)* and *B(x,y)*.

Both, System 1 and System 2 rely on 5 features to classify the target word as simple or complex. An almost similar set of features was previously employed by Matthew Shardlow (2013) to identify complex words using Support Vector Machines. The features we considered for the classification of words are:

- **Unigram Word Probability:** We used the Google Books Ngram Viewer[2] to obtain the word probability of the target word in the sentence. This information was considered only for the year 2000.
- **Length:** We considered length of the target word as a feature because longer words are likely to be complex.
- **Number of Senses:** A word with higher number of senses is relatively more ambiguous in comparison to a word with fewer senses. Number of senses of a word was obtained using WordNet[3] from NLTK[4] package.
- **Syllable Count:** A word with higher number of syllables[5] is likely to be more difficult to be read.
- **CD Count:** The number of films in which the target word had appeared was obtained from the SUBTLEX[6] corpus.

3 Experiments and Discussions

3.1 Data

We used the joint dataset provided by the task organizers for training. The training and test data consisted of 2,237 and 88,221 instances, respectively.

[2]https://books.google.com/ngrams
[3]http://www.nltk.org/howto/wordnet.html
[4]http://www.nltk.org/
[5]http://www.syllablecount.com/syllables.
[6]http://zipf.ugent.be/open-lexicons/interfaces/subtlex-uk/

Also, the training and test data comprised 200 and 8,929 unique sentences respectively.

3.2 Discussions

We discuss our experiments with respect to the training dataset. Experiments were run with a variety of machine learning algorithms using the Scikit-learn toolkit (Pedregosa et al., 2011). However, the Nearest Centroid Classification algorithm was found to outperform other algorithms like Random Forest and Support Vector Machines significantly over 5-fold cross-validation of the training dataset. We tried numerous combinations of features and finalized on the feature set described in the previous section.

There are a few features that we tried during our experimentation but did not include in the final system, as the results turned comparatively lower with their inclusion during cross-validation. We believe that the features are still worth discussing. They are:

1. *Average Word Length of Synonyms:* If the length of the given word is greater than the average length of all its synonyms, then the word is tagged as 1, else 0. This gives us a relative estimate on whether people would prefer to use a word more regularly in comparison to its synonyms. A similar feature could be tried using frequency or syllables in addition to length of the word. But for our experiments, we only used the length feature.

2. *Rank of a Sense:* We use the Lesk algorithm from Wordnet to find the sense of a word in the given sentence. We find the corresponding sense and the rank of the sense based on frequency using Wordnet. A lower rank of a sense suggests higher frequency of usage of that sense (Christiane Fellbaum, 1998). A higher frequency of a sense indicates that the word is likely to be inferred as simple. The rank of a sense is divided by the total number of senses of the word to get a normalized measure of the feature.

3. *Number of Synonyms:* Number of synonyms of a given word was considered. This information was obtained using Wordnet.

4. *Collocation Score:* Collocations are a set of words which occur together frequently. We find collocations for the target word in each sentence. We calculate a collocation score (C-score) defined as

$$C\text{-}score = \frac{C}{N}$$

where, C is the number of collocations matched in the sentence and N is the total number of tokens in the sentence. A higher C-score value indicates that the word usage is less ambiguous. Collocations[7] include Noun, Verb, Adjective, Adverb and Conjunction collocations.

We believe that the first two features are very critical in differentiating simple words from complex ones as words should be inspected in a relative frame with respect to their synonyms and senses. *A complex word may have higher number of senses but it does not necessarily imply that a word with higher number of senses is always complex.* It is fairly possible that a word (with many senses) in the given sentence may turn out to be the most frequently used sense and hence appear to be simple.

Thresholding based on frequency of a word is a common and effective way to identify complex words. The PLUJAGH team submitted a frequency threshold-based system and it stood first in this shared task when evaluated on F-score. In addition to simple thresholding, it is essential to consider relative frequencies of a word with respect to their synonyms. For example, consider the words *eldest* and *oldest*. From a manual experiment, we found that *eldest* is a complex word and *oldest* is a simple word. The frequency of *oldest* is higher than the frequency of *eldest*. However, the word *unused* (from the training dataset provided by organizers) which is less frequent (google ngrams) than *eldest* is a simple word. We found 110 words in the training dataset whose frequency is lower than *eldest* and are still simple. We claim that this may be due to the relatively higher frequency/usage of *oldest* with respect to the frequency/usage of *eldest*.

In conclusion, it becomes important to speculate words in a relative frame (of frequency/senses/length) with respect to their synonyms, to improve complex word identification.

[7] Collocations were collected from http://prowritingaid.com/free-online-collocations-dictionary.aspx

The effectiveness of this hypothesis could have been empirically better visible with the availability of a larger training dataset.

3.3 Evaluation Metric

The official evaluation metric of the task is G-score. It is the harmonic mean of Accuracy (A) and Recall (R).

$$G\text{-}score = \frac{2 * A * R}{A + R}$$

3.4 Results

Table 1 shows the average G-score obtained using different classifiers for 5-fold cross-validation on the training data. For experiments using the Nearest Centroid Classification, we explored 24 different distance metrics, of which Manhattan and Standardised Euclidean metrics performed the best. Based on G-score, our systems were ranked 15th and 23rd in the task. Our first system was able to beat all the baseline systems including the threshold-based and the lexicon-based systems. Table 2 shows the performance of System 1 and System 2 on the test data.

System	G
System 1	0.65
System 2	0.63
Decision Tree	0.54
Naive Bayes	0.61
AdaBoost	0.58
Gradient Boost	0.54
SVM	0.44
Random Forest	0.53

Table 1: Cross-validation results on Training Data

System	A	R	G
System 1	0.546	0.879	0.674
System 2	0.465	0.860	0.603
Decision Tree	0.734	0.613	0.668
Naive Bayes	0.404	0.924	0.562
AdaBoost	0.767	0.777	**0.772**
Gradient Boost	0.826	0.705	0.761
SVM	0.837	0.546	0.661
Random Forest	0.794	0.641	0.709

Table 2: Results on Test Data

The reason for choosing the Nearest Centroid Classifier was its consistent and comprehensive dominance over other classifiers tuned over various parameters and different feature combinations during the 5-fold cross-validation on the joint dataset. The results shown in Table 2 pertain only to the finalized feature set discussed in Section 3. Table 2 also shows the results obtained on the test data using different classifiers with the same set of features employed for System 1 and System 2. At this point of time, we are not sure about why few classifiers like AdaBoost and Random Forest performed way better than System 1 and System 2 on the test data. A possible reason for the poor performance of the Nearest Centroid Classifier in comparison to other systems could be the imbalance between the training and the test data size. The test data being highly skewed could probably be another reason. AdaBoost[8] classifier (Table 2) was found to achieve a G-score of 0.772 on test data. This suggests that both, the proposed feature set and the approach presented are competent.

4 Conclusion and Future Work

In this paper, we described a promising approach for identifying complex words for non-native English speakers using Nearest Centroid Classification technique. Our approach is simple in terms of both, features and the learning algorithms.

We emphasized that words should be inspected in a relative frame with respect to their synonyms and senses. Testing this supposition in depth will be the subject of future work. We further look to improve the system by incorporating phonetic and semantic features. We also look to explore the problem at sentence level, as the complexity of the sentence can influence a person in comprehending a word's meaning in that sentence.

Acknowledgment

We would like to thank the SemEval-2016 Task 11 organizers, Henry Gustavo Paetzold and Lucia Specia for conducting this interesting shared task. In addition, we thank the anonymous reviewers for their

[8]For the AdaBoost classifier, the number of estimators used was 10.

constructive comments and insights that helped us in improving this manuscript. We are grateful to Himani Chaudhry and Himanshu Sharma.

References

Jan De Belder, Koen Deschacht, and Marie-Francine Moens. 2010. Lexical simplification. In *Proceedings of ITEC2010: 1st international conference on interdisciplinary research on technology, education and communication.*

Stefan Bott, Luz Rello, Biljana Drndarevic, Horacio Saggion. 2012. Can Spanish Be Simpler? LexSiS: Lexical Simplification for Spanish. In *Proceedings of COLING, Mumbai, India* .

Colby Horn, Cathryn Manduca and David Kauchak. 2014. Learning a Lexical Simplifier Using Wikipedia. In *Proceedings of the ACL 2014*, page pp. 458-463.

Christiane Fellbaum. 1998. Wordnet: An Electronic Lexical Database. Cambridge: MIT Press.

Fabian Pedregosa, Gael Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas , Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, Edouard Duchesnay. 2011. Scikit-learn: Machine Learning in Python In *Journal of Machine Learning Research, 12*, pages 2825–2830.

Michel Jean-Baptiste, Yuan Kui Shen, Aviva P. Aiden, Adrian Veres, Matthew K. Gray, Joseph P. Pickett, Dale Hoiberg, Dan Clancy, Peter Norvig, Jon Orwant, Steven Pinker, Martin A. Nowak, and Erez Lieberman Aiden. 2011. Quantitative analysis of culture using millions of digitized books, science, 331(6014), pp.176-182.

Christopher Manning, Prabhakar Raghavan, & Hinrich Schutze. 2008. Introduction to information retrieval. Vol. 1. No. 1. Cambridge: Cambridge university press.

Gustavo Henrique Paetzold. 2015. Reliable Lexical Simplification for Non-Native Speakers. In *Proceedings of the NAACL-HLT 2015 Student Research Workshop (SRW)*, page p9.

Gustavo Henrique Paetzold and Lucia Specia. 2016. SemEval 2016 Task 11: Complex Word Identification. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval 2016)*.

Matthew Shardlow. 2013. A Comparison of Techniques to Automatically Identify Complex Words. In *Proceedings of the ACL (Student Research Workshop)*, page pp103-109.

Advaith Siddharthan. 2014. A Survey of Research on Text Simplification. In *ITL-International Journal of Applied Linguistics*, page pp165(2):259-298.

Lucia Specia, Sujay Kumar Jauhar and Rada Mihalcea. 2012. SemEval-2012 Task 1:English Lexical Simplification. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics-Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation. Association for Computational Linguistics.*