LTG at SemEval-2016 Task 11: Complex Word Identification with Classifier Ensembles

Shervin Malmasi[1] **Mark Dras**[1] **Marcos Zampieri**[2,3]
[1]Macquarie University, Sydney, NSW, Australia
[2]Saarland University, Germany
[3]German Research Center for Artificial Intelligence, Germany
{first.last}@mq.edu.au, marcos.zampieri@dfki.de

Abstract

We present the description of the LTG entry in the SemEval-2016 Complex Word Identification (CWI) task, which aimed to develop systems for identifying complex words in English sentences. Our entry focused on the use of contextual language model features and the application of ensemble classification methods. Both of our systems achieved good performance, ranking in 2nd and 3rd place overall in terms of F-Score.

1 Introduction

Complex Word Identification (CWI) is the task of identifying complex words in texts using computational methods (Shardlow, 2013). The task is usually carried out as part of lexical and text simplification systems. Shardlow (2014) considers CWI as the first processing step in lexical simplification pipelines. Complex or difficult words should first be identified so they can be later substituted by simpler ones to improve text readability.

CWI has gained more importance in the last decade as lexical and text simplification systems have been developed or tailored for a number of purposes. They have been applied to make texts more accessible to language learners (Petersen and Ostendorf, 2007); other researchers have explored text simplification strategies targeted at populations with low literacy skills (Aluísio et al., 2008). Finally, another relevant application of text simplification are people with dyslexia (Rello et al., 2013).

The SemEval 2016 Task 11: Complex Word Identification (CWI) provides an interesting opportunity to evaluate methods and approaches for this task. The organizers proposed a binary text classification task in which participants were required label words in English sentences as either complex (1) or simple (0). The task organizers provided participants with a training set containing sentences annotated with this information, followed by an unlabeled test set for evaluation. The assessment of whether words in a sentence are complex or simple was performed by human annotators required to label the data.[1]

2 Data

Based on the information available at the shared task's website[2]: "400 annotators were presented with several sentences and asked to select which words they did not understand their meaning."

The CWI task dataset was divided as follows:

- **Training set:** 2,237 judgments by 20 annotators over 200 sentences. A word is considered complex if at least one of the 20 annotators assigned it as so.
- **Test set:** 88,221 judgments made over 9,000 sentences (1 annotator per sentence).

[1]Here the term *complex* is used as a synonym for difficult. Unlike the Morphology term *complex* (antonym of *simplex*) that defines compound words or words composed of multiple morphs (Adams, 2001).

[2]http://alt.qcri.org/semeval2016/task11/

Proceedings of SemEval-2016, pages 996–1000,
San Diego, California, June 16-17, 2016.

3 Methodology

The primary focus of our team's entry was the use of judgements from different annotators to create training data. We looked at how adjusting the threshold for inter-annotator agreement would affect the results and whether the combination of data created using different threshold values could improve performance.

Initially, the training data released by the organizers was labeled in a way that a word was marked as complex if any annotator judged it so. During the course of the shared task the organizers released additional information about the training data, chiefly the individual judgements of the 20 annotators that were used to derive the final labels for each word.

We attempted to use this data in our system. During development we noted that by increasing this threshold to two, the performance of our system under cross-validation improved by a small amount. Accordingly, we pursued this direction as the main focus of our experiments.

3.1 Classifiers

We utilize a decision tree classifier, which we found to perform better than Support Vector Machine (SVM) and Naïve Bayes classifiers for this data.

3.2 Features

Our core set of features are based on estimating n-gram probabilities using web-scale language models. More specifically, this data was sourced from the Microsoft Web N-Gram Service[3], although we should note that this service has been deprecated and replaced since the shared task.[4] These language models are trained on web-scale corpora collected by Microsoft's Bing search engine from crawling English web pages.

Given a target word w_t, we extract several probability estimates to use as classification features. These estimates, which we describe below, use the target word as well its preceding and following words, as shown in Figure 1.

[3] http://weblm.research.microsoft.com/

[4] It has been replaced by Microsoft's Project Oxford: https://www.projectoxford.ai/weblm

3.2.1 Word Probability

This is an estimate of how likely the target word is to occur in the language model:

$$P(w_t)$$

Rarer words would be assigned lower values and thus this feature can help quantify word frequency for the classifier.

3.2.2 Conditional Probability

We calculate the bigram probability of w_t:

$$P(w_t \mid w_{t-1})$$

Similarly, we estimate the trigram probability:

$$P(w_t \mid w_{t-1}, w_{t-2})$$

These values estimate the likelihood of the target word occurring given the previous one or two words. They can help quantify if the word is being used in a common or less frequent context.

3.2.3 Joint Probability

We also use the following joint probability estimates of the target word and its surrounding words:

$$P(w_{t-1}, w_{t-2}, w_t)$$
$$P(w_{t-1}, w_t)$$
$$P(w_{t-1}, w_t, w_{t+1})$$
$$P(w_t, w_{t+1})$$
$$P(w_t, w_{t+1}, w_{t+2})$$

The intuition underlying the use of all of these n-gram language model features is that the understanding of certain words depends on the context they appear in. A large number of English words are polysemous and their classification, without taking into account the specific sense being use, could lead to misclassifications. This can occur in scenarios where a learner knows the most frequently used sense of a polysemous word, but is confronted with a different sense that they have not encountered before. Additionally, even if a known word is used in an unusual context, it could be a cause of confusion for learners.

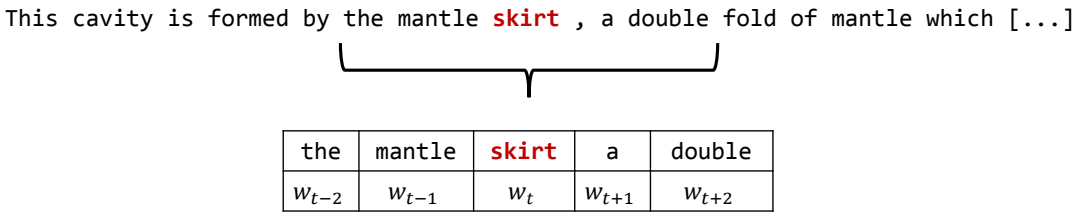

Figure 1: An example of the context extracted for a target word, which is "skirt" in this example.

3.2.4 Word Length

Guided by the intuition that the most frequent words in a language are usually shorter, we use the length of a word as a classification feature.

4 Ensemble Classifiers

Classifier ensembles are a way of combining different classifiers or experts with the goal of improving accuracy through enhanced decision making. They have been applied to a wide range of real-world problems and shown to achieve better results compared to single-classifier methods (Oza and Tumer, 2008). Through aggregating the outputs of multiple classifiers in some way, their outputs are generally considered to be more robust. Ensemble methods continue to receive increasing attention from researchers and remain a focus of much machine learning research (Woźniak et al., 2014; Kuncheva and Rodríguez, 2014).

Such ensemble-based systems often use a parallel architecture, as illustrated in Figure 2, where the classifiers are run independently and their outputs are aggregated using a fusion method. For example, *Bagging* (bootstrap aggregating) is a commonly used method for ensemble generation (Breiman, 1996) that can create multiple base classifiers.

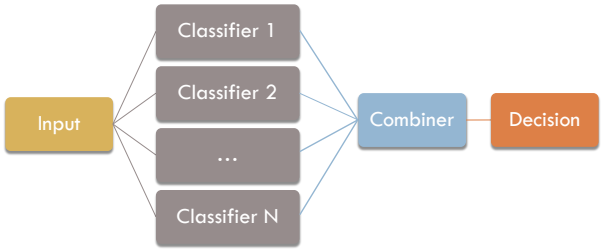

Figure 2: An example of parallel classifier ensemble architecture where N independent classifiers provide predictions which are then fused using an ensemble combination method.

It works by creating multiple bootstrap training sets from the original training data and a separate classifier is trained from each one of these sets. The generated classifiers are said to be diverse because each training set is created by sampling with replacement and contains a random subset of the original data.

Other, more sophisticated, ensemble methods that rely on meta-learning may employ a stacked architecture where the output from a first set of classifiers is fed into a second level meta-classifier and so on.

The first part of creating an ensemble is generating the individual classifiers. Various methods for creating these ensemble elements have been proposed. These involve using different algorithms, parameters or feature types; applying different preprocessing or feature scaling methods and varying (*e.g.* distorting or resampling) the training data.

5 Systems

In this section we describe the two systems we created and entered in the shared task.

5.1 System 1

Our first system was based on decision tree classifier trained on data where the minimum threshold for inter-annotator agreement was set to 3. Given that the testing data was only annotated by a single rater, we did not want to pick a value that was too high, even though this could improve cross-validation performance on the training data.

Additionally, we converted this setup to an ensemble by creating 100 randomized decision tree classifiers by using bagging, which we described earlier. The decisions of these learners were fused via plurality voting to yield the final label for an instance.

Rank	Team	System	Accuracy	Precision	Recall	F-score	G-score
1	PLUJAGH	SEWDFF	0.922	0.289	0.453	0.353	0.608
2	LTG	System2	0.889	0.220	0.541	0.312	0.672
3	LTG	System1	0.933	0.300	0.321	0.310	0.478
4	MAZA	B	0.912	0.243	0.420	0.308	0.575
5	HMC	DecisionTree25	0.846	0.189	0.698	0.298	0.765
6	TALN	RandomForest_SIM.output	0.847	0.186	0.673	0.292	0.750
7	HMC	RegressionTree05	0.838	0.182	0.705	0.290	0.766
8	MACSAAR	RFC	0.825	0.168	0.694	0.270	0.754
9	TALN	RandomForest_WEI.output	0.812	0.164	0.736	0.268	0.772
10	UWB	All	0.803	0.157	0.734	0.258	0.767

Table 1: The top 10 systems in task, ranked by their F-score.

5.2 System 2

For our second system we extended the threshold-based approach to an ensemble of decision trees trained on different data.

We created four individual classifiers, each trained with a different minimum threshold[5] ranging between 1–4. The outputs of these classifiers, a binary prediction, were then combined using a plurality voting combiner. It should also be noted that having an even number of base classifiers also introduces the possibility of ties occurring.

6 Results

The top 10 task submissions, ranked by the F-score, are shown in Table 1 with our systems highlighted. Both of our systems achieved very competitive results, ranking in second and third place overall.

Our second system, an ensemble of classifiers trained on distinct data derived using different levels of inter-rater agreement, performed slightly better than the first system. This could be interpreted as this evidence the second approach is slightly better, and we hypothesize that combining annotations from different combinations of annotators may help the classifier learn reliable models of the phenomenon, since individual annotations (as well as the original combined annotation) were noisy. However, determining this requires further experiments. This is due to the fact that with four classifiers in the ensemble, voting resulted in a tie for some 6% of the testing data. These ties were broken arbitrarily, introducing an element of stochasticity to our results.

In hindsight this does not appear to have been the most intuitive or robust way of dealing with such ties since the distribution of classes is not balanced. In fact, this distribution is highly skewed, as we discuss in the next section.

[5]Setting the threshold to 1 is equivalent to using the original training data.

6.1 Conclusion

We developed two ensemble-based systems for this task, both of which achieved competitive results in the final rankings. Our results indicate that the use of contextual features, as well as language models, are promising for this task.

Analysis of the gold standard labels release after the task shows that only 4.7% of the 88k samples belonged to the positive class. This is a very highly skewed distribution that can make it hard to train effective classifiers. It also means that accuracy cannot be used as the sole evaluation metric here; a balanced measure of precision and recall like the F-score is required. Alternatively, the balanced accuracy measure (Brodersen et al., 2010) could also be used. Such a high data imbalance can result in training classifiers that are biased towards the majority class. This bias can be more problematic if the distribution of classes is different in the test set. Accordingly, future work in this area could look at the use of methods for dealing with unbalanced datasets (He and Garcia, 2009). The application of such methods, in conjunction with ensembles, could potentially result in greater performance.

Future work could attempt to integrate additional language resources for this task. Analyzing the text produced by learners could provide insight into the limitations of learner vocabulary. Learner corpora, widely used in the task of Native Language Identification (Malmasi and Dras, 2014; Malmasi and Dras, 2015) could be useful here.

Acknowledgments

We would like to thank the CWI task organizers for managing the organization of this event. We also thank the anonymous reviewers for their insightful comments.

References

Valerie Adams. 2001. *Complex words in English*. Routledge.

Sandra M Aluísio, Lucia Specia, Thiago AS Pardo, Erick G Maziero, and Renata PM Fortes. 2008. Towards brazilian portuguese automatic text simplification systems. In *Proceedings of DocEng*.

Leo Breiman. 1996. Bagging predictors. In *Machine Learning*, pages 123–140.

Kay H Brodersen, Cheng Soon Ong, Klaas E Stephan, and Joachim M Buhmann. 2010. The balanced accuracy and its posterior distribution. In *Proceedings of ICPR*.

Haibo He and Edwardo A Garcia. 2009. Learning from Imbalanced Data. *IEEE Transactions on Knowledge and Data Engineering,*, 21(9):1263–1284.

Ludmila I Kuncheva and Juan J Rodríguez. 2014. A weighted voting framework for classifiers ensembles. *Knowledge and Information Systems*, 38(2):259–275.

Shervin Malmasi and Mark Dras. 2014. Chinese Native Language Identification. In *Proceedings of EACL*.

Shervin Malmasi and Mark Dras. 2015. Multilingual Native Language Identification. In *Natural Language Engineering*.

Nikunj C Oza and Kagan Tumer. 2008. Classifier ensembles: Select real-world applications. *Information Fusion*, 9(1):4–20.

Sarah E Petersen and Mari Ostendorf. 2007. Text simplification for language learners: a corpus analysis. In *Proceedings of SLaTE*.

Luz Rello, Ricardo Baeza-Yates, Stefan Bott, and Horacio Saggion. 2013. Simplify or help?: text simplification strategies for people with dyslexia. In *Proceedings of W4A*.

Matthew Shardlow. 2013. A comparison of techniques to automatically identify complex words. In *Proceedings of the ACL Student Research Workshop*.

Matthew Shardlow. 2014. A survey of automated text simplification. *International Journal of Advanced Computer Science and Applications (IJACSA)*, (Special Issue on Natural Language Processing).

Michał Woźniak, Manuel Graña, and Emilio Corchado. 2014. A survey of multiple classifier systems as hybrid systems. *Information Fusion*, 16:3–17.