Tetum Sentiment Analysis

Thien Hai Nguyen
nhthien@jaist.ac.jp

December 31, 2015

1 Build from scratch

We can build the model by ourselves using nltk and scikit-learn libraries. We can choose supervised or unsupervised approach. As you know, Tetum is a scare language, there is no syntactic parsing, dependency parsing, or sentiment resources (Wordnet or Sentiwordnet) for this language. Therefore, we cannot utilize complicated methods from English language. However, we can build a model which is language independent. In other words, we do not consider syntactic, semantic features.

1.1 Supervised Approach

1.1.1 Approach 1

Using SVM to classify each documents to positive, neutral, negative. Features are bag-of-words, n-grams.

- **Pros**: It's very fast to build the model. For online learning, we can use incremental training algorithm in scikit-learn library (SGDClassifer).
- **Cons**: We need to manually annotate sentiment labels for each documents by ourselves (need a large annotated documents for good performance). However, to overcome the lack of training data for Tetum, we can use translation method to generate Tetum traning data from English data. Of course, the accuracy of translation will affect the final result. Another method to overcome the lack of training data is to use co-training method [1].

1.1.2 Approach 2

Joint training with English sentiment analysis. ⇒ Need parallel corpus ⇒ Using machine translation.

- **Pros**: Compared with Approach 1, the accuracy could be improved
- **Cons**: The model could be complicated and take times to build. The translation accuracy will affect the final result. In addition, we have to design the incremental training method (online learning) for this model.

[1] https://en.wikipedia.org/wiki/Co-training

1.2 Unsupervised Approach

1.2.1 Approach 3

Use Tetum sentiment resources (opinion word lists) to calculate the sentiment score for each document. Because there is no such resource, we can generate them using a bilingual dictionary. (Utilize the translation systems from Google, Bing, Babylon or your translation system).

- **Pros**: It's very fast to build the model.
- **Cons**: The accuracy translation affects the final result.

2 Google Prediction API (Approach 4)

We just extract features (bag-of-words, n-grams) and use Google prediction API. We don't know what kind of algorithms will be used. The algorithms are black-boxes in here, usually linear models. Only supervised algorithms can be used to do sentiment analysis. $\Rightarrow$ Fast, easy, similar to Approach 1.

- **Pros**: It's very fast to build the model. Google prediction API supports incremental training for online learning.
- **Cons**: Similar to Approach 1. In addition, because the algorithm is black box, we can not modify the algorithm

3 Conclusion

We can use Approach 1, Approach 3 or Approach 4 to build the model quickly. Which method will produce the better results? It could be Approach 1 or Approach 4 (Need to empirical evaluation from test set).