Complex Word Identification Report

Xin Sun

Abstract

In this report, we describe how to develop a system for the of Complex Word Identification task. We will try to choose the best model and extract some new feature which could be useful. There are some main parts in this report: the detail and importance of the task; the description of the baseline system; the description of method to implement the improved system and the motivation. The report also concentrates on how the system performs on the prediction set, and the learning curves for the trainable systems. At the end of the report, there is a conclusion about the useful points from the system for the future work.

1 Introduction

The aim of the text simplification system is to help different kinds of readers have better comprehension of the text, especially for the non-native speakers, native speakers with low education. The first key task in simplification system is predicting which words might be complex for target people. This task is known as complex word identification (CWI). This task has got lots of attention and development because of it plays a role key in the Nature Language Processing area.

The goal of the complex word identification (CWI) is to predict which is the difficult word for the non-native speakers in the sentence. In order to achieve this task, there are four train datasets in the complex word identification (CWI):

- English monolingual CWI;
- German monolingual CWI;
- Spanish monolingual CWI;
- Multilingual CWI with a French test set.

In this project, we will concentrate on the binary classification task for the English monolingual CWI and Spanish monolingual CWI. Firstly, we build the model according to the trainset. Then, we improve the model by the development set. At last, we can test our improved system on the test set.

In this task, an accurate system is very important because it can help determine the really complex word. Then some appropriate measures can be applied to these complex words, which could help non-native speakers have better comprehension of the text.

2 Baseline system

To build the Baseline system, the first step is to choose the suitable model and the features.

Model: Logistic Regression model

We choose the logistic regression model in the baseline system. Logistic Regression model have many advantages. Logistic Regression model can be simple to implement. Its computation is very small, fast, and low in storage. So it can be a good model for the baseline system.

Features: length of the character and length of the tokens

Then we need consider which features can be used in the baseline system. The key of the task is to predict whether the target word is the complex word or not in the given sentence. Therefore, some basic feature of the target word could to applied to the baseline system. We choose two features in the baseline system: length of the character and length of the tokens.

Length of the character: The word with long character tends to be complex, so the length of the character might be a useful feature.

Length of the tokens: the phrase tends to be more difficult to understand than the single word, so the length of tokens might be a useful feature.

After choosing the model and the features, we build the baseline system. In our case, the F1 score could be calculated to evaluate the system. We can use development set and test set to evaluate the baseline system.

Experiments on development set:

```
english: 27299 training - 3328 dev
macro-F1: 0.69

spanish: 13750 training - 1622 dev
macro-F1: 0.72
```

Figure 1: evaluate the baseline system on the development set

As we can see from the Figure 1, the baseline system gets 0.69 macro-F1 for the English dataset and 0.72 macro-F1 for the Spanish dataset. These scores are pretty good which means that the idea is as expected and the baseline system does predict well with the development set.

Then we evaluate the baseline on the test set:

```
english: 27299 training - 4252 test
macro-F1: 0.68

spanish: 13750 training - 2233 test
macro-F1: 0.70
```

Figure 2: evaluate the baseline system on the test set

As the Figure 2 shows, the baseline system gets 0.68 macro-F1 for the English dataset and 0.70 macro-F1 for the Spanish dataset when evaluate on the test set, which means that the baseline system could also work and predict well on the test set. However, there was a slight drop in the macro-F1 score. It means that the baseline system performs a bit worse on the test set than on the development set. The reason might be that the size of the test set is larger than the development set. The logistic regression model might have a bit lower accuracy when handle the large size of data. On the other hand, although the F1 score seems not bad, there is also a potential to improve the baseline system to get better predictions by choosing more suitable models and extracting better features combination. These are the limitation of this model and the baseline system. We need to development an improved system to get better predictions.

3 Improved system

As mentioned above, the baseline system performs a bit worse on the test set and there is a possibility to achieve a better system. These are the motivations for us to get an improved system.

3.1 Model description

The key to build the improved system is to choose the suitable model and feature features combination. As we know, the model in the baseline system is logistic regression model, which is easy to implement and calculates fast. The model also has some disadvantages such as low accuracy. It is a key step to choose the model for the improved system. In this project, we would choose one best model from these follow 5 models:

Logistic Regression

Merit: good anti-noise ability and stable performance; easy to implement; could avoid overfitting; could process data with high dimensions

Demerit: the parameters are more complex; calculate slowly

Random Forest Classifier

Merit: good anti-noise ability and stable performance; easy to implement; could avoid overfitting; could process data with high dimensions

Demerit: the parameters are more complex; calculate slowly

SVM

Merit: wide range of application; low generalization error; easy to explain; low computational complexity.

Demerit: sensitive to the selection of parameters and kernel functions

K Neighbors Classifier

Merit: high accuracy; not sensitive to outliers

Demerit: Large computation; problem of sample imbalance; requires a lot of memory

Gradient Boosting Classifier

Merit: high accuracy

Demerit: sensitive to the selection of parameters.

3.2 Feature description

In the baseline system, we only use two features-length of character and length of tokens. According to the result of experiments on the development set and test set for basement system, we can know that these two features really help the system predict whether the target word is complex or not. There are also some other features wo can use to get an improved system. At First, we can propose some features, then, experiments with these features on development set to make sure which of them could help predict better. In this project, we would propose the follow 6 features and find the best feature combination based on the baseline system.

Feature1: word frequency

If a word is frequent in text, it tends to mean that the word is not complex. So the word frequency might be a useful feature in the improved system.

Feature2: POS tagging

The word with a common POS tends to be more easy to understand. So the POS tagging might be a useful feature in the improved system.

Feature3: the number of synonyms

The word with more synonyms tends to be easy to understand. So the number of synonyms might be a useful feature in the improved system.

Feature4: the length of character per token

If the length of character per token is large, the word is always complex. So the length of character per token might be a useful feature in the improved system.

Feature5: the bigram based on the word

The complexity of words may be related to the bigram based on the word.

Feature6: the bigram based on the character

The complexity of words may be related to the bigram based on the character in the word.

After proposing these 6 features, we would experiment them on the development set based on the five models. The method is adding one feature to the feature combination each time. The original feature combination is the baseline system features: length of the character and length of the tokens. The figures 1-5 represents the feature1-5. The feature 0 means original feature combination.

3.3 experiments on development set

We calculate the F1 score for different features combinations and model on the development set. The figure 3 and figure 4 shows the results.

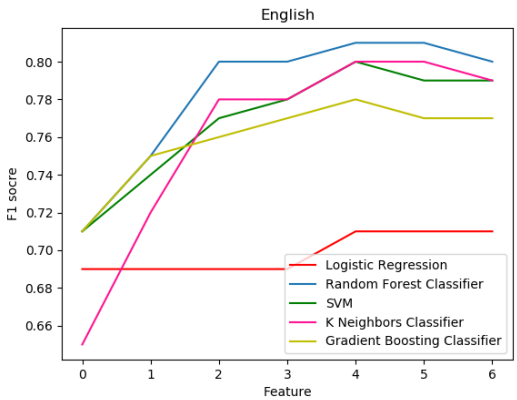


Figure 3: English

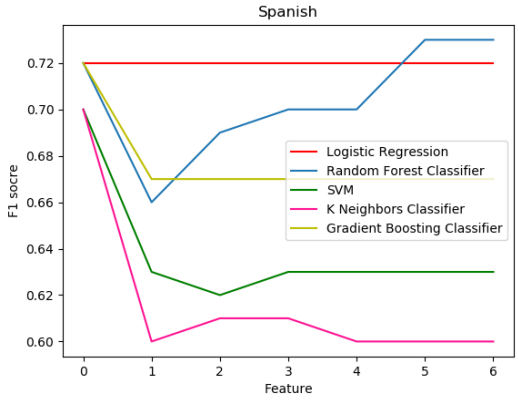


Figure 4: Spanish

The figure 3 shows the F1 score of different models experiments on the English Development set. Firstly, concentrate on the comparison of different models. We can know that the Random Forest Classifier model gets the highest F1 score which means the Random Forest Classifier model could be the better model in these 5 models for the improved system. Then turn to the feature combinations. We can see the F1 score increase with the addition of the feature, but when add the last feature adds, there is a drop of F1 score in the figure3. This means the feature1-5 really help predict, but the last feature seems not helpful for the improved system. The last feature (the bigram based on the character) has an adverse effect on the performance of the improved system. And there are also some other models predict worse when add the fifth feature to the combination.

In conclusion, the fifth and the sixth features might be useless and we choose the feature1-4 and the two baseline feature as our features combination. The system's model is the Random Forest Classifier model.

3.4 evaluate on the test set

After we get the improved system, we evaluate the improved system on the test set.

```
english: 27299 training - 4252 test
macro-F1: 0.82

spanish: 13750 training - 2233 test
macro-F1: 0.71
```

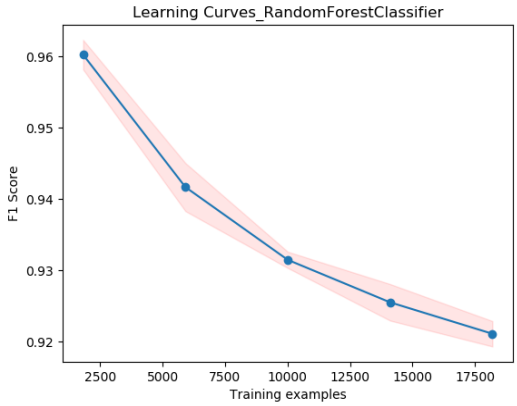
Figure 5: evaluate on the test set

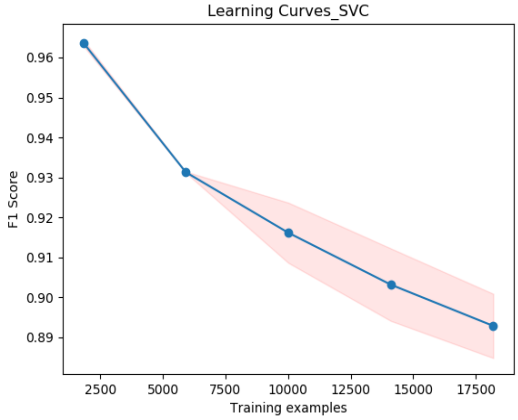
We can see from the result that the improved system really helps predict the complex word.

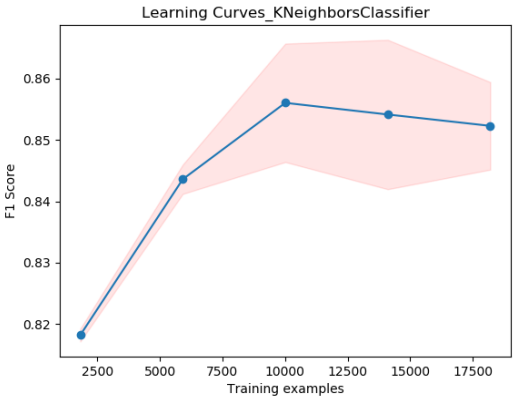
We can loop our predictions of improved system and baseline system, and then compare the prediction to the label. We can find which the target word the system predicts right and which the word

the system predicts wrong. An example shows why the improved system is better: In a sentence "Syrian troops shelled a rebel-held town on Monday, sparking intense clashes that sent bloodied victims flooding into hospitals and clinics, activists said.", and "hospital" is the target word. The baseline predicts the "hospital" as complex word, which is wrong. The reason might be that the length of the target word is long and the baseline only predicts according to the length. But the improve system predicts right because the improve system could also identify the frequency and the POS of the word, it really helps predict the "hospital" as easy word.

4. Plot learning curves







As we can see from the learning curves, the Random Forest Classifier and SVM better than others when less training data is available.

5. future work

An example shows the improved system predicted wrong: the sentence is "The Britain-based Syrian Observatory for Human Rights and the activist network called the Local Coordination Committees said the latest shelling of Rastan started", and the target word is "called". The improved the "called" as complex word. The reason might be that the "called" is the past tenses which is rare in the train set. The method to solve this problem is to optimize the system so that the system can identify the prefix, stem and suffix of the target word. In that way, the system could predict more accurate

6. conclusion.

In my project, I build a baseline system and improved system for the task of CWI. In the baseline system, the model is Logistic Regression and the feature is length of the character and length of the tokens. The system got 0.68 F1 score in the English Test set and 0.70 F1 score in the Spanish Test set. Then I build an improved system. After the experiments on the development set, I found the Random Forest Classifier and a best features combination. So the improved system uses these model and features. After comparing the baseline system predictions and the improved system predictions with the label, we could learn that the key to this task is to find the effective features which could represent the complex word such as the word frequency, the prefix, stem and suffix of the target word and so on.

The code could be found at:
https://github.com/leoXinsun/Xin_NLP_Project_CWI