Who Likes Whom or What?

Aditi B 1213175832

Harini R 1213221839

Harish A 1212953090

Pranav D 1213414668

Rohit B 1213168981

{anbarask, hramamu2, hananth2, phdeshpa, rbalas10}@asu.edu

1 Motivation

The volume of data available on social media platforms like twitter is immense. Users express their opinions on a variety of subjects like movies, products, socio-political events etc. We can leverage this data to take informed decisions. By applying relevant extraction method and suitable NLP techniques, we can identify customer attitudes, opinion about products and services. It might be helpful for the other individuals to make better decisions based on these reviews. Organizations could use this information to identify areas where they are exceling and areas where they need to improve, based on the reviews from customers. They could also gauge the performance of their competitors to develop strategies for increasing their profit. We intend to use twitter data to identify the opinion of the users towards a set of 6 airlines (American, Delta, Southwest, United, US Airways and Virgin America).

2 Problem Definition

The input is the short text of twitter data.This data is sent to the Bigram Topic Model which will extract all the topics. The same twitter short sentence is taken as the input and the data is preprocessed and the cleaned data is sent to the classifier for sentiment analysis.The topics extracted from the topic modelling is also considered as a feature for sentiment analysis.The output of the system would be the sentiment associated with the topic for the particular Airlines.

3 Examples

@United now what?!? http://t.co/5hpSqVRjK8 flight was gone when I got off plane! #BusinessTravel #goodenoughmother'
Output:

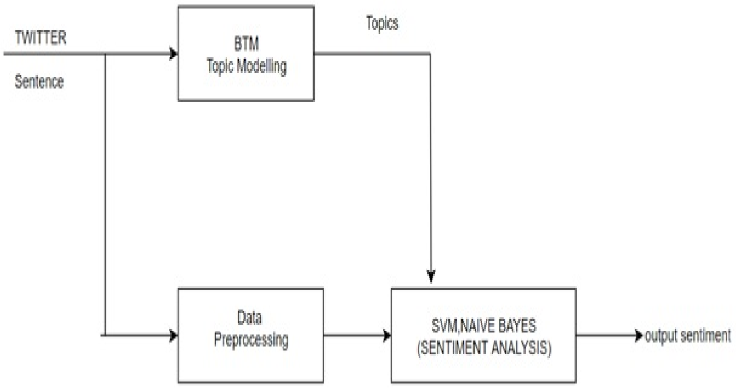

Figure 1: System Architecture

Topic: Bad Flight
Sentiment: Negative

4 Inference

The project focuses on extracting comments mentioned on social media about various airlines and associate the sentiment of the extracted tweet using a Topic based approach. We feed tweets for which the topics are extracted using a Topic based models.

- LDA
- BTM

Once the tweets are categorized into topics we will train a sentiment analysis model by aggregating all tweets that belong to each topic. So a separate model is developed for each topic (using all tweets belonging to that model as training set) and this is then used to classify the sentiment. This training model used for performing the sentiment analysis is:

- SVM,
- Naive Bayes,

The pre-processing step, feature extraction and used for the topic identification and sentiment analysis are explained below

4.1 Pre-processing Step

We perform preprocessing in two stages, for the initial step of topic extraction we do the following:

- Remove the stop words from the tweet,
- Stemming and lemmatization. Example: The word cleaning in the sentence will become clean.,
- Replacing acronyms with full forms. Example: lol will be replaced with laughing out loud,
- Removing the user handle and replacing it with a <User>tag,
- Remove urls from the tweet.
- Converting the entire tweet to uniform case. We would use lowercase for our case,
- Removing the multiple characters from a word. For example: The word Coooooool will be replaced with Cool,
- Removing the multiple characters from a word. For example: The word Coooooool will be replaced with Cool,
- Hashtag Analysis - extract the word or phrase from the hashtag, instead of passing the entire hashtag.

In-order to achieve precision, we want to retain the relevant features from the original tweet. Hence, we restrict the number of preprocessing steps for sentiment analysis. We only plan to perform the following steps for sentiment analysis:

- Remove the stop words from the tweet,
- Stemming and lemmatization,
- Replacing acronyms with full forms. Example: lol will be replaced with laughing out loud,
- Removing the user handle and replacing it with a <User>tag,
- Remove urls from the tweet.
- Remove digits,
- Hashtag Analysis - extract the word or phrase from the hashtag, instead of passing the entire hashtag.

4.2 Features for sentiment analysis

- Positive-emoticon : Presence of positive emoticon
- Negative-emoticon : Presence of negative emoticon
- Capitalization : Presence of words in uppercase
- Elongated words : Presence of multiple characters in a word
- Exclamation mark : Presence of !
- Question mark : Presence of ?
- Negation : Presence of negation words
- Retweet Count : Number of times the tweet has been retweeted
- Sarcasm : Does the tweet hint of sarcasm
- Hashtags : Analysed hashtag
- Topic : The topic of the tweet
- Laugh : Presence of words like haha
- POS-tags : Word to part-of-speech correspondence
- POS-position : Match between the word position and part-of-speech

4.3 Methodology

Topic modelling is the task of identifying which underlying concepts are discussed within a collection of documents, and determining which topics each document is addressing. In our approach after the features for the topic extraction are performed, we then feed the pre-processed data to a generative models for extraction of topics which uses LDA or BTM. LDA is the most commonly used topic based model which is used for topic extraction. It takes features extracted from each tweet as a document and finds the distribution of topics associated to the tweet. We can arrive at the most dominant topic which is represented in the document and classify the document as belonging to that topic. We will then perform the same using the approach of BTM, to overcome the problem of sparse document which arises when we use LDA. BTM tackles this problem by learning topics over short text by directly modeling the generation of all the unordered word-pairs co-occurring within a document (i.e. biterms) across the corpus. Topic inference in a document is based on the assumption that the topic proportions of a document equals to the expectation of the topic proportions of biterms generated from the document:

If d represents the document and z the topic and W_i, W_j represent the two words in the bigram -:

$$P(\frac{z}{d}) = P(\frac{z}{b}) * P(\frac{b}{d}) \tag{1}$$

P(z / b) can be calculated via Bayes formula based on the parameters estimated in BTM:

$$P(\frac{z}{b}) = \frac{P(z) * P(\frac{W_i}{z}) * P(\frac{W_j}{z})}{\sum\limits_z P(z) * P(\frac{W_i}{z}) * P(\frac{W_j}{z})} \tag{2}$$

where

$$P(z) = z\theta$$

$$P(\frac{W_i}{z}) = (\frac{\varphi_i}{z})$$

The empirical distribution of biterms in the document as the estimation

$$P(\frac{b}{d}) = \frac{n_d(b)}{\sum\limits_b n_d(b)} \tag{3}$$

where $n_d(b)$ is the frequency of the biterm b in the document d. In short texts,

$$P(\frac{b}{d}) \tag{4}$$

is nearly an uniform distribution over all biterms in the document d. Biterm is an unordered word-pair co-occurred in a short context. The data generation process under BTM is that the corpus consists of a mixture of topics, and each biterm is drawn from a specific topic.

Once the subtopics are identified they are identified to belong to either of the 10 topics which are mentioned below:

- Bad Flight
- Cancelled Flight
- Customer Service Issue
- Damaged Luggage
- Flight Attendant complaints
- Flight Booking
- Late Flight
- Longliness
- Lost Luggage
- Others

We take all tweets belonging to each topic and separately train a sentiment analysis model using SVM and Naive Bayes. This will generate multiple models for each topic which will give you optimized sentiment specific to topics. So when any new tweet is introduced, the topic will first be identified and then based on this the specifically model trained for that topic is used to identify the sentiment.

5 Training

The data for our model analysis consists of 14485 tweets, where each tweet is uniquely defined by 15 attributes. The entire data can be seen as a 2 dimensional matrix of size 14485x15. This data contains 14485 tweets for the Airlines in consideration. For the training phase of the project, we plan to use 80% of the data (approximately 11500 tweets) and use the remaining 20% of the data (approximately 3000 tweets) for the testing phase. For validation purpose, data is scrapped from Twitter website using a number of tools. Primarily python, selenium and twitter api's were used to fetch data and store it in a readable .csv format. For simplicity, we only fetched tweets which were in English.

6 Evaluation

The majority of libraries and implementation rely on NLTK libraries. The algorithm applied for Sentiment Analysis, Nave Bayes (NB) and Support Vector Machines (SVM) is also implemented using NLTK libraries. For comparison between the 2 models used for sentiment analysis, we make use of three parameters: Precision, Recall and F-Measure.

The Precision can be calculated using TP and FP rate as shown below:

$$Precision = \frac{TP}{TP + FP} \tag{5}$$

where,
TP = True Positive
FP = False Positive
TP is used for sentences, which are correctly classified, and FP is for those sentences, which are wrongly classified.

Recall can be calculated as shown below:

$$Precision = \frac{TP}{TP + FN} \tag{6}$$

where,
TP = True Positive
FN = False Negative
FN is used for non-classified sentences and TP is for correctly classified sentences. F-Measure can be computed as below:

$$Precision = \frac{2 * Precision * Recall}{Precision + Recall} \tag{7}$$

7 Related Works

Uncovering latent semantic structure from a text corpus has been long discussed and many topic models have been proposed. Latent Semantic Analysis (LSA) is a method to mine the semantic structure of a text collection that is based on Singular-valued decomposition of the document-term to determine the major associative words patterns. Another similar approach was the Probabilistic Latent Semantic Analysis (PLSA) which is the probabilistic variant of the LSA method. In PLSA document is considered as a mixture of topics, and a topic is considered as a probabilistic distribution of words. The Latent Dirichlet Allocation (LDA) model extends the PLSA by incorporating Dirichlet priors to improve and result in a more complete generative model. Many extensions of the PLSA and LDA have been proposed such as author-topic model, Bayesian non parametric topic model and supervised topic model. The recently proposed regularized topic model and the generalized Polya model are similar to BTM and also employs word co-occurrence statistics to enhance topic learning. However, both utilize word co-occurrences as structure priors for topic-word distribution, rather than directly modeling their generation process. All these models concentrate mainly on normal texts instead of short text specifics.

Early models on short texts concentrated primarily on exploiting external knowledge to enrich the representation of short texts. For example, Sahami et al.[28] suggested a search-snippet-based similarity measure for short texts. Phan et al.[24] learned hidden topics from large external resources to enrich the representation of short texts. Jin et al.[19] learned topics on short texts via transfer learning from auxiliary long text data. But these methods are domain specific and does not work for a general scenario as external data set might not be available. There are many approaches where the short-text of Twitter are combined into a large pseudo-documents with the help of additional features, and the traditional topic models are trained on them. The problem with the above listed approaches to topic mining is the lack of data and suffer from sparse patterns in the document level.

BTM model focuses on general-domain short texts, without exploiting any external knowledge. This method also overcomes the sparsity problem. In this model the topics are learnt by directly modelling the generation of word co-occurrence patterns in the entire corpus. Biterm is an unordered word-pair co-occurred in a short context. The data generation process under BTM is that the corpus consistS of a mixture of topics, and each biterm is drawn from a specific topic. Advantages of BTM:

- BTM enhances the topic learning by modelling the word co-occurrence patterns (i.e. biterms), rather than documents
- BTM solves the problem of sparse patterns at document-level by using the aggregated patterns in the whole corpus for learning
- It is easy to implement and scales up well

Topic components and a global topic distribution of the corpus, except the topic distribution of each individual document can be obtained by learning BTM. However, we show that the topic distribution of each document can be naturally derived based on the learned model.