Project 4: Sentiment Analysis

Array3

Hongfei Li | Meng Du | Yiqin Wu

Data Pre-processing

A) Punctuations.

For each review, we remove HTML tags and punctuations and keep numbers.

B) Stop words.

Most documents contain some general words (i.e. stop words) that appear very often but do not carry much meanings, such as it, the, as, on. If we do not ignore these stop words when classifying documents, our prediction could be biased as a result of those words with high occurrence getting high weight. Thus we remove stop words during tokenizing process.

C) TF-IDF.

TF–IDF, short for term frequency–inverse document frequency, reflects how important a word is to a document. In IMDB dataset, movie terminologies appear frequently in these reviews. Some terms appear often but is not related to people's attitude while other terms appear less but are important to sentiment analysis. The basic idea here is to filter out common terms, and assign high weights to a term that has a high frequency in the given document and a low document frequency in the whole collection of documents.

D) N-gram model.

An n-gram is a contiguous sequence of n items from a given sequence of text or speech. With n-gram model, we keep information about sequence of tokens. In our model we choose bi-grams as features.

Model : logistic

Finally we choose logistics model with l2 regularization to predict reviews' sentiment as this model needs least running time and returns highest accuracy.

Accuracy

Submission_decimal.csv:

In this submission file, we output the probability of a document being positive or negative, which are all decimals. In other words, The prediction is between [0,1]. The kaggle score for accuracy is 0.95163.

Submission_binary.csv:

In this submission file, we output either 0 or 1 as indicators of a document being negative or positive. The accuracy is 0.88288.

Model –tried but eventually discarded

A) Recurrent Neural Networks

We tried to apply recurrent neural networks on this dataset. It takes 20 minutes to train on GPU, which is much longer than other simple methods. However, the long running time does not pay and the accuracy is around 0.80. The reason is that this dataset is too small for neural networks to have any advantages over simple methods. Overfitting is a serious issue and causes the relatively low accuracy. Eventually we gave up neural networks due to its long running time and not-so-good accuracy.

B) Ensemble method with Linear SVM, Logistics and Naïve Bayes

By sampling from train data sets respectively for each model and get average predictions for final classification, the combination model returns accuracy almost equal to logistics model, with fluctuation of

0.02. This model needs more running time compared with logistics one. Thus, we choose logistics one as final model.

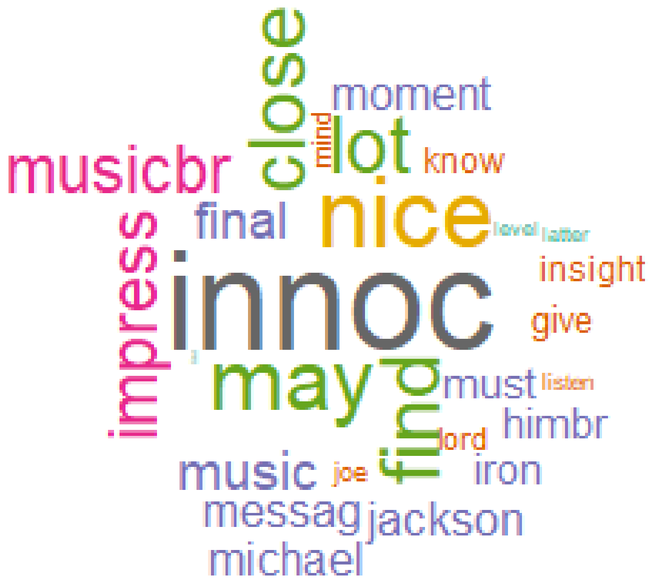
Visualization

A) Word Cloud for each document

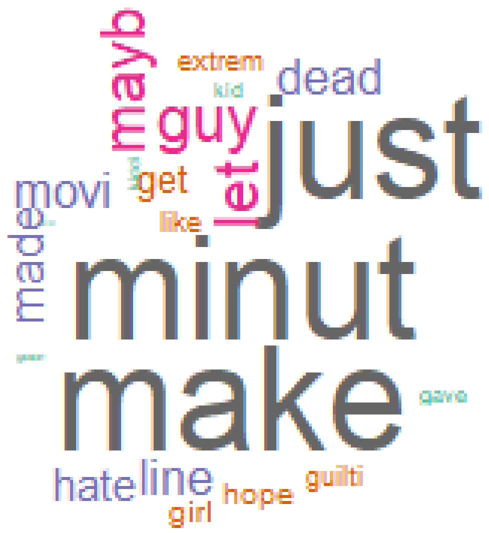
We visualize each test document by two Word Cloud figures. One is for positive words in the given document and the other is for negative ones. We chose the most important positive or negative words to visualize. The importance is measured by the word frequency and how strong the attitude it reflects.

For example:

Positive Words in a document:



Negative Words in a document:



B) Histogram for all predictions

We use histogram to show the prediction results. In other words, how positive or negative these documents are.

Acknowledgments

Thanks to sweezyjeezy's original idea of using logistics model, which is from a topic on the Kaggle forum: https://www.kaggle.com/c/word2vec-nlp-tutorial/discussion/11261