April. 20, 2018

CIS 668/IST 664 - Natural Language Processing

Analyzing the change of public sentiments on Facebook scandal with Reddit data

Professor Lu Xiao

Name
Removed On
Purpose

Contents

1. Abstract

Reddit today has become one of the most popular bulletin board systems in the United States. People are enthusiastic about writing, posting and commenting in Reddit to let others know their opinions and sentiments on some specific social issues, sports games or daily lives. The aim of the project is to experiment with machine learning algorithms to compare sentiment scores and stock price of Facebook during the data leak scandal. When researchers analyze the stock prices, they often overlook Reddit as a data source. The reason is that, although there is large data in Reddit, the majority of them are unstructured text data and hard to be analyzed without processing. We use people's comments in Reddit about Facebook as our corpus and want to see if there is some relationship between public opinions and stock prices. If there is, how public opinions make influences on stock price is also one of our research topics.

2. Introduction

Online communication occupies the irreplaceable position in the modern public discussion and its influences are ever growing. Texts on internet are traces of our everyday conversation and hence, valuable materials for researches. Reddit is one of the prominent online platforms where people exchange their thoughts and affect on them each other. The number of unique visitors to the site is close to 1700 million as of March 2018[1]. In 2017, users generated more than 900 million comments and 12 billion upvotes (vote to contents such as comments).[2] As the service puts itself, Reddit is indeed "the front page of the internet" to many people.

These features make texts collected from the platform ideal to perform opinion mining and sentiment analysis. People can initiate a thread about almost any topics on this platform. Every expression on the site is evaluated through the upvote system. Users and researchers can easily identify which posts or comments got endorsement or votes from audiences and the information presents the agreement of public on the idea.

Among numberless such topics, Facebook-Cambridge Analytica data breach scandal is one of the most attention-drawing and controversial subjects in a past month. Facebook, the biggest social network service(SNS) handed over personally identifiable information of up to 87 million users to the social media strategy consulting firm, Cambridge Analytica from 2014 to 2015.[3] The information was used for political campaign to influence voter opinion and then-president candidate Donald Trump camp was one of the clients who hired consultants of the firm and enjoyed the outcome of exploiting the data. The popularity of Facebook, seriousness of the problem, unmeasurable scale of the similar cases and the dire consequences of the breach made the scandal gigantic issue. Obviously, Reddit users exchanged various opinions on the issue from March 17th when the scandal was first reported by several news outlets including New York Times[4] to the recent. We performed opinion mining and sentiment analysis using Reddit data on this issue. Then, we compared the result with the stock price of Facebook. We wanted to identify similarity or dissimilarity between the trends of two data in time and tried to explain the reason.

Three elements make this research valuable, interesting and feasible. First, the topic holds huge political and societal meaning because the social media which stores massive amount of personal data has potential to be used a tool to alter public opinion and the power has kept growing. Second, the comparison of sentiment and stock price can depict the correlation of the public opinion which is represented by Reddit sentiment score and market's reaction to the scandal represented by the price. At the same time, it also can be a tool to verify the reliability of our sentiment analysis model.

Finally, the mentions in Reddit on the issue was big enough to develop, test and improve our sentiment analysis model.

3. Research

Since social media is playing an important role in our life and everyone is sharing their opinions on it, it's becoming a field of interest for researchers to do sentiment analysis on people's online posts and comments.

When we are choosing the training sets, we found there are several labeled corpuses, including the movie review, rotten tomatoes movie reviews, twitter data.

Jean Y. Wu and Yuanyuan Pao[5] used the rotten tomatoes movie review as their corpus to predict the sentiment for the upcoming reviews. The corpus is labeled with the sentiment score scaling from 1 – 5(weak to strong).

Alexander Pak[6]used the twitter corpus to analyze twitter data. In their essay, they took all the emoticons into consideration.

However, since we think that it's better to have negative scores and positive scores for negative sentiments and positive sentiments, we are not going to use the 1-5 scaled score. Moreover, there are some voting system in the Reddit which is that you can support someone's post by clicking "up" and disagree with someone's post with clicking "down". Thus, we believe if we use both positive and negative numbers, we can develop a formula for a new customized sentiment score. Especially for the negative numbers, if more people disagree with a negative comment, it means that people actually hold a positive attitude towards the original topic, then we'll use negative number to multiply "downvote" which is also a negative number, it makes more sense. Also, there is no emoticons in our data, so we didn't use the twitter corpus. And we finally decided to choose movie review as our training set.

When we are deciding which classifier to choose, we found in other research papers that there are different models for classifiers to train our corpus.

One researcher Mishne[7]used emoticons in LiveJournal posts to train a mood classifier at the document level. He used support vector machine (SVM) as the classifier and identified the intensity of the community mood.

Mao and Lebanon[8]trained conditional random field (CRF) classifiers on sequential sentiments with a movie review dataset.

According to the article "Emotion classification using web blog corpora" (Yang et al., 2007)[9], the authors focus on blog corpus for sentiment analysis and use emotion icons assigned to blog posts as indicators of users' mood. They used SVM and CRF to classify sentiments at the sentence level and then applied the classifiers to the document level.

Lee, Kathy, et al[10]classified Twitter Trending Topics into 18 general categories such as sports, politics, technology, etc. They construct word vectors with trending topic definition and tweets, and classify the topics with using a Naive Bayes Multinomial classifier.

We tried SVM and CRF models and found that those models are not accurate enough for our dataset. So finally we used Multinomial NB Classifier as our classifier.

4. Data Questions

(1) What is the change of public sentiment in Reddit during the Facebook data leak issue? As we know, Facebook data leak scandal was exposed in 17th March. We want to know how the peoples' opinions changed from before the exposure to one month after the

scandal. Reddit is a very popular BBS in the US, from which, we can have a rough vision of public opinions on some specific social issues. One of objectives of this final project is to draw a line chart to show

(2) What is the difference of public sentiment in Reddit and the news from official agencies?
To answer this question, we want to know more about the similarities and differences between public opinion and news. Furthermore, that will be even better if we can analyze the relationship between this two and understand how news from official agencies make influences on public opinions.

(3) What is the relationship between public sentiment on Facebook in Reddit and the stock price of Facebook?
If we can visualize the public sentiment on Facebook accurately, we can compare the line chart of public sentiment and the line chart of Facebook stock price to see if there are some interesting findings. What's more, if the relationship between public sentiment and stock price is strong, we can even make predict of stock price of Facebook based on the comments in Reddit.

5. Methodology

5.1 Data Extraction

Initially, we checked reddit search engine website: https://elasticsearch.pushshift.io. The search strategy we used is "q=(zuckerberg facebook) AND num_comments: > 500&sort = created_utc : desc&size=800". To make it clear, the information we want to retrieve is those posts that titles cover Zuckerberg or Facebook and numbers of comments are more than 500. After acquiring those URLs, we scraped the title, comments and points of every post from the official API of Reddit.

5.2 Data Cleaning and Preparation

After scraping information from Reddit API, we deleted those posts that have nothing to do with Facebook and keep the only one post with the most comments per day as the experiment data.

5.3 Data Modeling

In this modeling part, we first used SentiWordNet to analyze the sentiment polarity score, then use a trained Multinomial Naive Bayes classifier to predict the sentiment probability distribution, as shown in figure.1

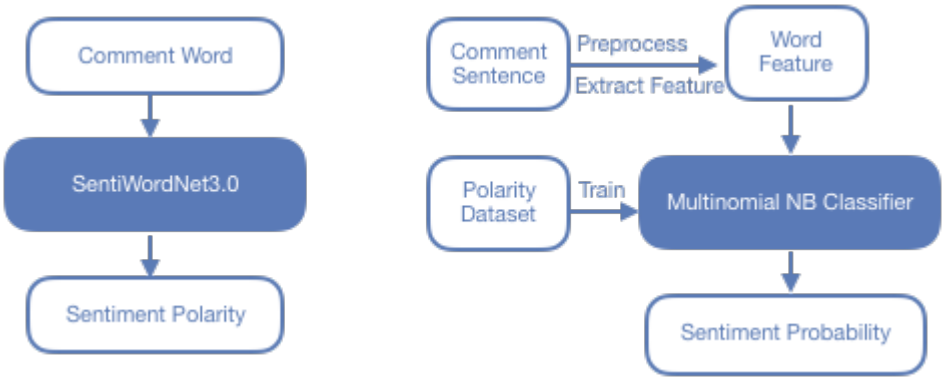

Figure.1 Different Models Based on Sentiment Polarity and Probability

(1) Model1 based on SentiWordNet3.0
For the modeling1 using SentiWordNet3.0 to calculate polarity score, first we tokenized word from comments and used formula(1) to calculate the wordScore of each

word and then combined them together into sentence. For the notation, pos is noted for the positive score of this word, and neg is noted for the negative score.

$$wordScore = \sum_{1}^{n} \frac{pos - neg}{sentiScore}$$

(2) The result of model1 is included in the following. We also manually labeled 50 Reddit comments to validate and evaluate our model. The trend of polarity seems matching with our exception but it really has some obviously wrong prediction about the positive and negative. So after deep consideration and discussion, we decided to implement a classifier to predict the sentiment probability.

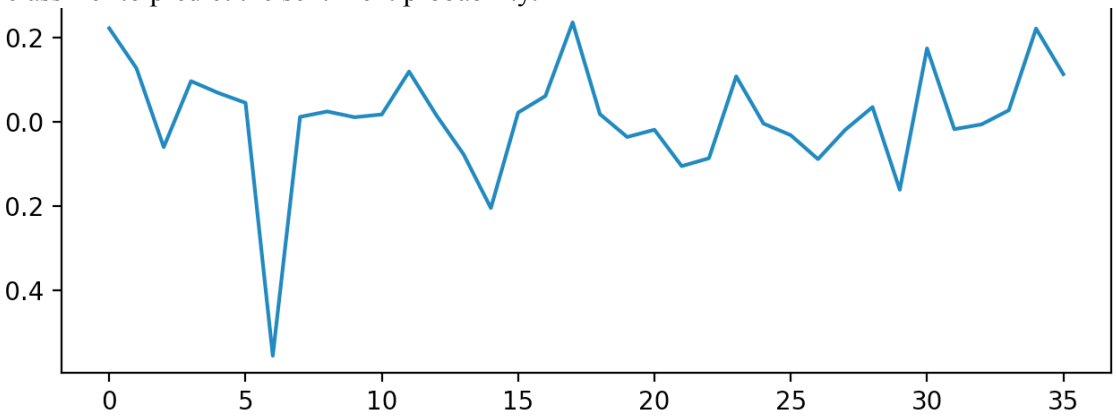

Figure.1 The sentiment polarity scores line chart

(3) Model2 based on Multinominal NB Classifier

a. Training Corpus:

For the training dataset, we decided to use sentiment polarity labeled dataset - Movie Review Data. It contains 1000 positive and 1000 negative processed reviews. Introduced in Pang/Lee ACL 2004. Released June 2004.

b. Feature Extraction and Selection:

In the feature engineering, the our method is based on tf-idf model. First we filtered the words by stopwords, Negation word and then extract the feature of these words in comment level.

For the Tf-idf model, term frequency (tf) is basically the output of the BoW model. For a specific comment, it determines how important a word is by looking at how frequently it appears in the comment. Term frequency measures the local importance of the word. If a word appears a lot of times, then the word must be important.

And for the Feature Selection part, we used the chi2 method to select top 2000 important features for the training, while chi2 is Chi-squared stats of non-negative features for classification tasks. We finished all the training part in the pipeline, which is provided by sklearn package.

c. Classifier Selection:

We tried to training different classifiers and their accuracy of them is shown in figure.2. We also used K-fold Cross Validation to here. So based on the final accuracy shown in table.1, we decided to use the best classifier - Multinominal NB Classifier.

Classifiers	Accuracy
NaiveBayes Classifier	72.5%
Bernoulli NaiveBayes	71.0%
Multinomial NaiveBayes	76.2%
LogisticRegression	70.4%
SGDClassifier	68.6%
LinearSVC	73.1%

Table1. Accuracy of different classifiers

d. Classifier Testing:

After training part is finished, we used the classifier to analyze our test data. We also proposed a Formula.2 to compute comment score.

$$\begin{cases} s = \left(Prob_{pos} - 0.5\right) * 2 \\ ComScore = \sum_{1}^{n} \frac{p_i * s_i}{\sum_1^n p_j} \end{cases}$$

Noted that prob is the probability of positive prediction of our classifier, which ranges from 0 to 1, and we simply use it minus 0.5 and multiple 2 to make it in the range of -1 to 1. pi represents the points of the comments, which is the number of upvote minus the number of downvote. The testing result is attached in the later part.

5.4 Data Visualization

In data visualization part, we used several tools to draw different graphs. Firstly, we use Excel to draw a line chart. One line in this chart represents the users' sentiment scores on Facebook scandal. The other line represents the stock price of Facebook.

Secondly, we drew a word cloud which includes the most common words in users' comments on Facebook scandal to show what people talked about commonly.

Finally, we drew another line chart that shows the relationship between stock price and number of comments. The reason why we want to include this graph is that we can analyze how people's concern extent affects the stock price.

5.5 Validation of the modeled data

To validate our model No.1 data, we manually evaluate 50 comments and give a sentiment score for each comment. We calculate the average of our manual points and compared these points with sentimental scores calculated by machine to see the gap.

6. Result

Figure 1 sums up the whole Reddit data set we aggregated through the data extraction. As explained before, we limited the dataset to one post for each day which has the most comments of the day and its top-level comments. Top-level comment means a comment which attached directly to the post. Similar to many other online board, each comment can have sub-comments (we may be able to call them as second-level or third-level comments matching the depth of hierarchy from the post) and they consist a thread. Each number of comments is drawn as pink bars and the left axis represents the

scale.

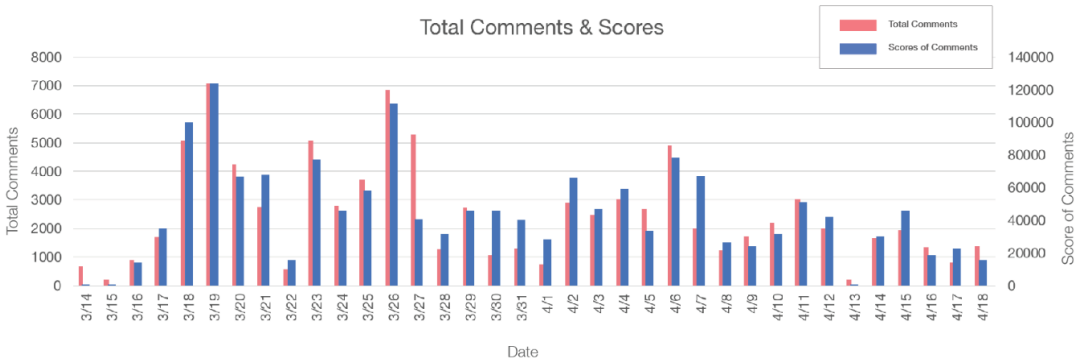

Figure 2 Total comments and score of the post of each day

The other bar(blue) represents the score of each post. On Reddit, user can give a vote to each post or comment (called 'upvote') and one vote signifies one point of a 'score' of each post or comment. That means higher score represents more endorsement to the post from the audiences.

As we can see, two days after the break of the scandal (March 17th) recorded the highest amount of comments as well as the score. The result makes sense since when an incident burst, it takes time to disseminate the information through public.

Another high peak in this graph is on March 26th. The title of the post is 'Facebook has lost $100 billion in 10 days – and now advertisers are pulling out.' It's a link to the news article of a news agency, Reuter with same title of the post. The punishing consequence of the data breach which the article depicted could have drawn huge attention of public.

The overall shape of this graph demonstrates the trend of declining interest of the public on this issue. One interesting finding of this data is the one on April 10th. The day can be noted as the most important day of this scandal besides the beginning day because Mark Zuckerberg, the founder and CEO of the Facebook testified at the Congress on the day. But, different to the major media's attention, the public on Reddit didn't show much interest on the testimony. However, the amount of the data and score did begin to rise after the hearing.

By applying our model to these data, we could get the following sentiment probability scores of each day. We matched the chart with the chart of close stock price of Facebook, Inc. on each day. The range of the sentiment possibility score from -1 to 1. All the scores were below 0, which means sentiment toward Facebook and Zuckerberg were negative all the time but the degree was different by each date.

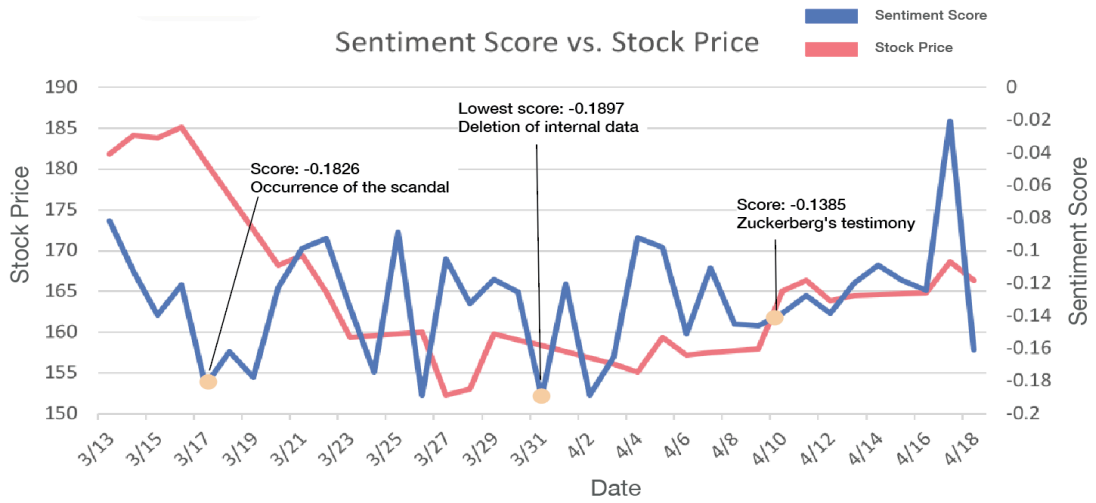

Figure 3 Sentiment score vs. Stock price

We can say the two charts are somewhat similar from a broad perspective. Both the stock price and the sentiment possibility score began to drop down around the point of scandal beginning on March 17th. Although the sentiment score shows much sharp fluctuation, both of them went through low status and began to rise around a month after the occurrence of the event.

Before starting the regression analysis of the two data, we closely examined the data of some critical or odd points. The first odd points are sudden rise of the sentiment score on March 22nd and 23rd. The topics of the both dates were same but in different sub-category of Reddit: “The CLOUD Act would let cops get our data directly from big tech companies like Facebook without needing a warrant. Congress just snuck it into the must-pass omnibus package.” After looking through the comments on the day, we could find the possible explanation that substantial amount of them were about how to protect your data rather than blaming the big tech companies or representatives who were related to the act.

March 31st was the day when the sentiment score recorded the lowest mark(-0.18971). The topic of the post on the day was “Facebook Employees Are Reportedly Deleting Controversial Internal Messages,” with a link to Fortune magazine article with the same title. While people’s anger to the company for treating their personal data poorly kept growing, the report of possible deletion of internal memo infuriated the public.

Possibility score on the date of Zuckerberg’s testimony didn’t show any significant difference compared the day before and after. This somewhat apathetic result is consistent with figure 2. Lastly, the final day of the subject data showed a heavy fall. The title was “Facebook is a tyranny – and our government isn’t built to stop it. America’s founders didn’t envision the power of the corporation. We need a new structure for self-governance that can counter.” Passing a month of the scandal occurrence, while the interest of the public is gradually diminishing, authors of the contents demonstrated a tendency to use strong words to draw attention. The sudden nosedive of sentiment score reflects such tendency.

7. Conclusion and Further Research

We can conclude that, our final sentimental model basically follows the same trend of stock price. However, from the validation that we did, the accuracy of sentiment score is unsatisfied. From our error analysis, the drawbacks of our sentimental model are that: 1) The sample data scale is not big enough. 2) Reddit has many sarcasms in comments, so it's very hard for our model to distinguish positive comments and negative comments. 3) Training dataset for our machine learning is about movie reviews, which leads to the awkward situation that machine don't have enough corpus and training data to recognize the actual feature in sentences and documents in Reddit.

Through the course of this study we identified areas to explore for the further research regarding the classification of the sentiment. A noticeable part of the area is how we can make the model reckon with the context of the posts or the comments. As the result showed, sentiment score data which our model produced did not have cogent correlation with the stock price data. Among many explanations for the reason, a compelling one is that our current model is not precise enough to capture the sentiment of corpus yet.

Similar to many other utterances online, Reddit corpus holds many meanings which can be appreciated only when we read between the lines such as sarcasm, pretense, or irony. Our current model is based on the bag-of-the-word feature selection mechanism. That means the classifier parses target text (a sentence or a document) to word level and classifies each word to calculate the target's sentiment possibility score based on. A weakness of this approach is that the classifier can't recognize the context of the text. Ideally, if we can incorporate understanding of hidden meaning in our sentiment analyzing model, the result may draw different picture and produce different correlation diagnosis result.

To raise up the classifier to the pragmatically functional level, we need to incorporate topic modelling in our model. Topic model is useful tool to discover hidden semantic structures in a text body. Through this model, one can extract probabilistically meaningful topics from large corpus. In this case, if we can expand our data to substantial amount of posts and their comments on a certain date and identify the topics related to the scandal first, then we can recognize which sub-topics were being discussed and analyze them in pragmatic level.

Another strong solution to this problem is deep learning. If we can collect large enough data to perform unsupervised training to classify subtle expressions such as sarcasm, the classifier could produce more reliable data with higher accuracy.

The step after the highly reliable classification model development will be the prediction. Our research identified empirical similarity between the sentiment analysis score and stock market price. Highly elaborated sentiment model analysis may be adopted to predict seemingly unrelated data such as stock price in the future.

The final goal of our project is to make prediction of stock price based on the sentiment score in reddit. Referring to our previous results, we tried to calculate the linear and quadratic regression formula by Excel. Unfortunately, the results of regression are not as our expectation. The R square of linear regression is 0.005410198 and the R square of quadratic regression is 0.008070642. So, statistically, it seems like sentiment scores and stock prices are not linearly or quadratically dependent.

We still believe there are some specific relationship between public opinions and stock

prices. In our further research, one of the most challenging and important problems is to find the potential associative relationships between these two factors. Also, the sample data we used is just one post with most comments per day. If we cover more data in reddit and other big social media and BBS platform, the relationship may be more obvious.

8. Reference

[1] Reddit.com. (April 30, 2018). Unique visitors 2018 | Statistic. Retrieved from https://www.statista.com/statistics/443332/reddit-monthly-visitors/

[2] Reddit.com. (December 19, 2017). The Best of Reddit in 2017. Retrieved from https://redditblog.com/2017/12/19/the-best-of-reddit-in-2017/

[3] The Guardian. (April 04, 2018). Facebook says Cambridge Analytica may have gained 37m more users' data. Retrieved from https://www.theguardian.com/technology/2018/apr/04/facebook-cambridge-analytica-user-data-latest-more-than-thought

[4] The New York Times. (March 17, 2018). How Trump Consultants Exploited the Facebook Data of Millions. Retrieved from https://www.nytimes.com/2018/03/17/us/politics/cambridge-analytica-trump-campaign.html [5] Wu, Jean Y., and Yuanyuan Pao. "Predicting Sentiment from Rotten Tomatoes Movie Reviews."

[6] Pak, Alexander, and Patrick Paroubek. "Twitter as a corpus for sentiment analysis and opinion mining." LREc. Vol. 10. No. 2010. 2010.

[7] G. Mishne, “Experiments with Mood Classification in Blog Posts”, Proceedings of 1st Workshop on Stylistic Analysis of Text for Information Access, 2005.

[8] Y. Mao and G. Lebanon, “Sequential Models for Sentiment Prediction”, Proceedings of ICML workshop on Learning in Structured Output Spaces, 2006.

[9] Yang, Changhua, Kevin Hsin-Yih Lin, and Hsin-Hsi Chen. "Emotion classification using web blog corpora." Web Intelligence, IEEE/WIC/ACM International Conference on. IEEE, 2007.

[10] Lee, Kathy, et al. "Twitter trending topic classification." Data Mining Workshops (ICDMW), 2011 IEEE 11th International Conference on. IEEE, 2011.

9. Appendix: Part of codes

```
def dayComSenti1(com):
    '''
    :return polarity score
    :param com: list of comments perday [content, vote, polarity]
    '''
    for index, (cc, score) in enumerate(com):
        blob = TextBlob(cc)
        com[index].append(blob.sentiment.polarity)
    return com

def testAcr():
    '''
    :print rmse accuracy by the labeled test file
    '''
    df = pd.read_csv('test.csv')

    sentList = df['sent'].tolist()
    rate1 = df['rate1'].tolist()
    rate2 = df['rate2'].tolist()
    rates =[]
    pred = []
    for i, sent in enumerate(sentList):
        blob = TextBlob(sent)
        predict = blob.sentiment.polarit
        pred.append(predict)
        rate = (rate1[i] + rate2[i])/2
        rates.append(rate)
        print(sent)
        print('avg labeled rate: ', rate)
        print('predict rate: ', predict)
        print('\n')

    print('total accuracy RMSE', rmse(rate,pred))

def rmse(tgt,pred):
    '''
    :para: target and prediction label
    :return rmse score
    '''
    sqrerr = []
    for i in enumerate(tgt):
        err = tgt[i] - pred[i]
        sqrerr.append(err * err)

    from math import sqrt
    return sqrt(sum(sqrerr))/len(sqrerr)
```

```
def trainSet():
    '''
    :return feature extracting trainset
    '''
    documents = [(list(movie_reviews.words(fileid)), category)
                 for category in movie_reviews.categories()
                 for fileid in movie_reviews.fileids(category)]

    random.shuffle(documents)

    all_words = []
    for w in movie_reviews.words():
        all_words.append(w.lower())

    stopwords = nltk.corpus.stopwords.words('english')
    newstopwords = [word for word in stopwords if word not in negationwords]
        # remove stop words from the all words list
    new_all_words_list = [word for word in all_words if word not in newstopwords]
    all_words = nltk.FreqDist(new_all_words_list)


    word_features = list(all_words.keys())[:3000]
    featuresets = [(find_features(rev), category) for (rev, category) in documents]
    train_set = featuresets[:2000]
    return train_set
```

```
def pipeClf(train_set):
    '''
    :return: trained classifier
    :param [] training set
    '''
    logger.info('Training MultinomialNB Model')
    pipeline = Pipeline([('tfidf', TfidfTransformer()),
                         ('chi2', SelectKBest(chi2, k=2000)),
                         ('nb', MultinomialNB())])
    pipecl = SklearnClassifier(pipeline)
    pipecl.train(train_set)
    return pipecl

def calTotalScore(senti):
    '''
    :return: final sentiment score of the post
    :param senti(polarity, vote)
    '''
    sumVote =0.0
    sumScore = 0.0
    for (c, v, p) in senti:
        sumVote += v
        sumScore += v * p
    return sumScore/sumVote

def dayComSenti(com, clf):
    ''''
    :return probability of sentiment prediction
    :param com: list of comments perday [content, vote, polarity]
    '''
    for index, (cc, score) in enumerate(com):
        pred1 = clf.classify(find_features(cc))
        prob_dist = clf.prob_classify(find_features(cc))
        pos_prob = round(prob_dist.prob("pos"), 3)
        com[index].append((pos_prob - 0.5)*2)

    return com
```