Pomona at SemEval-2016 Task 11: Predicting Word Complexity Based on Corpus Frequency

David Kauchak
Computer Science Department
Pomona College
Claremont, CA
david.kauchak@pomona.edu

Abstract

We introduce a word frequency-based classifier for the SemEval 2016 complex word identification task (#11). Words with lower frequency are predicted as complex based on a threshold optimized for G-score. We examine three different corpora for calculating frequencies and find English Wikipedia to perform best (ranked 13th on the SemEval task), followed by the Google Web Corpus and lastly Simple English Wikipedia. Bagging is also shown to slightly improve the performance of the classifier. Overall, we find word frequency to be a strong predictor of complexity. On the SemEval "test" set, a frequency classifier that uses the optimal frequency threshold performs on-par with the best submitted system and a system trained using only 500 labeled examples split from the test set achieves results that are only slightly below the best submitted system.

1 Introduction

Text simplification aims to transform text into more accessible versions while retaining the original meaning. A frequent subproblem of the general simplification problem is complex word identification: identify words in a text that are difficult to understand for the reader. Complex word identification is critical in lexical simplification algorithms where the simplification process is done a word at a time; frequently, simplification is broken into two steps, first identifying the complex words that need simplifying and, second, determining substitutions for these words (Shardlow, 2014). Even for simplification systems that make sentence-level transformations (Siddharthan, 2014) complex word identification can be used as an additional feature function in the model and as a development tool to help measure progress. Additionally, in some domains such as health and medicine, accuracy is critical and semi-automated simplification tools are common (Leroy et al., 2012). In these domains, complex word identification is useful to help guide the simplification process by both identifying which words need to be simplified and filtering/ranking possible candidate substitutions (Leroy et al., 2013).

In this paper, we explore the use of word frequency as a predictor of the complexity of that word. Corpus studies have shown that simpler texts contain more frequent words than more complicated texts (Breland, 1996; Pitler and Nenkova, 2008; Leroy et al., 2012). User studies have also shown a correlation between word frequency and whether users know the definition of a word (Leroy and Kauchak, 2013). In semi-automated text simplification approaches, replacing less frequent words with higher frequency synonyms has been shown to produce text that people view as simpler and is easier to understand (Leroy et al., 2013).

2 Bagged Frequency Classifier

Given a sentence $S = s_1 s_2 ... s_m$ and a word in that sentence, s_j, the complex word identification task is to predict whether that word is complex (1) or not (0). Labeled examples are triples consisting of the sentence, the word and the label, i.e. $\langle S, s_j, \{0, 1\}\rangle$, and unlabeled examples consist only of the sentence and word $\langle S, s_j\rangle$. We view the problem as a super-

Proceedings of SemEval-2016, pages 1047–1051,
San Diego, California, June 16-17, 2016.

vised classification problem: given a collection of training examples, the goal is to learn a classifier to predict the label of unlabeled examples. See the SemEval Task 11 description for more details (Paetzold and Specia, 2016).

We utilize bagging (bootstrap resampling) to learn and combine multiple basic classifiers that predict by thresholding the frequency of occurrence of the word in question (s_j) in an external corpus. Classification is then done by majority vote of these classifiers. The subsections below provide more details.

2.1 Basic frequency classifier

The basic frequency classifier predicts the word complexity using only a single feature, the frequency of the word in question (s_j) in a corpus. Given an unlabeled example, the classifier predicts based on a learned threshold, α:

$$predict(\langle S, s_j \rangle) = \begin{cases} 1 & \text{if } freq(s_j) < \alpha \\ 0 & \text{otherwise} \end{cases}$$

with the assumption that words that occur less frequently in a corpus are more complex.

To train the basic classifier, we select α in an exhaustive fashion by considering all possible frequencies of seen in the training examples as candidate thresholds. Specifically, for each training example $\langle S, s_j, \{0, 1\} \rangle$, we consider using $\alpha = freq(s_j)$. We select the α that maximizes the G-score on the training set, where the G-score is defined as:

$$\frac{2 * accuracy * recall}{accuracy + recall}$$

We chose to optimize the G-score since it was the metric used for evaluation in the SemEval task (Paetzold and Specia, 2016), though other metrics could be used instead.

2.2 Word frequencies

Word frequencies can be pre-calculated from any corpus. For this paper we examined three corpora: articles from Simple English Wikipedia[1], articles from English Wikipedia[2] and the Google Web Corpus (Brants and Franz, 2006). For the Wikipedia articles we used the document aligned data set created by Kauchak (2013) consisting of approximately 60K articles on the same topics from each Wikipedia.

To collect word frequencies for the two Wikipedia variants, tokenization was first performed using the Stanford CoreNLP PTBTokenizer (Manning et al., 2014) and then frequencies were calculated. For the Google Web Corpus, we used the unigram counts. In all corpora, all capitalization variants were aggregated, e.g. occurrences of "natural" and "Natural" would both be counted towards the same word form.

2.3 Bagging

To improve classifier performance and reduce variance we investigated the use of bagging (Breiman, 1996), also referred to as bootstrap resampling. An ensemble classifier is learned by training multiple basic frequency classifiers. Specifically, let $train$ be a training set consisting of $size(train)$ labeled examples and b be the number of basic classifiers to be learned and combined. The bagged classifier is trained by repeatedly

1) generating a new training sample $S \subseteq train$ containing $size(train)$ labeled examples by randomly sampling with replacement from $train$ and then
2) training a new basic frequency classifier on S.

These two steps are repeated b times resulting in b different classifiers. To classify a new, unlabeled example, each of the b classifiers makes a prediction and the final label is the label with the majority vote, with ties going to not complex (0), since this was the more frequent class.

3 Experiments

We submitted two systems to the SemEval Complex Word Identification challenge (Task 11), which used the same parameter settings and only differed in where the corpus frequencies were collected, English Wikipedia (NormalBag) and the Google Web Corpus (GoogleBag). Both systems used $b = 10$ rounds of bagging, which was shown experimentally to have the best scoring value, using repeated rounds of 10-fold validation. We also discuss results here for a system which used Simple English Wikipedia word frequencies, though we did not submit it to the challenge (for consistency, we denote it SimpleBag).

[1] https://simple.wikipedia.org/

[2] https://en.wikipedia.org/

	Test			Train		
System	G-score	Accuracy	Recall	G-score	Accuracy	Recall
NormalBag	0.714	0.603	0.872	0.684	0.665	0.705
GoogleBag	0.691	0.568	0.881	0.675	0.648	0.704
SimpleBag	0.674	0.542	0.891	0.674	0.630	0.725

Table 1: Results for the bagged systems ($b = 10$) with word frequencies calculated on the three different corpora. Results are shown for the test and training sets for systems trained on the training data.

The task consisted of a training data set (N = 2,237), which was available during development, and a test data set (N = 88,221) on which the competition was scored and the labels were only released after the competition (Paetzold and Specia, 2016). We use both data sets here to analyze the performance of the classifiers.

3.1 The impact of corpus choice

Table 1 shows the results for the classifiers trained using bagging with word frequencies calculated from the three different corpora. English Wikipedia performed the best, followed by Google Web Corpus and finally Simple English Wikipedia. The top two entries were entered into the SemEval competition and ranked 13th and 16th respectively out of 51 systems (42 team submitted systems and 9 baseline systems). We hypothesize that English Wikipedia represents a good compromise between size/coverage and corpus quality; even though NormalBag had slightly lower recall than the other two, it was able to achieve that recall with a significantly higher accuracy.

To verify that the differences in performance between the three systems were significant, we used bootstrap resampling with a paired sample t-test (Koehn, 2004). Based on 100 random samples, all differences between all metrics and all systems were significant ($p < 0.0001$, with Bonferroni correction to correct for testing multiple different comparisons).

Overall, relative to other systems that were submitted for the SemEval task, these frequency-based classifiers biased towards recall, e.g. the Google frequency and English Wikipedia frequency systems ranked 3rd and 5th with respect to recall (of the 42 team submitted systems).

	Train	
Corpus	basic G-score	bagged G-score
Normal	0.677	**0.680**
Google	0.668	0.667
Simple	0.665	**0.669**

Table 2: Comparison of the basic frequency classifiers and their bagged counterparts with $b = 10$ on the training data averaged over 100 random 10-fold samples. Systems that were significantly different between the basic and bagged are in bold.

3.2 The impact of bagging

To measure the impact of bagging on the prediction performance of the systems, for each corpus source, we compared the basic frequency classifier to the bagged variant. We generated 100 random 10-fold partitions of the training data and performed 10-fold cross validation on each for each system variant. We averaged the results across the each 10-fold set resulting in 100 scores for each of the systems.

Table 2 shows the averages over these 100 scores. For both of the Wikipedia variants (Normal and Simple) bagging provided a small increase in performance ($p < 0.0001$ based on a paired t-test). For the Google frequencies the performance actually decreased, though this decrease was not significant.

To understand the effect that b (the number of bootstrap samples) has on the performance of the classifier, we compared the performance of the classifier with $b = 1, 2, ..., 100$. Figure 1 shows a plot of the G-score versus the number of bags used by the classifier for the NormalBag classifier on the training set. As with the previous experiment, to partially mitigate noise, we generated 100 randomly 10-fold sets and averaged the results across all of these to generate the data.

For small b, increasing the number of classifiers voting does increase the performance of the classifier. However, after around $b = 10$ adding more

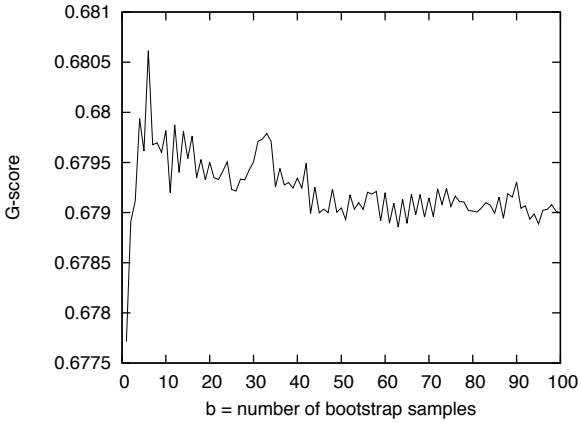

Figure 1: G-score for the NormalBag classifier with varying number of boostrap samples. Results are averages over 100 random 10-fold samples of the training data.

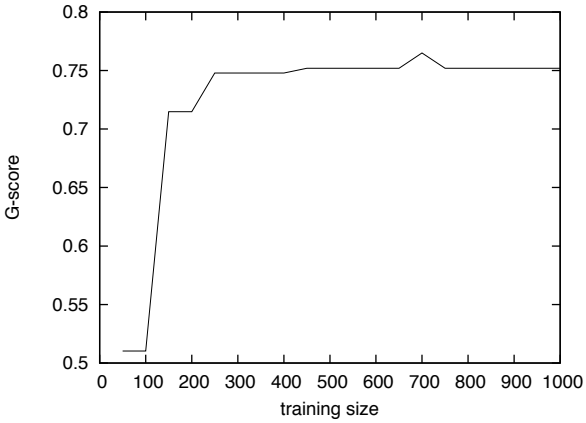

Figure 2: G-score for the basic threshold classifier using English Wikipedia frequencies for increasing training data size. Here the training data is a subset of the SemEval "test" set.

classifiers degrades the performance with the classifier. Although the difference is small for $b = 1$ vs. $b = 10$ (0.677 vs. 0.680), the difference is statistically significant.

3.3 Limits of frequency-based classification

Using English Wikipedia frequencies and the optimal frequency threshold (i.e. α), the basic threshold classifier achieves a G-score of 0.779 on the test data set. This is slightly higher than the best scoring SemEval system, which achieved 0.774. Clearly frequency provides a strong signal for word complexity.

The previous experiment assumes an unreasonable scenario where we know the labels and can pick the optimal value. To better understand the impact of frequency, we split the test data into 10-folds and performed 10-fold cross-validation analysis using the basic threshold classifier, training the threshold on 90% of the SemEval "test" data and then testing on the remaining 10%. In this scenario, the threshold classifier still achieves a G-score of 0.764, only slightly less than the score achieved using the optimal threshold.

0.764 is still significantly higher than the score achieved by the system when trained on the SemEval "training" set. Two possible differences exist between the training and testing data. First, the test data is two orders of magnitude larger than the original training data. This additional data could result in a more reliable classifier. Alternatively, train and test were generated in different ways and could have different characteristics.

To investigate this, we held out 10% of the "test" data set as testing data and trained the basic threshold classifier on increasing amounts of the remaining 90%. Figure 2 shows the G-score for training sizes up to 1,000 (the G-score mostly stabilized beyond 1,000 with only minor variation). Even with only 250 training examples, the classifier already achieves a G-score of 0.748 and with 500 training examples, it achieves 0.752, only a little less than the final score using all of the training data of 0.760. For the frequency classifier, more the data domain, and less the size, accounts for the differences in performance seen.

4 Conclusion

In this paper, we described our entry for the complex word identification SeEval 2016 task (#11). We utilize word frequency to classify complexity, with less frequent words being classified as complex. As has been seen in previous corpus studies, frequency is a very strong predictor of the complexity of a word. However, the corpus where those frequencies are measured does play a role in performance. We found that English Wikipedia performed best for this particular task. Future research is needed to investigate this phenomena more broadly.

References

Thorsten Brants and Alex Franz. 2006. Web 1T 5-gram version 1. Linguistic Data Consortium, Philadelphia.

Leo Breiman. 1996. Bagging predictors. *Machine learning*.

Hunter M Breland. 1996. Word frequency and word difficulty: A comparison of counts in four corpora. *Psychological Science*.

William Coster and David Kauchak. 2011. Simple English Wikipedia: A new text simplification task. In *Proceedings of ACL*.

David Kauchak. 2013. Improving text simplification language modeling using unsimplified text data. In *Proceedings of ACL*.

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *EMNLP*.

Gondy Leroy and David Kauchak. 2013. The effect of word familiarity on actual and perceived text difficulty. *Journal of American Medical Informatics Association*.

Gondy Leroy, James Endicott, Obay Mouradi, David Kauchak, and Melissa Just. 2012. Improving perceived and actual text difficulty for health information consumers using semi-automated methods. In *American Medical Informatics Association (AMIA) Fall Symposium*.

Gondy Leroy, James E. Endicott, David Kauchak, Obay Mouradi, and Melissa Just. 2013. User evaluation of the effects of a text simplification algorithm using term familiarity on perception, understanding, learning, and information retention. *Journal of Medical Internet Research (JMIR)*.

Christopher D Manning, Mihai Surdeanu, John Bauer, Jenny Rose Finkel, Steven Bethard, and David McClosky. 2014. The stanford corenlp natural language processing toolkit. In *ACL (System Demonstrations)*.

Courtney Napoles and Mark Dredze. 2010. Learning simple Wikipedia: A cogitation in ascertaining abecedarian language. In *Proceedings of HLT/NAACL Workshop on Computation Linguistics and Writing*.

Gustavo H. Paetzold and Lucia Specia. 2016. Semeval 2016 task 11: Complex word identification. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval 2016)*.

Emily Pitler and Ani Nenkova. 2008. Revisiting readability: A unified framework for predicting text quality. In *Proceedings of the conference on empirical methods in natural language processing*. Association for Computational Linguistics.

Matthew Shardlow. 2014. A survey of automated text simplification. *International Journal of Advanced Computer Science and Applications*.

Advaith Siddharthan. 2014. A survey of research on text simplification. *ITL-International Journal of Applied Linguistics*.