HMC at SemEval-2016 Task 11: Identifying Complex Words Using Depth-limited Decision Trees

Maury Quijada and **Julie Medero**
Harvey Mudd College
340 E Foothill Blvd
Claremont, CA, 91711, USA
{mquijada, jmedero}@hmc.edu

Abstract

We present two systems created for SemEval-2016s Task 11: Complex Word Identification. Our two systems, a regression tree and decision tree, were trained with a word's unigram and lemma word counts, average age-of-acquisition, and a measure of concreteness. The systems ranked 5th and 6th, respectively, on the test set by G-score (the harmonic mean between accuracy and recall). With the regression tree's predictions earning a G-score of 0.766, and the decision tree's earning 0.765, the two systems scored within 1 percent of the score of the best-performing system in the task.

1 Introduction

Text simplification is the process of reducing the complexity of a text while preserving the original meaning. Text simplification may include syntactic or pragmatic aspects (Siddharthan, 2014), but much of the work that has been done has focused on lexical simplifications. In lexical simplification, difficult words or phrases are replaced to make a text more accessible. This kind of simplification can benefit several reader populations, including second-language learners (Petersen and Ostendorf, 2009).

One important first step in lexical simplification is complex word identification. This step predicts which words in a text will be difficult for a reader so that they can then be targeted for simplification.

The International Workshop on Semantic Evaluation for 2016 (SemEval-2016) hosted Task 11: Complex Word Identification (CWI), which asked participants to identify words that would be challenging for non-native English speakers (Paetzold and Specia, 2016). Task participants were given 2,237 training examples. Each example contained a word, the sentence containing the word, and the word's index in the sentence. In addition, each word was labeled as *complex* or *not complex* (or *simple*) by 20 human annotators, and each word was given a binary label of *complex* if at least one annotator thought the word was hard to read. The individual labels of the 20 annotators were also made available to participants.

Submissions were evaluated on a test set of 88,221 words. Test set items had the same format as the training set, except that they were only annotated by one person, so the labels indicated whether that annotator alone labeled it *complex*.

2 Previous Work

This is the first year of the CWI Task, so no previous work has been done on exactly the same task. However, substantial work has been done previously in the general area of characterizing word difficulty.

Traditional readability measures like Flesch-Kincaid (Kincaid and others, 1975) and Gunning Fog (Gunning, 1952) rely primarily on word length, while the Lexile framework considers word length and unigram frequency. More recent work has incorporated n-grams and part of speech information (Petersen and Ostendorf, 2009; Graesser and others, 2004), word clusters (Deane and others, 2006) to improve the accuracy of reading level predictions, while other work has used orthographic and phonemic features (Mostow and others, 2002) to predict where children would have reading difficulty.

Proceedings of SemEval-2016, pages 1034–1037,
San Diego, California, June 16-17, 2016. ©2016 Association for Computational Linguistics

3 Methodology

For each word in the training set, we extracted features that we predicted would be good indicators of a word's complexity. We also used the labels from the individual annotators to generate continuous-valued labels for each of the training words. Finally, we trained several machine learning models as implemented using Python's `scikit-learn` package (Pedregosa and others, 2011).

3.1 Features

We investigated several potential features for the CWI Task, based on metrics across various features for words in the training set (see Table 1).

1. **Unigram** and **lemma frequency** from the Corpus of Contemporary American English (COCA) (Davies, 2008), using the WordNet lemmatizer in Python's `nltk` package (Loper and Bird, 2002). *Complex* words are, on average, less frequent than *simple* words.

2. **Word age-of-acquisition** according to the average age-of-acquisition (AoA) for a word in a list of roughly 30,000 English words (Kuperman and others, 2012). *Complex* words, have a higher mean AoA.

3. **Word concreteness**, on a scale of 1 to 5, according to a list of roughly 40,000 English words (Brysbaert and others, 2013). *Complex* words have higher scores, possibly due to the inclusion of technical words.

4. **Word length**, **stem length**, and **lemma length** in characters. For all three, *complex* words are longer on average than *simple* words.

5. **Number of word pronunciations** in Carnegie Mellon University's Pronouncing dictionary (Lenzo, 2007), accessed through `nltk`. *Complex* words have a lower number of pronunciations on average, possibly because these words tend to be more technical.

6. **Probability of the word's sequence of characters** according to a character-based trigram language model created with SRILM and trained on the COCA dataset (Stolcke and others, 2002). *Complex* words have lower log-probabilities. This may be because *complex* words have more "unlikely" character sequences that are hard to decode.

Feature	Simple	Complex
Unigram Count	443k (1M)	151k (597k)
Lemma Count	183k (742k)	68k (290k)
Age of Acquisition	8.9 (3.2)	9.8 (3.0)
Concreteness	2.8 (0.9)	3.0 (1.0)
Word Length	6.0 (2.5)	6.7 (2.5)
Pronunciation Count	1.4 (0.7)	1.2 (0.5)
Synset Count	9.5 (8.7)	6.7 (8.3)

Table 1: Summary of feature means and standard deviations.

7. **Number of synsets** in WordNet. *Complex* words belong to a lower number of synsets, possibly because they are more likely to be unique and domain-specific.

8. **Part-of-speech** (POS) given by `nltk`'s part-of-speech tagger. Nouns are more likely to be *complex* words, and verbs are more likely to be *simple* words. For models that do not support categorical features, we performed a one-hot encoding of the most common tags: NN, NNS, JJ, RB, and VBD.

We experimented with different combinations of features, but only five features were used in the predictions given by our models: unigram and lemma frequency, age-of-acquisition, concreteness, and word length. Therefore, those were the features that we used in our sybmitted system.

3.1.1 Labels

Originally, the training set included binary labels that correspond to whether at least one annotator thought a word was *complex*. However, because these labels are binary, they are not indicative of the extent to which the annotators agreed each word was difficult. It is useful to learn the difference, for example, between a word that almost all of the annotators agree is difficult, and one that 19/20 annotators felt was *simple*, but one annotator thought was *complex*.

To help with this, we replaced every binary label with a continuous label representing the percentage of annotators who found the word to be complex.

Description	Regression Tree	Decision Tree
Accuracy	0.838	0.846
Precision	0.182	0.189
Recall	0.705	0.698
F_1-score	0.290	0.298
G-score	0.766	0.765

Table 2: Summary of system performance on the CWI test set

We used thresholding to convert the continuous labels back to binary labels as needed.

3.2 Models

We experimented with Support Vector Machines, Logisitic Regression, and Perceptrons, but got the best results from `scikit-learn`'s implementation of depth-limited regression and decision trees with our features and a maximum depth of 3. These models gave the best performance on average in terms of cross-validation accuracy, F_1-score, and G-score. Regression trees and decision trees have the added benefit of providing a level of model interpretability that the other models do not.

Since decision trees requires discrete categories for training, we used a threshold of 0.25 when providing the labels to the model for training. That is to say, all examples in training were labeled *complex* if at least 25 percent of annotators marked it *complex*. For each cross-validation fold, we used the proportion of the first 19 annotators who labeled a word complex as labels during training, then evaluated our model on the labels provided by the 20th annotator.

We also trained our regression tree on the proportion of the first 19 annotators who labeled each word *complex*. But since regression trees train on and predict continuous-valued labels, we used the percent of annotators directly as labels during training, then thresholded the model's prediction for each word at test time. We got best results with a threshold of 0.05: words were interpreted as *complex* if the model predicted its measure of complexity to be 0.05 or greater. These thresholds were chosen because they gave the best G-score for 5-fold cross-validation on the training set.

4 Results

On the CWI test set, our regression tree and decision tree ranked 5th and 6th on G-score, respectively, out of the 40 system submissions (Paetzold and Specia, 2016). Table 2 breaks down the precision, recall, and G-score of each model. The models had G-scores of 0.765 and 0.766. By comparison, the top scoring system had a G-score of 0.774, and overall the average G-score was 0.620 with a standard deviation of 0.123 (Paetzold and Specia, 2016).

		Prediction	
		Complex	**Not Complex**
Truth	**Complex**	2913	1218
	Not Complex	13059	71031

Figure 1: Regression tree confusion matrix on the CWI test set.

		Prediction	
		Complex	**Not Complex**
Truth	**Complex**	2884	1247
	Not Complex	12355	71735

Figure 2: Decision tree confusion matrix on the CWI test set

Figures 1 and 2 depict the confusion matrices generated from comparing the trees' predictions to the testing labels. Over 90% of the misclassifications given by both trees were false positives. This indicates that the models tended to overpredict complex words, which is also seen in the relatively low precision of both systems.

5 Analysis

Our models relied most heavily on unigram and lemma frequency features. Even when the AoA, concreteness, and lemma length features are excluded, the regression tree and decision tree obtain a G-score of 0.735 and 0.770, respectively, on the test set. This indicates that corpus frequency alone is an extremely good indicator of a word's complexity.

Despite our submitted models' success with corpus-based features, we obtained low precision scores of 0.18 for each model. These scores were consistent with results for many of the other systems participating in the task. The average precision score was 0.123 with a standard deviation of 0.06 (Paetzold and Specia, 2016).

We posit that this problem is due in part to the difference in distribution between the training and testing sets for our models. Namely, our models must train on labels that are representative of the judgments of multiple annotators, but also be tested

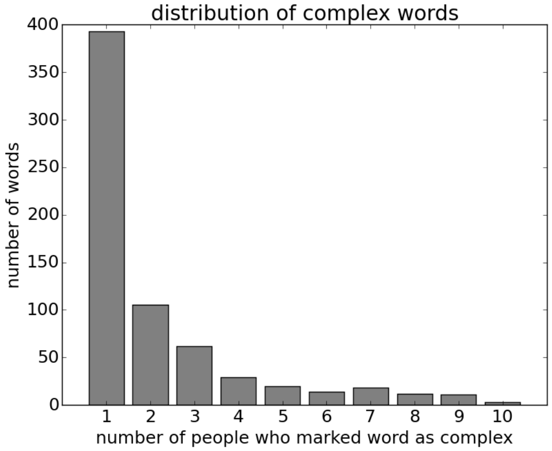

Figure 3: Distribution of how often a certain number of annotators marked a word *complex*.

on labels that are representative of the judgment of *only one* annotator. The difference in distribution between the training and testing set means that the model learns to predict word complexity for a group, but is evaluated on one person's judgments. This problem is exacerbated by the low agreement between annotators. Figure 3 shows that most words marked *complex* are labeled so by only one person.

In addition, some of the labels in the test set are possibly counterintuitive. For example, in:

> The Plan of Management is the main policy document for the Park and strives to balance strategic or long-term goals and tactical or day to day goals.

the word *strives* and both instances of the word *goals* were labeled *complex*. Intuitively, those words seem less difficult to us than the words *strategic* and *tactical*, which were both labeled *not complex* by the same annotator. This example illustrates what makes the task so difficult: not only are the testing and training set distributions different, but the labels for each are subjective and possibly conflicting.

6 Conclusion

This paper details our submission to SemEval-2016's Task 11: Complex Word Identification. We explored several potential features, eventually submitting a regression tree and decision tree based on unigram and lemma frequency, age-of-acquisition, concreteness, and word length. By incorporating annotator disagreement into our models through continuous-valued labels during training and testing, our models ranked 5th and 6th overall in the Task. Error analysis reveals that our models had trouble generalizing the judgments of multiple annotators in the training set to the judgment of one annotator in the test set, leading to low precision.

References

M Brysbaert et al. 2013. Concreteness ratings for 40 thousand generally known english word lemmas. *Behavior Research Methods*, 46(3):904–911.

M Davies. 2008. The corpus of contemporary american english: 520 million words, 1990-present.

P Deane et al. 2006. Differences in Text Structure and Its Implications for Assessment of Struggling Readers. *Scientific Studies of Reading*, 10(3):257–275, July.

A C Graesser et al. 2004. Coh-Metrix:Analysis of text on cohesion and language. *Behavior Research Methods, Instruments, and Computers*, 36:193–202.

R Gunning. 1952. *The Technique of Clear Writing*. McGraw-Hill.

J. P. Jr. Kincaid et al. 1975. Derivation of new readability formulas for Navy enlisted personnel. *Research Branch Report 8-75*.

V Kuperman et al. 2012. Age-of-acquisition ratings for 30,000 english words. *Behavior Research Methods*, 44(4):978–990.

K Lenzo. 2007. The cmu pronouncing dictionary.

E Loper and S Bird. 2002. Nltk: The natural language toolkit. In *Proc. ACL-02 Workshop on Effective Tools and Methodologies for Teaching NLP and Comp. Linguistics*, ETMTNLP '02, pages 63–70, Stroudsburg, PA, USA. Association for Computational Linguistics.

J Mostow et al. 2002. Predicting Oral Reading Miscues. In *Proc. ICSLP*.

G Paetzold and L Specia. 2016. Semeval2016, task 11: Complex word identification. In *Proc. Sem-Eval 20016 Shared Task 11: Complex Word Identification*.

F Pedregosa et al. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

S Petersen and M Ostendorf. 2009. A machine learning approach to reading level assessment. *Computer, Speech and Language*, 23(1):89–106.

A Siddharthan. 2014. A survey of research on text simplification. *ITL-International Journal of Applied Linguistics*, 165(2):259–298.

A Stolcke et al. 2002. Srilm-an extensible language modeling toolkit. In *Proc. INTERSPEECH*.