UWB at SemEval-2016 Task 11: Exploring Features for Complex Word Identification

Michal Konkol
NTIS – New Technologies for the Information Society &
Department of Computer Science and Engineering,
Faculty of Applied Sciences,
University of West Bohemia,
Univerzitní 8, 306 14 Plzeň
Czech Republic
konkol@kiv.zcu.cz

Abstract

In this paper, we present our system developed for the SemEval 2016 Task 11: Complex Word Identification. Our team achieved the 3rd place among 21 participants. Our systems ranked 4th and 13th among 42 submitted systems. We proposed multiple features suitable for complex word identification, evaluated them, and discussed their properties. According to the results of our experiments, our final system used maximum entropy classifier with a single feature – document frequency.

1 Introduction

This paper describes our participation in the Complex Word Identification (CWI) shared task of SemEval 2016. CWI is a subtask of text simplification. Text simplification changes the structure, grammar, and vocabulary of the text to make it easier to understand without losing information. CWI looks for the words that should be simplified. The task is motivated and described in detail by the organizers (Paetzold and Specia, 2016).

In this paper we apply a machine learning approach to CWI. Our main goal is to explore suitable features.

The paper has the following structure. Section 2 introduces the task in more detail. Section 3 presents the design of our system. In Section 4, we choose the optimal parameters and features for our final system. In Section 5, we discuss overall results of the task. Section 6 summarizes our contribution.

2 Task

The task is defined as a binary classification with classes *complex* and *simple*. The complex class represents words that should be simplified. Given a word in a sentence, a system decides whether the word is complex or not.

Paetzold and Specia (2016) prepared two training data sets. The first data set contains 20 decisions from unique annotators; we call it *all*. Second data set aggregates the annotations – the word becomes complex if at least one annotator considers it complex; we call it *aggregated*. Both data sets have 2 237 sentences (i.e. the all data set has 44 740 training examples, the aggregated set has only 2 237). The organizers announced that the test data set has 9 200 sentences annotated by a single annotator.

The task was evaluated using *G-score*: a harmonic mean of accuracy and recall of the complex class. Each team could submit two systems for evaluation.

3 System

CWI is a binary classification task. We chose the maximum entropy classifier for our system, but we believe the choice of classifier has only a small impact compared with the choice of features.

We propose features that may be suitable for the CWI task and motivate them in the following list.

Word frequency – A ratio of occurrences of the current token to the number of all tokens in the corpus. We expect less frequent words to be more likely complex. Also extended to n-grams.

Proceedings of SemEval-2016, pages 963–966,
San Diego, California, June 16-17, 2016.

Document frequency – A ratio of documents that contain the current token to the number of all documents. The motivation is the same as for word frequency. Also extended to n-grams.

Character n-gram frequency – The frequency of character n-grams of the current word. We expect that some character n-grams are typical for complex words.

Language model word probability – The probability of the current word given by a language model. The more probable words are not likely to be complex.

Part of speech – Part of speech tag of the current word. Some word categories are used less often by non-native speakers.

Word length – Length of the current word. Common words are usually shorter.

WordNet synset size – The size of WordNet synsets which contain the current word. Large synsets might be highly ambiguous and thus harder for a non-native speaker.

WordNet frequency ratio – The ratio between the current word and the most frequent word in a synset. If there is a lot more frequent word with the same meaning, the non-native speaker usually uses this word instead of the current word.

WordNet number of synsets – The number of synsets that contain the current word. Word with many meanings may be harder.

Language model sentence probability – The probability of the whole sentence. The overall probability of a sentence may indicate the sentence complexity. A complex sentence may affect the difficulty of the current word.

Average n-gram frequency – The average n-gram frequency of the sentence. If the sentence contains a higher number of complex words, then even a common word may become complex.

3.1 G-score decision boundary

A standard maximum entropy classifier is trained so that all types of errors have the same weight. This is not the case when G-score is used as an evaluation metric. With G-score and a skewed data set, it is necessary to give higher priority to errors where complex class is incorrectly classified as simple class. This type of errors lowers the recall of the system.

To deal with it, we propose an altered decision criterion (1) for the standard classifier, where y^* is the class chosen by the system, $p(y = c|d)$ is the probability of the complex class given the data, and t is the threshold.

$$y^* = \begin{cases} \text{complex} & p(y = c|d) > t \\ \text{simple} & \text{otherwise} \end{cases} \quad (1)$$

4 Optimizing parameters

Each team could submit two systems. We decided to submit two versions of the same system with different parametrization – the first one trained on the *aggregated* data and the second one on the *all* data.

4.1 Experimental setup

We estimated the n-gram frequencies from Wikipedia. We used the language model from (Brychcín and Konopík, 2015), Stanford CoreNLP for part-of-speech tags (Toutanova et al., 2003), the MIT Java Wordnet Interface (Finlayson, 2014), and the Brainy implementation of maximum entropy classifier (Konkol, 2014).

4.2 Procedure

We measure the improvement of our system with 5-fold cross-validation on the training data.

First, we optimized hyper-parameters (e.g. thresholds, discretization) of individual features.

Second, we iteratively added more features to the (initially empty) feature set. At each step we included the feature that improved G-score the most. We stopped if G-score decreased.

For each evaluation, we chose the optimal threshold t from (1).

4.3 Results

Table 1 shows the results for individual features (i.e. first iteration) on the *aggregated* data. We compare the features to a baseline that marks all words as complex.

We found that the unigram document and word frequencies predicted the complex words best from

System	Accuracy	Recall	G-score
all complex	31.6%	100.0%	48.0%
1-gram freq	69.5%	65.0%	67.2%
2-gram freq	54.6%	77.9%	64.2%
3-gram freq	51.1%	71.8%	59.7%
1-gram doc freq	64.2%	72.5%	68.1%
2-gram doc freq	52.5%	80.9%	63.7%
3-gram doc freq	36.2%	96.9%	52.7%
char 3-gram freq	58.2%	63.6%	60.8%
char 4-gram freq	50.5%	83.4%	62.9%
char 5-gram freq	58.5%	66.7%	62.3%
lang model	61.0%	61.0%	61.0%
pos	47.8%	67.0%	55.8%
word length	48.3%	80.7%	60.4%
WN synset size	46.9%	67.8%	55.5%
WN freq ratio	47.3%	56.4%	51.5%
WN num synsets	54.7%	62.2%	58.2%
global lang model	42.6%	91.8%	58.2%
avg word freq	45.1%	84.5%	58.8%

Table 1: Results for individual features with 5-fold cross-validation on the aggregated training data. The official metric for evaluation is called G-score and it is a harmonic mean of accuracy and recall of the complex class.

all the proposed features. The unigram document frequency performed slightly better. We believe document frequency better reflected the real frequency of the words. For example the word YouGov appears almost 3000 times at a single Wikipedia page, thus we overestimated the word frequency of YouGov.

We are ambivalent about the role of the context. The global language model and the average word frequency features show that the overall complexity of a sentence affected the complexity of its words. It suggests that the context of the word plays a role. However, unigrams performed better than higher order n-grams for document and word frequencies. The complexity of the word may be thus affected by some words in the sentence but not necessarily the closest words. We believe this phenomenon requires further research.

The WordNet features revealed that the number of meanings and number of synonyms influenced the complexity of the word.

The character n-gram features might indicate that some combinations of letters were typical for complex words, but it might also only identify the shorter words (and their frequencies). The latter option is supported by better scores of higher order character n-grams, because they could recognize longer words.

The word length feature showed that such a simple feature can be relatively good predictor of complex words.

The part-of-speech feature proves that some types of words are more complex than others, e.g. adverbs are more complex than verbs.

Even though all the features improved the all complex baseline, we found that adding more features to the unigram document frequency (best individual feature) did not improve the G-score. We did not expect this behavior, because the features represent different information sources. This might have many reasons, we elaborate two of them. First, we might get close to the inter-annotator agreement as the task seems to be highly subjective. Second, we use a heuristic procedure for finding the optimal feature set. This procedure might be flawed, especially choosing the discretization before we combined the features.

The final system used only the unigram document frequency. As we used only one feature, it might be better to drop the classifier and simply find a threshold for unigram document frequency. We believe the results would be the same, but the system could be even simpler.

5 Shared task evaluation

Paetzold and Specia (2016) presented complete results for all systems. 21 teams participated in the task and submitted 42 systems. The results show that the top 9 systems were over 76%, so they performed almost as good as the winner (77.4%).

We provide a summarized results of our systems in Table 2. Our team ranked third with systems on 4th (*all* data) and 13th (*aggregated* data) place. The system trained on the aggregated data predicts complex words for the worst English speakers (among the annotators); the other one for average speakers. The difference between the systems (7.4%) suggest that the difference between worst and average speakers (regarding vocabulary) is not that high.

TR	SR	System	Accuracy	Recall	G-score
1.	1.	SV000gg soft	77.9%	76.9%	77.4%
3.	4.	UWB All	80.3%	73.4%	76.7%
3.	13.	UWB Agg	56.9%	88.5%	69.3%
—	—	Average	73.69%	59.1%	62.0%

Table 2: The official comparison with other systems. TR stands for team rank, SR for system rank. There were 21 teams and 42 systems. The official metric for evaluation is called G-score and it is a harmonic mean of accuracy and recall of the complex class.

6 Conclusion

We proposed a set of features that might be suitable for the CWI task. We showed that most of the features had some information value, even though their combination may not lead to an improvement. We can conclude that the word frequency and document frequency were the best predictors for complex words.

Our final system used a maximum entropy classifier with a single feature – document frequency. We ended up as the third best team (with the 4th best system). It proves that it is possible to achieve a state-of-the-art G-score in the CWI task with a very simple system.

Acknowledgments

This publication was supported by the project LO1506 of the Czech Ministry of Education, Youth and Sports.

References

Tomáš Brychcín and Miloslav Konopík. 2015. Latent semantics in language models. *Computer Speech & Language*, 33(1):88 – 108.

Mark Finlayson. 2014. Java libraries for accessing the princeton wordnet: Comparison and evaluation. In Heili Orav, Christiane Fellbaum, and Piek Vossen, editors, *Proceedings of the Seventh Global Wordnet Conference*, pages 78–85, Tartu, Estonia.

Michal Konkol. 2014. Brainy: A machine learning library. In Leszek Rutkowski, Marcin Korytkowski, Rafa Scherer, Ryszard Tadeusiewicz, Lotfi A. Zadeh, and Jacek M. Zurada, editors, *Artificial Intelligence and Soft Computing*, volume 8468 of *Lecture Notes in Computer Science*, pages 490–499. Springer International Publishing.

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)*.

Kristina Toutanova, Dan Klein, Christopher D. Manning, and Yoram Singer. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, NAACL '03, pages 173–180, Stroudsburg, PA, USA. Association for Computational Linguistics.