OPINION SUMMARIZATION OF PERSON USING TWEETS

PROJECT REPORT – Text Mining for Social Media.

By,

Rohith Engu – 678923180.

Tushar Gupta – 373808005

Guided by,

Professor. Edmund Yu.

Abstract

Twitter is an online microblogging site which enables users to share their emotions, thoughts or opinions either positive or negative, relevant or irrelevant by tweeting a sentence up to 140 characters. One of the important feature of a tweet is that it enables users to share their opinions on someone. For a particular person over twitter, there are more than 1000's of tweets per week and it is very difficult to read all of them to frame an opinion. Due to this very nature of twitter we want extract some of them which convey the opinion that users have about the person. We describe an approach below to solve the problem of summarizing opinions based on tweets.

Problem Definition

Summarizing text can be viewed as a generic problem of an application which takes in input as documents or text related to a topic and summarizes into condensed form of most important content in the input documents.

Whereas for this project we can briefly describe the problem statement as below: given a number of tweets based on a person we tried to represent all the tweets into a summary which takes into consideration the sentiment ratio of input tweets. For Example: if we are given about 100 tweets with 70 tweets as positive and about 30 tweets as negative about "Justin Trudeau" by various twitter users we will be summarizing all the tweets into condensed threshold number (30%) of tweets (which is summary) by also maintaining the ratio of the positive to negative sentiment.

Another important feature in our approach is finding relevancy of a tweet. Finding tweets which are relevant to expressing opinion is an important task because there could be lot of tweets about a person on twitter, but not all of them try to convey an opinion. We also describe an approach below to solve the problem of relevance of tweet.

Several approaches are currently in use and have been implemented for automatic summarization of documents and most of them consider a particular topic as a basis of extraction. There are very few summarizers which try summarizing Social Media like tweets and there is not much work on summarizing tweets based on a person with maintaining their positive to negative sentiment ratio in the summary.

In this document we further discuss our approach in finding a summarized representation of all the tweets about a particular person.

Data

Using Twitter search API we collected approximately 4000 tweets on various personalities over twitter like Donald Trump, Justin Bieber, Bill Gates, Katy Perry, JK Rowling, Adolf Hitler, Miley Cyrus, Charlie Sheen, and Russel Crowe. Also, while collecting the tweets we got the entity count for the Person in the tweet and the length of the tweet. We wrote all the data into the MongoDB. This data will act as our training data. The data is available in file TrainingTweets.csv provided in the submission.

For ex: "Name: Donald Trump, Tweet: `RT @AllKnowA: We all know a dumbass named Donald Trump`, NEREntityCount: 2, Length of Tweet:54"

Test Data: We collect tweet at runtime for each of the person we want to summarize. Just specify the name of the person and number of tweets to get. It will save the data in the file "name person".csv

Attributes

1. Length Of Tweet (LengthofTweet)
   The number of characters in the tweet after we removed the url's.

2. Number of Named entity (NEREntityCount)
   We count the number of named entities in the tweet. To do this we first used the NLTK NER system but the results were not very good so we used the Stanford NER system.
   For ex: "`@CBCAlerts @JustinTrudeau when will his royal highness Justin Trudeau start working?`"
   NLTK NER - (PERSON Justin/NNP Trudeau/NNP) i.e only 1 person entity
   Stanford NER – JustinTrudeau, Justin, Trudeau i.e 3 entity
   The NLTK NER failed to classify `@JustinTrudeau as an entity.`

3. Number of Positive, Negative words and emoticons (positiveNegative)
   This attribute defines the total number of positive unigrams, negative unigrams and the number of emoticons in the tweet. We will discuss more of how we extract them in algorithm.

Class Labels

4. Sentiment of the Tweet (Sentiment)
   For the training set we manually assign a label of 0 or 1. 0 being positive and 1 being negative.
   For ex: "`RT @AllKnowA: We all know a dumbass named Donald Trump`" sentiment 1. "`RT @Brittany0180: Wow so many pictures of Donald Trump`" sentiment 0.

5. Relevance of the Tweet (Relevance)
   For the training set we manually labelled each tweet being relevant or irrelevant. Relevant being 1 and irrelevant being 2. When our proposed system will produce a final summary then it will consider only the relevant tweets and discard all irrelevant tweets.
   For ex: "`Up like Donald Trump!`" relevance 1. "`RT @CarolHello1: Natl Security Donald Trump Whatever It Takes Albany`" as irrelevant 2.

Algorithm

We used a No-SQL database MongoDB for faster processing of data.

1. Pre Processing

1.1 Filtering

Initially the number of tweets were 4170. We filtered these tweets down to 1411 after removing all the redundant tweets as most of them were re-tweets or duplicates. In order to filter these tweets we pre-processed the tweets and checked for redundancy and later we manually filtered from the rest of them.

1.2 Basic Pre-processing

1.2.1 Removed Url:

Removed all the https and http and other url's from the tweet.

1.2.2 Removing all the Username

Removed all the usernames from the tweets. The usernames start with @ and they were irrelevant to sentiment or relevance of the tweet.

1.2.3 Removing all the "RT"

From all the tweet remove the word "rt" if the tweet starts with it.

1.2.4 Remove numbers and wide spaces

1.2.5 Trim the tweet

1.3 Remove Stop Words

We used NLTK to remove the stop words. We didn't remove words like "not" from the tweet as they have a significant impact on what a tweet is about. For ex: "not good", if we remove not then the whole sentiment switches from negative to positive.

1.4 Remove Person Entity

Remove all the permutation of the name of the person from the tweet. For ex: if we collected tweet about Donald Trump then we remove the following from the tweet: "donald, donald's, trump, trump's, donaldtrump".

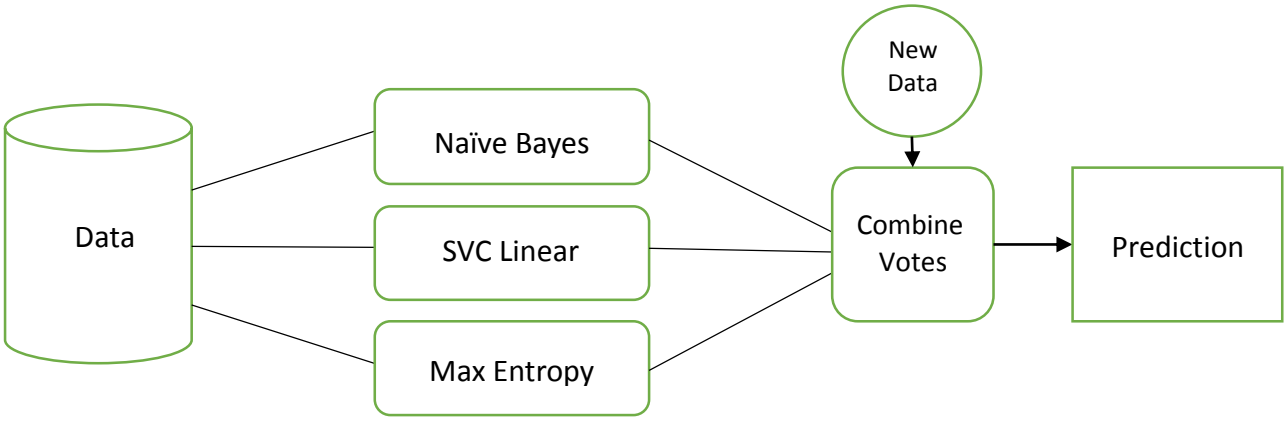
1.5 Stemming

We used English Stemmer to perform the stemming on the tweet.

2. Sentiment Analysis

For sentiment analysis we are using an ensemble based approach and the number of word features we are using are 954. We got 1000 features but we eliminated few features whose length is less than 2.

We tried using Naïve Bayes, Decision Tree, SVC linear, SVC poly and Maximum entropy. Out of them we selected Naive Bayes, SVC Linear and Max entropy for ensemble. We didn't use Decision Tree as a classifier as the number of features are high and it was taking a lot of time to build the tree. It may be because of the fact that the tree being generated is quite complex.



10 fold cross validation was used to measure the accuracy for all of these classifiers and the results are below:

Cross Validation	Naïve Bayes	SVC Linear	Max Entropy	SVC Poly
1	**0.7207**	0.7207	**0.7837**	0.6396
2	0.7027	**0.7117**	**0.7567**	0.5405
3	**0.8018**	0.7927	**0.7927**	0.6036
4	**0.7297**	0.6756	**0.8018**	0.6667
5	**0.6306**	0.6216	**0.7387**	0.5675
6	**0.7477**	0.7027	**0.8288**	0.6576
7	**0.6846**	0.6576	**0.7837**	0.6216
8	**0.6846**	0.6667	**0.7567**	0.5135
9	0.7387	**0.7387**	**0.8198**	0.7207
10	0.5855	**0.6216**	**0.7117**	0.5495
Average	70.27%	69%	78%	61%

The Naïve Bayes and the maximum entropy performs the best. And the reason we think why SVC is not performing well is that the data is very sparse.

Once all the models are generated using the training data what we do is, we pick each tweet (a record/document in MongoDB) from the test data and extract its features and then voting is performed on the prediction output of classifiers. This sentiment is added as an attribute in the Mongo DB. For ex: if 2 of the classifiers predict it as a positive sentiment and the other one predicts it as a negative sentiment then the tweet is classified as positive.

3. Get the positive, Negative and emoticons count for each tweet

To get the number of positive and negative words in a tweet we are using 2 lexicons:

a) Regular Lexicon [1]: Contains regular dictionary words with a score between the range of -5(negative) to 5 (positive).
b) Twitter Lexicon [2]: Contains words that are relevant to twitter social media. For ex: aww, yayy, #love etc. The scores are in the range of -1 to 1.

For both the files we enriched them and added some of the words relevant to tweets related to people.

We also count the number of emoticons `":-)",":-*",":-(",":-|"` in the tweet.

We then set the attribute “positiveNegative” as sum of number of positive words + negative words + emoticons

4. Logistic Regression

Now for predicting the relevance of the tweet we are using Logistic Regression. We wanted to use linear regression but the class labels in linear regression comes out to be continuous values and they were all very close to each other. That is why we switched to Logistic Regression. The attributes we are using to train the model is NEREntityCount, LengthofTweet and positiveNegative.
Below are the result of one of the 10 iterations of cross validation:

Class Label	Precision	Recall	F1-score	Support
1	0.62	0.81	0.70	161
2	0.56	0.33	0.41	118
Avg/total	0.56	0.61	0.58	279

Confusion Matrix

	Relevance 1	Relevance 2
Relevance 1 (Predicted)	130	31
Relevance 2 (Predicted)	79	39

From the confusion matrix we can interpret that a lot of irrelevant (Relevance =2) are predicted as relevant (Relevance=1) tweets. If we improve the accuracy of the logistic regression we would be able to get more refined tweets for the output of the summarizer since we are only concerned about the relevant tweets in the output.

Accuracy 0.605734767025

The accuracy of the system as we can see is not very good. The reason behind this the number of attributes and the number of training samples we have in our dataset is very low.

Now, once the model is trained, then for each of the test tweet (a document/record in Mongo DB) we predict whether the tweet is relevant or irrelevant based on the features extracted for it. Below is a sample of the result on the test tweets:

Example:

- `#justintrudeau is a class act` – **Relevant.**
- `@Commodity52now @sunlorrie @nationalpost @JustinTrudeau And Jews &amp; women like me mistaken as a Jew, repeatedly spat on by ignorant Muslims` – **Irrelevant**
- "`05. I Want to Hold your hand - Chris Colfer via @YouTube`" – this tweet is irrelevant to us but is marked as relevant by the model.

5. Summarizing using Sum Basic

In order to produce the summary we use a threshold value (percentage of tweets you want in your summarizer output). We are using 0.3 as our threshold value i.e 30%.

5.1 The first step is to get the total number of tweets we have collected for that person (Test Set).
5.2 We get the number of positive tweets and negative tweets in the input
5.3 Calculate the ratio of positive tweets to the negative tweets. We do this as we want both the negative and the positive tweets to be a part of the summary maintaining the ratio.
5.4 Calculate the Probabilities distribution of all words in input.

$$P(w_i) = n/N$$

n – occurrences of word, N – Total number of words.

5.5 For each of the tweet (S_i) we assign the weight (sumBasic score) by:

$$Weight(S_i) = \sum_{W_i \in S_j} \frac{P(W_i)}{|\{W_i | W_i \in S_j\}|}$$

5.6 From Relevant (class label : 1) tweets we pick the best scoring Positive and Negative Sentiment tweet containing highest probability word.
5.7 For tweet which is picked we update the probability distribution of the words in this tweet by:

$$pnew(wi) = pold(wi) * pold(wi)$$

This will help eliminate the redundant tweets because if there is a tweet having the same words the overall sumBasic score will be reduced for that tweet

5.8 If the threshold is reached we exit otherwise we go back to 5.5

Evaluation

In this section we describe the evaluation of our implementation using three methods Precision, Recall, and F-Measure. Evaluation for our Summarizer is carried out by comparing the above mentioned values to a Manual summary made by us by collecting a threshold limit of tweets.

The evaluation procedure is described in detail below:

For each Person for whom tweets are to be summarized we manually extract 30% of the tweets for which we feel summarizes opinions on that selected person. We compare these manually extracted tweets to the tweets which our summarizer gives as the output. We manually compare each tweet from the manually selected tweet and the summarizer output and calculate the values of Precision, Recall and F-Measure.

Precision

Precision can be defined as a method which calculates the percentage of items that are relevant.

TP – True Positive in our approach is the number of tweets which were manually selected by us and have also been picked by our summarizer.

FP – False Positive is the number of tweets which appeared in our summarizer but have not been picked manually.

Recall

Recall can be defined as a method which calculates the percentage of items that are selected and are relevant.

FN – False Negative is the number of tweets that are picked manually but have not been picked by our summarizer.

F-Measure

F-Measure or F1-score can be defined as a measure of a test's accuracy, it considers both Precision and Recall values and is defined as a harmonic mean of both values.

Results

We evaluated the above measures considering different personalities because of diverse opinions on them. We only extract the tweets that Have Relevance = 1.

	Mark Zuckerberg	**Chris Colfer**	**Rachel Roy**	**Justin Trudeau**
Total Tweets Collected	54	120	100	100
Tweets Extracted	15	36	30	30
Relevance 1 count in Input	34	95	79	50
Relevance 2 count in Input	18	25	21	50
Positive Tweets in Summary	14	32	20	28
Negative Tweets in Summary	1	4	10	2
Precision	**0.53**	**0.58**	**0.48**	**0.53**
Recall	**0.60**	**0.61**	**0.4667**	**0.53**
F Measure	**0.56**	**0.59**	**0.4732**	**0.53**

For Justin Trudeau the number of irrelevant tweets are pretty good. If we can classify more tweets as irrelevant then the system will perform pretty decent. The values of precision and recall are what we expected out of the proposed system. However, looking at the results above us we can say that if we look at the positive and negative tweets the positive tweets are dominating in number. This is because of 2 factors, if we consider the neutral tweets to be positive in nature then it is really difficult to find a large number of negative tweets and the other factor is the accuracy of the sentiment analyzer.

Note:

- For calculating all the above measures we manually extracted a threshold of 30% of the tweets given as input for that person.
- Along with our Source code we are submitting all output files of Persons for whom we tried to summarize the tweets.

  Ex: for Justin Trudeau as input through our Main.py, we collected tweets into a CSV file named "Justin Trudeau.csv" then we manually assigned TP, FP and FN values to the tweets by comparing them to what our summarizer predicted and saved them into an Excel file with same name.
- Our output file (summarized form) is a text file generated through our application as "<PERSON_NAME> Summary.txt".
- We have provided a folder "Results", it contains all the excel sheets for different people for which we tested our summarizer. Every workbook has three sheets which summarizes all the results calculated manually and also by our summarizer.

Conclusion

In this project we described an approach to summarize tweets based on opinions from different users. We represented the problem as a logistic regression and used ensemble method to predict Sentiments of tweets for one of our class label. The approach was then evaluated with various methods like Precision, Recall and F- Measure.

From our results and the accuracies shared above, our logistic regression model has an accuracy of ~60%, which is a bit less from expected, this is due to the less number of dataset. We can improve the accuracy by adding more dataset and extracting more features and also we can use an ensemble based approach for the prediction of Relevancy class label as we did for sentiment.

Sentiment analysis does not have much significance in our approach because we use it to maintain the ratio of Positive to Negative opinions in the summarizer output. This is because we want to capture both the positive and negative opinions about the person. Relevance of tweet does not depend on the sentiment of the tweet.

Considering different accuracies for the people we have summarized tweets we have an average of 0.53 for precision and 0.55 for recall which is comparable to other works in similar areas of text summarization.

References

- Nawal El-Fishawy, Alaa Hamouda, Gamal M. Attiya, Mohammed Atef - Arabic Summerization inTwitter Social Network [Ain Shams Engineering Journal (2014) 5, 411 – 420]
- Kavita Ganesan, ChengXiang Zhai and Jiawei Han - Opinosis: A Graph-Based Approach to Abstractive Summarization of Highly Redundant Opinions [University of Illinois at Urbana-Champaign]
- Vishal Gupta, Gurpreet S. Lehal - A Survey of Text Mining Techniques and Applications [JOURNAL OF EMERGING TECHNOLOGIES IN WEB INTELLIGENCE, VOL. 1, NO. 1, AUGUST 2009]
- Dipanjan Das, Andr´e F.T. Martins - A Survey on Automatic Text Summarization [November 21, 2007]
- David Inouye, Jugal K. kalita Comparing Twitter Summarization Algorithms for Multiple Post Summaries [2011]
- [1] http://www2.imm.dtu.dk/pubdb/views/publication_details.php?id=6010
- [2] http://saifmohammad.com/WebPages/lexicons.html