An Ensemble Sentiment Classification System of Twitter Data for Airline Services Analysis

by

Yun Wan

Submitted in partial fulfilment of the requirements for the degree of
Master of Electronic Commerce

at

Dalhousie University
Halifax, Nova Scotia
March 2015

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

In airline service industry, it is difficult to collect data about customers' feedback by questionnaires, but Twitter provides a sound data source for them to do customer sentiment analysis. However, little research has been done in the domain of Twitter sentiment classification about airline services. In this thesis, an ensemble sentiment classification strategy was applied based on majority-votes principle of multiple classification methods, including Naive Bayes, SVM, Bayesian Network, C4.5 Decision Tree and Random Forest algorithms. In our experiments, six individual classification approaches, and the proposed ensemble approach were all trained and tested using the same dataset of 12864 tweets, in which 10 fold evaluation is used to validate the classifiers. The results show that the proposed ensemble approach outperforms those individual classifiers in this airline service Twitter dataset. From our observations, the ensemble approach can improve the overall accuracy of individual approach in twitter sentiment classification in this domain.

LIST OF ABBREVIATIONS USED

IG	Information Gain
API	Application Program Interface
KNN	K-Nearest Neighbor
SVM	Support Vector Machine
GUI	Graphic User Interface
NB	Naïve Bayes
POS	Part-of-Speech
NLP	Natural Language Processing
CTM	Correlated Topics Models
VEM	Variational Expectation-Maximization
AQR	Airline Quality Rate
ID3	Iterative Dichotomiser 3

ACKNOWLEDGEMENTS

First of all I would like to gratefully acknowledge my supervisor, Dr. Qigang Gao. Without his invaluable guidelines, kind patience, strong encouragement for my research and overriding the academic requirements for me to transfer from project option to do this thesis. I am very pleasant that, after more than one year's systematic training from his insight supervision and enthusiastic concerning, I have greatly improved my research skill and start to publish paper, which I have never thought it before I met Dr. Qigang Gao.

I would also like to give my deep thanks to Dr. Vlado Keselj. Thank him for reviewing the thesis and his grant to me to switch from project to thesis option. His excellent instruction in the Natural Language Processing course gave me a lot of inspirations and a comprehensive coverage of the existing NLP techniques. My deep appreciation also goes to Dr. Jacek Wolkowicz for his instructions in the Data mining course and his advices for me in the twitter analysis research area.

Without these people, this thesis would not exist.

Yun Wan
Halifax March 27th 2015

CHAPTER 1 INTRODUCTION

1.1 Motivation

Airline service companies must interpret a substantial amount of customer feedback about their products and services. However, conventional methods to collect customers' feedback for airline service companies is to investigate through distributing and collecting questionnaires, which is time consuming and inaccurate. It needs labour to distribute and collect questionnaires to customers and also it will take too much effort to record and file those questionnaires considering how many passengers take flights every day. Beyond that, not all customers take questionnaires seriously and many customers just fill them in randomly and all of this brings noisy data into sentiment analysis. Unlike investigation questionnaires, twitter is a much better data source for sentiment classification for feedbacks of airline services. Because of the Big Data technologies, it has become very easy to collect millions of tweets and implement data analysis on them. This has saved a lot of labour costs which questionnaire investigations need. More than that, people post their genuine feelings on Twitter, which makes the information more accurate than investigation questionnaires. The other limitations for questionnaire investigations are that the questions on questionnaires are all set and it is hard to reveal the information which questionnaires do not cover.

As a result, text sentiment analysis has become very popular in recent years for automatic customer satisfaction analysis of online services. Sentiment analysis is a sub domain of data mining, which are exploited to analyze large-scale data to reveal hidden information. Obviously, the advantages of automatic analysis of massive datasets make sentiment analysis preferable for airline companies.

Sentiment classification techniques can help researchers and decision makers in airline companies better understand customer feedback and satisfaction. Researchers and decision makers can utilize these techniques to automatically classify customers'

feedback on micro-blogging platforms like Twitter. Business analysis applications can be developed from these techniques as well.

There have been much research on text classification and sentiment classification, but there has been little on Twitter sentiment classification about airline services. Except applying popular sentiment classification approaches to tweets on airline services domain, it is also desirable to develop a new approach to further improve the classification accuracy.

1.2 Research Objectives

Twitter is a really good source to get customers' feedback and marketing information in airline services, but there has been no perfect solution to automatically classify the massive amount of tweets, which leaves room for doing research in this area. This thesis focuses on comparing the performance of different sentiment classification approaches and developing a new sentiment classification approach to classify the tweets about airline services.

In this thesis, seven approaches are presented including an ensemble approach, which consist of a Naive Bayes classifier, a Support Vector Machine classifier, a Bayesian Network classifier, a C4.5 Decision Tree classifier and a Random Forest classifier. The ensemble classification approach takes into account classification results of the five classifiers and uses the majority vote method to determine the final sentiment prediction. The comparison of different sentiment classification approaches and an analysis are given in this thesis.

1.3 Thesis Organization

The thesis is organized as follows. In chapter 1, the motivation are explained and the thesis objective is introduced. In chapter 2 the relevant work are discussed and major poplar methods are presented. Chapter 3 presents the data collection, data pre-processing and feature selection procedure. In chapter 4, the methodologies for sentiment classification are explained and the proposed approach is presented. In chapter 5, the

evaluation plan, the accuracy evaluation and an analysis are presented. In chapter 6, the conclusion is drawn and my contributions are described.

CHAPTER 2 RELATED WORK

2.1 Text Classification

“Text categorization (a.k.a. text classification) is the task of assigning predefined categories to free text documents.” (Yang and Joachims 2008) Text classification has applications in many areas such as spam filtering, email routing, language identification, topic classification and sentiment classification. Because of the development of electronic and information technologies, the volume of electronic text files has become too large for people to process manually. It has brought challenges and opportunities for the development of Natural Language Processing techniques such as text classification. Text classification techniques can use statistical or probabilistic algorithms to automatically classify massive electronic text files with computing technology.

Text classification is also a sub-domain of data classification. However, the text classification problem has some unique characteristics from the regular data classification problem. Most regular data classification applications deal with digits or nominal attributes but text classification applications deal with text data, which includes letters, words or phrases. The most common way to apply regular data classification techniques to text classification is to transform the text data into regular numeric data and then to implement data classifications. For example, we can transform every word appearing in a text dataset to an attribute and every text document to a vector of binary values which indicates the occurrences of the words in the document. Nevertheless, the dimensionality of the transformed digital dataset will still be too large for classification tasks. Even a small text dataset can contain more than a thousand distinct words, not to mention the phrases and longer grams. This problem is called the “curse of dimensionality" (Wikipedia 2014).

Feature selection is a process in text classification. In feature selection process, we select the features in the text dataset with feature selection algorithms based on the text classification goal. By only selecting the useful features for classification tasks, the dimensionality of the text classification dataset can be reduced to a reasonable size.

They are several popular text classification approaches (G. and RM. 2012) which exhibit efficiency, accuracy and scalability. They are the Lexicon-based approach, the Naive Bayes approach, the Bayesian Network approach， the Support Vector Machine (SVM) approach and the Decision Tree approach.

In data classification, there are two kinds of classification, supervised classification and unsupervised classification. In supervised classification, pre-labelled data are provided and classification models are trained on the labelled data. Unsupervised classification is a classification method which does not need pre-labelled data.

2.2 Non-Topic Text Classification

According to the objectives of text classification, text classification can be divided into topic classification and non-topic classification. Topic classification is used to classify different text files into different topic groups. Topic classification is used in many real world applications such as the Google search engine, auto-recommendation systems and library management. In text data, the topics of the text files are highly related to the word frequency distribution and topic classification applications have shown very good performance with traditional probabilistic and statistical methods (G. and RM. 2012).

Non-topic classification has been developed to classify text files in different groups based on properties which are not topics, such as genre classification and sentiment classification. Genre classification has been developed to classify text files into different genre groups such as classifying them as newspaper or research articles. Sentiment classification has been developed to classify text files into different sentiment groups, which are usually keyed to positive sentiment, negative sentiment and neutral sentiment.

2.3 Related Work in Sentiment Classification

Sentiment mining is a division of text mining, which includes information retrieval, lexical analysis and many other techniques. Many methods widely applied in text mining are exploited in sentiment mining as well. But the special characters of sentiment expression in language make it very different from standard factual-based textual analysis

(Pang and Lee 2008). The most important application of opinion mining and sentiment classification has been customer review mining. There have been many studies recorded on different review sites.

Sentiment classification has become very popular research area in recent years (G. and RM. 2012) not only because it is more difficult than other text classification problem but also because it has wide applications in real world. For example, customer review sentiment classification can be very important to online sales stores such as Amazon.com.

The simplest way to do sentiment classification is using the lexicon-based approach (Pang and Lee 2008), which calculates the sum of the number of the positive sentiment words and the negative sentiment words appearing in the text file to determine the sentiment of the text file. Intuitively, it is supposed to perform well since people do use sentiment words to express their sentiments. However, it does not work as well as we expect considering people do not always express their feelings in this way. People may use objective words to show sentiments, for example "AirCanada has seriously tested my patience today". People also may express their complaints in an ironic way, for example "Thank you Delta for having the rudest employees and almost making me miss my flight".

Rather than categorizing sentiments into three groups, there also have been works that categorize sentiment into six groups. This work develops an approach for sentiment classification of tweets about airline services, which is sentiment classification research in a specific domain and in a specific platform.

In the survey done by (Pang and Lee, 2008), a broad view of sentiment classification methods are discussed, including the machine learning techniques and traditional classification methods. The machine learning techniques have widely applied in text classification area and most of them are supervised learning classification methods. In the supervised learning methods, two datasets are provided (Han, Kamber and Pei 2012). One is the training dataset and the other one is the test dataset. The training dataset is used to train the models, in which process the differentiating characteristics of the documents are identified. The test dataset is used to validate the performance of the

model which is trained by the training dataset. Several machine learning sentiment classification methods have been developed such as the Naïve Bayes (NB) method, the maximum entropy (ME) method, and the support vector machine (SVM) method (Han, Kamber and Pei 2012, 327). These text classification methods have shown very good performance in text categorization.

The Naïve Bayes method has been a very popular methods in text categorization because of its simplicity and efficiency. (Melville, Gryc and Lawrence 2009). The theory behind is that the joint probability of two events can be used to predict the probability of one event given the occurrence of the other event. They key assumption of the Naive Bayes method is that the attributes in classification are independent to each other, which considerably reduces the computing complexity of the classification algorithm.

The Support Vector Machine (SVM) method was considered the best text classification method. (Xia, Zong and Li 2011). The Support Vector Machine method is a statistical classification approach which is based on the maximization of the margin between the instances and the separation hyper-plane. This method is proposed by Vapnik (History of SVM 2014).

Different from other machine learning methods, the K-nearest neighbors (KNN) method does not extract any features from the training dataset but compare the similarity of the document with its neighbors (Han, Kamber and Pei 2012, 423). For a document d, the KNN classifier finds the k-nearest documents and calculates the numbers of the documents in different classes and the document will be classified to the class which hold most neighbors.

Many comparative research have been done for different sentiment classification approaches. Songbo Tan (Tan and Zhang 2008) compared four feature selection approaches and five machine learning methods on Chinese texts. He concluded that the Information Gain algorithm outperforms other feature selection approaches and the Support Vector Machine approach works best in sentiment classification. Yi et al. (Yi and Niblack 2005) also discovered that the Support Vector Machine approach performs better than the Naïve Bayes approach and an N-gram model do.

A comparative study on feature selection in text categorization by Songbo Tan, the Information Gain algorithm outperforms other algorithms in feature selection in text categorization (Tan and Zhang 2008). In their work, they evaluated the different feature selection algorithms by applying the features to a K-Nearest Neighbor (KNN) classification model and a linear regression model. So in our work, we adopted the Information Gain algorithm to select features for sentiment classification.

Prabowo and Thelwall combine the ruled-based classification and the machine learning methods, and proposed a hybrid method (Pak and Paroubek 2010). Their method yielded satisfactory results when applied to movie reviews, product reviews and Myspace comments (Pak and Paroubek 2010).

Li, Feng and Xiao used a multi-knowledge based approach in mining movie reviews and summarizing sentiments, which proved very effective in applications (Li, Feng and XiaoYan 2006). Ding, Bing and Philip proposed a holistic lexicon based approach to classify customer' sentiments towards certain products and achieved high accuracy (Ding, Bin and Yu 2008). This approach is content dependent and needs to select feature words, phrases from training data.

Lin and He proposed a probabilistic modeling framework called Joint-sentiment model, which adopted the unsupervised machine learning method (Lin and He 2006). In their research, they applied their model in movie reviews and classify the review sentiment polarities.

The ensemble classification approach is a combination of different classification approaches and classify the documents based on the classification output with the majority vote method. Rui Xia build an ensemble sentiment classification model which integrates two feature sets and three sentiment classification approaches (Xia, Zong and Li 2011). He adopted the features based on the Part-of-Speech tags and the features based on the word relations, and the classification method are the Naive Bayes method, the Maximum Entropy method and the Support Vector Machine method.

2.4 Related Work in Twitter Sentiment Classifications

Depending on what text files are used to apply sentiment classification, sentiment classification can be categorized to many different specific application groups, such as movie review sentiment classification, product review sentiment classification, blog sentiment classification and social network sentiment classification and so on (Pang and Lee 2008).

Movie review, and product review sentiment classification apply to reviews or comments on certain objects and services. Because these sentiment classification techniques can be applied to many real world companies such as Amazon, there have been much research work on review sentiment classification. Blog and social network sentiment classification are applied to the posts that are published on the Internet. Unlike reviews sentiment classification, these sentiment classification work is not about feedback toward certain products or service but can be the authors' opinions about anything. Many approaches (Pang and Lee 2008) have been developed for blog sentiment classification and social network sentiment classification.

They are many different social network platforms such as Facebook, Twitter and Instagram. They have their own unique characteristics from each other and different sentiment classification approaches have been developed for them (Pak and Paroubek 2010). For example, Twitter allows users to post no more than 140 characters for each post, which makes Twitter sentiment classification different from other text sentiment classification because many text files like blogs are much longer than 140 characters. Many techniques used in text file sentiment classification do not perform well in Twitter sentiment classifications because of its length restrictions. For example, Information retrieving and summarization approaches that perform well in paragraph sentiment classification are not very useful for twitter sentiment classification because there is not much information to retrieve and summarize to classify its sentiment. Besides that, traditional and simple classification approaches such as the Lexicon-based approach also perform better in long length text files than in tweets because there are much higher

probabilities to see sentiment words appearing in long paragraphs than in tweets, which are limited to 140 characters.

Because Twitter provide public access to its streaming and historical data, it has become a very popular data source for sentiment analysis and much work has been done in this area.

J.Read used emoticons, such as ":-)" and ":-(", to collect tweets with sentiments and to categorize them into positive tweets and negative tweet. They adopted Naive Bayes approach and the Support Vector Machine approach, both of which reached accuracy up to 70% (Read 2005).

In the research of Wilson et al, they used hashtags to collect tweets as the training dataset. They tried to solve the problem of wide topic range of tweet data and proposed a universal method to produce training dataset for any topic in tweets (Wilson, Wiebe and Hoffmann 2005). Besides that, Wilson et al. also considered three polarities in tweets sentiment classification, which includes positive sentiment, negative sentiment and neutral sentiment. Unigrams, bigrams and POS features were taken into account as classification features, and emoticons and other non-textual features were also considered. In their experiments, it showed that training data with hashtags could train better classifiers than regular training data do. But in their research, the dataset were from libraries and they neglected the fact that tweets with hashtags are only a small part of real world tweets data.

Pak and Paroubek proposed an approach, which can retrieve sentiment oriented tweets from the twitter API and classify their sentiment orientations (Pak and Paroubek 2010). From the test result, they found that the classifier using bigram features produces highest classification accuracy because it achieves a good balance between coverage and precision. Their work in tweets sentiment mining is not domain specific, which means applying their methods in domain specific mining will yield different results. Besides that, the data source is biased as well because they retrieved only the tweets with emoticons and neglected all other tweets that didn't contain emoticons, which are the majority of

tweets. In this work, they didn't consider the existence of the neutral sentiment and classifying these tweets is very important for tweet sentiment analysis.

2.5 Related Work in Sentiment Classification on Twitter Data about Airline Services

The challenges in twitter sentiment classification not only come from the fact that each post is not allowed to exceed 140 characters but also because the sentiment of the tweets can be very dependent on the scenarios the users are involved in but the context of the scenarios is not provided in the tweets. For example, "Cancelled again, It's the fourth time" can be a tweet with negative sentiment if it is about taking flights but also can be a neutral sentiment tweet if it is talking about the user frequently cancelling some subscriptions. Because of this, Twitter sentiment classifications are very domain dependent.

In sentiment classification, features are important because they are the attributes that determine texts' sentiments (Pang and Lee 2008). Features can be unigrams which are words, or N-grams. Twitter sentiment classifications are domain dependent because those features are domain dependent, and sentiment features in one domain may not be sentiment features in other domains at all. For example, in the stock market area, the word "bear" means negative sentiment since it is a term describing bad performances in the stock market but it means no sentiment at all in most other domains. So the unigram "bear" can be extracted as a feature in the stock market area but not in other areas such as airline services.

There have been several works about twitter sentiment classification, and most of them are not domain dependent. Researchers have been trying to develop approaches to classify twitter sentiment in a general way but have not achieved an outstanding result (Pang, Lee and Vaithyanathan 2002).

Little work has been done on twitter sentiment classifications of airline services. Conventional sentiment classification approaches, such as Naive Bayes approach, have

been applied to some tweet data and the performance was not bad (Pak and Paroubek 2010)

Lee et al. used twitter as the data source to analyze consumers' communications about airline services (Pang, Lee and Vaithyanathan 2002). They studied tweets from three airline brands: Malaysia Airlines, JetBlue Airlines and SouthWest Airlines. They adopted conventional text analysis methods in studying twitter users' interactions and provided advices to airline companies for micro-blogging campaign. In their research, they didn't adopt sentiment classification on tweets, which will be more salient for airline services companies to understand what customers are thinking.

In the handbook of "Mining Twitter for Airline Consumer Sentiment", Jeffery Oliver illustrates classifying tweets sentiment by applying sentimental lexicons (Oliver 2012). This handbook suggests retrieving real time tweets from Twitter API with queries containing airline companies' names. The sentiment lexicons in this method are not domain specific and there is no data training process or testing process. By matching each tweet with the positive word list and the negative word list, and assigning scores based on matching result to each tweet, they can be classified as positive or negative according to the summed scores. The accuracy is unknown since it is not considered in this book. In our work, this method was applied and tested with labeled data. It can yield inaccurate testing results because sentiment classifications are highly domain specific.

Adeborna et al. adopted Correlated Topics Models (CTM) with Variational Expectation-Maximization (VEM) algorithm (Adeborna and Siau 2014). Their lexicons for classification were developed with AQR criteria. In Sentiment detection process, the performances of the SVM classifier, the Maximum Entropy classifier and Naive Bayes classifier were compared and Naive Bayes classifier was adopted. Besides that, tweets are categorized by topics using the CTM with the VEM algorithm. The result of this case study reached 86.4% accuracy in subjectivity classification and displayed specific topics describing the nature of the sentiment. In this research, the overall dataset they used contains only 1146 tweets, which includes only three airline companies. Besides, the author only used unigrams as sentiment classification features in Naive Bayes classifier,

which can cause problems because phrases and negation terms can change sentiment orientation of the unigrams in sentences. In my work, more than 100,000 tweets are collected, and Unigrams, Bigrams, Trigrams and the information gain algorithm will be applied into feature selection, which is much less biased. Besides that, their work did not present details about the classification approaches and comprehensive evaluations. However, my work not only contains the analysis of tweets with different sentiments but also includes the comparison of the performances of different approaches.

CHAPTER 3 DATA PREPARATION

3.1 Introduction

I adopts seven sentiment classification approaches including one approach presented by myself. The proposed approach can be divided into two parts. The first part is the model construction and the second part is the class prediction. The model construction consists of the feature selection process and the model training process. Because the high dimensionality of the word vectors transformed from raw text data is not computationally efficient, it is necessary to select features that play an important role in determining documents' sentiment classes. For example, the stop words distribute evenly in all of the text files and they are usually useless for sentiment classification. In the feature selection process, not only the unigrams are considered, bigrams and trigrams are also considered because phrases can have different meanings from the single words making them up. In this approach, I used the Information Gain algorithm to select features against the sentiment classes because the transformed gram vectors are binary dataset and the Information Gain algorithm performed better than the Gain Ratio algorithm in the data with low cardinality.

3.2 Data Collection

Twitter provides free connections to Twitter API to everyone. There are the Twitter Search API and the Twitter Streaming API (Twitter.com 2014). The Twitter Streaming API gives people low latency access to Twitter's global stream of tweet data. As part of Twitter's REST API, the Twitter Search API allows people to query against the indices of recent or popular tweets, which means people can retrieve historical tweet data with keywords and other feature restrictions.

Considering that I do not need exhaustive data retrieval of real time tweet data about airline services, I only used the Twitter Search API to retrieve tweets data about airline services.

Using Twitter Search API to retrieve tweets by key words might cause ambiguity. For example, searching tweets with the key word 'Delta', which is the biggest airline brand in North America, might collect tweets that convey geographic information other than Delta airline services feedback. In our work, we search each airline brand with a combination of two key words including the brand's name and 'flight' to collect tweets that convey airline services feedback. For example, I search tweets data about Delta airlines, I used the keyword “flight” and “Delta” combined to retrieve tweets from the Search API. For some airline companies, I just used the brand name to retrieve tweets data since the brand name brings no ambiguity in collecting tweets. In my work, I used only the keyword “Aircanada” to retrieve tweet data about Air Canada’s services. In this case, it might cause ambiguity but it can retrieve more data. To get a full and comprehensive coverage of English tweets about airline services, most of the airline services brands in North America were considered. Based on the list, the largest airlines in North America are: Delta Airlines, JetBlue Airways, United Airlines, Air Canada, SouthWest Airlines, AirTran Airways, WestJet, American Airlines, Frontier Airlines, Virgin Airlines, Allegiant Air, Spirit Airlines, US Airways, Hawaiian Airlines, Skywest Airline, Alaska Air Group (Major Canadian and US Airlines 2014). Retrieving tweets about those brands can build the best dataset for sentiment analysis of airline services. The list of airlines the tweet retrieved about is shown in table 1.

For sentiment analysis, only the text of tweets was considered; there was no other constraint for retrieved tweets except the language is set to English. I retrieved tweets with those sixteen brands' names and the key word 'flight' from Twitter Search API. However, Twitter Search API only returns 3000 tweets in maximum and 200 tweets in minimum for a single query each time. Because timing factors were not considered in my work, I kept retrieving tweets randomly in different periods until the data volume meets my requirement. At the end, I got 25086 tweets for Delta Airlines, 22060 tweets for United Airlines, 16211 for SouthWest Airlines, 16567 tweets for Air Canada and 13807 tweets for JetBlue Airways and 14135 for rest of the airline companies. Because the volume of tweets returned from Twitter Search API for each brand indicates that its market share, the fractions of tweets for each brands were not adjusted. In total, there was a dataset containing 107866 tweets in my work

Table 1 List of Airline companies

	Airline	Nationality
1	Delta Airlines	US
2	JetBlue Airways	US
3	United Airlines	US
4	Air Canada	Canada
5	SouthWest Airlines	US
6	AirTran Airways	US
7	WestJet	US
8	American Airlines	US
9	Frontier Airlines	US
10	Virgin Airlines	US
11	Allegiant Air	US
12	Spirit Airlines	US
13	US Airways	US
14	Hawaiian Airlines	US
15	Skywest Airline	US
16	Alaska Air Group	US

These tweets include original tweets and retweets. I discarded the irrelevant tweets and labeled each relevant tweet in the dataset as positive sentiment, negative sentiment or neutral sentiment manually. In the dataset, 4288 tweets were labeled positive, 35876 tweets were labeled negative, 40987 tweets were labeled neutral and 26715 tweets were discarded for being irrelevant. It reveals that customers prefer to complain than to give appraise on Twitter about their flight experiences.

Table 2 Sentiment distribution of the tweets

class	positive	negative	neutral	irrelevant
tweets	4288	35876	40987	26715

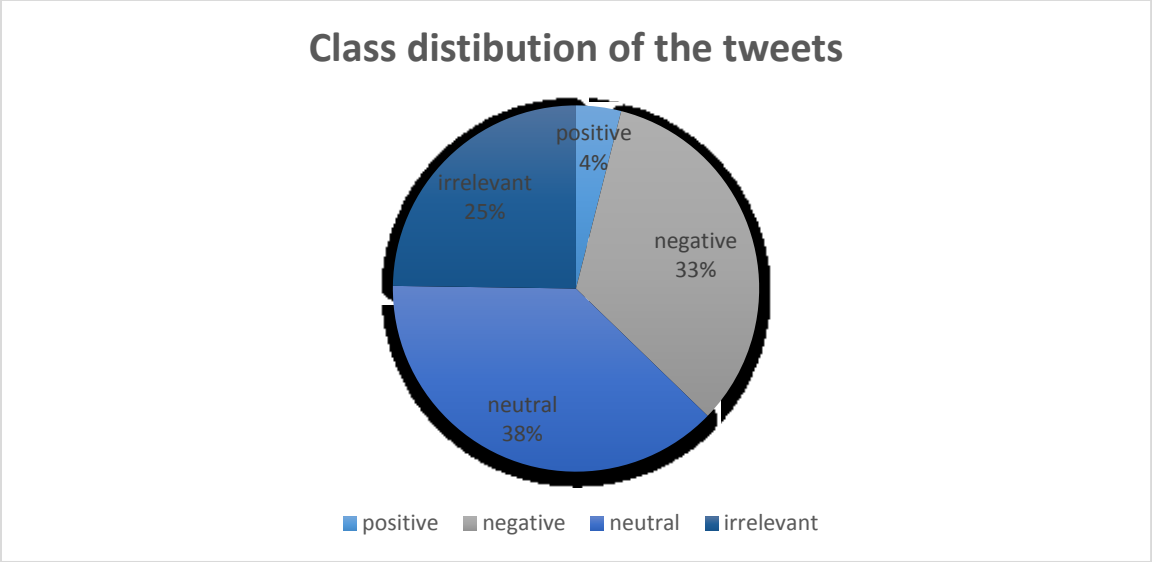

Figure 1 Sentiment distribution of the tweets

3.3 Data Pre-processing

In data analysis, data pre-processing is very necessary because the raw data contains noise and duplicates, or the original data are not suitable for analysis methods. This is more important in text classification because text data are so different from numeric data, which means we must convert text data into a format that can be analyzed. This also requires data to be labelled if the classification approaches are supervised learning classification approaches and model evaluations are involved. Besides that, real world text data often contain a lot of typos, abbreviations and symbols, which makes classification results inaccurate. For example, in social network postings, people type "THX", "Thanks" or "Thenks" to say thanks, in which the first one is abbreviation and the last one is a typo. Even though it is very simple for human beings to understand that these words mean the same thing, this brings huge difficulties for machine learning algorithms to figure it out.

The first procedure in data pre-processing is to remove duplicate documents. As a social network application, Twitter allows users to retweet other users' posts and share common topics. There also exists many robots which keep posting same contents. This yields numerous tweets with exactly the same contents and those duplicate tweets can change

the weights of the original single tweet. To remove the duplicate tweets, I developed a program in R, which removes the duplicate tweets by two steps. In the first step, the program sorts the tweets alphabetically so all of the duplicate tweets are grouped together. In the second step, the program scans each tweet and deletes the following tweets which are identical to the proceeding tweet. I set the threshold to 0.8 for the tweets similarity evaluation, which means if two tweets are 80% identical then they are considered duplicates. The original dataset collected from the Twitter Search API has 146532 tweets and 107866 tweets left after removing duplicates.

The second procedure in data pre-processing is to label the tweets I collected from the Twitter Search API. I developed a graphic user interface (GUI) in R to read and label the tweets one by one. I set four categories for the tweets to be labelled as, which are positive, negative, neutral and irrelevant.

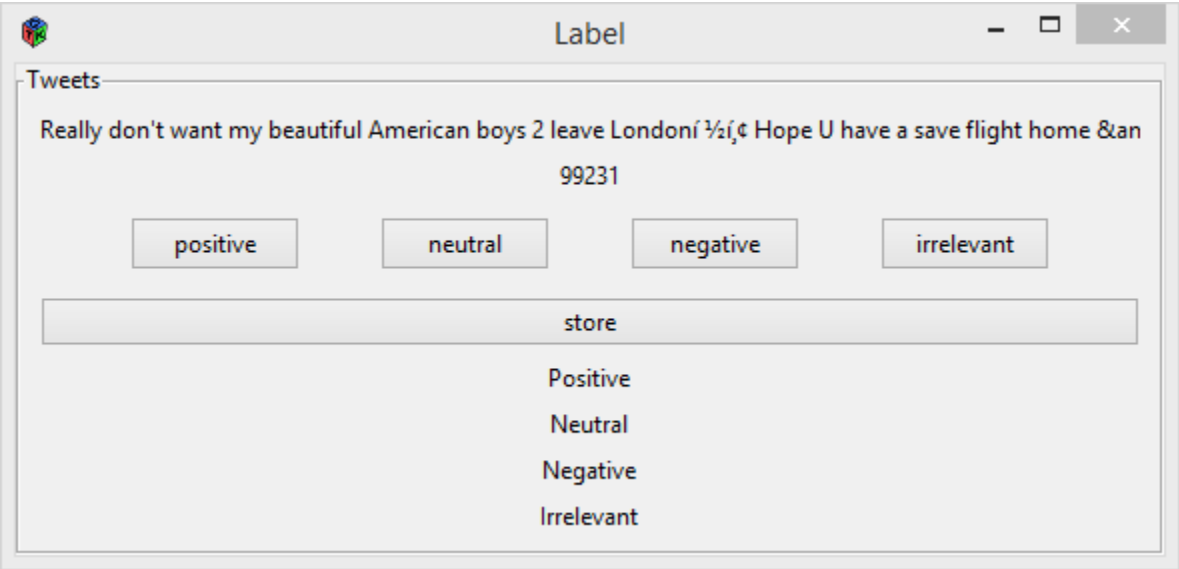

Figure 2 Tweet labelling Graphic User Interface

The graphic user interface allows me to click the button to label a tweet and display the tweet next to it. I scanned and labelled the 107866 tweets. Even though the tweet retrieval strategy is very successful, many tweets were labelled as irrelevant since they do not represents passengers' sentiment towards airline services. For example, news posts about airline and flights were labelled irrelevant since they are objective facts posted by the

news publishers. Airlines companies' assistant accounts' replies to customers were also labelled irrelevant since they represent the airline services companies and their sentiments are meaningless in this research. The class distribution of the tweets I collected is extremely uneven, there are 4288 positive tweets, 35876 negative tweets, 40987 neutral tweets and 26715 irrelevant tweets in my dataset. The irrelevant tweets are discarded and the remaining dataset is resampled to produce a dataset with an even class distribution.

For model training and classification, balanced class distribution is very important to ensure the prior probabilities are not biased caused by the imbalanced class distribution. For example, in the Naive Bayes classification model training, as shown in Equation 1.

$$p(S|D) = \frac{p(S)}{p(D)} \prod_i^n p(w_i|S) \quad (1)$$

The probability of the document D being classified as the sentiment class S is $p(S|D)$, which is determined by $p(S)$, $p(D)$ and $p(w_i|S)$. If the class distribution in the training data is not balanced, then $p(S|D)$ will be biased because $p(S)$ are different for different classes. Another extreme example is that a dataset which contains 100 document and 99 of them are in the negative sentiment class and only 1 is in the positive sentiment class, then this single positive document will have the probability of 99% being classified as negative in the classification model without considering other factors. Besides that, a dataset with balanced class distribution is also important for classification evaluation because the overall accuracy for all of classes will be biased caused by different weights of the accuracy of different classes. For example, considering the dataset I mentioned previously, which consist of 99 documents in the negative class and 1 document in the positive class, we can get 99% accuracy by just classify all of documents as negative. However, this kind of high accuracy for classification is meaningless since it does not tell the difference between two classes.

I randomly resampled a dataset with exact same number of documents for each sentiment class. I got a dataset with 4288 documents for each sentiment class and 12864 documents in total.

The next step for tweet pre-processing is to remove noise and useless characters in the tweets. I developed a program in R to scan all of the tweets and remove the punctuations, symbols, emoticons and all other non-alphabet characters from the tweets. Besides that, I also removed web links and decapitalized all of tweets because these features provides little information in sentiment classifications.

3.4 Stemming

Unlike formal publications, the texts on social networks and blogs are unedited texts, which means they are not bound to strict grammar rules and the requirements of correct spelling. Typos and abbreviations happen a lot in social network postings, especially in tweets. For example, the tweet "Thx Aircanada, ur flight is canceled again, screw u" actually means "Thanks Air Canada, your flight is cancelled again, screw you". In this tweet, the word "thanks, your, you" are abbreviated to "Thx, ur, u" and the word "canceled" has another form which is "cancelled". Different forms, typos and abbreviations bring more difficulties in sentiment classification because they decrease the features' power in determining the documents' sentiment and make the features sparser than they really are.

To solve this problem, I implemented stemming techniques to stem the different inflections of the words to their word stem. For example, all of the different forms and inflections of the word "cancel" such as "cancelling", "cancelled" and "canceled" can be converted to an identical stem word "cancel" though stemming techniques. However, stemming techniques cannot solve all of problems brought by the informal typing in social network texts. For example, one of the abbreviation of "thanks", 'Thx", may not be stemmed into one word stem, and the abbreviation "u" and the word "you" will not be stemmed together. Nevertheless, stemming techniques still can considerably reduce the sparsity of the features. I adopted the Snowball stemmer algorithm in Weka.

3.5 Transformation

As discussed before, texts cannot be used to implement data analysis directly because almost all text classification algorithms only deal with digital data. So before sentiment

classification, the tweet dataset must be transformed to a form which is analyzable. The most common way to do that is to convert text data into a matrix, in which the columns represent the words appearing in the dataset and the rows represent the documents.

The first step of data transformation is to make all distinct words appearing in the tweet data a set. This set contains all the distinct words appearing in the tweet dataset and no duplicate words exist in this set. The second step of data transformation is to make a matrix and in this matrix, each column represents a word appearing in the word set from the previous step. By doing this, we can convert each of the tweet documents into a binary row of the matrix. For any word appearing in a tweet document, there must be a column in the matrix representing it. When converting tweet documents to binary matrix rows, for each column in the row, if the word it represents appears in this tweet document then its value is set to 1, and if the word it represents does not appears in this tweet document then the value is set to 0.

Table 3 Samples of raw tweet data

1	aircanada im glad you are because this isn't how one ought to treat premium customers
2	aircanada if you ever talk shit about phil kessel again ill stop flying with you
3	aircanada whats going on with light to sfo
4	delta flight in more days
5	Flight thank you for being such a great representative of uga youre a dgd now go make us proud you gradygrad
6	United we made flight thank you from all connecting passengers

Table 3 and Table 4 shows an example of how data transformation works. After stemming and transformation, I got the matrix of documents with 60 columns and 6 rows. These 60 columns are all the distinct stemmed words appearing in the text dataset and the rows represent the tweet documents. For example, the first three documents contain the word "aircanada, and the stemmed word for it is "aircanad", and the values in the column

of “aircanad’ in the three rows of matrix are 1 and the values of other rows in this column is 0.

Table 4 Transformed tweet data

1	about	again	aircanad	ar	……..	unit	us	we	your
2	0	0	1	1	……	0	0	0	0
3	1	1	1	0	……..	0	0	0	0
4	0	0	1	0	……..	0	0	0	0
5	0	0	0	0	……..	0	1	0	1
6	0	0	0	0	……..	1	0	1	0

In topic classifications, the frequency of words appearing in documents are considered because the more frequently a feature appears in one document than it does in another document means this document is more related to the topic this feature represents than it is related to other topics. The length of these text files is often very long, such as articles and blogs. However, the restriction to the number of characters allowed for each tweet is 140. Each one rarely contains the same unigrams except the stop words. So in my data transformation process, only the occurrences of the words but not the frequencies of the words are considered. Besides the fact of rarity of repeating unigrams in a single tweet document, another good reason to convert tweet documents into a binary matrix is that binary matrices are much more computing efficient and inexpensive than numeric matrices considering the huge dimension of the word matrices.

3.6 N-grams

In sentiment classification, features can be unigrams, bigrams, trigrams and more. A unigram is a single word, a bigram is a phrase made of two single words and a trigram is a phrase made of three single words. A tweet document can be transformed into many kinds of different features. For example, the sentence “How are you” has three unigrams, “How”, “are” and “you”, two bigrams “How are”, “are you”, and one trigram, “How are you”.

The reason for taking N-gram features from text documents is because N-gram features indicate different sentiment information than the unigrams do. Sometimes it is because the preceding word in a N-gram phrase is a negation, which can reverse the sentiment orientation of the unigrams in the phrase to the opposite sentiment orientation and give the N-gram phrase the opposite sentiment orientation to the unigrams in it. For example, in the sentence "I am not happy", the unigram "happy" has a positive sentiment meaning but the negation "not" reverses the sentiment orientation of the sentence and makes this sentence a negative sentiment sentence. More than that, objective unigrams can make subjective bigrams. For example, in the sentence " AirCanada, I will stop flying with you", the unigrams "stop" and "flying" are objective but the bigram "stop flying" is a subjective phrase in the airline service domain.

In my data transformation process, I do not transform the tweet documents to a matrix with columns represent only unigrams, but a matrix with columns representing unigrams, bigrams and trigrams. The transformation for unigrams is discussed in the last section, and here I explain how to transform bigrams and trigrams to a matrix. In the bigram transformation, every two consecutive words in a tweet document are considered a bigram. So for a tweet document with N unigrams, there are (N-1) bigrams for this tweet document. In the trigram transformation, every three consecutive words in a tweet document are considered a trigram. So for a tweet with N unigrams, there are (N-2) trigrams.

Actually, we can consider even longer multi-grams in sentiment classification, such as four-grams or five-grams. However, there are several reason for not doing that. First of all, it will make the transformed matrix even sparser and make the sentiment classification not implementable. Besides, as the length of the N-gram becomes longer, the N-gram features for each tweet document will be more distinct from the N-gram features from other tweet documents. For example, in the sentence "Thank you Aircanada, it was a really smooth flight and I will definitely fly you again", the unigram features such as "smooth" and the bigram features such as "Thank you" are highly likely to appear in other tweet documents as well. However, the six-gram feature "I will definitely fly you again" is unlikely to appear in other documents. There has been research that

from bigrams to multi-grams, the information gain for each level of N-gram decreases as the length of the multi-grams increases. (Cheng, et al. 2007) shows that the Information Gain decreases as the feature length increases.

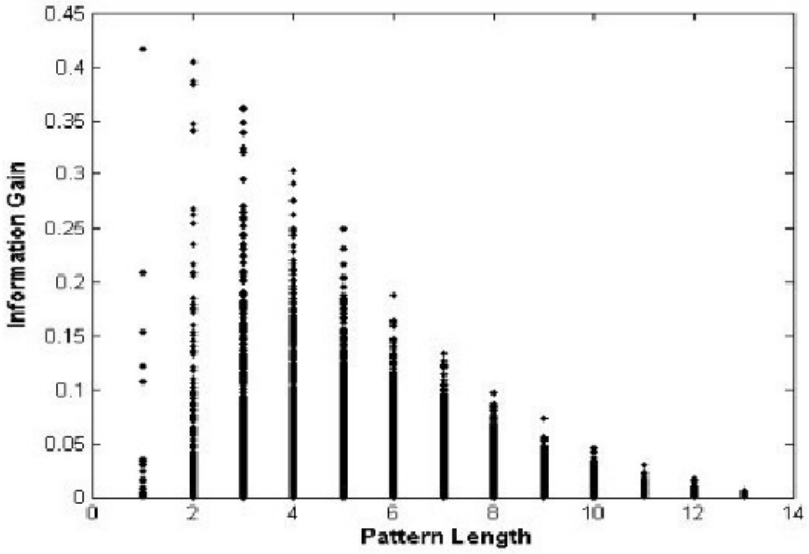

Figure 3 Information Gain vs. Feature Length on UCI data (Cheng, et al. 2007)

Table 5 is an example of data transformation with unigrams, bigrams and trigrams. The dataset is from the previous example.

Table 5 Transformed data of Unigrams, Bigrams and Trigrams

1	about	again	……..	being such	flight thank	……..	Youre a dgd	Us proud you	……..
2	0	0	……	0	1	……	0	0	……
3	1	1	……..	0	0	……..	0	0	……..
4	0	0	……..	0	0	……..	0	0	……..
5	0	0	……..	1	1	……..	1	1	……..
6	0	0	……..	0	1	……..	0	0	……..

3.7 Feature Selection Algorithms

The one of the most important parts in sentiment classification is the feature selection. The word matrix resulting from the data transformation process is too big for classification and too many features can also cause over-fitting problems. In my experiment, the tweet dataset has 12,864 tweet documents. Even though I only consider the unigrams, the bigrams and the trigrams as features, I still get 129,220 features, which make the matrix 129,220*12,864 elements. It is very expensive and not implementable for a regular computer to do sentiment classification of such a big size. So it is necessary to select the features that play more important roles in sentiment classification. There are several methods of evaluating the importance of the features in sentiment classification, such as Information Gain algorithm, Gain Ratio algorithm and Gini Index algorithm.

A) Information Gain Algorithm

Information Gain algorithm was developed based on the work done by Claude Shannon on Information theory, which studies the value of the "information content" of messages (Han, Kamber and Pei 2012, 336). Information Gain is an evaluation of how well an attribute performs in classifying the documents and how much entropy has been reduced after classification. The entropy of a dataset means the "impurity" or the extent of messiness of the classes grouping. For example, in each group of a three group tweet documents, the sentiment classes are identical within the group but different from other groups, then this dataset entropy is very low. An attribute that can considerably reduce the entropy of the dataset produces high information gain and should be selected as a feature for sentiment classification. The information needed to classify the dataset D can be expressed as:

$$Info(\mathrm{D}) = -\sum_{i=1}^{m} p_i \log_2(p_i) \qquad (2)$$

Where p_i is the probability that an arbitrary tweet document in the tweet dataset belongs to the sentiment class C_i. m is the number of classes in the tweet dataset. So $Info(\mathrm{D})$ is the average amount of information to get the sentiment class of a tweet document in the tweet dataset D.

By applying an attribute A to classify the tweet documents in the tweet dataset D, we can get a new dataset in which the entropy is changed. For this new tweet dataset, the information needed to reach a 100 percent correct classification can be written as:

$$Info_A(\mathrm{D}) = \sum_{j=1}^{v} \frac{|D_j|}{|D|} Info\big((D_j\big) \tag{3}$$

where v is the number of distinct values in the attribute A. The term $\frac{(D_j)}{(D)}$ is the weight for each partition D_j. $Info_A(\mathrm{D})$ is the information needed to classify an arbitrary tweet document from the new dataset.

To evaluate the performance of this attribute A in classifying the tweet documents in the tweet dataset D, we can calculate the difference between $Info_A(\mathrm{D})$ and $Info(\mathrm{D})$ to get the information gain of classification with the attribute A.

That is:

$$Gain(A) = Info(\mathrm{D}) - Info_A(\mathrm{D}) \tag{4}$$

In my feature selection, each unigram, bigram and trigram are evaluated against the sentiment class individually and ranked by the value of information gain.

B) Gain Ratio Algorithm

Information Gain algorithm is very intuitive but also has disadvantages in some cases. Information Gain algorithm does not take into account the cardinality of the attribute values. For example, the attribute ID for the tweet dataset, which contains only distinct values and can perfectly separate the tweet documents to different partitions and each partition has an identical class. However, the result of this classification is meaningless because it does not provide any useful information in predicting new tweet documents even though this attribute has very high information gain value in classification.

To solve this problem, the Gain Ratio algorithm was developed as an extension of the Information Gain algorithm (Han, Kamber and Pei 2012, 340). It normalizes the information gain by using a "split information" value, which is:

$$Splitinfo_A(\mathrm{D}) = -\sum_{j=1}^{v} \frac{|D_j|}{|D|} \times \log_2(\frac{|D_j|}{|D|}) \quad (5)$$

The value of the $Splitinfo_A(\mathrm{D})$ means the information generated by splitting the dataset, D, into v partitions based on the attribute A. The gain ratio is defined as:

$$GainRatio(\mathrm{A}) = \frac{Gain(A)}{Splitinfo_A(\mathrm{D})} \quad (6)$$

C) Gini Index Algorithm

For attributes with multiple values, there is another way to evaluate the attributes than the Gain Ration algorithm such as the Gini Index algorithm (Han, Kamber and Pei 2012, 341). The Gini Index algorithm considers that combinations of the different values in the attributes and does a binary split for each possible value combination. For example, for attribute A with v distinct values, there are (2^v-2)/2 possible subsets of values with the full set and the empty set removed. The Gini Index algorithm computes a weighted sum of the entropy of each split. The overall Gini Index for the attribute A can be written as:

$$Gini_A(\mathrm{D}) = \sum_{i=1}^{n} \frac{|D_i|}{|D|} Gini(D_i) \quad (7)$$

Then the reduction of impurity of classifying the tweet document D with the attribute A can be calculated by the formula:

$$\Delta Gini(A) = Gini(D) - Gini_A(\mathrm{D}) \quad (8)$$

The attribute that maximizes the impurity reduction has the minimum Gini index.

In my data matrix, all of the attributes are transformed to binary attributes, which indicate the occurrences of the attributes. The class is a nominal attribute with three classes.

Since Gain Ratio algorithm and the Gini Index algorithm were developed to evaluate the performance of multi-value attributes in classification. Information Gain algorithm is selected to evaluate and rank my features considering the attributes are binary and the Information Gain algorithm is very efficient.

3.8 Feature Selection Implementation

I used Weka to compute the Information gain for each attribute and rank them in decreasing order. In Weka, I select the supervised filter, Attribute Selection to implement feature selections. In the Attribute Selection filter, I selected the InforGainAttributeEval algorithm for the evaluator option and the Ranker algorithm for the search option. I kept the default value of the threshold for the Ranker algorithm, which is -1.7976931348623157E308. By keeping the threshold default value, the algorithm ranks all of the attributes decreasingly without removing any attributes. If the default value is set to other positive values, the attributes, of which the information gain are less than the threshold value will be removed.

I exported the ranking results and plotted them in a line chart to see the rates of information gain decreasing. As shown in Figure 4, the x coordinate is the ranks for the attributes and the y coordinate is the information gain for each attribute.

There is a cutoff of Information Gain around the 18,000th feature in the feature rank. There is a cutoff of the Information Gain between the feature which ranked 1,386 and the features ranked below, the value is 0.002. There is a cutoff of the Information Gain between the feature which ranked 656 and the features ranked below and the value is 0.03. So for my experiment, the features that ranks above the 656^{th} feature are selected.

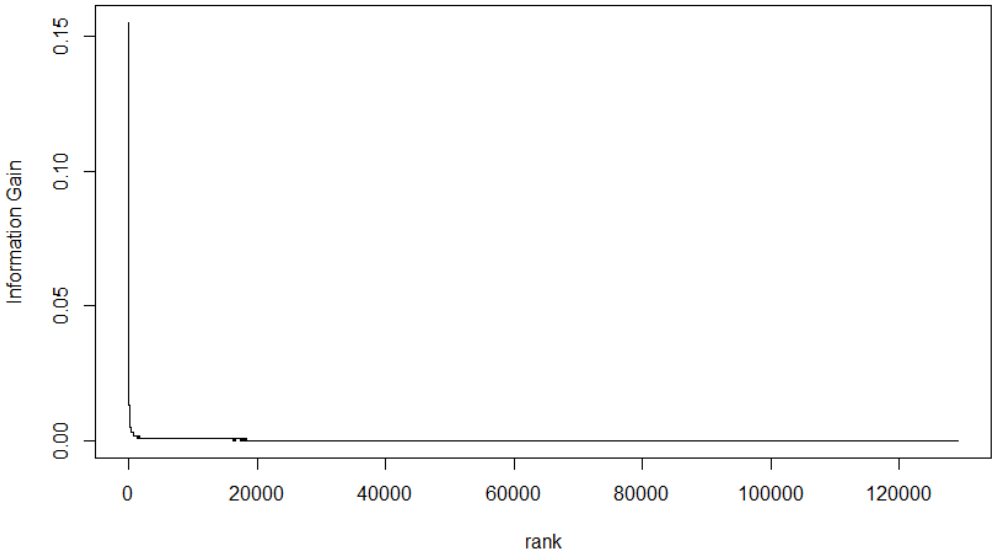

Figure 4 IG of all the features

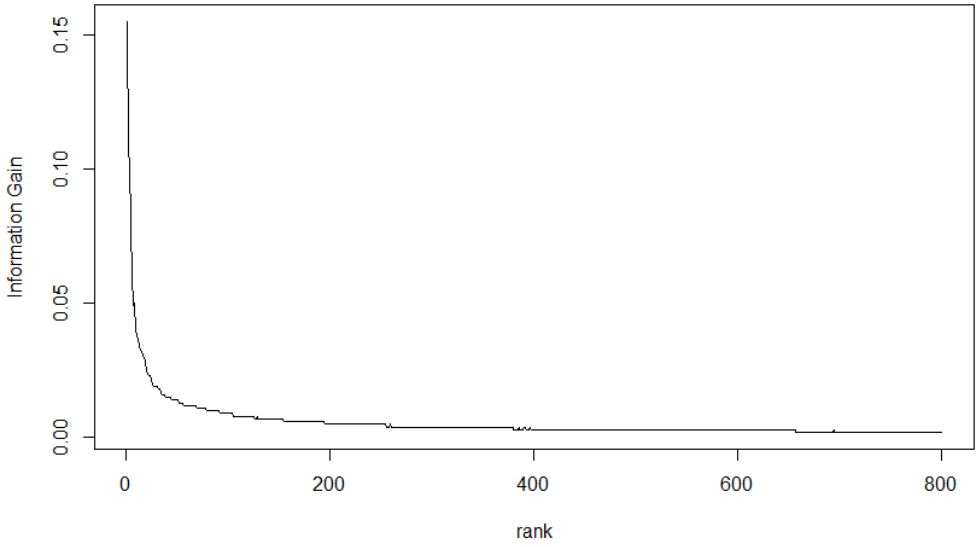

Figure 5 IG of top 800 features

CHAPTER 4 METHODOLOGY AND SYSTEM DESIGN

4.1 Introduction

In the model training process, all the data are divided into training data and test data. The whole dataset will be used to train models and 10-fold validation is used to validate the classification models. The ensemble classifier consists of these five classifiers and each of them predicts the test data classes individually. In the class prediction part, each tweet document has five class prediction results, which come from the five different classifiers. The sentiment class of each document is determined by the five class prediction results with the majority vote method.

4.2 Classification Methods

For machine learning sentiment classification approaches, the second part is model training after the feature selection. However, for some traditional approaches such as the Lexicon-based approach, there is no machine learning techniques involved and no modeling training is required for these approaches. Here I discuss the sentiment classification methods in my work.

4.2.1 The Lexicon-based Method

The Lexicon based sentiment classification method is the simplest and the most intuitive method in sentiment classification. Even though there exist many different versions of the lexicon based classification method, the methodologies are all the same. The Lexicon-based sentiment classifier matches each document with the sentiment word lists, which usually consist of a positive sentiment word list and a negative sentiment word list. Then the classifier counts the number of matches for the positive sentiment and the negative sentiment, and calculates the sentiment scores for each sentiment class. At the end, the classifier compares the scores for each sentiment class and classifies the document to the class with the largest sentiment score. If the sentiment scores are the same for both of the sentiment classes, or no sentiment word matches the sentiment word lists, the document is classified to the neutral sentiment class. Besides that, in some Lexicon-based sentiment

classification methods, different sentiment words have different weights in determining the sentiment classes because, in a same sentiment class, different words express the same sentiment feelings to different extents. For example, in the sentence "You are good at computer science but Peter excels in it", the word "excel" is stronger than the word "good" when indicating their computer science skills. The Lexicon based sentiment classification is applied to the tweet documents that are not transformed to a binary matrix.

In my thesis, I use the word list, "A list of positive and negative opinion words or sentiment words for English" from (Minqing and Bing 2004). This word list has around 6,800 words, and about 3,400 words for each of the two sentiment classes. This word list contains adjectives, adverbs, nouns and even verbs for their sentimental subjectivity. For example there are the words "celebrate", "afford" in the positive word list and the words "difficulty", "disable" in the negative word list. In my work, every word in the word list has same weight in sentiment scoring. In the matching process, every occurrence of a match to the positive word list adds one to the positive score and every occurrence of a match to the negative word list adds minus one to the negative score. For each tweet document, the total sentiment score is the sum of the positive scores and the negative scores.

- If the total sentiment score is larger than zero, the tweet document is classified to positive.
- If the total sentiment score is less than zero, the tweet document is classified to negative.
- If the total sentiment score is equal to zero, the tweet document is classified to neutral.

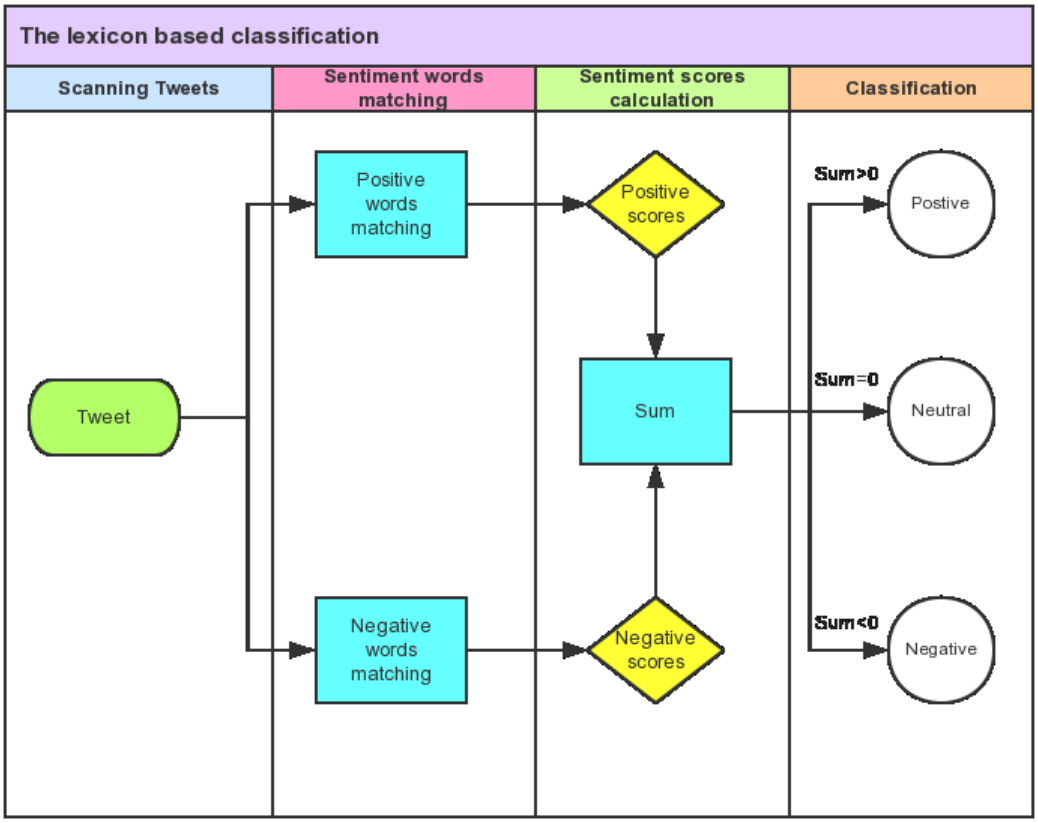

Figure 6 The Lexicon based classification

4.4.2 Probabilistic Sentiment Classification Methods

A) Bayesian Theorem

Bayes' theorem is the foundation for Bayesian classification methods including the Naive Bayes classification method and the Bayesian Network classification method. It was developed by a nonconformist English clergyman, Thomas Bayes, who did early work in probability and decision theory in the 18th century (Han, Kamber and Pei 2012, 350).

Let X be a data document. For the Bayesian theorem, the data document X is considered "evidence" and X is made of a group of attributes. Let H be some hypothesis such as that the data document belongs to a certain class C. For the purpose of classification, we want

to know the probability of $P(H|X)$, which means the probability of the data document belongs to the class C, given the occurrence of the evidence X.

In Bayesian theorem, $P(H|X)$ is the posterior probability, or a posteriori probability for the event H given the evidence of X. For example, suppose the tweet documents contain three attributes, the word "delay", the word "cancel" and the word "smooth", and that X is a tweet document containing the word "delay" and the word "cancel" but not the word "smooth". Suppose that H is the hypothesis that a tweet being a tweet with negative sentiment, then $P(H|X)$reflects the probability that a tweet document being a negative sentiment tweet given it contains the words "delay" and "cancel" but not "smooth"

In contrast, $P(H)$is the prior probability, or a priori probability, of H. For our example, this is the probability of any of the tweet documents being a negative sentiment document regardless of the words it contains. The posteriori probability, $P(H|X)$is dependent on the condition of the event X and, however, the priori probability, $P(H)$is independent of the event X.

Similarly, $P(H|X)$ is also the posteriori probability of X conditioned on the event H. That is, given a tweet document which is a negative sentiment tweet, the probability of this tweet document containing the words "delay" and "cancel" but not the word "smooth".

$P(X)$ is the priori probability of X regardless of the event of H. In this example, it is the probability of a tweet document containing the word "delay" and "cancel" but not the word "smooth".

Because the probability of the co-occurrence of H and X is set, we have the equation that:

$$P(H|X)P(X) = P(X|H)P(H) \tag{9}$$

So the posteriori probability for the hypothesis given the condition of X can be written as:

$$P(H|X) = \frac{P(X|H)P(H)}{P(X)} \tag{10}$$

In our example, the posteriori probability of the tweet document belonging to the class C given the tweet document containing the word “delay”, ”cancel” but not the word “smooth” is the product of the posteriori probability $P(X|H)$ and the priori probability $P(H)$divided by the priori probability $P(X)$.

B) Naive Bayes Classification Method

The Bayes theorem is easy to understand. However, it becomes very impractical when applying the formula in the real world classifications. Because X represents a pattern of values for a group of attributes, when the number of attributes becomes very large, the distribution of X becomes very sparse and it is not implementable. In the example mentioned in the previous section, there are only three attributes, which are the words “delay”, “cancel” and “smooth”. So there are 2^3 possible events of X. However, for sentiment classification, the dimensionality can easily outnumber thousands of attributes and will have 2^N possible events of X, in which N is larger than a thousand.

To overcome this problem, Naïve Bayes method was developed. It assumes that the attributes are independent to each other but only correlated to the target class, which considerably reduces the complexity of computing and solves the problem of sparsity.

Let D be a training set of tweet documents and their associated class labels. Each of the tweet document is represented by an n-dimensional attribute vectors, X = (w_1, w_2 $w_3 \cdots$ w_n), and for the tweet document a word is an attribute.

Suppose there are m classes C_1, C_2 $C_3 \cdots C_m$. Given a tweet document, X, the classifier will classify the tweet document X to the class which have the highest posterior probability, conditioned on X. That is, the Naive Bayes classifier classifies the tweet document X as the class C_i if and only if:

$$P(C_i|X) > P(C_j|X) \tag{11}$$

for $1 \leq j \leq m, j \neq i$

For the class C_i , if the posteriori probability is the maximum posteriori probability among all of the classes, then the tweet document will be classified to C_i.

Since P(X) is constant for all classes, only $P(X|C_i)P(C_i)$ needs to be maximized to classify the tweet document. For Naïve Bayes classification, the distribution of classes are better to be balanced to get unbiased classification result and $P(C_i)$ is identical for all of classes. To classify the tweet document, we only need to exam the value of $P(X|C_i)$.

$$P(X|C_i) = P(w_1, w_2, w_3 \cdots w_n|C_i) \quad (12)$$

Because of the assumption that all of the attributes are independent to each other, this posteriori probability can be rewritten as:

$$P(X|C_i) = \prod_{k=1}^{n} P(w_k|C_i)$$

$$= P(w_1|C_i) \times P(w_2|C_i) \times P(w_3|C_i) \times \cdots \times P(w_n|C_i) \quad (13)$$

It is very easy to calculate the probabilities of $P(w_1|C_i)$, $P(w_2|C_i)$, $P(w_3|C_i)$, $\cdots$ $P(w_n|C_i)$ from the training dataset. Naive Bayes method is one of the most widely used methods to classify text data. Like the lexicon-based classifier, the Naive Bayes classifier treats each tweet document as a bag-of-words. In our work, we calculate the sentimental orientation probabilities based on the Naive Bayes algorithm for each word appearing in the training dataset and set up the sentiment distribution matrices for all the words in the training dataset.

In our work, we utilize the Naive Bayes algorithm and the smoothing algorithms embedded provided in Weka to implement experiments and tests.

C) Bayesian Network Classification Method

Like Naïve Bayes method, Bayesian Network also derives from Bayes' theorem, but Naive Bayes method assumes that the features are independent to each other. However, Bayesian Network method takes consideration of the relationships between the features.

In the case of my work, the features come from unigrams, bigrams and trigrams appearing in the dataset. There exists systematic dependency between the features. For example, the bigram "delayed again" is highly dependent on the unigram "delayed". Bayesian network classification methods specify joint conditional probability distributions.

Let X = (w_1, w_2 $w_3 \cdots w_n$) be a tweet document described by the attributes A_1, A_2 $A_3 \cdots A_n$. Any of the attribute A_i is conditionally independent of its non-descendants, given their parent attributes. The complete representation of the existing joint probability distribution can be expresses by the following equation:

$$P(w_1 \cdots w_n) = \prod_{i=1}^{n} P(w_i \,|Parents(A_i)) \tag{14}$$

Where $P(w_1 \cdots w_n)$ is the probability of a particular combination of feature values of X and the values for $P(w_i|Parents(A_i))$ is the posteriori probability of w_i given the probability of the attribute A_i.

The Bayesian Network classifier scans each single tweet and calculates the probability for each class: positive, negative and neutral. Each tweet will be classified to the class which gets the highest probability.

4.2.3 Support Vector Machine Classification Method

Support Vector Machine (SVM) method is a classification method for both linear and nonlinear data. Support Vector Machine (SVM) method maps the dataset with high dimension, which in my case is with many attributes, to a higher dimensional space and searches for a linear optimal separating hyper-plane which can separate the data of one class to another.

To explain how Support Vector Machine (SVM) method works, suppose we have a 2 dimensional dataset, any of the data can be expresses as (X_1, X_2).

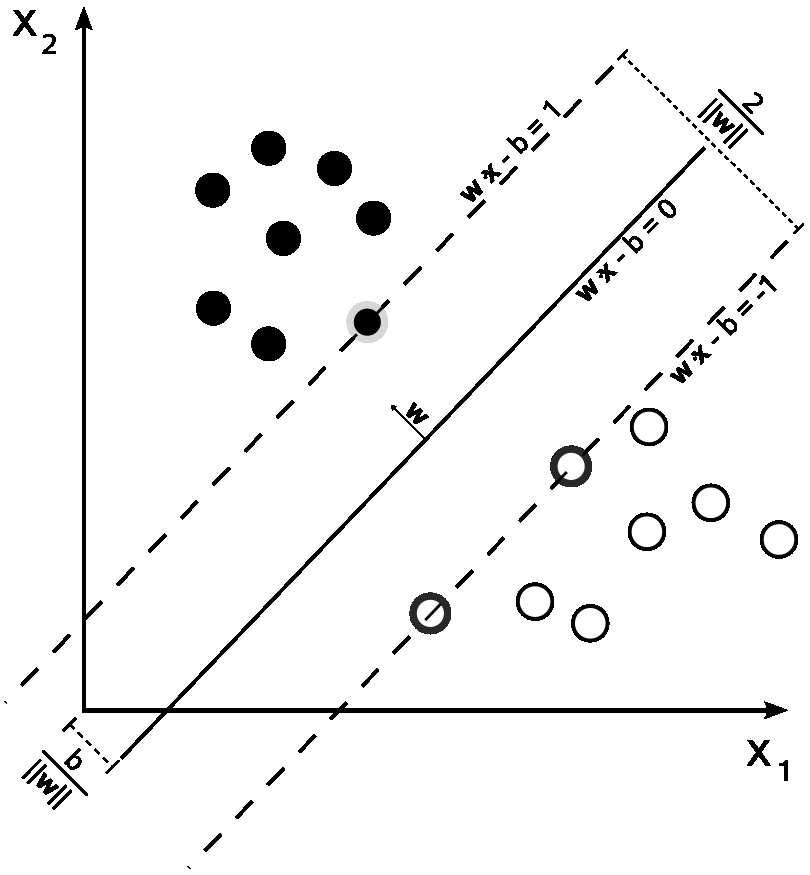

Figure 7 Support Vector Machine (contributors 2015)

Support Vector Machine method classifies the dataset by searching a separation line that can achieve the maximum margin between two classes. As shown in the figure 2. The separation line $w \cdot \mathrm{x} - b = 0$ gets larger margin than the other two lines do.

In multi-dimension classification, the Support Vector Machine classifier builds a high dimension space to map these data and searches for a hyper-plane to separate the classes and get the largest margin.

Support Vector Machine classifiers are supervised machine learning models used for binary classification and regression analysis. However, in our work, we aim to build classifiers, which can classify tweets into three sentiment categories. Based on the study done by Hsu and Lin, the pairwise classification method outperforms the one-against-one classification method in multiclass Support Vector Machine classification. In the

pairwise classification method, each pair of classes will have one SVM trained to separate the classes.

We adopt pairwise classification approach in SVM classification method. We utilize the libSVM algorithm in Weka, which uses pairwise classification for multiclass SVM classification.

4.3.4 Decision Tree Methods

A decision tree is a flowchart-like tree structure, in which each internal node represents a test on an attribute and each branch represents an outcome of the test, and each leaf node represents a class. For example, the sentiment classification for a text can be illustrated as figure X. The word “not” is the first classification node and it is called the root node, which used to split based on the occurrence of the word “not”. The word “smooth” and the word “delay” are the internal nodes, which are used to test the occurrences of the two words. The leaf nodes represent the sentiment classes.

During sentiment classification, a tweet document is tested along the decision tree until its sentiment is classified. For example, the tweet “The flights always delay” passes the root node to the branch of no. Then the tweet passes the internal node of “delay” and it is classified to negative.

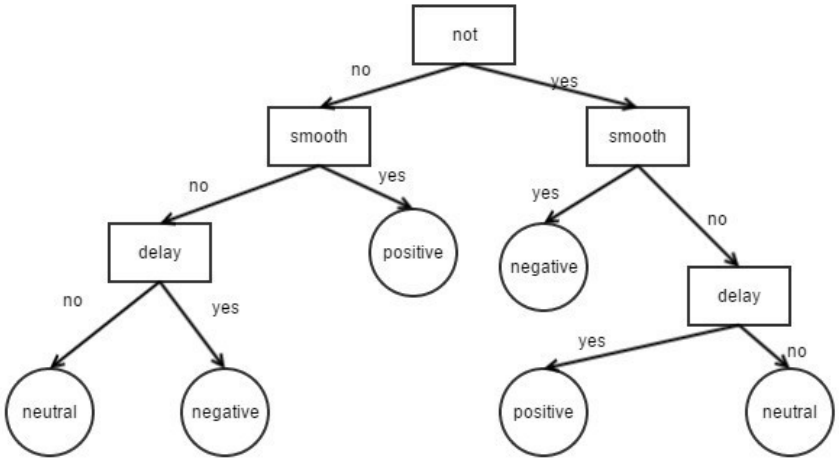

Figure 8 Decision Tree classifier

Decision Tree is a very popular method in knowledge discovery area because it does not need domain knowledge and parameter setting. Besides that, Decision Trees can be displayed and it is intuitive for people to understand the mechanics behind them. Decision Trees are very efficient to build and often produce good performance. However, one disadvantage of Decision tree is the over- fitting problem. Besides that, at each step of splitting data, decision tree only considers the optimum attribute for current splitting but not the optimum splitting attribute for the whole tree. This is called greedy algorithm and we may not get the best Decision Tree.

A) C4.5 Decision Tree Method

The first popular Decision Tree algorithm was Iterative Dichotomiser 3 (ID3), developed by J. Ross Quinlan. Iterative Dichotomiser 3 (ID3) is an algorithm used to generate a decision tree and it is most used in statistical learning and NLP domains.

ID3 algorithm selects attributes to split data by using Information Gain algorithm. ID3 algorithm selects the features which produce the biggest information gain to split the dataset and keep splitting the data by iterating this process and selecting from the remaining features. As mentioned in the previous chapter, Information Gain is used to evaluate how much impurity reduced by using the selected attribute to split the data. Iterating stops when all documents in the split subsets belong to the same class or no more features can be used to split the training data.

Because the ID3 algorithm keeps iterating the process of splitting subset data, it can cause the over-fitting problem. Besides, ID3 algorithm cannot deal with continuous attributes or attributes containing missing values. As an extension of ID3, C4.5 was developed by Ross Quinlan to solve these problems.

C4.5 discretizes the continuous attribute by setting a threshold and splitting the data to a group whose attribute value is above the threshold and another group whose attribute value is below or equal to the threshold. C4.5 handle missing values in attribute by just not using the missing values in Information Gain calculations. C4.5 handles over-fitting problems by using the post-pruning approach.

C4.5 uses a post-pruning approach called pessimistic pruning, which uses the Cost Complexity pruning algorithm and uses the training data to estimate the error rate. Error rate of the tree is the percentage of misclassified instances in the tree. The Cost Complexity of a tree is calculated based on the number of the leaves and the error rate of the tree. For example, for any internal node, the Cost Complexities of the node and its subtree are calculated and compared. If the Cost Complexity of the node is smaller than its subtree, the subtree is pruned.

B) Random Forest Method

Because the decision tree generated by ID3 algorithm and C4.5 algorithm are not necessarily the best decision tree for classification, Random Forest was developed as an ensemble approach based on many decision trees. Random Forest uses the Majority Vote method and returns the class with most votes.

Random Forest uses the Bagging approach in building classification models. For a dataset, D, with N instances and A attributes, the general procedure to build a Random Forest ensemble classifier is as follows. For each time of building a candidate decision tree, a subset of the dataset D, d, is sampled with replacement as the training dataset. In each decision tree, for each node a random subset of the attributes A, a, is selected as the candidate attributes to split the node. By building K Decision Trees in this way, a Random Forest classifier is built.

In classification procedure, each Decision Tree in the Random Forest classifiers classifies an instance and the Random Forest classifier assigns it to the class with most votes from the individual Decision Trees.

In my experiment, a Random Forest classifier and a C4.5 Decision Tree classifier are both built and tested.

4.4.4 The Ensemble Method

4.4.4.1 Candidate Classifier Selection

The ensemble classification method is a combination of different classifications. Because every sentiment classification method has its advantages and disadvantages, the overall accuracy of many different sentiment classifiers, with Majority Vote, is expected to be higher than any individual sentiment classifier. For example, a single tweet document is classified to positive by the Lexicon based classifier, classified to negative by the Naive Bayes classifier, classified to negative by the Bayesian Network classifier, classified to negative by the Support Vector Machine (SVM) classifier, classified to positive by the C4.5 Decision Tree classifier and classified to negative by the Random Forest classifier. From the classification results, the tweet document has a bigger probability of being a tweet with negative sentiment than the probability of being a tweet with positive sentiment. This is called the Majority Vote method and, in this example, there are four votes for negative and only two votes for positive.

However, to build an ensemble classifier, there are a lot of options. If there are N classifiers available, then there are 2^Npossible subsets of theses classifiers. Because an ensemble classifier requires more than one classifier to be built, there are $2^N - \mathrm{N}$ possible ensemble classifiers. We cannot just take all available classifiers to build the ensemble classifier because some of the classifiers that perform really badly can lower the accuracy of the ensemble classifier. In my thesis, I select the classifiers by implementing a sentiment classification test on the six classifiers: the Lexicon-based classifier, the Naive Bayes classifier, the Bayesian Network classifier, the Support Vector Machine (SVM) classifier, the C4.5 Decision Tree classifier and the Random Forest classifier.

I use a test dataset with 31,888 tweets in the classifier selection process. These tweets include original tweets and retweets. I discard the irrelevant tweets and label each relevant tweet in the dataset as positive sentiment, negative sentiment or neutral

sentiment manually. In the dataset, 2,502 tweets are labeled positive, 7,039 tweets are labeled negative, 13,074 tweets are labeled neutral and 9,273 tweets are discarded for being irrelevant.

Table 6 Label distribution of Tweets

class	positive	negative	neutral	irrelevant
tweets	2502	7039	13074	9273

In the Bayesian approaches, the model training process requires the class distribution to be balanced. So I resampled the data with 2,500 tweets for each class: positive sentiment, negative sentiment and neutral sentiment. For evaluation purpose, the dataset with 7,500 tweets was used for every classification approach in my test experiment.

In my test experiment, I removed all symbols, hashtag signs, links, emoticons and punctuation from tweets since I don't regard those factors as classification features. I also adopted text clean techniques by using the program in R to remove duplicates and clean tweets. I used Weka as my data mining tool to implement my experiment.

I conducted experiments with the six classification models. I used the 10-fold validation plan to evaluate the machine learning classification approaches including: the Naive Bayes classifier, the SVM classifier, the Bayesian Network classifier, the C4.5 Decision Tree and the Random Forest classifier. Test results for the three-class classification experiment are shown in table 4. The Lexicon-based classifier got the lowest accuracy, which is 60.5%. The accuracy of the Naive Bayes model classification reached 81.8%. The Bayesian Network classifier outperformed the Lexicon-based classifier and the Naive Bayes classifier by reaching an accuracy of 85.1%. The SVM classifier got an accuracy of 74.7%, the C4.5 Decision Tree got an accuracy of 82.9% and the Random Forest classifier got an accuracy of 82.4%.

Table 7 Accuracy of three class classification

Classifier	Positive accuracy	Negative Accuracy	Neutral Accuracy	Overall Accuracy
Lexicon-based	70.8%	56.2%	54.6%	60.5%
Naive Bayes	87.0%	81.5%	76.9%	81.8%
Bayesian Network	87.5%	85.3%	83.4%	85.1%
SVM Classifier	74.1%	85.2%	64.8%	74.7%
C4.5 Decision Tree	80.0%	85.7%	83.0%	82.9%
Random Forest	81.3%	85.6%	80.4%	82.4%

Because there has been much work done in two-class dataset, I also implemented the sentiment classification algorithms in the two-class classification experiment, in which the training data and the test data only contain two classes: positive sentiment and negative sentiment. In my experiment, the accuracy of the Lexicon based classifier is 67.9%, the accuracy of Naive Bayes classifier is 90.0%, the accuracy of Bayesian Network classifier is 91.4% and the accuracy of SVM classifier is 84.6%. The C4.5 Decision Tree classifier got an accuracy of 86.0% and the Random Forest classifier got an accuracy of 89.8%. The results show that, the Lexicon based classifier performs much worse than other candidate classifiers.

Table 8 Accuracy of two class classification

Classifier	Positive accuracy	Negative Accuracy	Overall Accuracy
Lexicon-based	77.8%	58.0%	67.9%
Naïve Bayesian	90.3%	89.8%	90.0%
Bayesian Network	91.4%	91.5%	91.4%
SVM	79.7%	89.5%	84.6%
C4.5 Decision Tree	84.3%	87.8%	86.0%
Random Forest	90.2%	89.4%	89.8%

For the Lexicon based classifier, in both of the two-class classification and the three-class classification experiments, the positive accuracies are much higher than the negative accuracies. In both experiments, the Bayesian Network classifier produced the highest accuracies. After comparing the experiment results of the six sentiment classifiers. I selected Naive Bayes method, Bayesian Network method and Support Vector Machine (SVM), C4.5 Decision Tree and Random Forest to build the ensemble classifier.

4.4.4.2 The Ensemble Classifier

The ensemble classifier uses the Majority Vote method to classify each document's class. The five classifiers have the same weights in the majority vote process. In my thesis, I use 10-fold validation. I use the same dataset to produce subsample dataset to train the Naive Bayes classifier, the Bayesian Network classifier, the Support Vector Machine (SVM) classifier, the C4.5 Decision Tree classifier and the Random Forest classifier individually. The process of the model training and classification are implemented with Weka.

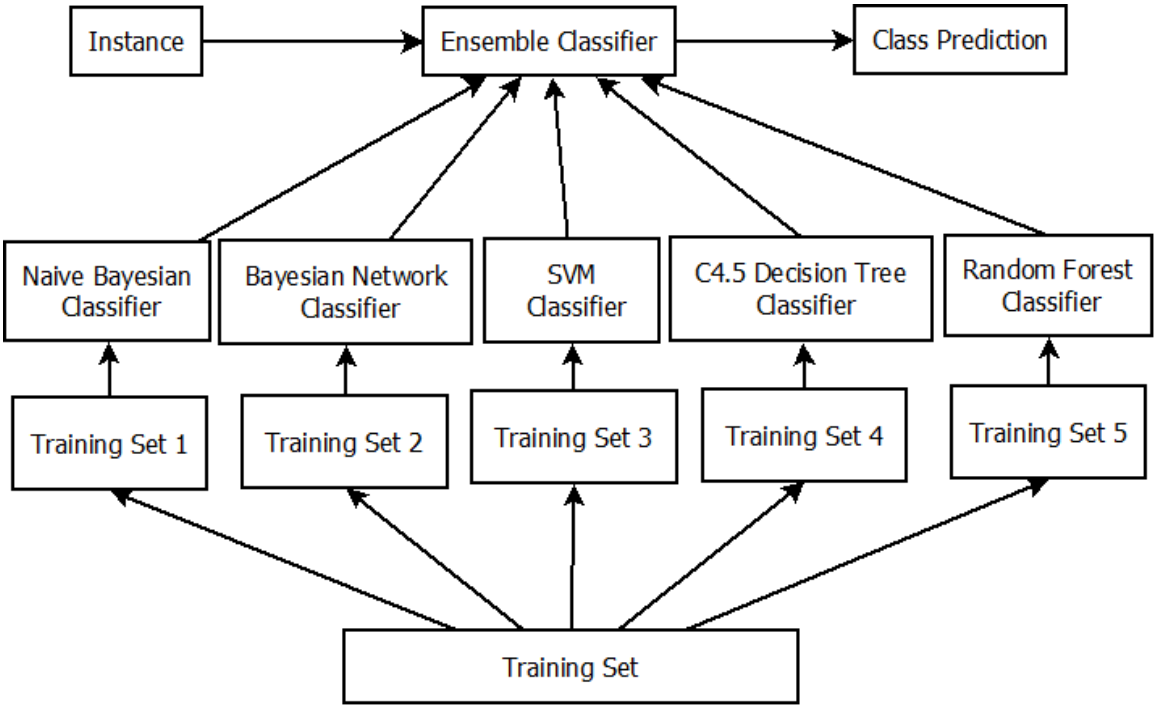

Figure 9 The ensemble classifier

As shown in Figure 9, each classifier is trained on a subsample of the overall dataset and classify the tweets independently.

The combination rule of the ensemble classifier is Majority Vote algorithm. I use a table 9 to illustrate how the ensemble classification method works. In table 9, the first column is the classification results of the Naive Bayes classier, the second column is the classification results of the Bayesian Network classifier, the third column is the classification results of the Support Vector Machine (SVM) classifier, the fourth column is the classification results of the C4.5 Decision Tree classifier, the fifth column is the classification results of the Random Forest classifier and the sixth column is the sentiment labels of the tweet documents. For every tweet document in the test dataset, there are a row of classification results used to predict the sentiment classes and the sixth column represents the results of the ensemble classifier.

Table 9 The Ensemble Classification

Tweet No	Naïve Bayesian	Bayesian Network	SVM	C4.5	Random Forest	Ensemble
1	1	1	-1	1	0	1
2	0	0	-1	1	0	0
3	-1	-1	-1	1	1	-1
4	-1	0	-1	-1	0	-1
5	1	0	-1	1	-1	1
6	1	-1	1	1	-1	1
7	0	0	-1	0	1	0

As shown in the pseudo code, a tweet document is assigned an arbitrary class if the classification results of the five classifier cannot be determined by the Majority Vote method. The algorithm of the classifier can be expressed in follow:

Require: C (five classification results for one instance), S (sentiment classes: Positive, Negative and Neutral)

if for any class i in S, C_i in C

count(C_i) >=3

then Class=C_i

else if for any two class j, k in S, C_j and C_k in C

count(C_j) = count(C_k)=2

then Class=C_j or Class= C_k

return Class

For example, if a tweet document is classified to negative by the Naïve Bayesian classifier, classified to neutral by the Bayesian Network classifier, classified to positive by the Support Vector Machine (SVM) classifier, classified to positive by the Decision Tree classifier and classified to negative by the Random Forest. Then this tweet document is assigned arbitrarily to either of the two classes, which are positive or negative, because this tweet document has same probabilities of being either of the two classes.

CHAPTER 5 EXPERIMENT AND EVALUATION

5.1 Evaluation Plan

5.1.1 Classification Validation

Generally speaking, over-fitting happens when the training data is relative small (Wikipedia, Cross-validation, 2014), and cross-validation is a good solution to avoid this. In my research, I take 12,864 tweets data, which is relatively not small, but it is still a good choice to implement cross-validation.

Cross-validation is a method for model validation, which samples a subset of data to do model training and another subset of data to do model validation. 10-fold validation is one cross-validation method. In 10-fold validation, the dataset is randomly partitioned into 10 subsets with equal sizes. In the model training and validation process, each 9 subsets of data is used as a training dataset to train a model and the remaining 1 subset is used to validate the model. After repeating 10 times, each 9 subsets have been used as a training dataset to train a model and 10 classification validation results are produced. The overall validation result of the 10-fold validation is the average validation result of the 10 models. In the data mining research area, 10-fold validation is a popular validation method and it is used in my experiment.

5.1.2 Accuracy Evaluation for Different Classes

In sentiment classification, there are three sentiment classification results for the text: positive, negative and neutral. So there will be 6 different classification errors in the experiment:

- Positive tweets being classified as negative or neutral;
- Negative tweets being classified as positive or neutral;
- Neutral tweets being classified as positive or negative.

Different classification errors tell different information about the classifiers. For example, too many neutral tweets being classified as positive or negative means the classifier is over-sensitive and too many positive or negative tweets being classified as neutral means the classifier is under-sensitive. The accuracy for positive tweets, negative tweets, neutral tweets and the overall accuracy will be displayed. Besides that, the incorrect results, like positive tweets being classified as negative or neutral will be displayed. This information will be analyzed to evaluate the advantages and disadvantages of each classifier:

The overall accuracy of the seven classifiers will be compared, and if the ensemble classifier gets the highest accuracy, it means that the ensemble system can improve the accuracies of the classifiers it is made up of. If the ensemble classifier does not get the highest accuracy, this implies that other classification algorithms are more applicable for the sentiment classification for tweets about airline services.

5.1.3 Accuracy Evaluation Based on F-measure

In accuracy evaluation of classification, there are Recall, Precision and F-measure to evaluate the overall accuracy of the classifier.

A) Recall

Recall is the fraction of the correctly classified instances for one class of the overall instances in this class (Han, Kamber and Pei 2012, 366). For example, if 900 tweets are classified to positive and 800 of them are correct, and in the dataset there are 1000 tweets which are positive, then the recall for the positive class is 800/1000, which equals to 0.8.

B) Precision

Precision is the fraction of the correctly classified instances for one class of the overall instances which are classified to this class (Han, Kamber and Pei 2012, 366). For example, if 900 tweets are classified to positive and 800 of them are correct, and in the dataset there are 1000 tweets which are positive, then the Precision for the positive class is 800/900, which equals to 0.89.

C) F-measure

To get a comprehensive evaluation of the classification, F-measure is developed to integrate the Recall and the Precision. The F-measure can be expressed as

$$F_\beta = (1 + \beta^2) \times \frac{precision \times recall}{\beta^2 \times precision + recall} \quad (15)$$

This is a general form of F-measure and the parameter β is used to change the weights for Precision and Recall in calculating the F-measure value. In my thesis, because recall and precision are equally important (Han, Kamber and Pei 2012, 364). I set β to 1, and it is called the harmonic mean of precision and recall. The formula can be rewritten as:

$$F_1 = 2 \times \frac{precision \times recall}{precision + recall} \quad (16)$$

5.2. Experiment Result Evaluation

5.2.1 Accuracy for Different Classes

After the experiment, the classification results are listed in table 10. From the classification results, I discovered the accuracy of the other six classifiers, the Naive Bayes classifier, the Bayesian Network classifier and the Support Vector Machine classifier, the C4.5 Decision Tree classifier and the Random Forest classifier are much higher than the accuracy of the lexicon-based classifier. The Lexicon-based classifier got the lowest overall accuracy, which is 61.5%. The accuracies for different classes are different. In this table, the accuracy means recall, and the recalls for positive are significantly higher than the accuracy in the negative class and the neutral class. This means the Lexicon-based classifier is over-sensitive. The reason for that is because many Twitter users express their feelings in a sarcastic way. For example, in the tweet document “Thanks Delta for cancelling my flight, that’s pretty good.” There are two positive words “Thanks” and “good”, and there is one negative sentiment word “cancelling”. This tweet document is classified to positive class but it is actually a negative tweet. Other classifiers perform similarly in different sentiment classes

compared to the Lexicon-based classifier, which means that these classification methods perform better in handling sarcasms. The results also show that the ensemble classifier and the Decision Tree classifiers have more balanced accuracies in different sentiment classes than other machine learning classifiers.

Table 10 Accuracy for different classes

Classifiers	Positive accuracy			Negative accuracy			Neutral accuracy			Overall accuracy
Classified Result	Correct	Incorrect		Correct	Incorrect		Correct	Incorrect		Correct
		Negative	Neutral		Positive	Neutral		Positive	Negative	
Lexicon-based	72.9%	10.8%	16.3%	62.8%	24.0%	15.6%	48.8%	23.7%	31.2%	61.5%
Naïve Bayesian	87.4%	5.0%	7.6%	82.5%	9.4%	8.1%	76.7%	11.6%	11.7%	82.2%
Bayesian Network	87.7%	4.8%	7.4%	82.4%	9.6%	7.9%	76.6%	11.8%	11.6%	82.3%
SVM	77.2%	13.2%	9.5%	68.2%	9.0%	22.8%	85.5%	9.0%	5.5%	77.0%
C4.5 Decision Tree	82.4%	10.0%	7.5%	80.7%	7.0%	9.3%	84.9%	7.0%	8.1%	83.6%
Random Forest	83.7%	9.1%	7.2%	81.8%	7.1%	11.2%	84.8%	7.2%	8.0%	83.4%
Ensemble	87.2%	6.2%	6.6%	81.6%	8.1%	10.3%	83.7%	9.3%	7.0%	84.2%

In terms of the overall accuracy, the Support Vector Machine (SVM) classifier got the lowest accuracy of the six machine learning classifiers, which is 77.0%. The Naive Bayes classifier got the second lowest overall accuracy, which is 82.2%. The Bayesian Network classifier outperforms the Naive Bayes classifier because the Bayesian Network algorithm takes the correlations of the features into account. The correlations of the features in my dataset are mainly caused by the data transformation because the bigrams and trigrams are made of unigrams. The Random Forest classifier got the second highest accuracy during the six classifiers by adopting an ensemble strategy in building a classifier with many random Decision Trees. The C4.5 Decision Tree classifier

outperforms the Random Forest classifier and got the highest overall accuracy of the six machine learning classifiers, which is 83.6%. The C4.5 Decision Tree performs better than the Random Forest because it adopts the post-pruning algorithms. Unlike the Naive Bayes classifier and the Bayesian Network classifier, the C4.5 Decision Tree classifier is not a probabilistic classifier. This result indicates that C4.5 Decision Tree classifier is more accurate than probabilistic classifiers.

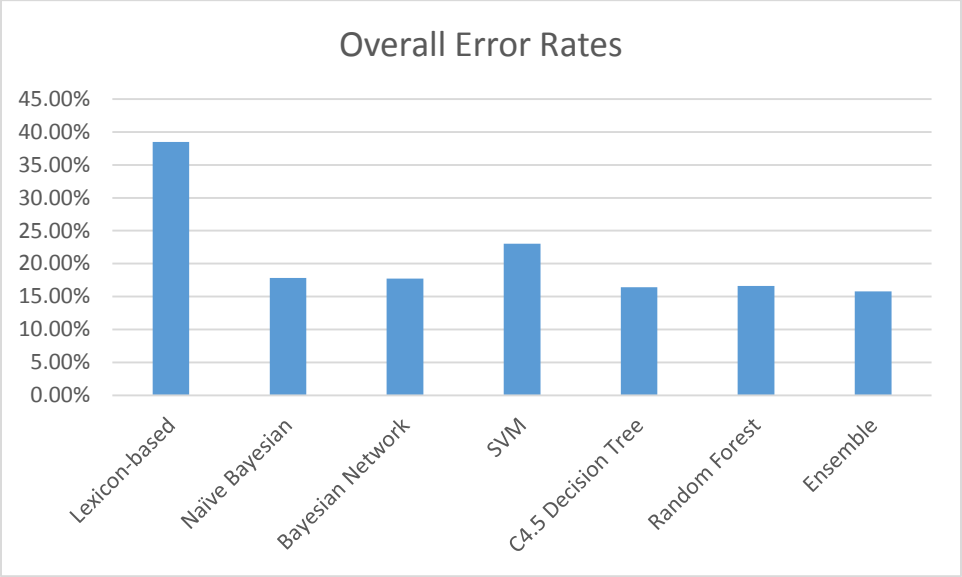

Figure 10 Error rates for different classes

The ensemble classifier got the highest accuracy of the seven classifiers. This shows that the ensemble classifier can improve the accuracy more than the individual classifiers it consists of. Besides that, the ensemble classifier also shows a very balanced distribution of accuracies in different sentiment classes.

5.2.2 Performance Measure

In the table of the recall, precision and F-measure, the Lexicon-based classifier got the lowest F value for the classification accuracy.

The C4.5 Decision Tree classifier, the Random Forest classifier and the Ensemble classifier I proposed have a more balanced accuracy distribution in Precision and Recall compared to the probabilistic classifiers and the Support Vector Machine Classifier.

Classifiers	Positive accuracy			Negative accuracy			Neutral accuracy			overall		
	Precision	Recall	F-measure	Precision	Recall	F-measure	Precision	Recall	F-measure	Precision	Recall	F-measure
Lexicon based	60.5%	72.9%	66.7%	60.5%	62.8%	61.7%	60.5%	48.7%	54.6%	60.5%	61.5%	61.0%
Naïve Bayesian	80.6%	87.4%	83.9%	81.1%	82.0%	81.8%	85.4%	76.7%	80.8%	82.4%	82.2%	82.3%
Bayesian Network	80.4%	87.7%	83.9%	81.2%	82.5%	81.9%	85.7%	76.6%	80.9%	82.2%	82.3%	82.2%
SVM classifier	81.1%	77.3%	79.1%	82.0%	68.2%	74.5%	70.3%	85.5%	77.2%	77.8%	77.0%	77.4%
C4.5 Decision Tree	85.4%	82.4%	83.9%	84.3%	83.6%	83.9%	81.4%	84.9%	83.1%	83.7%	83.6%	83.6%
Random Forest	85.5%	83.7%	84.6%	84.4%	81.7%	83.0%	80.6%	84.8%	82.7%	83.5%	83.4%	83.4%
Ensemble	83.4%	87.2%	85.3%	85.7%	81.6%	83.6%	83.5%	83.7%	83.6%	84.2%	84.2%	84.2%

Table 11 F-measure of accuracy

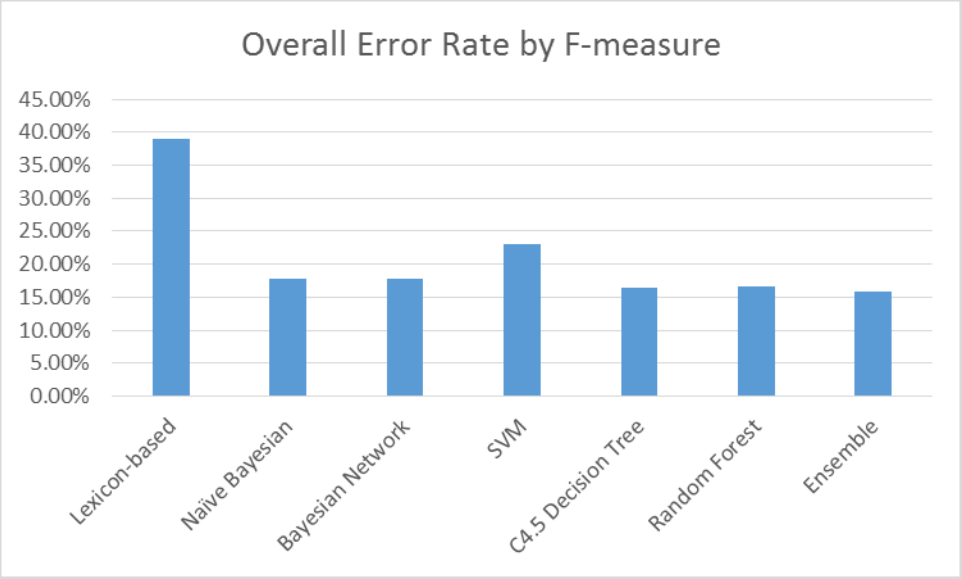

Figure 11 Error rate with F-measure

The result is listed in table 11. The Support Vector Machine (SVM) classifier got a very imbalanced distribution of accuracy of Precision and Recall. This means that, when

applying the Support Vector Machine (SVM) classifier, the weights of Precision and Recall need to be adjusted if the sentiment classification has special preferences for the accuracy.

In terms of Precision, Recall and the F-measure, the ensemble classifier I proposed got the highest accuracy among the seven classifiers. This is a strong indication that an ensemble classification system can improve the classification accuracy than the classification methods it is made up of in the tweet data about airline services.

5.2.3 Two Classes Sentiment Classification

The second part of the experiment is to do two class classification.

Table 12 F-measure accuracy for two class classification

Classifiers	Positive accuracy			Negative accuracy			overall		
	Precision	Recall	F-measure	Precision	Recall	F-measure	Precision	Recall	F-measure
Lexicon-based	0.64%	72.9%	68.9%	69.9%	62.8%	66.4%	67.3%	67.9%	67.6%
Naïve Bayesian	90.4%	90.5%	90.4%	90.5%	90.3%	90.4%	90.4%	90.34	90.4%
Bayesian Network	90.3%	90.7%	90.5%	90.6%	90.3%	90.5%	90.5%	90.5%	90.5%
SVM	89.6%	83.8%	86.6%	84.8%	90.2%	87.4%	87.2%	87.0%	87.1%
C4.5 Decision Tree	88.1%	86.3%	87.2%	86.6%	88.4%	87.5%	87.4%	87.3%	87.3%
Random Forest	90.2%	91.%	90.9%	91.4%	90.1%	90.7%	90.8%	90.8%	90.8%
Ensemble	91.2%	92.3%	91.8%	92.2%	91.1%	91.7%	91.7%	91.7%	91.7%

I discarded the tweets with neutral sentiment class and get a dataset with 8576 tweets and 4288 for negative and 4288 for positive and I reselected features. Because there are only two classes in the classification experiment, I only use a Recall Precision, F-value table. The results of two-class sentiment classification are list in table 12.

In the two-class classification experiment, every classifier got higher accuracy but the Naive Bayes classifier and the Bayesian Network classifier got a surprising increase in accuracy than the three-class classification experiment. This means that Bayesian classifiers performs better in the two-class classification than the three-class classification in this dataset. The Lexicon-based classifier is still the most inaccurate classifier. The ensemble classifier got the highest accuracy in Precision, Recall and the F-measure, and it achieved a good balance in the accuracy distribution.

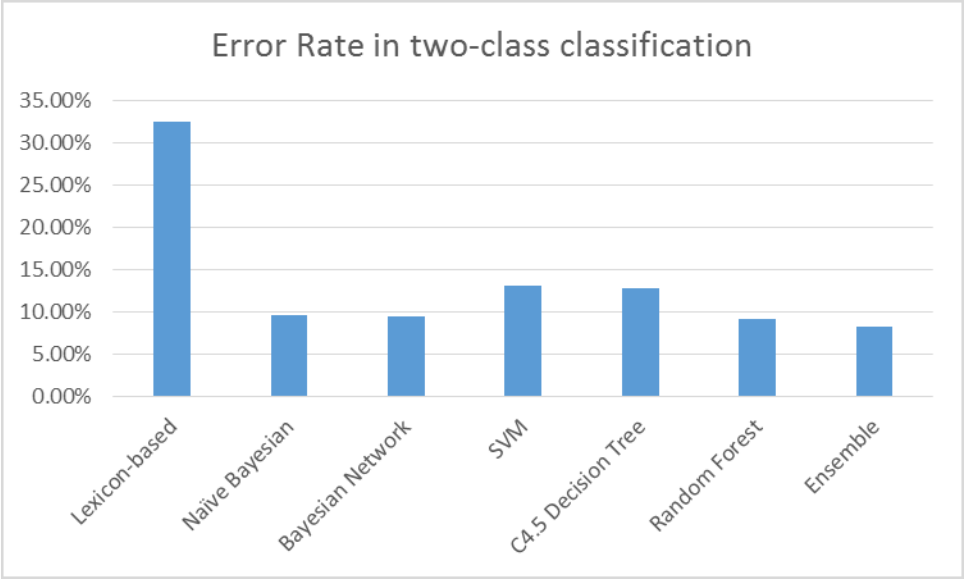

Figure 12 Two Class Error rate with F-measure

CHAPTER 6 CONCLUSION

Sentiment classification has been intensively studied by researchers and professionals from different domains. Because of the wide applications in the business areas, many approaches have been developed for sentiment classifications. Every industry is getting into the big data era and applying data technologies to dig new opportunities to build better businesses. One of these technologies is the sentiment classification technology which can automatically classify the customer sentiments and provide comprehensive understanding of customer feedback from raw data on the Internet. In all of the social network platforms, Twitter has been one of the most popular sources for marketing information research and sentiment classification.

This thesis makes empirical contributions to this research area by comparing the performance of different popular sentiment classification approaches and developing an ensemble approach, which further improves the sentiment classification performance. Besides that, the results of the experiments and the analysis on the tweets collected reveal much useful information for airline services improvements. Finally, the imbalanced accuracies of the classifiers in different sentiment class also reflects the customers' behaviors on Twitter.

6.1 Empirical Contributions

Previous work in Twitter sentiment classification traditionally focused on sentiment classification in general but did not focus on a specific domain. The classification results indicate that the accuracy of sentiment classification in the airline services domain is higher than twitter sentiment classification in general. The Lexicon based sentiment classifier is a general sentiment classifier and the other six supervised sentiment classifiers are domain-specific because the models are trained in the dataset of tweets about airline services.

In the domain of twitter sentiment analysis about airline services, little work has been done. This past work compares several different traditional classification methods and selects the most accurate individual classification method to implement sentiment

classification (Adeborna and Siau 2014). However, the ensemble approach I present improves the accuracy by combining these sentiment classifiers. For the airline services domain, the sentiment classification accuracy is high enough to implement customer satisfaction investigation.

Secondly, this research also reveals that the class distribution of the sentiments are highly imbalanced and the tweets with negative sentiment and neutral sentiment outnumber the tweets with positive sentiment. This indicates the fact that Twitter users prefer to tweet their bad feelings than their good feelings.

6.2 Practical Implications

The objectives of the thesis have been achieved by having:

- Compared the performances of six traditional sentiment classification methods in the tweets about airline services and showed that the supervised machine learning methods are much better than the Lexicon based sentiment classification method.
- Developed and presented an ensemble sentiment classifier and applied it to the tweets about airline services.
- Showed that the ensemble classifier outperforms the classifiers it is made up of.
- Discovered that Twitter users like to express their complaints toward airline services in a sarcastic way. This reveals the linguistic customs on the Internet.

The overall accuracy rate of the ensemble classifier is 84.2%, which is a satisfying result considering the complexity of sentiment classification. This approach is applicable for the airline companies to analyze the twitter data about their services.

6.3 Future Work

There are certain limitations in the thesis. First of all, the tweets collected from the Twitter API are not as pure as required for the sentiment classification. By searching tweets with keywords “flight” and the airline brand, I got a dataset with 40% irrelevant

tweets. Airline companies still need to further improve the accuracy of tweet data retrieval. Secondly, compared to the tweet data existing in Twitter, the dataset I collected and used in this thesis is a very tiny part. This is a problem to solve for doing scale sentiment classification by applying big data techniques. However, it requires expensive infrastructure investment to do this kind research and application. Besides that, my tweet data are very messy and they contain a lot of typos and abbreviations. Even though I adopted stemming techniques to reduce the dimensionality, it still cannot group all words with the same roots into one stem. It is desirable to auto correct all the typos and to extend the abbreviations to regular words, which requires high level Natural Language Processing techniques. More than that, the balanced class distribution is not a real world case and it might cause over-fitting problem in positive class. In the future work, more complicated models are expected to be built to solve this problem.

Moreover, for different airline brands, the features for sentiment classification might be different from each other and it is valuable to train sentiment classification models for different airline brands. To give more specific and valuable information for the decision makers of the airline companies, sentiment classification can be applied to the tweets about their airline services and produce detailed reports.

In the opposite of drilling down the domain for sentiment classification, a general method for twitter sentiment classification is more desirable because of its applicable value.

Last but not least, there are also many further research directions, which can be worked on. In my thesis, only the texts of the tweets are considered and other information like the users who tweet them, the times of the retweets and other factors are also potentially useful. The time series analysis of the twitter sentiment about airline services is also an interesting topic and the time data is available in the tweets retrieved from the Twitter API.

REFERENCES

[1] Adeborna, Esi, and Keng Siau. 2014. "An approach to sentiment analysis-The case of airline quality rating." *Pacific Asia Conference on Information Systmem.* 5.

[2] Cheng, Hong, Xifeng Yan, Jiawei Han, and Chih-Wei Hsu. 2007. *Discriminative Frequent Pattern Analysis for Effective Classification.* IEEE 23rd International Conference on IEEE.

[3] contributors, Wikipedia. 2015. *Support vector machine.* March 5. http://en.wikipedia.org/wiki/Support_vector_machine.

[4] Ding, Xiaowen, Liu Bin, and Philip S. Yu. 2008. "A holisitic lexicon-based approach to opinion mining." *Proceedings of the 2008 International Conference on Web Search and Data Mining.* ACM.

[5] G., Vinodhini, and Chandrasekaran RM. 2012. "Sentiment Analysis and Opinion Mining: A Survey ." *International Journal of Advanced Research in Computer Science and Software Engineering* 6.

[6] Han, JiaWei, Micheline Kamber, and Jian Pei. 2012. In *Data Mining, Concepts and Techniques*, 364.

[7] 2014. *History of SVM.* http://www.svms.org/history.html.

[8] Li, Zhuang, Jing Feng, and Zhu XiaoYan. 2006. "Movie review mining and summarization." *Proceedings of the 15th ACM international conference on Information and knowledge management .* ACM.

[9] Lin, Chenghua, and Yulan He. 2006. "Joint sentiment/topic model for sentiment analysis." *Proceedings of the 18th ACM conference on Information and knowldge management .* ACM.

[10] 2014. *Major Canadian and US Airlines.* http://www.nationsonline.org/oneworld/Airlines/airlines_north_america.htm.

[11] Melville, Prem, Wojciech Gryc, and Richard D. Lawrence. 2009. "Sentiment Analysis of Blogs by Combining Lexical Knowledge with Text Classification." *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining.*

[12] Minqing, Hu, and Liu Bing. 2004. "Mining and summarizing customer reviews." *Proceedings of the tenth ACM SIKDD international conference on Knowledge discovery and data mining.* ACM.

[13] Oliver, Breen Jeffery. 2012. *Mining twitter for airline consumer sentiment.*

[14] Pak, Alexander, and Patrick Paroubek. 2010. "Twitter as a Corpus for Sentiment Analysis and Opinion Mining." *LREC.*

[15] Pang, Bo, and Lillian Lee. 2008. "Opinion Mining and Sentiment Analysis." *Foundations and Trends in Information Retrieval* 4.

[16] Pang, Bo, Lillian Lee, and Shivakumar Vaithyanathan. 2002. "THumbs up? sentiment classification using machine learning techniques." *Proceedings of the Conference on Empirical Methods in Natural Language Processing(EMNLP).*

[17] Read, Jonathon. 2005. "Using emoticons to reduce dependency in machine learning techniques for sentiment classification." *The Association for Computer linguistics.* ACL.

[18] Tan, Songbo, and jin Zhang. 2008. "An empirical study of sentiment analysis for chinese documents." *Expert Systems with Applications.*

[19] Twitter.com. 2014. *REST APIs.* https://dev.twitter.com/rest/public.

[20] Wikipedia. 2014. *Curse of dimensionality.* http://en.wikipedia.org/w/index.php?title=Curse_of_dimensionality&oldid=639281699.

[21] Wilson, Theresa, Janyce Wiebe, and Paul Hoffmann. 2005. "Recognizing contextual polarity in phrase-level sentiment analysis." *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing.* ACL. 354.

[22] Xia, Rui, Chengqing Zong, and Shoushan Li. 2011. *Ensemble of feature sets and classification algorithms for sentiment classification.*

[23] Yang, Yiming, and Thorsten Joachims. 2008. *Text categorization.* http://www.scholarpedia.org/article/Text_categorization.

[24] Yi, Jeonghee, and Wayne Niblack. 2005. "Sentiment mining in WebFountain." *21st International Conference on IEEE.*

APPENDIX A: FEATURES AND THEIR INFORMATION GAIN

IG	Rank	Features
0.155 +- 0.00	1 +- 0	thank
0.105 +- 0.00	2.3 +- 0.46	delay
0.104 +- 0.00	2.7 +- 0.46	cancel
0.078 +- 0.00	4 +- 0	great
0.061 +- 0.00	5.5 +- 0.5	unit
0.061 +- 0.00	5.5 +- 0.5	thank you
0.049 +- 0.00	7.5 +- 0.5	airas
0.05 +- 0.00	7.5 +- 0.5	hour
0.04 +- 0.00	9.1 +- 0.3	great flight
0.038 +- 0.00	10.1 +- 0.54	rt
0.037 +- 0.00	10.9 +- 0.54	airasia flight
0.034 +- 0.00	12.7 +- 1.49	my
0.034 +- 0.00	13 +- 0.63	awesom
0.033 +- 0.00	13.8 +- 1.08	you
0.032 +- 0.00	15.3 +- 0.46	thanks for
0.031 +- 0.00	15.6 +- 1.28	dela
0.03 +- 0.00	16.8 +- 1.08	amaz
0.029 +- 0.00	17.9 +- 0.7	lov
0.028 +- 0.00	18.9 +- 0.54	for
0.025 +- 0.00	20.1 +- 0.54	qz
0.024 +- 0	21.1 +- 0.7	crew
0.023 +- 0.00	22.7 +- 1	no
0.023 +- 0.00	22.9 +- 1.3	not
0.022 +- 0.00	23.7 +- 1.1	bag
0.022 +- 0	24.5 +- 0.81	aircanad
0.02 +- 0.00	27.1 +- 1.14	flight qz
0.019 +- 0.00	28.3 +- 1.55	and
0.019 +- 0.00	28.5 +- 2.29	my flight
0.019 +- 0.00	28.8 +- 1.78	thanks for th
0.019 +- 0.00	29.4 +- 2.73	for th
0.018 +- 0.00	31.5 +- 0.92	airasia flight qz
0.018 +- 0.00	32.1 +- 1.87	rebook
0.018 +- 0.00	32.3 +- 2.28	search
0.017 +- 0.00	33.1 +- 2.7	a great
0.016 +- 0.00	36.1 +- 2.07	servic
0.016 +- 0.00	36.3 +- 1.73	jetblue thank
0.016 +- 0.00	38 +- 3.35	qatar

0.016 +- 0.00	38.4 +- 2.69	flight delay
0.015 +- 0.00	40.1 +- 2.39	wait
0.015 +- 0.00	41.9 +- 3.62	airlines flight
0.015 +- 0.00	42.3 +- 3.32	customer
0.015 +- 0.00	42.5 +- 4.34	missing airas
0.015 +- 0.00	43 +- 3.63	connect
0.014 +- 0.00	44.8 +- 2.79	worst
0.014 +- 0.00	45.7 +- 4.47	best
0.014 +- 0.00	46.5 +- 5.5	hr
0.014 +- 0.00	46.6 +- 5.12	delayed flight
0.014 +- 0.00	47.7 +- 5.2	flight cancel
0.014 +- 0.00	49 +- 3.49	divers
0.014 +- 0.00	50 +- 4.4	a great flight
0.014 +- 0.00	50.6 +- 4.1	now
0.013 +- 0.00	52.3 +- 5.78	nic
0.013 +- 0.00	53.7 +- 3.82	american airl
0.013 +- 0.00	54.4 +- 4.15	i
0.013 +- 0.00	55.5 +- 6.41	but
0.012 +- 0	57.3 +- 3.03	customer servic
0.012 +- 0.00	57.8 +- 4.79	ver
0.012 +- 0.00	59.6 +- 4.29	wa
0.012 +- 0.00	60.4 +- 5.31	flight attens
0.012 +- 0.00	62.5 +- 6.68	attens
0.012 +- 0	64.5 +- 5.57	kudo
0.012 +- 0.00	64.7 +- 6.26	american
0.012 +- 0.00	64.7 +- 6.02	southwest
0.012 +- 0	64.9 +- 4.74	an hour
0.012 +- 0.00	64.9 +- 5.56	was cancel
0.012 +- 0	64.9 +- 4.93	us milit
0.012 +- 0.00	65.7 +-11.3	ges
0.012 +- 0.00	66.6 +- 6.48	mh
0.011 +- 0.00	67.3 +- 7.21	new
0.011 +- 0.00	67.3 +- 6.97	indones
0.011 +- 0.00	70.1 +-10.13	th
0.011 +- 0	73.6 +- 5.43	cancelled flight
0.011 +- 0.00	74.7 +- 6.56	is
0.011 +- 0	75.4 +- 6	becaus
0.011 +- 0	76.5 +- 4.61	airway
0.011 +- 0.00	76.5 +- 7.86	american airlines flight
0.011 +- 0.00	77.3 +- 7.09	qatar airway
0.011 +- 0.00	77.9 +-10.1	lug
0.01 +- 0	79.7 +- 5.59	milit
0.01 +- 0.00	80.1 +- 6.33	me

0.01 +- 0.00	80.3 +- 7.52	upgrad
0.01 +- 0.00	81.5 +- 8.08	missing airasia flight
0.01 +- 0	82.7 +- 5.35	been
0.01 +- 0	85.1 +- 4.39	enjoy
0.01 +- 0.00	85.7 +- 8.98	ridicl
0.01 +- 0	86.1 +- 5.36	you for
0.01 +- 0.00	86.7 +- 7.51	flight crew
0.01 +- 0.00	89.1 +-10.12	the best
0.01 +- 0	89.3 +- 4.92	thank you for
0.01 +- 0	90.8 +- 4.14	is delay
0.01 +- 0.00	90.8 +- 7.97	again
0.009 +- 0.00	92 +- 6.65	flight w
0.009 +- 0	92.6 +- 5.44	americanair
0.009 +- 0.00	94.2 +- 7.97	delt
0.009 +- 0	94.9 +- 4.85	hilar
0.009 +- 0	95 +- 5.9	was delay
0.009 +- 0	96.1 +- 6.33	smooth
0.009 +- 0	96.4 +- 5.06	disappoint
0.009 +- 0.00	98 +- 8.27	why
0.009 +- 0.00	98.6 +- 8.52	lost
0.009 +- 0.00	103 +- 8.1	then
0.009 +- 0.00	103.3 +- 8.27	airl
0.009 +- 0.00	106.4 +-11.23	hold
0.009 +- 0	106.5 +- 8.37	thanks t
0.008 +- 0	108.8 +- 8.23	mis
0.008 +- 0	109.9 +- 9.3	on hold
0.008 +- 0.00	110.6 +-11.98	never
0.008 +- 0	111.7 +- 4.52	us
0.008 +- 0.00	112.3 +-10.83	for a great
0.008 +- 0.00	113.6 +-11.15	united m
0.008 +- 0.00	114.6 +-13.73	fantast
0.008 +- 0.00	115.4 +-13.02	my connect
0.008 +- 0.00	116.1 +-12.59	search for
0.008 +- 0	116.7 +- 8.16	the worst
0.008 +- 0.00	116.8 +-12.37	had
0.008 +- 0.00	116.9 +-13.57	to
0.008 +- 0.00	117.8 +-12.16	min
0.008 +- 0	118.1 +- 9.92	ruin
0.008 +- 0	118.1 +- 7.29	illuminat
0.008 +- 0	118.7 +- 6.65	was awesom
0.008 +- 0	120 +- 7.36	excel
0.008 +- 0	121.3 +- 7.51	your
0.008 +- 0	123.2 +- 7.63	flight is delay

0.008 +- 0	124.8 +- 9.4	flight divers
0.008 +- 0	126.4 +- 9.12	suck
0.007 +- 0	128.3 +-12.01	qatar qatar
0.007 +- 0	128.6 +- 9.6	for mis
0.008 +- 0.00	128.9 +-15.09	so
0.007 +- 0.00	129.7 +-17.64	safe flight
0.007 +- 0	130.1 +-12.68	jetblue thanks for
0.007 +- 0.00	130.2 +-14.76	and now
0.007 +- 0	131.5 +- 9.69	flight was cancel
0.007 +- 0.00	131.7 +-15.4	turbl
0.007 +- 0.00	132 +-21.11	saf
0.007 +- 0	133.4 +- 8.64	even
0.007 +- 0	133.5 +- 9.02	experi
0.007 +- 0.00	138.6 +-14.52	du
0.007 +- 0.00	141.3 +-15.02	due t
0.007 +- 0	141.5 +-13.43	crew on
0.007 +- 0	141.9 +-13.01	ter
0.007 +- 0	143.5 +-10.84	the
0.007 +- 0	143.6 +-10.75	kudos t
0.007 +- 0	145.3 +-11.49	you jetblu
0.007 +- 0	145.8 +-11.31	thank you jetblu
0.007 +- 0.00	145.8 +-16.44	vi
0.007 +- 0.00	147.1 +-21.58	rus
0.007 +- 0	148.2 +- 7.65	vide
0.007 +- 0	149.8 +-10.58	awesome flight
0.007 +- 0	150.8 +-12.38	hav
0.007 +- 0	152.3 +- 6.56	aircanada great
0.007 +- 0	154.7 +- 9.03	bod
0.007 +- 0.00	155.4 +-24.04	stuck
0.007 +- 0	155.5 +- 6.48	shot down
0.007 +- 0.00	155.8 +-16.69	wonder
0.006 +- 0	157.5 +- 9.76	shot
0.006 +- 0	158.1 +-13.03	told
0.006 +- 0	158.7 +-11.93	united thank
0.006 +- 0	161.1 +-13.16	can
0.006 +- 0.00	161.6 +-20.82	my bag
0.006 +- 0	162 +-14.85	appreci
0.006 +- 0	162.2 +- 8.92	hor
0.006 +- 0	163.6 +- 8.38	flight
0.006 +- 0	164.3 +-15.81	refund
0.006 +- 0	166.9 +-21.7	united cancel
0.006 +- 0	167.2 +-15.79	passenger
0.006 +- 0	167.2 +-14.45	delayed for

0.006 +- 0	168.6 +-15.19	staff
0.006 +- 0.00	171.6 +-20.17	fail
0.006 +- 0	172.8 +-10.41	got
0.006 +- 0	173.4 +- 9.6	the flight
0.006 +- 0.00	173.9 +-22.35	jetblu
0.006 +- 0	174.9 +-19.19	jetblue for
0.006 +- 0	175.3 +-15.63	search for mis
0.006 +- 0	175.4 +-15.74	overbook
0.006 +- 0	176.1 +-17.89	reuter
0.006 +- 0.00	176.2 +-24.3	of airas
0.006 +- 0	178.9 +-14.56	on
0.006 +- 0.00	180.7 +-25.32	minut
0.006 +- 0	181.1 +-23.61	delta for
0.006 +- 0	184.3 +-19.1	enjo
0.006 +- 0	186.7 +-17.73	stil
0.006 +- 0	188 +-13.39	cant
0.006 +- 0	189.4 +-13.66	diverted t
0.006 +- 0	190.9 +-19.68	mad
0.006 +- 0	191.1 +-20.41	my flight
0.006 +- 0	191.4 +-21.4	crash
0.006 +- 0	191.6 +-12.88	the upgrad
0.006 +- 0	191.9 +-22.26	crew on flight
0.006 +- 0	192 +-17.15	service on
0.006 +- 0	193.1 +-24.41	screw
0.006 +- 0	193.9 +-18.59	down
0.006 +- 0	195.6 +-15.43	dl
0.006 +- 0	196.1 +-18.19	good
0.006 +- 0	196.1 +-20.57	gat
0.005 +- 0	197.9 +-23.47	flight thank
0.005 +- 0	198.1 +-17.81	am
0.005 +- 0	198.3 +-14.76	fun
0.005 +- 0	199.5 +-23.69	southwest airl
0.005 +- 0	200.1 +-20.85	tim
0.005 +- 0	200.1 +-20.82	for your
0.005 +- 0	201.1 +-16.78	qatar qatar airway
0.005 +- 0	201.4 +-15.88	cancelled m
0.005 +- 0	201.7 +-15.68	breakingnew
0.005 +- 0	202.9 +-14.44	the us
0.005 +- 0.00	204 +-27.58	our flight
0.005 +- 0	204.3 +-15.79	aircanada for
0.005 +- 0	209.5 +-25.06	plan
0.005 +- 0	210.9 +-31.33	connecting flight
0.005 +- 0	213.6 +-24.69	you cancel

0.005 +- 0	214.1 +-19.55	for hour
0.005 +- 0	214.7 +-10.06	you delt
0.005 +- 0.00	215.1 +-33.87	later
0.005 +- 0.00	215.9 +-35.08	lat
0.005 +- 0	216.4 +-16.79	agent
0.005 +- 0	216.5 +-15.09	clas
0.005 +- 0	216.9 +-26.8	los
0.005 +- 0	218 +-12.19	thank you delt
0.005 +- 0	219.3 +-25.34	delta thank
0.005 +- 0	219.3 +-26.53	any
0.005 +- 0	220.1 +-18.94	wreck
0.005 +- 0	220.1 +-24.66	great servic
0.005 +- 0	220.9 +-22.64	flight mh
0.005 +- 0	223 +-19.43	cancelled and
0.005 +- 0	223.5 +-23.64	hour dela
0.005 +- 0	227.5 +-11.79	you aircanad
0.005 +- 0	228.4 +-17.95	thank you t
0.005 +- 0	230 +-27.26	delayed hour
0.005 +- 0	230.8 +-11.16	airlines flight attens
0.005 +- 0	231.4 +-24.03	miss m
0.005 +- 0	232 +-18.61	thankyou
0.005 +- 0	232.2 +-18.1	southwest airlines flight
0.005 +- 0	232.7 +-24.33	we
0.005 +- 0.00	234.7 +-32.94	at
0.005 +- 0	235.4 +-25.6	an awesom
0.005 +- 0	236.9 +-19.63	flight was delay
0.005 +- 0.00	237.3 +-30.1	trying t
0.005 +- 0.00	238.1 +-33.11	had t
0.005 +- 0	242 +-21.35	jetblue thank you
0.005 +- 0	246.5 +-17.84	is cancel
0.005 +- 0	246.7 +-35.55	leav
0.005 +- 0	247.8 +-25.04	united my flight
0.005 +- 0	247.9 +-13.97	didnt
0.005 +- 0	248.2 +-29.61	our bag
0.005 +- 0	249 +-32.38	the great
0.005 +- 0	251.9 +-36.08	not hap
0.005 +- 0	252.7 +- 9.98	thank you aircanad
0.005 +- 0	257.1 +-45.33	much
0.005 +- 0	258.2 +-27.22	deltaas
0.005 +- 0	259.1 +-33.07	southwest flight
0.005 +- 0	259.5 +-32.25	hold for
0.005 +- 0	259.8 +-34.66	hom
0.005 +- 0	260.1 +-30.75	friens

0.005 +- 0	260.2 +-19.26	for mak
0.005 +- 0	260.3 +-37.63	aircanada thank
0.004 +- 0	262 +-22.49	for missing airas
0.004 +- 0	262.7 +-25.71	former
0.004 +- 0	263.1 +-28.17	destroyer
0.004 +- 0	263.4 +-29.5	delayed b
0.005 +- 0	263.8 +-34.88	ever
0.004 +- 0	264.6 +-32.35	for the upgrad
0.004 +- 0	268.5 +-33.82	hap
0.004 +- 0	269.3 +-29.49	compens
0.004 +- 0	269.4 +-25.9	to ges
0.004 +- 0	270.5 +-29.86	you jetblue for
0.004 +- 0	271.4 +-26.35	great customer
0.004 +- 0.00	273.4 +-36.39	frustr
0.004 +- 0	274.7 +-34.07	smooth flight
0.004 +- 0	275 +-19.52	been delay
0.004 +- 0	275.9 +-27.86	great crew
0.004 +- 0	276.8 +-30.17	delta great
0.004 +- 0	278.1 +-19.24	airasia plan
0.004 +- 0	279.3 +-12.08	you aircanada for
0.004 +- 0	279.5 +-31.75	got cancel
0.004 +- 0	279.5 +-39.81	on hold for
0.004 +- 0	282.8 +-60.36	no on
0.004 +- 0	283.4 +-49.69	ves
0.004 +- 0	283.5 +-50.43	of airasia flight
0.004 +- 0	284.7 +-33.39	try
0.004 +- 0	284.7 +-22.21	tel
0.004 +- 0	285.8 +-39.31	the fre
0.004 +- 0	288.7 +-41.24	another
0.004 +- 0.00	291.5 +-59.69	me t
0.004 +- 0	292.9 +-22.61	get on
0.004 +- 0	293.5 +-37.62	for dela
0.004 +- 0	294.5 +-34.95	you unit
0.004 +- 0	294.7 +-35.04	thank you unit
0.004 +- 0	295.3 +-40.9	emergency land
0.004 +- 0	296.3 +-21.04	united thanks for
0.004 +- 0	297.1 +-46.63	aw
0.004 +- 0	298.7 +-24.58	cancelled my flight
0.004 +- 0	300.2 +-30.44	shot down b
0.004 +- 0	300.5 +-49.04	need
0.004 +- 0	300.7 +-29.14	down b
0.004 +- 0	302 +-28.47	to rebook
0.004 +- 0	304.2 +-43.07	for the great

0.004 +- 0	304.9 +-44.43	hotel
0.004 +- 0	308 +-66.77	worst airl
0.004 +- 0	308.7 +-19.75	up
0.004 +- 0	314.5 +-38.51	canceling m
0.004 +- 0	315.3 +-51.93	care of
0.004 +- 0	315.5 +-30.8	you for th
0.004 +- 0	315.6 +-22.91	safes
0.004 +- 0	316.7 +-41.83	for ges
0.004 +- 0	317.7 +-47.19	to hear
0.004 +- 0	317.8 +-43.79	dont
0.004 +- 0	320.5 +-37.77	someon
0.004 +- 0	322.7 +-40.87	caus
0.004 +- 0	323.9 +-22.86	abc
0.004 +- 0	324.1 +-42.64	as search
0.004 +- 0	324.2 +-27.18	never fl
0.004 +- 0	324.3 +-34.52	united what
0.004 +- 0	324.7 +-36.33	on dl
0.004 +- 0	324.9 +-44.04	for the fre
0.004 +- 0	325 +-33.75	nav
0.004 +- 0	326.8 +-46.26	like book
0.004 +- 0	327.7 +-35.23	strand
0.004 +- 0	327.7 +-61.6	our
0.004 +- 0	330.3 +-36.87	morn
0.004 +- 0	331.1 +-26.7	toda
0.004 +- 0	332.5 +-47.49	over
0.004 +- 0	333 +-45.77	bc
0.004 +- 0	335.3 +-37.08	flight and
0.004 +- 0	335.6 +-33.39	great customer servic
0.004 +- 0	336.4 +-48.03	by
0.004 +- 0	336.7 +-46.37	love th
0.004 +- 0	337.1 +-55.06	delayed
0.004 +- 0	339.3 +-47.3	worst flight
0.004 +- 0	339.8 +-54.44	reschedl
0.004 +- 0	339.9 +-47.68	the crew
0.004 +- 0	340.4 +-43.92	missing flight
0.004 +- 0	340.7 +-47.97	notif
0.004 +- 0	340.7 +-29.74	canceled flight
0.004 +- 0	340.8 +-54.04	to th
0.004 +- 0	341.4 +-40.64	flight hour
0.004 +- 0	342.7 +-42.37	oversold
0.004 +- 0.00	343.2 +-72.83	ear
0.004 +- 0	343.2 +-44.73	us sens
0.004 +- 0	344.8 +-33.23	why

0.004 +- 0	344.8 +-44.14	lost m
0.004 +- 0	348.2 +-34	military shot
0.004 +- 0	349.3 +-32.12	us military shot
0.004 +- 0	350.5 +-55.92	injur
0.004 +- 0	351 +-73.91	thanks delt
0.004 +- 0	351.4 +-55.64	jetblue flight
0.004 +- 0	352.6 +-40.76	cn
0.004 +- 0	353.3 +-56.83	nys
0.004 +- 0.00	353.9 +-91.63	brok
0.004 +- 0	354.4 +-55.7	java se
0.004 +- 0	355.2 +-55.97	jav
0.004 +- 0	355.6 +-59.84	via nys
0.004 +- 0	355.7 +-42.88	united thank you
0.004 +- 0	356.6 +-48.77	thanks jetblu
0.004 +- 0	357.6 +-74.18	over an
0.004 +- 0	359 +-59.11	what
0.004 +- 0	359.5 +-95.07	best flight
0.004 +- 0	359.9 +-59.14	united airlines flight
0.004 +- 0	361.7 +-35.33	to wait
0.004 +- 0	361.9 +-39.55	a flight t
0.004 +- 0	364.3 +-22.01	that
0.004 +- 0	364.3 +-73.61	a
0.004 +- 0	365 +-46.88	supposed t
0.004 +- 0	367 +-51.35	ar
0.004 +- 0	367.6 +-48.41	delayed and
0.004 +- 0	370.9 +-59.84	great flight and
0.004 +- 0	371.1 +-67.69	wonderful flight
0.004 +- 0	371.3 +-69.27	need t
0.004 +- 0	378.1 +-26.73	united th
0.004 +- 0	378.1 +-62.61	airtran
0.004 +- 0	380.6 +-38	missing plan
0.004 +- 0	382.2 +-84.98	a southwest
0.004 +- 0	384.3 +-68.04	im
0.004 +- 0	384.9 +-61.48	very disappoint
0.004 +- 0	385.4 +-36.77	whit
0.004 +- 0	386.7 +-55.97	flight dela
0.004 +- 0	386.8 +-71.09	until
0.004 +- 0	389.5 +-64.37	of missing airas
0.004 +- 0	389.6 +-66.19	to miam
0.004 +- 0	390.8 +-48.34	flight thank you
0.004 +- 0	391.1 +-38.84	pra
0.003 +- 0	392.3 +-23.77	been on
0.003 +- 0	392.8 +-54.95	inflight

0.003 +- 0	395.3 +-57.49	ua
0.003 +- 0	395.6 +-50.81	and n
0.003 +- 0	395.9 +-61.92	miam
0.004 +- 0	396.5 +-55.68	by us
0.003 +- 0	396.6 +-56.03	you guy
0.003 +- 0	397.5 +-39.27	a cancel
0.003 +- 0	402.6 +-81.04	grat
0.003 +- 0	402.6 +-54.61	thanks to th
0.004 +- 0	403.1 +-107.69	thx
0.004 +- 0.00	403.3 +-100.42	fight
0.003 +- 0	405.4 +-56.26	is not
0.003 +- 0	410.8 +-88.07	wtf
0.003 +- 0	411.5 +-71.82	united
0.003 +- 0	413.6 +-45.43	a safe flight
0.004 +- 0.00	413.9 +-144.65	my connecting flight
0.003 +- 0	414.3 +-65.45	poor
0.003 +- 0	415.4 +-45.02	mh us
0.003 +- 0.00	415.5 +-99.77	land
0.003 +- 0	415.8 +-49.65	traveling t
0.003 +- 0	416.3 +-44.02	hours on
0.003 +- 0	416.5 +-54.85	help as search
0.003 +- 0	416.6 +-46.62	mh us milit
0.003 +- 0	417 +-55.46	service thank
0.003 +- 0	417.2 +-43.79	flight is cancel
0.003 +- 0	417.4 +-55.25	labor
0.003 +- 0	417.9 +-49.9	us nav
0.003 +- 0	418.3 +-49.07	johnspatricc
0.003 +- 0	418.4 +-32.31	you delta for
0.003 +- 0	418.5 +-53.07	jordan
0.003 +- 0	418.6 +-52.27	still n
0.003 +- 0	420 +-62.97	say
0.003 +- 0	421.5 +-55.72	malays
0.003 +- 0	422.9 +-57.79	thanks aircanad
0.003 +- 0	423.7 +-94.8	westjet for
0.003 +- 0	425.5 +-59.81	fabl
0.003 +- 0	426.9 +-78.94	in java se
0.003 +- 0	427.9 +-79.37	in jav
0.003 +- 0	428.4 +-60.87	almost
0.003 +- 0	429.8 +-82.85	flight to miam
0.003 +- 0	430.4 +-64.41	going on
0.003 +- 0	430.4 +-53.03	united for
0.003 +- 0	430.9 +-30.99	report
0.003 +- 0	434 +-117.84	amazing flight

0.003 +- 0	434.1 +-73.46	the flight attens
0.003 +- 0	434.2 +-51.36	a saf
0.003 +- 0	434.8 +-61.76	on flight
0.003 +- 0	440.8 +-83.98	delta thanks for
0.003 +- 0	441.9 +-53.99	avi
0.003 +- 0	443.3 +-26.1	making m
0.003 +- 0	445.5 +-73.81	nice flight
0.003 +- 0	452.7 +-100.44	because of
0.003 +- 0	456.3 +-118.08	onl
0.003 +- 0	457.3 +-38.04	you to th
0.003 +- 0	457.3 +-107.78	read
0.003 +- 0	457.3 +-46.74	white flight
0.003 +- 0	457.9 +-71.78	fuck
0.003 +- 0	458 +-66.32	big thank
0.003 +- 0	459.6 +-69.68	glad t
0.003 +- 0	459.9 +-72.2	bestairl
0.003 +- 0	460 +-51.18	has been delay
0.003 +- 0	460.5 +-63.27	an american airl
0.003 +- 0	460.8 +-84.36	loving th
0.003 +- 0	461.5 +-75.99	unaccept
0.003 +- 0	461.7 +-59.42	like booking
0.003 +- 0	462.1 +-65.52	destroyer t
0.003 +- 0	462.6 +-61.23	as search for
0.003 +- 0	462.9 +-82.28	tail
0.003 +- 0	467.5 +-71.9	to the crew
0.003 +- 0	468.2 +-82.42	aircanada
0.003 +- 0	468.6 +-83.65	work
0.003 +- 0	469.7 +-98	over an hour
0.003 +- 0	470 +-92.09	delta m
0.003 +- 0	470.9 +-67.56	ive been
0.003 +- 0	472.8 +-116.91	stuck in
0.003 +- 0	473.7 +-122.31	a southwest flight
0.003 +- 0	474.6 +-83.34	bump
0.003 +- 0	476 +-88.06	vouches
0.003 +- 0	477.7 +-78.49	join
0.003 +- 0	480.8 +-79.66	of mis
0.003 +- 0	482.9 +-62.13	miss my connect
0.003 +- 0	483 +-85.11	aircanada for th
0.003 +- 0	483.4 +-73.83	booking a flight
0.003 +- 0	483.7 +-85.36	suppos
0.003 +- 0	483.8 +-79.96	to mis
0.003 +- 0	484.7 +-89.62	fir
0.003 +- 0	486.3 +-72.11	enjoy your

0.003 +- 0	487.2 +-93.72	my flight got
0.003 +- 0	487.7 +-74.24	first clas
0.003 +- 0	489.7 +-89.01	i understand
0.003 +- 0.00	494.4 +-187.82	is the best
0.003 +- 0	495.7 +-55.07	glob
0.003 +- 0	496.8 +-77.17	deserv
0.003 +- 0	497 +-55.07	airways flight
0.003 +- 0	498.9 +-51.53	united you
0.003 +- 0	502.6 +-89.19	impres
0.003 +- 0	503 +-48.34	this morn
0.003 +- 0	504.1 +-75	the associ
0.003 +- 0	504.3 +-70.67	rt johnspatricc
0.003 +- 0	504.3 +-46.4	united flight delay
0.003 +- 0	504.4 +-60.12	is ridicl
0.003 +- 0	505 +-77.75	hilarious southwest
0.003 +- 0	505.2 +-55.49	delayed an
0.003 +- 0	505.7 +-53.39	is awesom
0.003 +- 0	506.5 +-59.28	delaying m
0.003 +- 0	506.5 +-70.98	wouldnt
0.003 +- 0	506.7 +-55.09	military shot down
0.003 +- 0	507 +-62.91	cancelled th
0.003 +- 0	507 +-69.75	down by us
0.003 +- 0	507.4 +-70.08	happyholiday
0.003 +- 0	507.7 +-57.98	boe
0.003 +- 0	508.2 +-55.22	mh mh
0.003 +- 0	508.7 +-75.04	associated pres
0.003 +- 0	509.4 +-73.03	enters da
0.003 +- 0	509.5 +-91.74	good flight
0.003 +- 0	509.5 +-70.49	very much
0.003 +- 0	510.7 +-80.07	the associated pres
0.003 +- 0	510.8 +-100.29	tried t
0.003 +- 0	511.1 +-115.12	this
0.003 +- 0	511.5 +-91	returns t
0.003 +- 0	512.8 +-94.61	the great flight
0.003 +- 0	514.4 +-58.4	southwest flight attens
0.003 +- 0	514.5 +-76.47	airlines flight divers
0.003 +- 0	514.5 +-115.7	pleasur
0.003 +- 0	514.7 +-52.14	noth
0.003 +- 0	516.2 +-132.06	great flight on
0.003 +- 0	516.3 +-109.86	sit
0.003 +- 0	516.8 +-96.33	do
0.003 +- 0	517.4 +-90.37	understand
0.003 +- 0	520.1 +-83.42	mechan

0.003 +- 0	523.2 +-106.86	a hour
0.003 +- 0	526 +-155.43	an amaz
0.003 +- 0	526 +-123.19	or
0.003 +- 0	527.4 +-54.48	upgrade on
0.003 +- 0	529.1 +-139.22	waiting on
0.003 +- 0	530.2 +-49.52	don
0.003 +- 0	530.3 +-121.87	out
0.003 +- 0	532 +-79.15	rep
0.003 +- 0	532.5 +-99.03	amp
0.003 +- 0	533.2 +-70.76	got delay
0.003 +- 0	537 +-121.35	offic
0.003 +- 0	537.2 +-73.07	other
0.003 +- 0	538.9 +-121.17	out of
0.003 +- 0	540.1 +-160.81	wont
0.003 +- 0	540.3 +-94.43	smuggl
0.003 +- 0	547.9 +-71.69	watch
0.003 +- 0	555.8 +-75.39	a cancelled flight
0.003 +- 0	556 +-79.15	westjet thank
0.003 +- 0	557 +-88.27	upgrade t
0.003 +- 0	559.6 +-88.28	been on hold
0.003 +- 0	559.7 +-61.39	was amaz
0.003 +- 0	559.9 +-126.43	cant ges
0.003 +- 0	561.1 +-60.32	is amaz
0.003 +- 0	562.5 +-163.52	found
0.003 +- 0	563.1 +-79.32	us military ves
0.003 +- 0	563.2 +-54.08	to get on
0.003 +- 0	563.4 +-71.1	one
0.003 +- 0	563.4 +-49.74	great flight from
0.003 +- 0	563.7 +-81.68	to fight
0.003 +- 0	564.1 +-51.29	expans
0.003 +- 0	564.2 +-78.77	military ves
0.003 +- 0	564.2 +-101.08	san francisc
0.003 +- 0	564.3 +-79.9	fight is
0.003 +- 0	564.4 +-101.04	francisc
0.003 +- 0	566.8 +-108.63	turbulence injur
0.003 +- 0	567.4 +-96.55	global flight
0.003 +- 0	567.6 +-94.87	rt johnspatricc mh
0.003 +- 0	567.7 +-90.41	santas global flight
0.003 +- 0	567.8 +-76.05	for missing plan
0.003 +- 0	568.1 +-67.62	ebay us
0.003 +- 0	568.1 +-60.82	to pay for
0.003 +- 0	568.6 +-104.53	flight diverted after
0.003 +- 0	568.8 +-81.13	isis lik

0.003 +- 0	569.2 +-85.89	isis like book
0.003 +- 0	569.3 +-85.39	fight isis lik
0.003 +- 0	569.3 +-83.19	vet traveling t
0.003 +- 0	569.8 +-82.4	to fight is
0.003 +- 0	570 +-79.81	traveling to fight
0.003 +- 0	571 +-86.05	vet travel
0.003 +- 0	571.6 +-84.06	military vet travel
0.003 +- 0	571.8 +-126.99	for cancel
0.003 +- 0	572 +-87.93	johnspatricc mh
0.003 +- 0	572.4 +-106.52	diverted after
0.003 +- 0	573.5 +-109.7	can you
0.003 +- 0	574.4 +-93.83	santas glob
0.003 +- 0	575.9 +-142.21	flight got
0.003 +- 0	576.8 +-100.74	had
0.003 +- 0	581.2 +-102.31	and
0.003 +- 0	583.4 +-53.24	aircanada thanks for
0.003 +- 0	584.6 +-108.27	wasnt
0.003 +- 0	585.8 +-159.17	to d
0.003 +- 0	590.7 +-103.02	for another
0.003 +- 0	590.9 +-70.1	flight delayed for
0.003 +- 0	595 +-105.93	yvr
0.003 +- 0	595 +-73.05	been cancel
0.003 +- 0	595.4 +-83.35	doh
0.003 +- 0	600.8 +-122.27	flight attendants
0.003 +- 0	605.7 +-151.83	never f
0.003 +- 0	608.6 +-121.19	effort
0.003 +- 0	611.9 +-92.4	tell m
0.003 +- 0	613.5 +-129.73	get hom
0.003 +- 0	614.9 +-83.13	with you
0.003 +- 0	617.2 +-94.31	has been
0.003 +- 0	619.2 +-86.96	delayed becaus
0.003 +- 0	619.6 +-92.99	explan
0.003 +- 0	620.5 +-123.86	whes
0.003 +- 0	621.4 +-83.06	sends destroyer
0.003 +- 0	621.9 +-108.24	my am
0.003 +- 0	623.4 +-113.36	sever
0.003 +- 0	624.2 +-89.55	sends destroyer t
0.003 +- 0	625.2 +-66.1	mh mh us
0.003 +- 0	625.3 +-128.19	wors
0.003 +- 0	625.6 +-83.09	nicest
0.003 +- 0	626.2 +-79.74	flight awesom
0.003 +- 0	626.6 +-124.86	final flight
0.003 +- 0	626.8 +-73.19	miami us

0.003 +- 0	626.9 +-86.29	destroyer to help
0.003 +- 0	627.7 +-61.19	united c
0.003 +- 0	627.8 +-79.65	chief
0.003 +- 0	627.9 +-88.99	enters day thre
0.003 +- 0	628.1 +-85.85	plane enters da
0.003 +- 0	628.6 +-78.05	to miami us
0.003 +- 0	628.7 +-82.84	plane enter
0.003 +- 0	629.4 +-82.42	how whit
0.003 +- 0	629.6 +-72.29	cancelled due t
0.003 +- 0	630.4 +-66.95	usbound
0.003 +- 0	631.2 +-134.21	my flight w
0.003 +- 0	631.3 +-105.04	i had
0.003 +- 0	631.9 +-78.26	missing plane enter
0.003 +- 0	631.9 +-79.94	blam
0.003 +- 0	631.9 +-88.58	black
0.003 +- 0	632.2 +-69.01	thanks for dela
0.003 +- 0	632.2 +-58.92	helicopter
0.003 +- 0	633.4 +-82.18	how white flight
0.003 +- 0	633.7 +-74.51	cancelled du
0.003 +- 0	633.7 +-104.52	victim
0.003 +- 0	634.2 +-62.36	been wait
0.003 +- 0	635 +-70.23	being delay
0.003 +- 0	635.2 +-96.74	hilarious southwest airl
0.003 +- 0	635.5 +-87.14	flight great
0.003 +- 0	636.3 +-82.2	us sends destroyer
0.003 +- 0	636.9 +-80.64	the delay
0.003 +- 0	637 +-83.49	missing airasia plan
0.003 +- 0	637.8 +-112.4	an american
0.003 +- 0	638.2 +-53.75	hours l
0.003 +- 0	638.2 +-265.29	desk
0.003 +- 0	638.5 +-140.96	do you
0.003 +- 0	639.4 +-102.99	test
0.003 +- 0	640.4 +-86.24	day thre
0.003 +- 0	642.1 +-132.43	just cancel
0.003 +- 0	642.2 +-101.42	by the associ
0.003 +- 0	642.6 +-124.54	wer
0.003 +- 0	644.9 +-155.87	be cancel
0.003 +- 0	645.8 +-102.87	and not
0.003 +- 0	647.2 +-110.06	a flight
0.003 +- 0	648.7 +-145.5	the bag
0.003 +- 0	655.2 +-140.47	cabin
0.003 +- 0	657.9 +-75.15	have a great
0.003 +- 0	659.4 +-88.18	the phon

0.003 +- 0	659.6 +-196.96	tr
0.003 +- 0	662.8 +-120.46	after
0.003 +- 0	663.7 +-96.45	tarmac
0.003 +- 0	663.9 +-64.02	better
0.003 +- 0	664.4 +-89.89	alway
0.003 +- 0	664.4 +-70.86	sham
0.003 +- 0	665.2 +-94.34	incompes
0.003 +- 0	665.4 +-163.11	shit
0.003 +- 0	669.4 +-113.54	flight crew w
0.003 +- 0	671.3 +-161.62	kind
0.003 +- 0	671.8 +-153.32	such
0.003 +- 0	672.5 +-122.69	pacific southwest airl
0.003 +- 0	673.3 +-122.06	pacific southwest
0.003 +- 0	673.8 +-136.03	yesterda
0.003 +- 0	675.4 +-66.93	usairway
0.003 +- 0	677.5 +-192.9	united your
0.003 +- 0	677.8 +-136.03	guy