# PLUJAGH at SemEval-2016 Task 11: Simple System for Complex Word Identification

**Krzysztof Wróbel**
Jagiellonian University
ul. Golebia 24
31-007 Krakow, Poland
AGH University of Science and Technology
al. Mickiewicza 30
30-059 Krakow, Poland
kwrobel@agh.edu.pl

## Abstract

This paper presents the description of a system which detects complex words. It solely uses information regarding the presence of a word in a prepared vocabulary list. The system outperforms multiple more advanced systems and is ranked fourth for the shared task, with minimal loss to the best system. F-score optimization guaranteed the first place in this measurement. Different features are considered and evaluated. Maximal bounds are predicted. The rule "the simplest methods give the best results" is confirmed.

## 1 Introduction

The goal of Complex Word Identification (CWI) is to detect words in a text that are complex (not easy to understand) for some group of people. CWI is one of the tasks of SemEval-2016 (Paetzold and Specia, 2016).

CWI can be treated as the first step of Lexical Simplification (LS). LS was a task of SemEval-2012 (Specia et al., 2012). Complex words were identified using n-grams, the length of the word, and the number of syllables (Ligozat et al., 2012; De Belder et al., 2010; Biran et al., 2011). The resources exploited in this task include Wikipedia, WordNet, Google Web 1T corpus (Sinha, 2012; Paetzold and Specia, 2015). Additional annotation of input sentences was performed by: a part-of-speech tagger, and word sense disambiguation (Amoia and Romanelli, 2012; Jauhar and Specia, 2012).

A similar task is the prediction of the readability of a whole text. In comparison, in CWI, each word has to be scored. The applied methods are summarized in (Dębowski et al., 2015).

This paper presents findings regarding the necessary data and the performed experiments. For the final submission, a simple system was chosen, which scored at fourth place.

## 2 Task Data Analysis

It is important to notice the difference between training and test data. Each sentence in the training set was annotated by 20 annotators. If at least one of them classified a word in a sentence as complex, it was marked as complex. The training data consists of 2237 classified words. On the other hand, each sentence in the test data (88221 classified words) was annotated by only one annotator.

Complex words represent 31.56% of the words in the training data. Fortunately, organizers published the unaggregated annotations – every word in a sentence has 20 annotations. In this scenario, only 4.55% instances are classified as complex.

A priori probability of the word being complex is important knowledge for the classification task.

What is more, the organizers shared the baseline results for test data (Table 1). It shows that complex words represent 4.7% of instances in the test data – similar to training.

## 3 Resources and Methods

Knowledge bases are essential to this task. Wikipedia is one of the most popular sources of text used in NLP. Using the cycloped.io (Smywiński-Pohl and Wróbel, 2014) framework the English and Simple English Wikipedia were preprocessed. The



*Proceedings of SemEval-2016*, pages 953–957,
San Diego, California, June 16-17, 2016. 

**Table 1:** Scores for baseline systems on the test data. 1) All complex – all words are classified as complex, 2) All simple: all words are classified as simple, 3) Ogden's lexicon: words present in Ogden's Basic English vocabulary are classified as simple, others as complex. G-score is defined as a harmonic mean of accuracy and recall.

| System | Accuracy | Recall | G-score |
|---|---|---|---|
| All complex | 0.047 | 1.000 | 0.089 |
| All simple | 0.953 | 0.000 | 0.000 |
| Ogden's lexicon | 0.248 | 0.947 | 0.393 |

text extracted from articles allowed the calculation of term frequency (TF) and document frequency (DF). TF represents the total number of times a word appears in the corpora; DF is the number of documents in which the word occurred at least once.

It was required to apply the same tokenization of corpora as in the data from the organizers.

For every word which needed classification, many features were created:

- TF and DF for the word and its lemma use,
  - English Wikipedia,
  - Simple English Wikipedia,
  - corpora created from training and test sentences,
- length of sentence (number of words),
- length of word (number of characters),
- position of word in sentence,
- GloVe word embedding (Pennington et al., 2014).

For quick development, sklearn (Pedregosa et al., 2011) was used. Many supervised machine-learning algorithms were tested using cross-validation:

- decision trees with maximum depth from 1 to 6,
- linear classifier with stochastic gradient descent (SGD) training,
- k-nearest neighbors classifiers for k=3,5,10,20,
- random forest,
- extremely randomized trees,
- AdaBoost,
- GradientBoostingClassifier,
- LinearSVC.

**Table 2:** Ranking of features in terms of G-score. The last position presents the score for all features used in one model.

| Feature | G-score |
|---|---|
| DF of Simple English Wikipedia | 0.781 |
| lemma TF of Simple English Wikipedia | 0.781 |
| TF of Simple English Wikipedia | 0.780 |
| lemma TF of English Wikipedia | 0.778 |
| TF of training corpus | 0.774 |
| TF of English Wikipedia | 0.767 |
| GloVe word embeddings | 0.767 |
| TF of CHILDES Parental Corpus | 0.738 |
| length of word | 0.618 |
| position of word in sentence | 0.556 |
| length of sentence | 0.505 |
| all features | 0.784 |

## 4 Evaluation

All experiments were conducted by employing cross-validation on raw vote data. Training data were aggregated – a word is labeled as complex if at least two annotators marked it accordingly.

### 4.1 Metrics

The results are scored using a harmonic mean of accuracy and recall (marked as G-score). In comparison to F-score (a harmonic mean of precision and recall), it is higher if more instances are predicted as complex.

### 4.2 Experiments

Tree-based classifiers achieved the best results (except for word embeddings). Table 2 presents the G-scores obtained by training a classifier with each of the features. Combining features gives only a slightly better score.

#### 4.2.1 Upper Bounds

Complex word identification is a subjective task. The understanding of a word depends on the knowledge of a particular person. Therefore, 100% G-score is impossible to achieve. Due to the fact that the training data was annotated by multiple annotators, it was possible to measure the inter-annotator agreement. Two theoretical systems were scored on the training data. Both systems have knowledge regarding the annotators' assessment of the words in

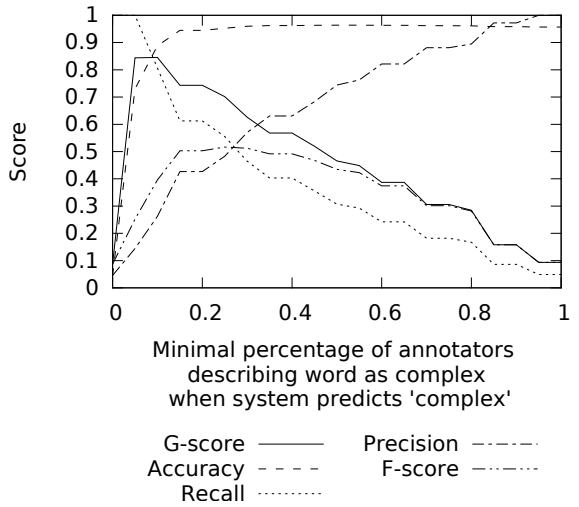


**Figure 1:** Results for the first theoretical system using classification with information about context.

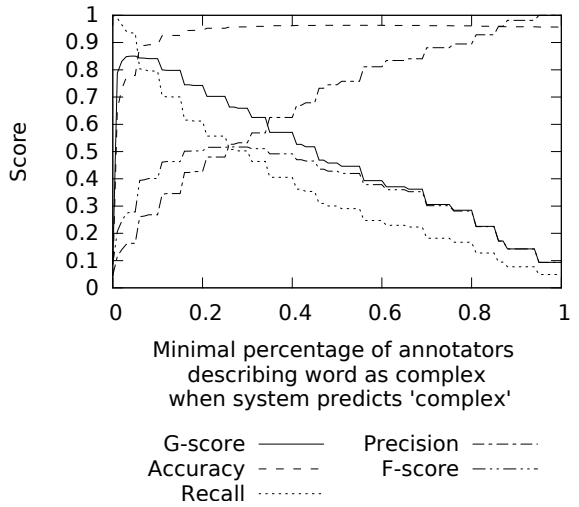


**Figure 2:** Results for the second theoretical system using classification without information about context.

sentences. The first one has information regarding the context (whole sentence) – for each sentence, it knows how many annotators recognized each word as complex. The second one knows how many times each word was assessed as complex (without context).

1. The problem can be treated as simple classification and not sequence labeling. For every word in every sentence, the system predicts words as complex if at least X people annotated it as complex. The maximum G-score is 84.54% for X=10% and the F-score is 51.66% for X=25%. This system has information regarding the word and the sentence. However, it is still not sequence classification – it has no information regarding the predictions of the other words in the sentence. Figure 1 presents results in a function of X.

2. Going further input data can be solely words, without the sentence, so that we can aggregate annotations for the same words, but in different sentences. The system describes a word as complex if at least X people annotated it as complex (this system has no information regarding the context of the sentence). The maximum G-score is 85.04% for X from 4% to 5%, and the F-score is 51.71% for X from 26% to 27%. This system has information only about the word. Figure 2 presents results in a function of X.

The results above show that a G-score of 86% can not be exceeded on this data.

#### 4.2.2 Final Submission

The experiments showed a minimally increased score for more advanced classifiers using more features in comparison to the simple one-rule algorithm with one feature. Simple models are usually more difficult to overfit. The complexity of this algorithm is $\mathcal{O}(1)$ for every word using hashing.

The final submission uses DF of Simple English Wikipedia. The scores, as a function of threshold, are presented in Figure 3.

The main submission is optimized for G-score, and its threshold is 147. Words with a DF exceeding this threshold are considered simple, and others are considered complex. A set of simple words contains almost 11 thousand tokens (without sanitization). The size of the model is 78 kilobytes.

The second submission was optimized for F-score and the threshold was 18.

## 5 Results and Discussion

Table 3 shows the top 10 results of the systems on the test data in terms of G-score. The system placed fourth with two other systems.

The best system, SV000gg, ensembles 23 distinct systems using 69 morphological, lexical, semantic, collocation, and nominal features. The system is much more advanced than the one presented in this

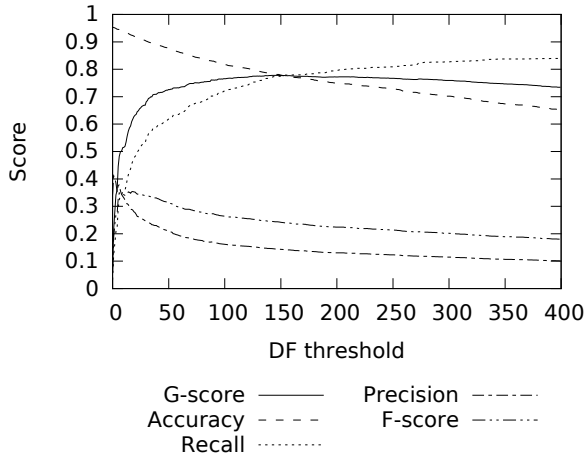


**Figure 3:**

**Table 3:** Top 10 systems in terms of G-score. Additionally, the average scores of all systems and their standard deviations are provided.

| System | Accuracy | Recall | G-score |
|---|---|---|---|
| SV000gg-Soft | 0.779 | 0.769 | 0.774 |
| SV000gg-Hard | 0.761 | 0.787 | 0.773 |
| TALN-WEI | 0.812 | 0.736 | 0.772 |
| UWB-All | 0.803 | 0.734 | 0.767 |
| **PLUJAGH-SEWDF** | 0.795 | 0.741 | 0.767 |
| JUNLP-NaiveBayes | 0.767 | 0.767 | 0.767 |
| HMC-RegressionTree | 0.838 | 0.705 | 0.766 |
| HMC-DecisionTree | 0.846 | 0.698 | 0.765 |
| JUNLP-RandomForest | 0.795 | 0.730 | 0.761 |
| MACSAAR-RFC | 0.825 | 0.694 | 0.754 |
| TALN-SIM | 0.847 | 0.673 | 0.750 |
| MACSAAR-NNC | 0.804 | 0.660 | 0.725 |
| Average | 0.737 | 0.591 | 0.620 |
| Standard deviation | 0.130 | 0.202 | 0.123 |

**Table 4:** Top 3 systems in terms of F-score. Additionally, the average scores of all systems and their standard deviations are provided.

| System | Precision | Recall | F-score |
|---|---|---|---|
| **PLUJAGH-SEWDFF** | 0.289 | 0.453 | 0.353 |
| LTG-System2 | 0.220 | 0.541 | 0.312 |
| LTG-System1 | 0.300 | 0.321 | 0.310 |
| Average | 0.123 | 0.590 | 0.193 |
| Standard deviation | 0.061 | 0.202 | 0.073 |

paper. Its result is higher by almost one percentage point.

The next system in the ranking, TALN-WEI, uses external resources, i.e. WordNet, simple/complex word lists, tools, i.e. part-of-speech tagger, and a dependency parser. A random forest classifier is then trained.

JUNLP-NaiveBayes employs word sense disambiguation and features extracted from an ontology. Also, a random forest classifier is used. Additional word lists are developed, i.e. scientific, geographical, and non-English.

Surprisingly, UWB-ALL is almost the same as the one presented in this article (the English version of Wikipedia is used, not Simple English).

The presented system took first place in terms of F-score. The higher score is probably due to this submission being optimized for F-score with no other teams doing this.

Beating 85% G-score is not possible without more information. It is possible that having the possibility to model every person's knowledge would improve the results. However, this approach needs historic data annotated by a specified user and the predictions would be only relevant for this user.

## References

Marilisa Amoia and Massimo Romanelli. 2012. Sb: mmsystem-using decompositional semantics for 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*, pages 482–486. Association for Computational Linguistics.

Or Biran, Samuel Brody, and Noémie Elhadad. 2011. Putting it simply: a context-aware approach to lexical simplification. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2*, pages 496–501. Association for Computational Linguistics.

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*.

Łukasz Dębowski, Bartosz Broda, Bartłomiej Nitoń, and Edyta Charzyńska. 2015. Jasnopis – a program to compute readability of texts in Polish based on psycholinguistic research. In *Natural Language Processing and Cognitive Science. Proceedings 2015*, pages 51–61.

Sujay Kumar Jauhar and Lucia Specia. 2012. Uow-shef: Simplex–lexical simplicity ranking based on contextual and psycholinguistic features. 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*, pages 477–481. Association for Computational Linguistics.

Anne-Laure Ligozat, Anne Garcia-Fernandez, Cyril Grouin, and Delphine Bernhard. 2012. Annlor: a naïve notation-system for lexical outputs ranking. 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*, pages 487–492. Association for Computational Linguistics.

Gustavo Henrique Paetzold and Lucia Specia. 2015. Lexenstein: A framework for lexical simplification. *ACL-IJCNLP 2015*, 1(1):85–90.

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)*.

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.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Ravi Sinha. 2012. Unt-simprank: Systems for lexical simplification ranking. 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*, pages 493–496. Association for Computational Linguistics.

Aleksander Smywiński-Pohl and Krzysztof Wróbel. 2014. The importance of cross-lingual information for matching Wikipedia with the Cyc ontology. In *9th International Workshop on Ontology Matching*, pages 176–177.

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*, SemEval '12, pages 347–355, Stroudsburg, PA, USA. Association for Computational Linguistics.