Sentiment Analysis of Twitter Data for a Tourism Recommender System in Bangladesh

Najeefa Nikhat Choudhury

School of Science

Thesis submitted for examination for the degree of Master of Science in Technology.

Otaniemi 28.11.2016

Thesis supervisor:

Assoc. Prof. Keijo Heljanko

Aalto University
School of Science

AALTO UNIVERSITY
SCHOOL OF SCIENCE

ABSTRACT OF THE
MASTER'S THESIS

Author: Najeefa Nikhat Choudhury

Title: Sentiment Analysis of Twitter Data for a Tourism Recommender System in Bangladesh

Date: 28.11.2016 Language: English Number of pages: 8+65

Department of Computer Science

Master's Program in ICT Innovation

Supervisor and advisor: Assoc. Prof. Keijo Heljanko

The exponentially expanding Digital Universe is generating huge amount of data containing valuable information. The tourism industry, which is one of the fastest growing economic sectors, can benefit from the myriad of digital data travelers generate in every phase of their travel- planning, booking, traveling, feedback etc. One application of tourism related data can be to provide personalized destination recommendations. The primary objective of this research is to facilitate the business development of a tourism recommendation system for Bangladesh called "JatraLog". Sentiment based recommendation is one of the features that will be employed in the recommendation system. This thesis aims to address two research goals: firstly, to study Sentiment Analysis as a tourism recommendation tool and secondly, to investigate twitter as a potential source of valuable tourism related data for providing recommendations for different countries, specifically Bangladesh.

Sentiment Analysis can be defined as a Text Classification problem, where a document or text is classified into two groups: positive or negative, and in some cases a third group, i.e. neutral. For this thesis, two sets of tourism related English language tweets were collected from Twitter using keywords. The first set contains only the tweets and the second set contains geo-location and timestamp along with the tweets. Then the collected tweets were automatically labeled as positive or negative depending on whether the tweets contained positive or negative emoticons respectively. After they were labeled, 90% of the tweets from the first set were used to train a Naive Bayes Sentiment Classifier and the remaining 10% were used to test the accuracy of the Classifier. The Classifier accuracy was found to be approximately 86.5%. The second set was used to retrieve statistical information required to address the second research goal, i.e. investigating Twitter as a potential source of sentiment data for a destination recommendation system.

Keywords: Sentiment Analysis, Twitter, Sustainable Tourism, Big Data, Spark, Scala

Preface

I want to thank my Master thesis supervisor Professor Keijo Heljanko for his advice, guidance and support starting from selecting the thesis topic until the end of the thesis. I built the code for the experiments on top of the "Tweeather" project and therefore my thanks to Alexandru Roşianu for making his source code available through GitHub.

The I & E Study is an obligatory part of the EIT Digital Master Programme on Cloud Computing and Services (CSS); it is a part of the Final Degree Project that consists of an internship, a Master Thesis and an I & E Study report. I would like to thank my I & E project supervisor Olli-Pekka Mutanen for his guidance during my research. Also, I am grateful to all the 51 people who took time to participate in the market research survey conducted for this thesis and the I & E report. My special thanks to EIT Digital Master School for giving me the opportunity to write my final project on my own business idea. Thanks to all the staff, advisors and teachers of EIT Digital Master School, Technical University of Berlin and Aalto University who have made the last two years easy and educational.

I am extremely grateful to my parents, brother and friends for their constant moral support and unconditional love.

Otaniemi, 28.11.2016

Najeefa Nikhat Choudhury

Acronyms

ACID Atomicity, Consistency, Isolation and Durability. viii, 6–8

API Application Programming Interface. viii, 8, 10, 14–16, 32–35, 37–39, 41

BASE Basically Available, Soft state, Eventually consistent. 7

BSON Binary JavaScript Object Notation. 8

CAP Consistency, Availability, Partition Tolerant. viii, 7, 8

CRF Conditional Random Field. 17, 28

DStream Discretized Stream. 37

FCA Formal Concept Analysis. 28

FFCA Fuzzy Formal Concept Analysis. 28

FN False Negative. 43, 44

FP False Positive. 43–45

FPR False Positive Rate. 44

GB Gigabyte. 32, 40

GFS Google File System. 4

GFS Global Forecast System. 39

HDFS Hadoop Distributed File System. 4–6, 36, 38

HiveQL Hive Query Language. 5, 37

HSX Hollywood Stock Exchange. 16

HTTP Hypertext Transfer Protocol. 8, 33

ICS Index of Consumer Sentiment. 14

IDC International Data Corporation. 1

JSON JavaScript Object Notation. 8

LIWC Linguistic Inquiry and Word Count. 16

MAE Mean Absolute Error. 16

ML Machine Learning. 10, 19, 20, 27, 28, 37

MLlib Machine Learning Library. 37

NLP Natural Language Processing. 14, 28, 29

NN Neural Network. 39

NOAA National Oceanic and Atmospheric Administration. 39

NoSQL Not Only Structured Query Language. viii, 6–8

POS Parts of Speech. 17, 20, 21, 43

PPV Positive Predictive Value. 45

PR Precision-Recall. 43–45

RAM Random Access Memory. 32, 39

RDBMS Relational Database Management System. 6, 7

RDD Resilient Distributed Dataset. 37

REST Representational State Transfer. viii, 8, 32–34

ROC Receiver Operating Characteristic. 43–45

SQL Structured Query Language. 5–7, 36, 37

SVM Support Vector Machine. viii, 17, 18, 20, 24, 25

TN True Negative. 43, 44

TP True Positive. 43–45

TPR True Positive Rate. 44, 45

XML Extensible Markup Language. 8

YAML YAML Ain't Markup Language. 8

Contents

List of Figures

1 Introduction

The amount of data stored on the Internet is increasing rapidly and according to the International Data Corporation (IDC) report published in 2014, approximately 44 trillion gigabytes of data will be stored annually on the Internet by 2020 [57]. Every aspect of the human-computer interaction, be it sensors, medical equipment, social media, home appliances or GPS trackers, is contributing to the exponentially growing Digital Universe. The massive amount of data that contributes to the Digital Universe is referred to as "Big Data". The generated data may contain valuable insights into human behavior and beliefs. However, not all of these massive and diverse data contain valuable information and an attempt to process all of it might be hectic and a waste of time. Therefore, it is important to target high-value data to make analysis manageable and efficient. The IDC report [57] defines five criteria to identify high-value data:

Easy to Access: The data should be accessible and not stored in end user PCs or proprietary embedded systems.

Real-Time: The data should not be accessed when it is too late for taking time sensitive decisions and actions.

Footprint: Analysis of the data should affect a huge population, major part of the organization or majority of the customers.

Transformative: The data, if analyzed and acted upon, should benefit companies and individuals in a meaningful way.

Intersection Synergy: The data should have more than one of the aforementioned criteria.

1.1 Big Data

"Big Data" refers to data sets that exceed the processing capacity of traditional database systems [16]. Every digital data source, be it pictures, audios, videos, social media or sensors, can be dissected and analyzed as potential business drivers today. Most of these massive amounts of data sets are unstructured and messy, unlike data analyzed in traditional database systems. Therefore, Big Data necessitates tools and technologies that specifically address the storage and analysis of such data. The phrase "Big Data Analytics" is often inappropriately and ignorantly used to mean analyzing any large data set. However, it is important to identify the essential properties that separate any large data set from that which is suitable for Big Data Analytics. When a company or an individual identifies that they are in fact dealing with Big Data, they can efficiently use any of the established or emerging Big Data Technologies at their disposal [16].

1.1.1 Big Data Properties

The term "Big Data" initially was coined to mean a lot of data. Now, it not only means massive amounts of data, but diverse, unstructured data from various sources. In order to identify data fit for Big Data Analytics, data scientists at IBM have identified four essential dimensions or properties, more commonly knows as the "four V's of Big Data"[1]

Volume

The scale of data in the Digital Universe is expanding exponentially. Therefore, the fundamental appeal of Big Data Analytics is its capacity to process massive amounts of data. More data means better prediction models and higher accuracy, even with inferior algorithms. However, the magnitude of available data also represents a primary challenge for Big Data practitioners. Traditional database systems and analytical tools are not appropriate for storage or processing, but rather a distributed and scalable approach has to be adopted for such massive amounts of data.

Velocity

The speed with which new data is transmitted at every millisecond is unfathomable. In every millisecond there is a new status update on Facebook or a tweet on Twitter. Modern cars have about 100 sensors that monitors and records the car's condition whenever it is running. The billions of connected mobile devices, terabytes of trade information and multitudes of online retail interaction need to be analyzed in real-time to make effective decisions. It is possible to store fast moving data and process them later. However, some data might be too volatile or fast to be stored and so companies that can provide immediate analysis and feedback on the streaming data will have a competitive advantage over others.

Variety

One primary concern of Big Data Analytics is that most data in the Digital Universe come from diverse sources and is unstructured. If human users are involved in data input or generation, then it will also contain errors. Moreover, text on social media and blogs often contain colloquial language. This is the major issue for traditional database systems as they can only accommodate structured data. The data collected for analysis is never ready for processing. The unstructured data has to be cleaned and ordered so that meaningful information can be derived from it. The process of converting messy data into a formatted one may also cause loss of information. Therefore, data scientists have to take into consideration the different forms of data that they may need to process before choosing storage and analytics tools.

[1] `http://www.ibmbigdatahub.com/infographic/four-vs-big-data`

Veracity

Finally, it is important to consider the uncertainty of data available for processing. Data can be rife with inaccurate information. Low quality, inaccurate data may cost companies millions in expenses. Moreover, using such data for predictions and recommendations may cause long-term and dangerous harm starting from loss of customers to loss of life. Therefore, it is very important that companies and individuals trust the authenticity and reliability of the data used for analysis.

1.1.2 Big Data Technologies

The limited capability of traditional database systems to process huge volume, high velocity and different variety of data, has propelled many research over the past years dedicated to storage and analysis of Big Data. These research gave rise to many Big Data Technologies such as MapReduce [14], Hadoop[2], Hive[3], Spark[4], etc. While in the past analyzing Big Data was only viable for large corporations such as Google and Walmart, today technologies such as Hadoop have made Big Data Processing feasible and inexpensive also for companies and individuals with limited resources. This section provides an overview on some of the technologies that have had monumental impact in the evolution of Big Data Analytics.

MapReduce

MapReduce [14] is a programming model and execution environment originally created at Google to solve their web indexing challenges. It breaks down massive tasks into smaller ones and processes them in parallel over multiple nodes . Thus, MapReduce allows distribution of large data sets that cannot fit on a single machine. The MapReduce program is based on functional programming principles and contains two side-effect free functions- “map” and “reduce”. The master node creates the required or specified number of map and reduce worker nodes. In the MapReduce framework, the user provides a block of raw data to the “map” function. The master splits the input and divides them among multiple map workers. The “map” function then produces key-value pairs from the input and groups them together by key. So, the output of the “map” function is (key,(list of values)). The “map” output is then provided as input to the “reduce” functions of the reduce workers. The “reduce” function of each worker node processes the input and produces the final output. Figure 1 illustrates the execution process of a MapReduce program.

The functional design of MapReduce allows it to handle the complexities of parallel computations such as load balancing, fault tolerance and synchronization without programmer intervention. The programmer only has to provide the “map” and “reduce” functions and the run-time system manages the parallelism and execution on large clusters. In case of failure, the “map” and “reduce” functions are re-executed.

[2] `http://hortonworks.com/apache/hadoop/`
[3] `https://hive.apache.org/`
[4] `http://spark.apache.org/`

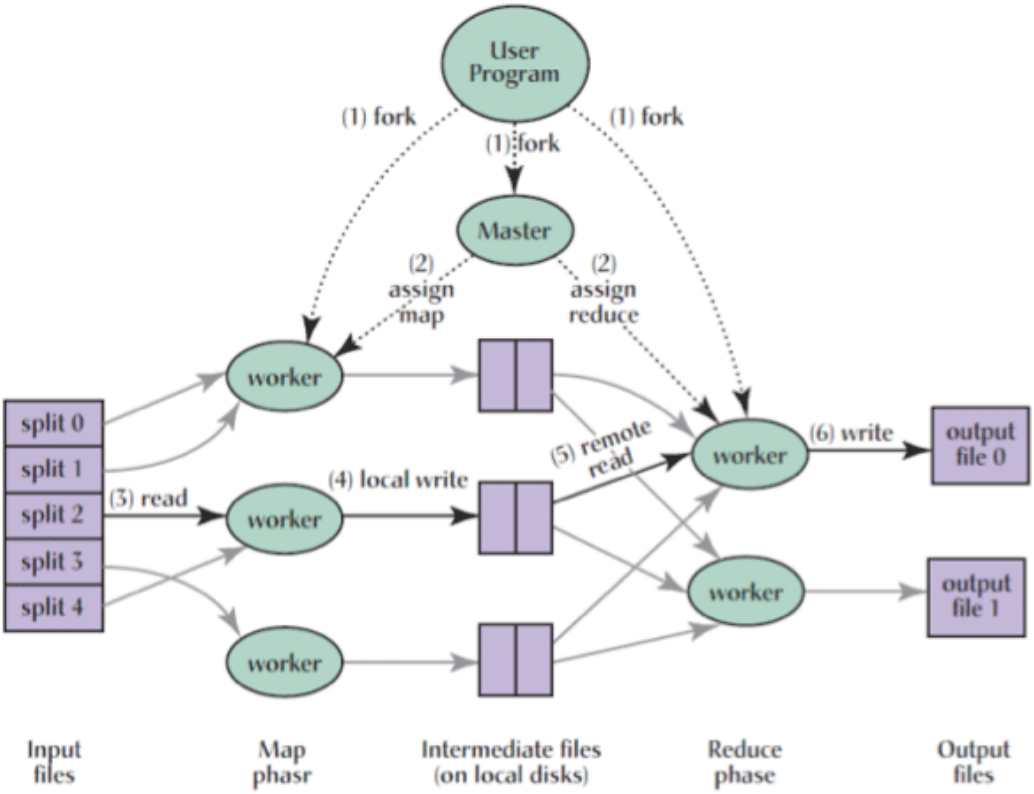


Figure 1: Execution Process of MapReduce Programs (Source: [14])

Hadoop

Apache Hadoop [64] has been created by Doug Cutting and is the most popular open source implementation of MapReduce. It is written in Java and today is maintained as a top-level project at Apache Software Foundation with a large community of contributors. Hadoop is designed for processing large-scale data and running processor-intensive Big Data Analytics. The Hadoop cluster consists of master node and worker nodes [50]. There might be one or several instances of a master node in a Hadoop deployment. The latter eliminates the risk of having a single point of failure. The Hadoop cluster may consist of hundreds or thousands of worker nodes.

There are three main processes or roles of a master node- JobTracker, TaskTracker and NameNode. The JobTracker interacts with client applications and distributes MapReduce jobs to other nodes. TaskTracker recieves tasks such as map and reduce from the JobTracker. The NameNode stores and tracks file directory trees and file metadata in the cluster. It also has control over access to files. Hadoop has its own file system known as the Hadoop Distributed File System (HDFS) [52] inspired by the Google File System (GFS) [18]. The worker nodes process and analyze the data by executing map and reduce jobs. Each worker node consists of two roles-

DataNode and TaskTracker. The DataNode's task is to store the data in HDFS and to replicate the data across clusters. In HDFS files are stored as blocks and the data is not cached. All data is replicated to three DataNodes for reliability and easy access. The Hadoop architecture and deployment is illustrated in Figure 2.

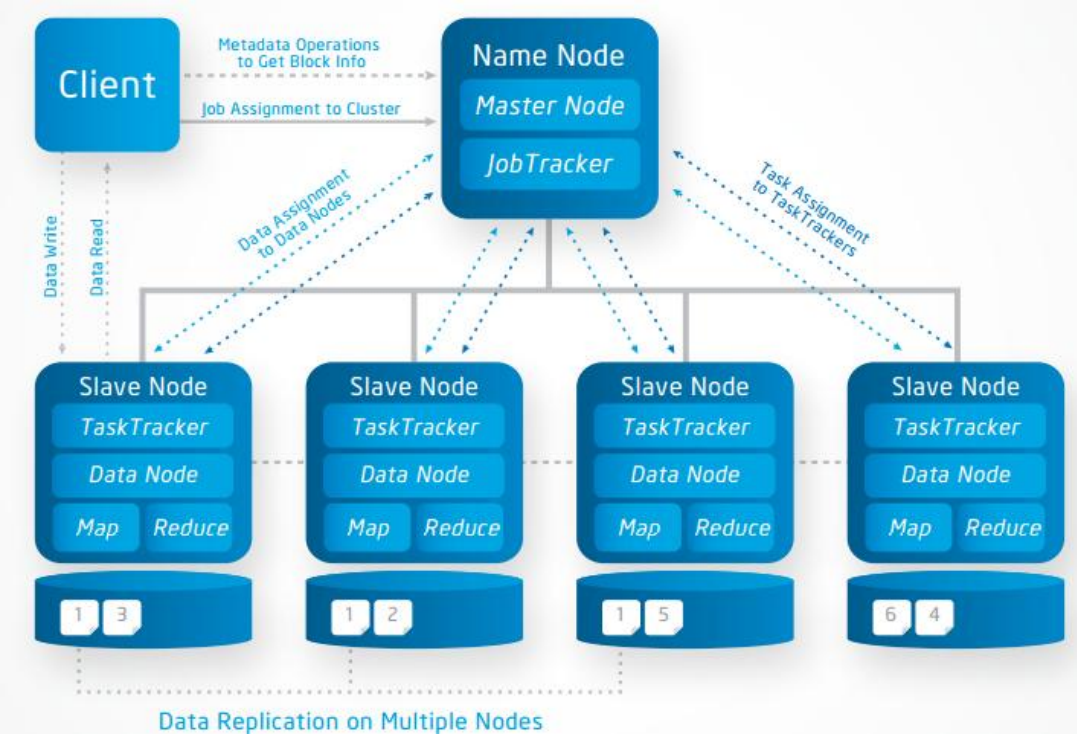


Figure 2: Hadoop Architecture and Deployment (Source: [3])

The Apache Software foundation built a large number of supporting tools that form the Hadoop ecosystem (see Figure 3) to make Big Data storage and analytics easier and more efficient. Apache HBase[5] is a distributed, scalable, column-oriented database that uses HDFS as the underlying data storage. It allows random reads and batch processing. Apache Zookeeper[6] is used as a coordination service by many applications on Hadoop clusters. Apache Pig[7] is a high-level data flow and programming tool. It is used on top of Hadoop with a programming language known as Pig Latin. Pig programs are highly parallelizable and thus massive data sets can be analyzed easily. Apache Hive[8] was developed by Facebook on top of Hadoop to provide data warehousing capabilities. Hive uses a language similar to Structured Query Language (SQL) known as Hive Query Language (HiveQL) to query, summarize and analyze data. Sqoop[9] and Flume[10] are used to integrate large

[5] https://hbase.apache.org/
[6] https://zookeeper.apache.org/
[7] https://pig.apache.org/
[8] https://hive.apache.org/
[9] http://sqoop.apache.org/
[10] https://flume.apache.org/

amounts of external data and transmit it to HDFS. Apachie Oozie[11] was designed for workflow scheduling. It stores the states, variables and workflow definitions of all active workflow instances in a database to manage Hadoop jobs. There are also many analytics related tools that can be used alongside Hadoop such as Datameer[12] and IBM BigSheets[13].

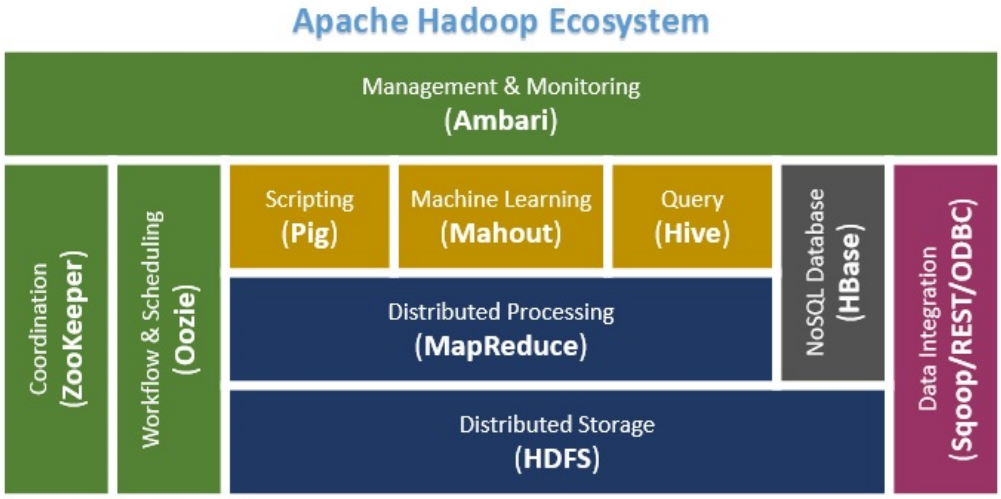


Figure 3: Apache Hadoop Ecosystem (Source: [4])

NoSQL

Relational SQL databases can be coupled with Big Data Technologies such as Hadoop to provide powerful analytics capabilities. However, the advent and rise of NoSQL [62] databases in the last decade greatly impacted the progress of Big Data Analytics. NoSQL, simply means it does not use SQL as a query language unlike traditional relational databases and is basically a non-relational database system. Relational Database Management System (RDBMS) [53] store and query structured data grouped into tables. The tables represent unique entities for example, "students" and "schools" for a university management system and have a pre-defined fixed schema. The tables consist of unique columns that represent properties of the entity and rows that represent individual records. The tables can have relationships among each other as shown with a blue line in Figure 4. One of the primary challenges of RDBMS is its inability to scale horizontally given its structured formatting and adherence to the ACID properties:

Atomicity: All or nothing of a transaction will succeed.

Consistency: A transaction will take the database from one consistent state to another.

[11] `http://oozie.apache.org/`
[12] `http://www.datameer.com/product/product-overview/`
[13] `http://www-01.ibm.com/software/ebusiness/jstart/bigsheets/`

Isolation: All transactions ate independent of each other.

Durability: A successful transaction will persist, even if application is. closed.

School Table

ID	Name
S001	University of Technology
S002	University of Applied Science

Student Table

School ID	ID	Name	DOB
S001	UT-1000	Tommy	05/06/1995
S001	UT-1000	Better	16/04/1995
S002	UAS-1000	Linda	02/09/1995
S002	UAS-1000	Jonathan	22/06/1995

Figure 4: Relational Database Example (Source: [2])

NoSQL addresses the challenges in RDBMS and makes scaling feasible. It does not rely on ACID properties which are incompatible with availability and performance requirements of large scale applications. NoSQL is built on the CAP theorem:

Consistency: Each operation will leave the database in a consistent state.

Availability: The database system is always available for modifications even in case of arbitrary network failure.

Partition Tolerant: In case the network is partitioned, the database system continues to function as before.

According to the CAP theorem, at most two of the CAP properties can be achieved at the same time in a distributed database system. In any distributed system, partition tolerance is a mandatory requirement for scalability. Therefore, distributed databases have to trade off between consistency and availability. NoSQL follows the Basically Available, Soft state, Eventually consistent (BASE) property. Thus, many NoSQL databases primarily focus on availability. Soft state means that the state of the database may change at any time. However, the database is eventually consistent, meaning after a certain time, if no input is added then the database using application will do the modifications needed to restore database consistency. Figure 5 illustrates the relationship between CAP theorem, ACID properties and NoSQL databases.

NoSQL databases are best for their schema-less or flexible schema design as new columns or properties can be added any time unlike SQL databases. There are several proprietary and open-source NoSQL data stores that are gaining popularity very rapidly. The NoSQL databases can be grouped according to the storage types-

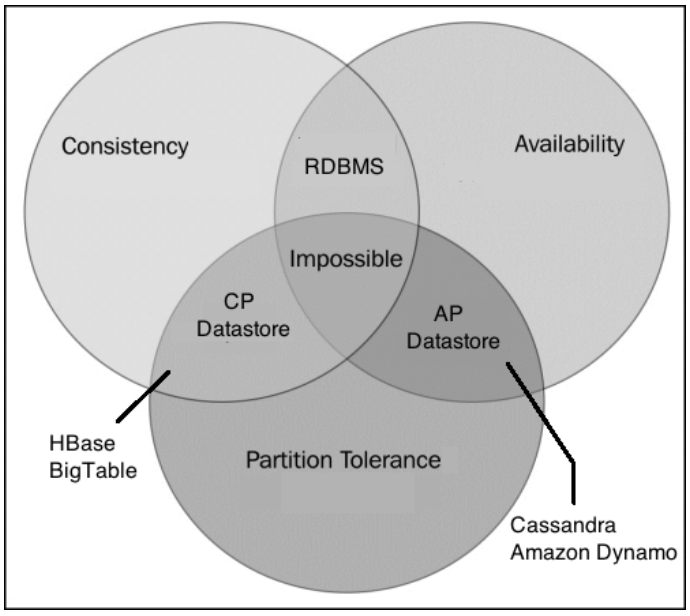


Figure 5: Relationship between CAP, ACID and NoSQL

Document Store	Key-Value Store	Graph	Other Cloud Datastores
MongoDB	Redis	Neo4J	BigTable
CouchDB	Membase	FlockDB	HBase
RavenDB	Voldemort	InfinteGraph	Cassandra
Terrastore	MemcacheDB		SimpleDB

Table 1: Popular NoSQL Databases

document store, key-value store, graph and column or row oriented cloud datastores. Document databases store and query semi-structured data in the form of Extensible Markup Language (XML), JavaScript Object Notation (JSON), Binary JavaScript Object Notation (BSON) or YAML Ain't Markup Language (YAML) accessed over Hypertext Transfer Protocol (HTTP) protocol using REST API. Key-value databases store a value against a key. The key has to be known to retrieve the value. Graph databases are specialized NoSQL databases designed for relation-heavy data sets. In graph databases, the relationships between nodes are represented as a graph. Finally, cloud datastores are databases that are provided as a service over a cloud computing platform. Some of the previously mentioned datastore types can also be run on the cloud. However, some cloud datastores are difficult to categorize by storage type. Some of these datastores are referred to as column oriented or row oriented or a hybrid of both. In column oriented databases, the data is stored and processed in columns instead of rows like traditional relational databases. Table 1 lists some popular NoSQL databases segmented according to the data model types [62].

1.2 Research Objective

Big data has benefited many industries as either a decision maker or business enabler. The tourism industry, which is one of the fastest growing economic sectors, can also benefit from the capabilities of Big Data Analytics. Travelers generate digital information in every phase of their travel- destination search, trip planning, travel booking, accommodation reservations and feedback on social media and travel apps. All of this information can be used to improve organizational operations of tourism companies, as well as the travel experience of tourists. One application of tourism related data can be to provide personalized destination recommendations. This thesis aims to study Sentiment Analysis as a method to provide context-based recommendations for tourism destinations. Twitter messages (known as tweets) regarding tourism and travel on twitter was analyzed to identify positive and negative sentiments. Also, the thesis explores the suitability of twitter as a source for sentiment data regarding tourist destinations in Bangladesh.

The primary objective of this research is to facilitate the business development of a tourism recommendation system for Bangladesh. Sentiment based recommendation is one of the features that will be employed in the recommendation system to promote local tourism. An online market research survey (see Appendix A) was conducted for the purpose of this thesis. 51 Bangladeshi adults participated in the survey. In the survey, 47.1% of the participants said that sentiment based recommendation is one of the top three interesting or important features that should be included in a travel application (see Figure 6). Therefore, due to the demand and potential for sentiment based recommendation as an innovative business driver, the thesis aims to address two research goals: firstly, to study Sentiment Analysis as a tourism recommendation tool and secondly, to explore twitter as a potential source of sentiment data in the context of tourism in different countries, with a focus on Bangladesh.

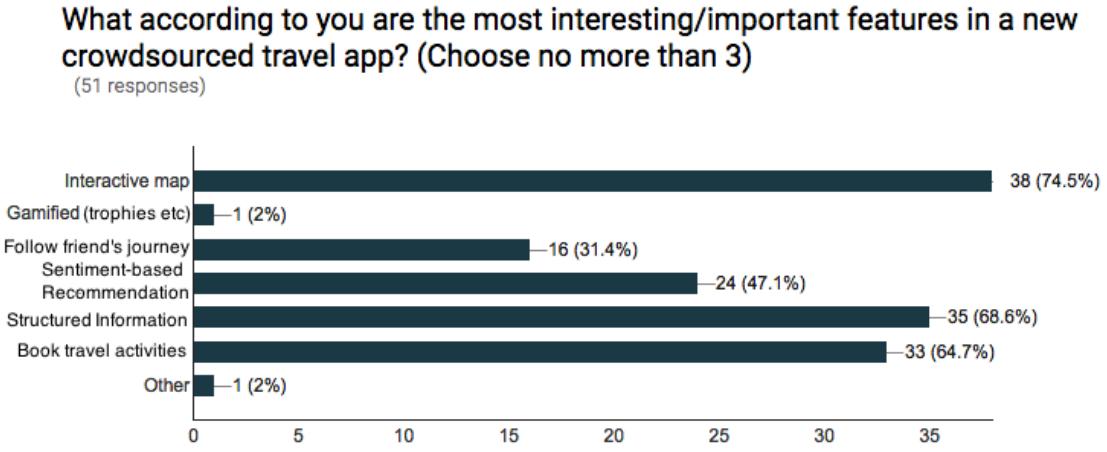


Figure 6: Market Survey Response for Tourism Recommender App

1.3 Research Methodology

The research methodology in this thesis is a combination of literature review and technical experiments. First, literature relevant to Big Data Technologies, Sentiment Analysis, Twitter Data Analysis and Machine Learning (ML) Classification were identified and collected. The collected material were a combination of books, magazine articles, white papers, journal proceedings, blogs and official application or tool websites. Then we verified the validity of the collected source by evaluating the authors' profile, number of citations and publisher reputation. After conducting a thorough literature review to understand the concepts, terms and related research in the aforementioned topics, the next phase of the research i.e. technical experiments were carried out. For the experiment, firstly data was collected from Twitter using the Twitter streaming API [60], secondly, Sentiment Analysis was carried out on the collected data and finally, the results were evaluated to answer the research questions. The experimental requirements, setup, procedure and results are discussed in details in Chapter 5 and 6.

1.4 Structure of Thesis

The thesis is arranged in 7 chapters. The rest of the thesis report is organized as follows: Chapter 2 gives an overview on different applications and research in Sentiment Analysis; Chapter 3 explains some common techniques used for classifying positive or negative sentiments in details; Chapter 4 outlines both general and thesis specific challenges and observations related to Sentiment Analysis; Chapter 5 gives a detailed overview of the environment and setup for analyzing the tweets; Chapter 6 explains the experimental procedure and evaluation of the results obtained in details; finally, Chapter 7 summarizes the thesis and suggests the future course of this project.

2 Background

The goal of this thesis is two-fold: first, to study Sentiment Analysis in the context of tourism recommendation and second, to investigate twitter as a potential source of valuable tourism related data for different countries, specifically Bangladesh. The web is rife with opinionated data about any topic imaginable, and tourism or travel is one such topic. Today, every human decision is based on the opinion of others and that includes where, when and how to travel. These opinionated data can be utilized for either business intelligence or application development by various organizations in the tourism industry. The primary motivation behind conducting this research is to use Sentiment Analysis in the development of an online tourism recommendation application, called "JatraLog". Twitter is a micro-blogging website and is a melting pot for opinions about almost everything- politics, technology, products, nations, people etc. Therefore, this thesis also aims to determine if Twitter contains valuable data about tourism which can be utilized for data analytics, specifically Sentiment Analysis, for providing tourism recommendation in "JatraLog". This Chapter provides general information about Sentiment Analysis research and applications.

Bing Liu defines Sentiment Analysis in his book "Sentiment Analysis and Opinion Mining" [29] as "the field of study that analyzes people's opinions, sentiments, evaluations, appraisals, attitudes, and emotions towards entities such as products, services, organizations, individuals, issues, events, topics, and their attributes". Sentiment Analysis is also interchangeably referred to as sentiment mining, subjectivity analysis, emotion extraction and most commonly opinion mining. Each of these terms might entail slightly different tasks, but overall relate to the same problem space, i.e. the identification of human opinion about a certain entity. While the field of emotion extraction may include identification of various human emotions such as anger, joy, sadness etc, Sentiment Analysis or opinion mining mostly refers to the identification of a positive, negative and sometimes neutral opinion towards an entity.

This Chapter is divided in Sections as follows: Section 2.1 gives an overview of different levels of granularity for segmenting the research domain of Sentiment Analysis; Section 2.2 discusses various applications that employ Sentiment Analysis and finally, Section 2.3 provides detailed information about the history and related research in the field of Sentiment Analysis.

2.1 Segmentation of Research Domain

Existing research in the field of Sentiment Analysis has mostly been segmented into three levels of granularity [29]: Document Level, Sentence Level, Entity and Aspect Level. The three levels for research in the field of Sentiment Analysis are explained below:

Document Level

In Document Level Sentiment Analysis the overall positive or negative opinion expressed in a document is identified. For instance, if a tech magazine publishes

a review of the new iPhone, then Document Level Analysis will extract an overall positive or negative sentiment in that article. This means, if an article highlights both positive and negative aspects of the product, it cannot be determined through Document Level Analysis. Also, in this level, the analyzer assumes that the document is only expressing about a single entity. While this kind of analysis might be appropriate for a product or service review application [41], [58], it may not work well for product comparison applications as each of the product's features should be analyzed for making purchase decisions. Thus, documents that compare multiple products, for example, iPhone 6 and Samsung Galaxy S4, cannot be analyzed in this level of analysis. Also, Document Level Analysis may be used to determine the alignment of a political blog, but in general blogs and forums may talk about multiple entities and therefore they cannot be analyzed on the Document Level.

Sentence Level

In Sentence Level Analysis, the opinion expressed in each sentence is determined. This level is also sometimes referred to as Phrase Level Sentiment Analysis. Usually researchers distinguish a neutral sentiment along with positive and negative sentiments in this level. This means, first objective sentences that express a fact are identified. Such sentences are said to have a neutral sentiment. Then subjective sentences, i.e. sentences that contain an opinion are further analyzed to identify positive or negative polarities [66]. Sentence level Sentiment Analysis are widely used while determining sentiments expressed within micro-blogs such as Twitter. Also sentiment extraction of reviews can benefit from sentence level analysis. While sentence level analysis can determine multiple opinions expressed in a single document, it however cannot determine feature based sentiments or comparisons similar to Document Level Analysis.

Entity and Aspect Level

Entity and Aspect Level Sentiment Analysis overcome the disadvantage of Document Level and Sentence Level Analysis in regards to feature based sentiment extraction. Thus, in case of Entity Level Analysis, language constructs such as documents, paragraphs, sentences or phrases are not analyzed unlike the other two levels. This level is more fine grained and extracts both the sentiment and the target of the sentiment [24], [25]. Thus, it is possible to determine exactly what features of a particular entity the author did or did not like. For instance, if a reviewer writes that a particular restaurant has excellent food, but poor services, then it is possible to determine positive sentiment for food and negative for service. Therefore, each aspect or feature can be analyzed. Also, it is possible to determine sentiments of documents that contain product or feature comparisons. This task introduces far more complexities and challenges in the Sentiment Analysis problem space.

2.2 Different Application Domains

Human society is driven by the opinion of other fellow human beings. Whether we buy a new product, read a book, watch a movie, eat at a restaurant, visit a new place

or even vote at an election, our decisions depend on positive and negative opinions of others. Before, we used to rely on opinions of friends and family before making purchase decisions and companies used to rely on survey or poll results to improve their products. However, now there is unlimited opinionated material available on the web in the form of blogs, news articles, micro-blogs, reviews, forum discussions, comments, etc. Opinions expressed online can provide valuable insights into the economical, social and political aspects of nations. So today, we not only rely on opinions from people we know, but also from complete strangers, regarding matters of both commercial and public interests. Companies also contain internal opinionated data in emails, call center communications, customer feedback etc. Sentiment from this plethora of opinionated web and internal data has been applied in many different commercial and research domains.

One of the most obvious applications of Sentiment Analysis is in review or feedback aggregation sites. Reviews of all kinds of consumer products and services are available all over the web that can be used to elicit sentiment information in order to provide review summaries. Recommendation systems can also augment feedback and review with Sentiment Analysis to recommend products that have not received too many negative feedback. Any kind of social or consumer trends can be analyzed through sentiment expressed in online communities and social media such as Twitter. The reviews do not have to be confined within the consumer space, it can also include opinions about political candidates or government policies. In fact, Sentiment Analysis can be used to predict election outcomes and analyze political trends or popularity on social media. Sentiment analysis can allow voters access to information such as what do the parties support, promote or oppose.

The summarization of reviews, trends and chatter about products, services or political parties does not only empower the public, but can also support business and government intelligence. Companies can asses their market situation, reception and the cause behind it. Similarly, political parties can figure out which campaign worked and which did not. Also, it is possible to analyze the tone of emails or texts to determine the sentiment of the sender. This for example, can help to discard or separate hate mails for famous public figures. Also, Sentiment Analysis can be useful in determining if an academic citation was made to support the findings of the cited material or to criticize it. This particular possibility will be of great value to academics and researchers in deciding the relevance or authority of a cited paper. Sentiment analysis might also have interesting applications in the field of sociology or psychology.

2.3 History and Related Research

Earliest research in the field of Sentiment Analysis and opinion mining can be traced back to the late 90s or early 2000s with the work of Hatzivassiloglou and McKeow [21], Das et al. [12], Morinaga et al. [36], Pang and Lee [41], Turney [58], Wiebe [65], Dini and Mazzini [15], Dave et al. [13]. One of the reasons why there has not been significant research in the area of Sentiment Analysis prior to this is because there

has not been enough opinionated digital content available before. While many of the research in Sentiment Analysis has been approached as a sub-topic of Natural Language Processing (NLP) or data mining research topics, other academic research approached Sentiment Analysis from an application-specific viewpoint. This thesis focuses on Sentiment Analysis for a specific application, i.e. tourism recommendation and therefore, this section reviews some of the application-oriented Sentiment Analysis research in details.

Moreover, different application-oriented sentiment research have utilized various different data sources such as social media, blogs, news articles, reviews etc to achieve their goals. In recent times, due to the popularity of Twitter and the myriad of user generated data available on it, many of the latest research have exploited Twitter as the data source. This thesis also utilizes tweets about tourism and travel for Sentiment Analysis. Therefore, the related literature on application-oriented Sentiment Analysis has been divided into two categories: Sub-section 2.3.1 gives a detailed overview of research that have used Twitter as the primary data source, while Sub-section 2.3.2 briefly discusses research that have conducted Sentiment Analysis with the means of other data sources.

2.3.1 Twitter Sentiment Analysis

Twitter messages, unlike most other data sources, are extremely short with a maximum of 140-character limitation and contain different languages, local slang, internet slang, informal language and misspellings. Thus, some Sentiment Analysis techniques and feature selections in case of other sources may not apply to Twitter and vice versa. Hong and Skiena [23] studied the relationship between the National Football League (NFL) betting line and public opinion expressed on Twitter as well as other sources, such as blogs and news media, thus aggreagating different types of data sources in their research. Shimada et al. [51] conducted a research very similar to this thesis, i.e. Sentiment Analysis of Twitter data for tourism. The primary objective of the research was to build a tourism information analysis system for Iizuka, a local city in Japan. They extracted tweets about popular tourist spots and events within the city using the Twitter API. They applied an unsupervised machine learning approach to build a naive Bayes classifier to identify positive and negative tweets. They also compared their results with dictionary-based classification approach (accuracy = 0.76) and showed that their method (accuracy = 0.89) achieved higher accuracy [51].

O'Connor et al. [38] found a correlation between political opinions of the public in polling surveys and sentiments expressed in Twitter messages during the years 2008 and 2009. The consumer confidence and political opinion polls were collected from several polling organizations such as Consumer Confidence Index, the Index of Consumer Sentiment (ICS) from the Reuters/University of Michigan Surveys of Consumers, the Economic Confidence index from the Gallup Organization, daily tracking poll for the approval rating of Barack Obama for the presidential position by from the Gallup Organization and various other voter polls taken during the 2008

U.S. presidential period. They collected 1 billion Twitter messages using the Twitter API for topic-based Sentiment Analysis over the years 2008 and 2009.

After collecting the Twitter messages, they identified the messages that contain the desired topic using keywords, such as, *economy, job, jobs* for "consumer confidence"; *obama* for "presidential approval" and *obama and maccain* for "elections". Then they used the subjectivity lexicon[14] provided by OpinionFinder[15] to count positive and negative messages. The lexicon contains a list of 1600 words labeled as positive and 1200 words labeled as negative. Each message may contain both positive and negative sentiment words. Therefore, they used the frequency of the sentiment words to calculate a sentiment score to classify messages as positive or negative. The research yielded very high correlation between consumer confidence and political opinion polls and sentiment trends on Twitter messages. The result varied for different datasets: in most cases the correlation was found to be higher than 70% (the best result was around 86%), while in some cases the correlation was found to be poor (around 10%). O'Connor et al. proposed that by improving the sentiment classification process, publicly available social media data can eventually replace time consuming surveys and polls [38].

Party	All mentions		Election	
	Number of tweets	Share of Twitter traffic	Election result*	Prediction error
CDU	30,886	30.1%	29.0%	1.0%
CSU	5,748	5.6%	6.9%	1.3%
SPD	27,356	26.6%	24.5%	2.2%
FDP	17,737	17.3%	15.5%	1.7%
LINKE	12,689	12.4%	12.7%	0.3%
Grüne	8,250	8.0%	11.4%	3.3%
			MAE:	1.65%

** Adjusted to reflect only the 6 main parties in our sample*

Figure 7: Share of Tweets and Election Results (Source: [56])

Another research on Sentiment Analysis in a political context was published by Tumasjan et al. [56]. In this case the research focused on investigating if Twitter is used extensively as a platform for political discussion and if political sentiments on Twitter can be used to predict the election outcome of the 2009 German federal election. They collected approximately 1 million tweets either mentioning the major political parties or the most popular politicians. The tweets were collected over a few months (August to September) prior to the election in 2009. They used a text analysis software, LIWC2007[16] [42] to automatically extract the sentiment of the

[14] http://mpqa.cs.pitt.edu/lexicons/subj_lexicon/
[15] http://mpqa.cs.pitt.edu/opinionfinder/
[16] http://liwc.wpengine.com/

tweets. In this research not only positive and negative sentiments were identified, but also other moods such as future and past orientation, sadness, tentativeness, certainty, work, achievement, anxiety, anger and money were considered to create political sentiment profiles for parties and candidates. The collected tweets were in German, but they were translated to English so that they can be processed using the Linguistic Inquiry and Word Count (LIWC) English dictionary.

Firstly, Tumasjan et al. found that while Twitter is extensively used as a platform for political discussion, the discussions are dominated by a small number of users (approximately 4% users created 40% of the messages). Secondly, they found that sentiment profiles for parties and candidates reflect the political proximity between parties regarding different issues before election, such as a potential coalition partnership after the election. Finally, they determined that the number of Twitter mentions of each party or political candidate closely resembled the election result and the Mean Absolute Error (MAE) of the prediction was 1.65% (see Figure 7) [56]. Another research by Bollen et al. [9] also extracted different moods such as calm, alert, sure, happy etc, instead of just positive or negative sentiments from Twitter to predict the stock market prices.

In 2010, Asur and Huberman published a research to predict box-office revenues for movies based on chatter and sentiments expressed on Twitter [7]. They used the Twitter Search API [61] to collected 2.89 million tweets about 24 Hollywood movies released over three months using keyword filtering and sanity checks. In the paper, they analyze the *critical period* for each movie, which they define as the time starting from one week before the movie release, typically consisting of promotional content to two weeks after the release, mostly consisting of viewer opinions. The research was developed based on two primary goals: first, to find a correlation between the amount of Twitter chatter about a movie prior to release and the box-office revenue outcome and second, to investigate the role of sentiments to predict future revenue outcome. Firstly, in the research they found that there is a strong positive correlation (correlation coefficient = 0.90) between rate of tweets about movie prior to release and box-office revenue generated in the opening week after release. They also compared their results to real box-office revenue information to gauge the accuracy of their prediction. They showed that the amount of attention and promotion on social media before release can directly impact the revenue outcome.

Asur and Huberman proved that their predictive model based on social media chatter has higher accuracy than that of the Hollywood Stock Exchange (HSX)[17], which is usually considered to be the gold standard. For the second part of their research goal, they built a sentiment classifier using the LingPipe linguistic analysis package[18] to identify positive, negative and neutral tweets. They manually labeled the training data with the help of workers from Amazon Mechanical Turk[19]. They obtained 98% accuracy for sentiment classification. They then used the trained classifier to predict

[17] `http://www.hsx.com/`
[18] `http://alias-i.com/lingpipe/`
[19] `https://requester.mturk.com/`

Predictor	Adjusted R^2	$p - value$
Avg Tweet-rate	0.79	8.39e-09
Avg Tweet-rate + thcnt	0.83	7.93e-09
Avg Tweet-rate + PNratio	0.92	4.31e-12
Tweet-rate timeseries	0.84	4.18e-06
Tweet-rate timeseries + thcnt	0.863	3.64e-06
Tweet-rate timeseries + PNratio	**0.94**	1.84e-08

Figure 8: Prediction of second weekend box-office gross (Source: [7])
(*PNRatio = ratio of positive to negative tweets for a movie. *thcent = number of theaters the movie was released in.)

the sentiment for the 24 movies during their respective *critical period*. They found that some movies, such as “The Blind Side" had lukewarm opening sales in the first week, but the sales boomed significantly in the second week. This outcome was strongly correlated to the increase in positive sentiments about the movie post-release. Thus, Asur and Huberman concluded that by adding sentiment information to the regression equation, the can improve the prediction (see Figure) instead of using only tweet-rate to build the regression model [7].

Several research on Sentiment Analysis that are not application-oriented have also utilized Twitter as the sentiment data source. Pak and Paroubek [40] extracted around 3 million tweets from Twitter and labeled them as positive, negative or neutral depending on the presence or absence of emoticons respectively. They constructed unigrams, bigrams and trigrams from the tweets to train their sentiment classifier and found that bigrams have the best accuracy. In order to choose the best sentiment classifier, they experimented with Naive Bayes classifier, SVM and Conditional Random Field (CRF) and found that Naive Bayes yields the maximum accuracy. Kouloumpi et al. [28] investigated how useful different features (e.g. n-grams, lexicons, Parts of Speech (POS)) that are used for Sentiment Analysis of formal text and micro-blogging features(e.g. emoticons, abbreviations, all-caps for emphasis) are for analyzing sentiments of Twitter messages. Go, Bhayani and Huyang [19] published one of the earliest works in Twitter Sentiment Analysis. They used unigrams, bigrams and POS tags as features for training the classifiers. They employed three different machine learning classifiers- Naive Bayes, SVM and Maximum Entropy to train emoticon tweets and found that SVM (accuracy = 82.2%) had slightly better accuracy than Naive Bayes (accuracy = 81.3%).

2.3.2 General Sentiment Analysis

Even though the proliferation of Twitter happened fairly recently, other forms of opinionated data were available on the web much before the advent of Twitter. Blogs,

emails, review sites, forums and books have been utilized to publish several application-oriented Sentiment Analysis research. For example, Liu et al. [30] extracted sentiments from blogs to predict product sales performance. Mohammad and Yang [35] conducted Sentiment Analysis on emails to track how emotions of genders differ in personal and workplace emails. Joshi et al. [26] published a research to predict the opening weekend revenue of movies based on sentiments extracted from reviews of film critics. Mohammad [34] used Sentiment Analysis to track emotions in novels and fairy tales. Sakunkoo and Sakunkoo [49] used sentiments to analyze the social influence in online book reviews. Groh and Hauffa [20] used sentiments in communicative texts such as e-mails to characterize social relations. Zhang and Skiena [68] extracted sentiments from blogs and news to study trading strategies.

Pang et al. [41] published a research on Sentiment Analysis using three different machine learning techniques- Naive Bayes classifier, Maximum Entropy and SVM. They used movie reviews by viewers as the data source and showed that SVM outperforms the other two techniques. This research was not application-oriented and aimed at identifying if machine learning techniques perform as well on Sentiment Analysis problems as other text classification problems.

3 Sentiment Classification

The plethora of structured and unstructured data stored online is a melting pot of information for various business and research domains. Much research effort in the field of data mining or data analysis over the past years has been dedicated to automatically classify text based on subject matter, genre, source, language etc in order to efficiently sort and manage the data. In some applications (discussed in Chapter 2) it might be useful to analyze the sentiment expressed in the text. Sentiment analysis is a classification process where the data is classified as having a positive or negative polarity. In some cases, a neutral sentiment is also identified during classification.

There are several algorithms that are used for traditional text classification and can also be applied for Sentiment Analysis. The methods can be broadly divided into two groups: ML techniques and Lexicon-based techniques. Figure 9 illustrates some of the most commonly used classification techniques [32]. The chapter provides a detailed overview of some popular techniques used for Sentiment Analysis.

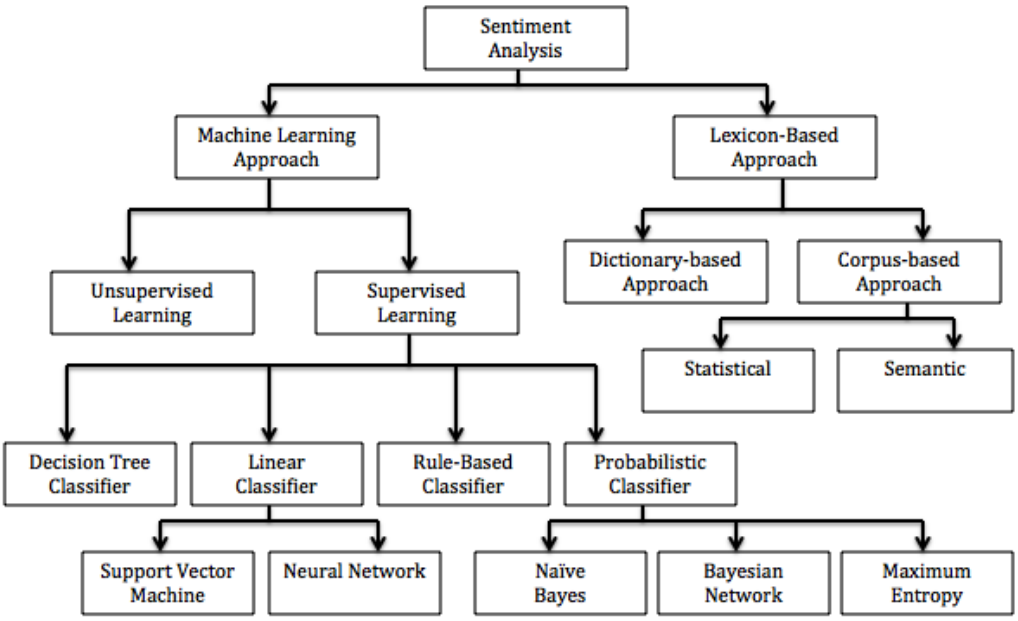


Figure 9: Text Classification Techniques (Source: [32])

3.1 Machine Learning Techniques

ML approach makes use of standard ML algorithms to solve Sentiment Analysis as a text classification problem, where the data is labeled by one of three classes: positive, negative or neutral. The "neutral" class might be discarded in certain

cases. Machine Learning techniques can be broadly divided into two categories: Unsupervised Learning and Supervised Learning.

Unsupervised ML techniques do not require any training data set. Therefore, unsupervised methods such as Clustering and Topic Modeling are useful in cases where training data is unavailable or difficult to find. These methods are used to automatically group similar type of data objects within a collection of objects. Supervised learning is the more commonly used method for Sentiment Analysis [29]. Sections 3.1.1, 3.1.2 and 3.1.3 describe three frequently used supervised ML techniques: Naive Bayes Classifier, Maximum Entropy and SVM respectively.

In supervised learning methods, a model is created using a set of training data, $D = \{X_1, X_2, ..., X_n\}$, where $X_1, X_2, ..., X_n$ are separate records, each manually labeled with a specific class. The records are classified according to syntactic or linguistic features or a combination of both. A classifier is then trained using one of the standard ML algorithms. After the training is complete, the classifier is used to predict the label of an instance of unknown class based on the selected features. The key to higher accuracy in Sentiment Analysis or any text classification problem is to select a set of useful features. Some possible features for sentiment classification are:

Terms Frequency and Presence: Term frequency count is the most commonly used feature for traditional text classification problems. These terms are individual words, called unigrams. Many researchers have used n-grams (bigrams and trigrams) to get better results [13]. However, in case of some domains, unigram may perform better than n-grams, for example, in case of sentiment classification of movie reviews [41]. Frequent appearance of terms may not always prove effective in overall Sentiment Analysis research as they do for traditional text classification, such as identifying the topic of a document. In such cases, term presence might be a better feature for sentiment classification rather than term frequency [41]. Term Presence is a binary value assigned to a term indicating if it is present or not in the text.

Part of Speech Tagging: The POS of individual words is a commonly used feature in Sentiment Analysis research. Some parts of speech, for example, adjectives are considered to be strong indicators of opinions or subjectivity in a sentence. One of the earliest research in Sentiment Analysis has been in identifying the semantic orientation of different adjectives [21] and subsequently finding a correlation between presence of adjectives and sentence subjectivity [22]. However, adjectives are not the only indicators of sentence subjectivity. Researchers have shown that other parts of speech such as verbs (e.g. *love*) and nouns (e.g. *gem*) can also indicate subjectivity [41]. Apart from using POS of individual words for subjectivity detection, Turney et al. [58] proposed using pre-selected phrases with specific parts or speech patterns, mostly containing an adjective or an adverb, in an unsupervised setting.

Sentiment Lexicon: Sentiment words or opinion words have been commonly used to effectively identify the polarity of sentences. For example, *amazing, good, happy* indicate positive sentiment, while *poor, sad, terrifying* indicate negative sentiment.

While most of the sentiment words are adjectives or adverbs, they can also be nouns and verbs. Apart from individual words, there are also sentiment phrases and idioms, such as "being over the moon", which means being extremely pleased, indicates positive sentiment. The sentiment words, phrases and idioms together form a sentiment lexicon or opinion lexicon which can be used to identify sentiment polarity. While a sentiment lexicon is a very useful feature, it is however not enough to identify sentiment polarity [29]. This maybe due to a number of reasons, for example, sentences without sentiment words or idioms may contain subjectivity, sentences with sentiment words may not be subjective (e.g. interrogative sentences) and positive or negative words might mean the opposite in certain domains.

Rules of Opinions: There are many expressions and compound statements that may indicate polarity of sentences depending on certain composition rules or domain knowledge. Apart from the sentiment words or phrases, these rules can also be used for increasing accuracy of sentiment classification results.

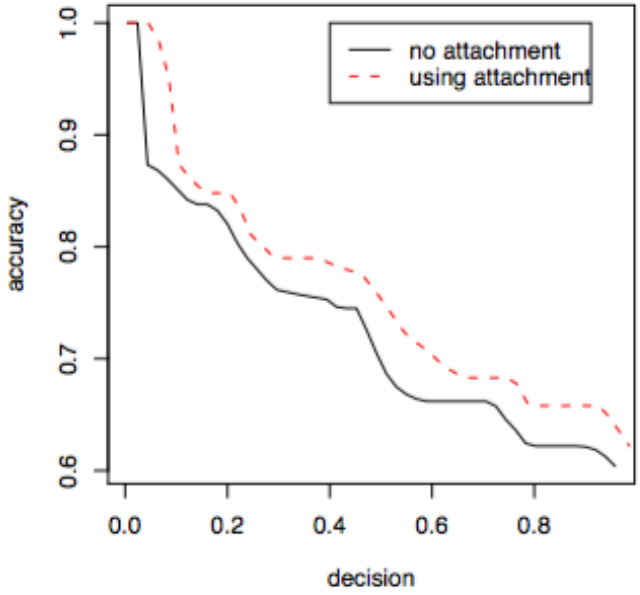


Figure 10: Impact of attaching negation words (Source: [40])

Sentiment Shifters: Sentiment shifters change the polarity of positive words to negative and vice versa. These shifters are very improtant features to consider during Sentiment Analysis, especially while using sentiment lexicon or POS tagging. Negation words, for example *not, don't, cannot*, are the most common types of sentiment shifters. In the following example, "I do not like apples", if negation word is not attached as a training feature, then it might be wrongly identified as positive due to the presence of the positive sentiment word *like*. Pak and Paroubek [40] show how the attachment of negation words during Sentiment Analysis can increase the prediction accuracy (see Figure 10). However, negation words need to be handled

with care as not all of them are sentiment shifters, for example, the word *not* in the phrase "not only good" does not shift the orientation of the phrase from positive to negative. There are other types of words that may act as sentiment shifters in certain cases. For example, modal auxiliary verbs (e.g. *could* be better, *should* improve) and presuppositional items (e.g. *barely* nice, *hardly* functions).

Emoticons: Emoticons are symbols used in digital communication such as emails, chatting or micro-blogging to convey human emotions. These emoticons can be used to identify sentiment polarity. For example, a smiling face or heart indicates positive sentiment, while a frowning or crying face indicates negative sentiment. Emoticons can be an effective feature to identify sentiments of social media data, reviews or emails. Pak and Paroubek [40] have used emoticons to collect and label a corpus of positive and negative tweets from Twitter. They then extracted other features, such as term frequency (unigrams, bigram, trigram) and negation words from the collected tweets for training a classifier to identify sentiment polarity. While emoticons are good features for labeling training data, they might not be effective by themselves for training the classifier.

3.1.1 Naive Bayes Classifier

Naive Bayes is a collection of several algorithms based on the Bayes Theorem [8], which is a probability theory that predicts the occurrence of an event due to a new evidence related to that event. Therefore according to Bayes Theorem, the probability of event A occurring given event or evidence B is true, P(A|B) is:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)},$$

where

1. $P(B) \neq 0$,
2. P(A) and P(B) are the probabilities of event A and B occurring independent of each other respectively, and
3. P(B|A) is the probability of event B occurring given event A is true.

In case of Naive Bayes for document classification, 'A' represents the "label" or "class", while 'B' represents the "feautures" [32]. The fundamental principle of Naive Bayes classifiers is that all features used to classify a document is assumed to be independent of each other [46]. For instance, if the features long, sweet and yellow are used to classify a fruit as a banana, then Naive Bayes treats each feature independently in calculating the probability of a fruit being a banana regardless of the relations between the features. However, in many real life scenarios, the selected features may not be independent of each other and this is a major disadvantage of Naive Bayes algorithm. In spite of this disadvantage, Naive Bayes outperforms many other algorithms and is simple to understand and easy to build with a small training data set.

Student Answers	Leader	Sincerely	Education	Total
Letter	0	15	10	30
Essay	40	35	45	50
Argument	10	15	5	20
Total	50	65	60	100

Table 2: Data Set for Naive Bayes Example

Thus, Naive Bayes is used to predict the class of a document using probability. In order to explain Naive Bayes, lets consider the following text classification example. Suppose, students are asked to write a letter to their teacher or an essay about their role model or an argumentative essay about free higher education in an English exam. We want to classify the student exam answers into the classes be "Letter", "Essay" and "Argument" based on three features or words that appear in the answers: "leader", "sincerely" and "education". Suppose we have 100 student answers for training. Table 2 shows the training data set and the numbers indicate how many answers of a class contain the corresponding feature.

With the help of the Bayes theorem and the information provided in the above table we can predict the class of a new student answer, given it contains the words leader, sincerely and education. In accordance with the "Naive" assumption that all features are independent, a probability equation can be derived from the Bayes theorem as follows:

$$P(class|features) = \frac{P(f_1|class)P(f_2|class)....P(f_n|class)P(class)}{P(features)},$$

where $f_1, f_2...f_n$ are the set of features used to categorize the documents into classes. The respective probabilities of whether the student answer is a "Letter", "Essay" or "Argument" can be calculated as follows:

$$P(Letter|Leader, Sincerely, Education)$$
$$= \frac{P(Leader|Letter)P(Sincerely|Letter)P(Education|Letter)P(Letter)}{P(Leader),P(Sincerely),P(Education)}$$
$$= \frac{0X0.5X0.333X0.3}{P(evidence)} = 0$$

$$P(Essay|Leader, Sincerely, Education)$$
$$= \frac{P(Leader|Essay)P(Sincerely|Essay)P(Education|Essay)P(Essay)}{P(Leader),P(Sincerely),P(Education)}$$
$$= \frac{0.8X0.7X0.9X0.5}{P(evidence)} = \frac{0.252}{P(evidence)}$$

$$P(Argument|Leader, Sincerely, Education)$$
$$= \frac{P(Leader|Argument)P(Sincerely|Argument)P(Education|Argument)P(Argument)}{P(Leader),P(Sincerely),P(Education)}$$
$$= \frac{0.5X0.75X0.25X0.2}{P(evidence)} = \frac{0.01875}{P(evidence)}$$

From the calculations above we see that the probability, P(Essay|Leader, Sincerely, Education) is greater than the probablities, P(Letter|Leader, Sincerely, Education) and P(Argument|Leader, Sincerely, Education). Thus, since it is more probable that the new answer is an essay about their role model according to the Bayes theorem, the Naive Bayes Classifier will label it as an "Essay". Naive Bayes can be used to solve various classification problems such as spam detection, topic categorization and Sentiment Analysis.

3.1.2 Maximum Entropy

Maximum Entropy Classifier [37] is a supervised probabilistic machine learning algorithm. Unlike Naive Bayes, Maximum Entropy classifier does not assume that the features are independent of each other. In many scenarios Maximum Entropy outperforms Naive Bayes, however, not in all cases as discussed by Nigam et al. [37]. Maximum entropy classifier is used to estimate probability distribution from data. It is based on the principle of maximum entropy which states that if there are no prior knowledge about some data, then it should be uniformly or randomly distributed. For example, suppose we have student answers from an English examination that are either of the four: letter, descriptive essay, argumentative essay or a story. Now, lets say that there is a 40% chance that the answer is an "argument" if it has the word opinion. So, according to uniform distribution principle, there is 20% chance for each of the other three classes. However, if we do not know anything about the student answer, then the probability is uniformly distributed among all four classes and hence there is a 25% chance that the answer is an "argument".

This method of uniform probability distribution is applied to various text classification problems such as language identification, Sentiment Analysis and topic categorization. Higher the uniformity of data distribution, the higher is the entropy. We should maximize the entropy while being consistent with the constraints of the data and this is why it is known as conditional distribution. In case of classification using maximum entropy, first we have to select a set of features that are necessary to categorize the documents. The features are usually frequency of classifying words in the document. Then we have to calculate the expected value of each feature for the training data and thus derive the constraint for the distribution model. After the classifier has been trained on the given constraint, it can take a new document and predict the class label.

3.1.3 Support Vector Machines (SVM)

SVM is a supervised classification algorithm and thus requires training data [11]. It is a linear classifier, unlike Naive Bayes and Maximum Entropy, which are probabilistic classifiers. In order to apply SVM, we first plot all data points on an n-dimensional graph, where n is the total number of features. The support vector are the co-ordinates of each data points and the goal of SVM is to find an optimal hyperplane that separates one class from another. The separation is called a margin and should be as large as possible. Suppose we have student answers from an English exam and

the answers are either a letter or an argumentative essay and we want to classify them based on the frequency of the words opinion and however.

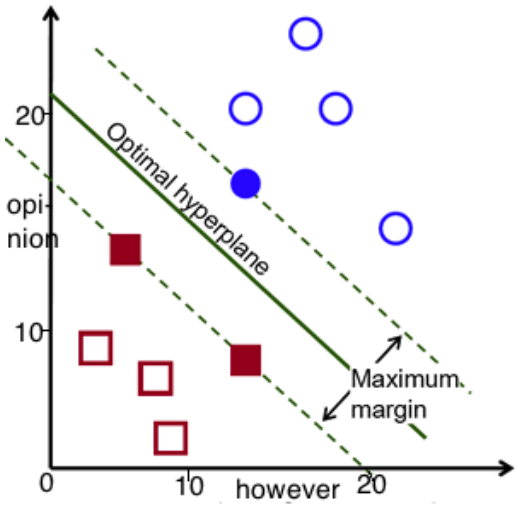


Figure 11: Classification using SVM (Source: [39])

In Figure 11, let the red squares represent letters and the blue circles represent argumentative essays. The line separating the two classes is a hyperplane. It is possible to calculate several hyperplanes that successfully separate the training data into distinct classes, however the hyperplane is optimal only when the margin between the training data is maximum. If the separation is not optimal, then classification of unseen data might be erroneous. To calculate the margin, we should find the distance between the hyperplane and closest data point and then double it as shown in Figure 11. Ideally, there should not be any data points within the margin. This requirement becomes a disadvantage when the data is noisy.

SVM can be applied to more than two dimensions or features and therefore the separating line is called a hyperplane. After the hyperplane is identified, a new answer can be labeled as “letter” or “argument” depending on which side of the hyperplane it is positioned. In cases where data points cannot be separated linearly as shown in figure 12, a new feature of higher dimensional input space is calculated. Thus, we introduce an additional feature ‘z’, $z = x^2 + y^2$ and plot a graph of ‘x’ against ‘z’. We can see in figure 13 that the classes can now be separated linearly. This is known as the “kernel” trick. SVM uses functions called kernels to automatically transform low dimensional input space to higher dimensional input space in cases where the classes cannot be linearly separated. There are many applications where SVM is used apart from Sentiment Analysis, for instance organizing reviews according to quality.

3.2 Lexicon-Based Techniques

Lexicon-based methods are widely used in Sentiment Analysis tasks. In this approach, positive and negative opinion words, phrases and idioms are used to classify sentiments.

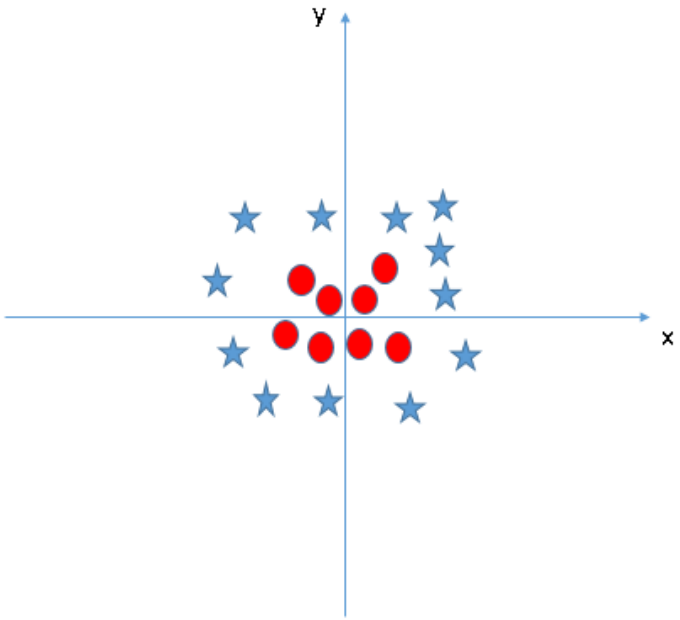


Figure 12: Non-linear Distribution of Data (Source: [45])

These opinion words, phrase and idioms together form an opinion lexicon. It is possible to manually create a lexicon. However, it is time consuming, inefficient and prone to human errors. Therefore, there are techniques to automatically create an opinion lexicon for Sentiment Analysis. The techniques can be broadly divided in two categories: Dictionary-based approach and Corpus-based approach. Sections 3.2.1 and 3.2.2 discusses the two techniques respectively. Usually, the manual approach is combined with the two automated techniques as a final correction of errors caused during the automated processes.

3.2.1 Dictionary-Based Approach

The Dictionary approach creates the opinion lexicon in an iterative process. In the first iteration a small set of opinion words with known positive or negative orientation are selected manually. Then the synonyms and antonyms for the selected words are searched and collected from a known corpora such as the thesaurus. This iteration continues until no new words are found. The selected and searched words are added to the seed list. Finally, the seed list is manually checked for any errors. The disadvantage of the dictionary-based approach is that it is not suitable to find word orientation based on context or domain.

3.2.2 Corpus-Based Approach

Unlike the dictionary-based approach, the corpus-based approach does not face the problem of finding context or domain based word orientation. In this approach,

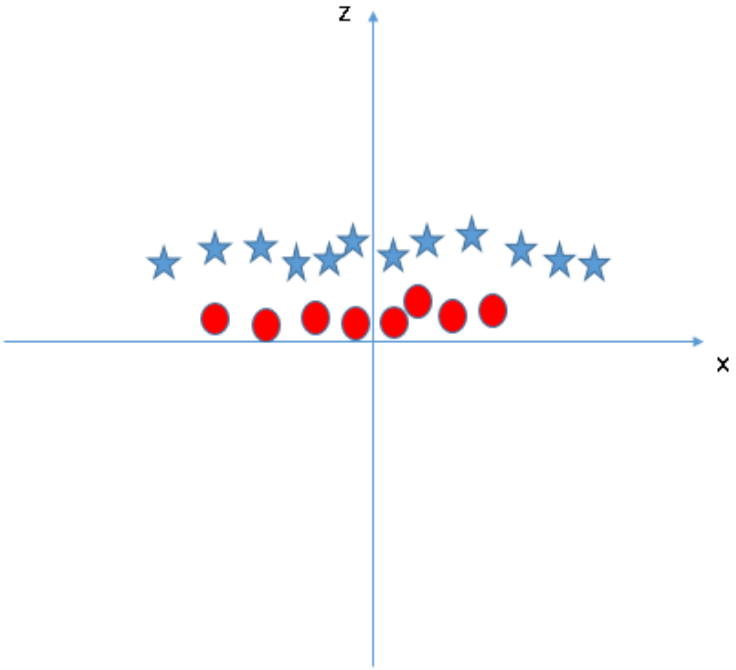


Figure 13: Convert Non-linear to Linear Distribution (Source: [45])

first a seed list of opinion words is selected. Then syntactic patterns that occur in association with the seed list of words are used to determine the orientation of other words. For instance, adjectives that appear after the conjunction "and" with a word from the seed list is considered to have the same sentiment orientation. Similarly, words such as "but" or "however" indicate a change in sentiment. A large corpus is used to learn if two conjoined words are of same or opposite sentiments. The link between the words form a graph. Clustering is applied on the graph to create a list of positive words and another list of negative words. One disadvantage of the corpus-based approach is that it requires a very large corpus for learning, which is difficult to prepare. Therefore, it may not be as effective as the dictionary-based approach if applied alone.

3.3 Other Techniques

All the ML techniques for sentiment classification discussed in Section 3.1 are supervised techniques. However, it is possible to use unsupervised ML methods for sentiment classification. Unsupervised algorithms compare the features of the target text with a word lexicon, where the polarities of the words are predetermined. The number of positive and negative words are counted. The presence of a higher number of positive words from the lexicon means the text is positive and similarly higher frequency of negative words classifies the target text as negative. Apart from the supervised techniques discuss in this Chapter, there are others such as Neural Networks and Bayesian Networks that can be used to train a sentiment classifier.

Some Sentiment Analysis research have used CRF to classify text sentiments [40]. CRF [55] is a probabilistic graphical model often used in NLP or computer vision to label sequential data. There are some classification techniques that combine several methods discussed in this chapter together to form hybrid techniques and have been shown to yield higher accuracy. There are also some methods for sentiment classification that cannot be categorized as either ML or lexicon-based technique, such as the Formal Concept Analysis (FCA) [17] or Fuzzy Formal Concept Analysis (FFCA) [69].

4 Challenges and Observations

This Chapter will discuss the challenges in Sentiment Analysis research and the observations made during the thesis. The Chapter is divided in two sections: Section 4.1 outlines the general challenges involved in Sentiment Analysis research mostly identified from the literature review and Section 4.2 gives an overview of the specific challenges and observations encountered while conducting the thesis-specific experiments.

4.1 General Challenges

Sentiment Analysis Classification has only two classes, i.e. "positive" and "negative", and sometimes a third class, "neutral". Even though Sentiment Analysis has less number of classes compared to other text classification problems, such as Topic-Based Classification, it has far greater challenges. For example, Topic-Based Classification can be done using keywords, however, this method does not yield similarly high accuracy for Sentiment Analysis as shown by Turney [58]. Over the years, many researchers have compiled lists of opinion or sentiment keywords, known as sentiment lexicons. While the sentiment lexicon is important for sentiment analysis, it is not sufficient when used alone as stated in Chapter 3. Some of the common challenges related to Sentiment Analysis are explained below:

- Some sentiment words may have opposite orientation in a specific context. For example, while the word *sad* is a negative word, it will indicate positive sentiment if used in the sentence, "The Titanic makes me *sad* every time I watch it", since the movie was successful in conveying the intended emotion. Thus, identifying the context of the text would be often crucial to accurately label it as positive or negative. This makes Sentiment Analysis quite challenging as there are no definitive way of understanding the context of a text, especially in Sentence-Level Sentiment Classification.

- Some sentences, such as interrogative sentences or conditional statements may contain sentiment words, but still may not express any sentiment at all. For example, the sentences "Is this movie *good*?" and "If the movie gets a *good* rating, I will watch it" do not convey any subjective information about any movie in spite of containing the positive sentiment word *good*. However, it is not necessary that all interrogative or conditional statements will be devoid of sentiment information. For example, the conditional statement, "If you want to watch a *good* thriller, watch The Prestige" expresses positive sentiment for the movie "The Prestige". Therefore, simply eliminating interrogative or conditional statements from consideration during Sentiment Analysis is not a viable option. Thus, it is challenging to identify a sentence as neutral when it contains either positive or negative words..

- One of the most difficult research challenge in NLP is to identify sarcastic statements. For example, the sentence "What an *amazing* lecture! I slept

through it like a baby.", conveys negative sentiment about the lecture even though it uses the positive word *amazing*. Thus, sarcasm can change the orientation of a word. Not being able to detect sarcasm in statements poses a challenge to Sentiment Analysis problems as sarcasm almost always changes the sentiment of a sentence.

- Factual or objective sentences usually do not contain any sentiment word and yet may express sentiments. For example, the statement "This phone needs to be charged several times a day", does not contain a single sentiment word, but expresses negative sentiment about the phone battery life. Thus, discarding sentences without sentiment words or labeling them as neutral could be detrimental to the accuracy of a Sentiment Analyzer.

- Another challenge in Sentiment Analysis is the presence of negation words, such as, *not*. One might assume that negation words always reverse the sentiment orientation. For example, "I do not like this movie" reverses the orientation of the word *like* from positive to negative. Thus, a lot of researches attach negation words to the sentiment words for training (e.g. not_like). However, as discussed in Section 3.1 there might be sentences where negation words do not reverse the orientation of the sentence for example, "I not only like the movie, I relate to it too.". The statement expresses positive sentiment and attaching negation word to the word *like* will falsely label the sentence as negative.

- Sentiment Analysis using social media data, such as Twitter tweets also present a challenge that is usually not found during Sentiment Analysis of formal text. Social media often contains internet jargon and short hands that change rapidly over time and cultural slang not only in different languages, but also within the English language. Social media text is also informal, therefore, there are many words which may entail different orientation depending on the age group of the author. Sentences such as "The movie was *wicked*" or "Russel Peters has a *ridiculous* sense of humor" convey positive emotions for the movie and Russel Peters, even though the words *wicked* and *ridiculous* are negative words.

4.2 Thesis-Specific Challenges

Apart from the general Sentiment Analysis challenges discussed in Section 4.1, there are many other obstacles encountered during carrying out the experiments for the thesis. Several scripts: collector, parser and trainer are mentioned below while explaining the thesis-specific challenges. These scripts are explained further in Chapter 5 and 6.

- The experiment was carried out using Aalto University computers. All students have a 10GB quota, which is enough memory for running the project. However, there is also a limit on the number of files, i.e. 200000. Since, the collection script collected and stored tweets in text files in batches, after a while, the file limit was reached even though there was still a lot of storage space left. Thus,

any other scripts could not be run on the machine. The solution was to zip the collected tweet files, delete the smaller text files and then read the zipped file for parsing.

- While running the parser script, a "Managed Memory Leak Detected" error was encountered and the script crashed. This error is a Spark bug[20]. The error is often misleading and may occur due to other reasons, such as, a task failure. Therefore, the suggested solution[21] was to change the error to a warning in the Spark source code.

- While running the trainer script, "java.lang.OutOfMemoryError: Java heap space" error was repeatedly encountered. Several solutions were suggested[22] for this issue, one of them was to increase the number of partitions to make the program more stable. Therefore, by increasing the number of partitions in the parser script the "Java Heap Space" error was solved. Even though the scripts were slower than before, it was completed successfully.

- The tweets were collected using tourism or travel specific keywords. Some of these keywords may also apply to domains that are not related to tourism. For example, the word *trip* was used in the keyword list and so the tweets: "Trip in Italy and meet with Cameron Dallas" and "Don't trip over something you can't control", were both collected for analysis even though the later is not related to tourism. Thus, special methods have to be applied to separate unrelated tweets from the ones that are related to the tourism domain.

- One of the initial thesis goal was to analyze tourism sentiment per country. However, in Twitter, not all tweets are geo-localized. Therefore, when the tweets were filtered first by presence of geo-location data and then by the tourism keywords, the collected dataset was too small for training a Sentiment Classifier. Due to this, the resulting accuracy of the classifier was less than 10% for the geo-localized dataset.

- Two datasets were collected. In both cases, the datasets were strongly imbalanced. This means, the number of positive tweets outweighed the number of negative tweets to a very high degree (greater than 3×). The imbalanced dataset might produce misleading evaluation metrics value in case of binary classification.

[20] `http://stackoverflow.com/questions/34359211/debugging-managed-memory-leak-detected-in-spark-1-6-0`

[21] `https://github.com/apache/spark/pull/11969`

[22] `http://stackoverflow.com/questions/21138751/spark-java-lang-outofmemoryerror-java-heap-space`

5 Environment and Setup

This chapter describes the environmental setup and tools used to carry out the experiments to study Sentiment Analysis. Section 5.1 describes the system requirements for carrying out the experiment; Section 5.2 describes the streaming and analysis tool and API; and finally Section 5.3 describes Tweeather- a machine learning project, used as the base to implement the experiment.

5.1 System Requirements

This section describes the hardware and software requirements for carrying out the experiments for this thesis:

- Java 1.7+ (installed version: 1.8.0_102)
- scala-sbt (installed version: 2.11.8)
- Apache Spark 1.6
- 8 to 14 Gigabyte (GB) Random Access Memory (RAM) (available RAM: 10 GB)
- Atleast "number or Twitter apps configured to collect tweets + 1" logical cores
- Apache Zeppelin 0.6.2

5.2 API and Tools

The sub-sections below provide detailed description of the data source, streaming API, analysis tool and library required for data collection and Sentiment Analysis.

5.2.1 Twitter APIs

Twitter[23] is a social networking and micro-blogging service created and founded by Jack Dorsey, Biz Stone, Evan Williams and Noah Glass in 2006. Registered users can post and view messages of a maximum of 140 characters in length called "tweets". Active users on Twitter has grown exponentially (see Figure 14) and is currently over 310 million[24]. Users on Twitter share there views, feelings and opinions about all sorts of brands, topics, places and people imaginable. They follow each other and various public figures to keep updated about current trends and products. The tweets are visible to the public by default, however, the users can restrict their tweets only to their followers. Many commercial applications and research institutes have utilized the publicly available information in the 140-character long tweets for product development and academic research.

[23] `https://twitter.com/`

[24] `https://about.twitter.com/company`

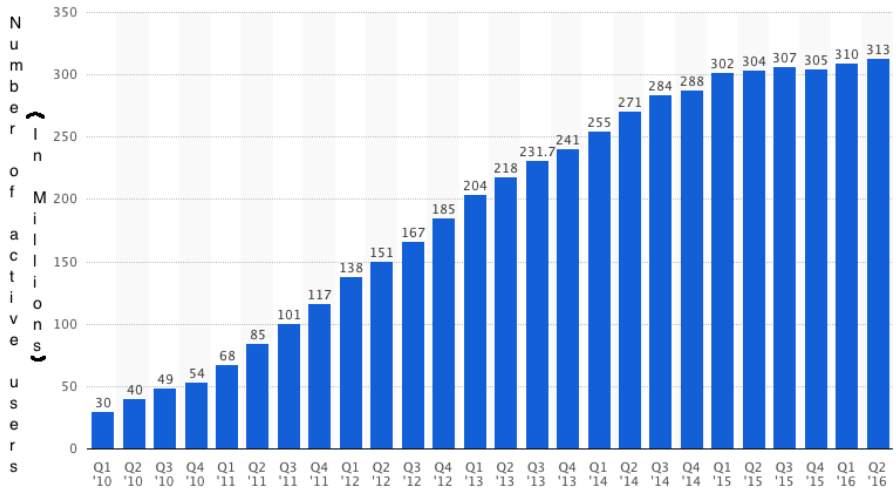


Figure 14: Number of monthly active Twitter users worldwide from 1st quarter 2010 to 2nd quarter 2016 (Source: [54])

In order to allow programmatic access to tweets for analysis, Twitter provides two different types of APIs- the REST APIs [59] and the Streaming APIs [60]. The REST APIs allow to post tweets, read specific user profile and their follower's data and make singular searches of historical tweets for upto past 7 days based on location, keywords etc. However, in order to access tweets in real-time, the Streaming API has to be used. The experiments for the thesis make use of the Streaming API and it is discussed in details below.

Twitter Streaming API

The Streaming API provides low latency access to real-time Twitter data and is the suitable API for this thesis. Unlike the REST APIs, the Streaming API requires to maintain a persistent HTTP connection. Figure 15 and Figure 16 show how the REST API and Streaming API respectively will handle a user's HTTP request differently for the same application. Thus, in case of the REST APIs, the user's HTTP request will establish a connection to Twitter's API. However, in case of the Streaming API, the HTTP process and the streaming connection process will run separately. The streaming process collects tweets in real time and then filters, parses and stores them in a data store. The HTTP process will query the data store when a user generates an HTTP request. Both the APIs use OAuth[25] to allow authorized access to users and applications.

Depending on the use case, Twitter provides three streaming endpoints for connection- Public streams, User streams and Site streams. The Public stream provides all publicly available data on Twitter. This endpoint is suitable for following specific users or topics and for data analytics. User streams provide almost all data corresponding to a

[25] `https://dev.twitter.com/oauth`

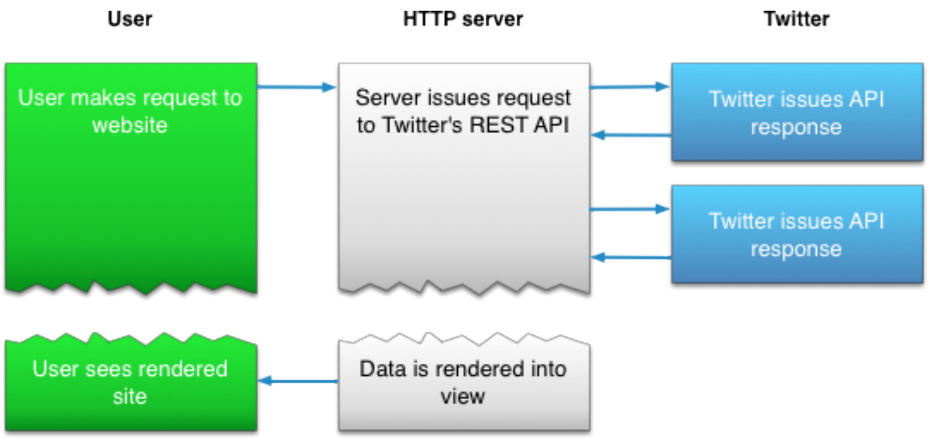


Figure 15: Twitter REST API Example (Source: [60])

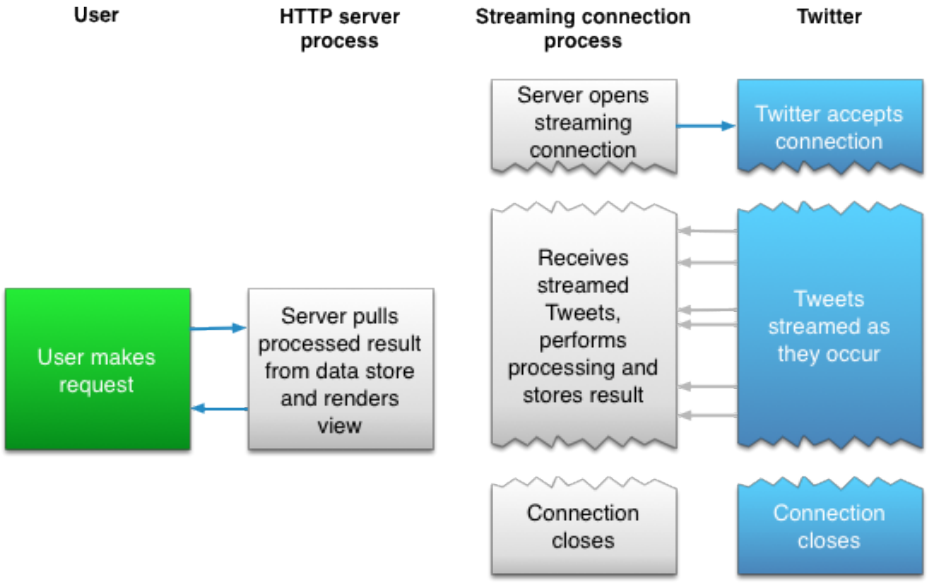


Figure 16: Twitter Streaming API Example (Source: [60])

single Twitter user. Finally, Site streams are similar to User streams, however, a server can connect to Twitter on behalf of multiple users. In order to establish a connection to the Twitter Streaming API, the application or user has to be authenticated using OAuth. The requesting application should provide four pieces of information from Twitter to access the API- API Key, API secret, Access token and Access token secret. These values can be obtained by logging into Twitter with the username and password and then creating a new application on Twitter's application settings page[26]. Once a connection is established with the API, Twitter servers will maintain

[26] `https://apps.twitter.com/`

an open connection as long as there are no server-side errors, network errors, multiple logins with same credentials or sudden drop in streaming rate.

The Streaming API currently provides 11 request parameters[27] that can be used to specify what data the API endpoints will return. Some of these parameters that are commonly used on all streaming endpoints are described below:

delimited: This parameter is set to the string "length" in order to indicate that the tweets will be delimited in the stream, so that the client knows how many bytes to read before the end of the tweet.

stall_warnings: If this parameter is set to "true" then the client will periodically receive warning messages if they run the risk of being disconnected due to slow streaming rate.

filter_level: This parameter is set to "none" by default, which means all available tweets will be displayed. The filter_level can also be set to "low" or "medium". The latter will deliver the tweets that appear as top results for searches on the Twitter website.

language: This parameter will indicate which language the returned tweets should be in. To specify the desired languages, this parameter should be set with a comma-separated list of BCP 47 [43] language identifiers.

follow: This parameter will contain a comma-separated list of user IDs indicating which user's tweet should be delivered to the stream. The delivered stream may consist of tweets created by the user, re-tweets by the user, replies to tweets created by the user and re-tweets of any tweets created by the user. It is not possible to follow protected users.

track: This parameter will contain a comma-separated list of phrases that will be used as keywords to filter the tweets to be delivered on the stream. The text of the tweet, user name mentions, text in hashtags and displayed urls will be checked for matches to the list of keywords.

locations: This parameter contains a comma-separated list of longitude, latitude pairs with a set of bounding boxes to filter tweets by geo-location. If the tweets fall within the bounding box then they will be delivered to the stream.

Other request parameters that can be used in some specific streaming endpoint or when application has elevated access are the following: "count", "with", "replies" and "stringify_friend_id". For the purpose of this thesis, only "language" and "track" parameters have been used for collecting tweets. The "language" parameter in the project code has been set to English ("en") and the "track" parameter contains phrases that are specifically used to tweet about travel and tourism. The tweets were collected using only one Twitter App and the collected tweets were saved locally in text files in batches with 5 minute intervals.

[27] `https://dev.twitter.com/streaming/overview/request-parameters`

5.2.2 Spark

Apache Spark[28] is a fast and general purpose cluster computing platform designed for large-scale data processing. Spark was developed in UC Berkley in 2009 as a research project in the AMPLab, in March 2010 it was made open source and then transferred to the Apache Software Foundation in June 2013. It is a framework trying to improve on Hadoop MapReduce and it especially aimed to address the inability of the MapReduce model to handle iterative algorithms and interactive queries efficiently [67]. Spark authors claim that it outperforms Hadoop Mapreduce by 100X in memory and 10X on disk. It is designed to process all workloads such as batch applications, streaming, interactive queries and iterative computations in the same engine. Since it does not require separate distributed systems, Spark is inexpensive and easy to manage. Apart from UC Berkley, Yahoo, Databricks and Intel are major contributors to the Spark project. Spark supports programming in Python, Java, Scala and SQL and has rich built-in libraries. It can run on Hadoop clusters, Mesos[29], standalone or in the cloud. Spark can also integrate with and access data from any other Big Data tools such as HDFS or Cassandra.

Spark Component Stack

Spark contains several closely integrated components that can interoperate among each other like libraries in a software project [27]. These different components form the Spark stack as shown in Figure 17 are briefly described below:

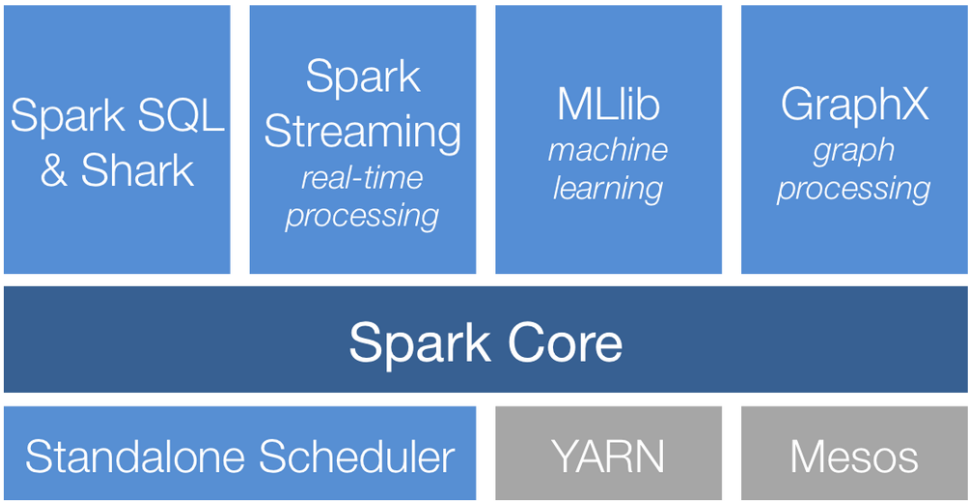


Figure 17: The Spark Stack (Source: [27])

Spark Core: All Spark platform functionality are built on top of the Spark Core, which is the basic execution engine for the platform. The Core handles in-memory

[28] `http://spark.apache.org/`
[29] `http://mesos.apache.org/`

computing capabilities, memory management, task scheduling, fault recovery and interaction with data stores. Spark's main programming abstraction and underlying data structure is the Resilient Distributed Dataset (RDD). RDD is a read-only collection of objects distributed as logical partitions across several nodes in the cluster. Spark Core provides the APIs to define, build and manipulate the RDD collections.

Spark SQL: Spark SQL is a component on top of Spark Core that provides support for structured and unstructured data. It provides an interface to interact with Spark through query languages such as SQL and HiveQL. The query languages are translated to Spark operations and the database tables are represented as RDDs.

Spark Streaming: The Spark Streaming component allows processing streams of real-time data. It provides similar fault tolerance capabilities, throughput and scalability as the Spark Core. Data can be ingested in from various different sources such as Kafka, Flume, Twitter etc, and stored in file systems, databases or dashboards as shown in Figure 18. The continuous stream of data called Discretized Stream (DStream) is divided into batches, represented internally as RDDs and then processed by the Spark engine.

Figure 18: Spark Streaming Architecture (Source: [6])

MLlib: Machine Learning Library (MLlib) is a Spark library that consists of common Machine Learning capabilities such as regression, binary classification, collaborative filtering, etc. These ML functions can scale across a cluster of computers.

GraphX: GraphX is a distributed library built on top of Spark and it extends the Spark RDD API. It provides an API for manipulating graphs and expressing graph computations and a library containing common graph algorithms such as PageRank.

Cluster Managers: The Cluster Managers allow Spark to run either on Hadoop YARN, Apache Mesos or Spark's own cluster manager called the Standalone Scheduler.

For the purpose of this thesis, the Spark Streaming [6] component was used in conjunction with the underlying Spark Core components to collect real-time tweets from Twitter. In order to take advantage of Spark streaming capabilities the "StreamingContext" library should be used. Spark Streaming provides a library called

"TwitterUtils" which was used to connect to the Twitter Streaming API and collect live tweets for the experiment. The Spark project code for the thesis was run in the Standalone mode.

5.2.3 Zeppelin

Apache Zeppelin[30] is an opensource, web-based notebook that allows interactive and collaborative data analytics and visualization for distributed, general-purpose data processing systems such as Apache Spark, Apache Flink[31], etc. The project entered the incubation stage on December 2014 and the first stable version was released on May 2016. The Apache Zeppelin Interpreter[32] system enables seamless integration with any language or data processing backend. Currently Zeppelin supports many interpreters such as Scala, Spark, Python, Apache HBase etc. The notebook provides built-in integration for Apache Spark, thus there is no need to build a separate plugin or library for it. Zeppelin provides some basic charts and tools, such as bar charts, pie charts, scatter plots, dynamic forms, etc., for data visualization and analytics using Spark SQL or other language backend queries.

5.3 Tweeather

"Tweeather[33]" [47] is a machine learning project started by Alexandru Roşianu in January 2016. He started "Tweeather", which is a combination of the word tweet and weather, to find a correlation between the sentiment expressed in tweets and the weather of the location where the tweets originated. Roşianu was inspired by a research that studies user behaviour on Twitter to build a predictive model for user income [44]. In his project Roşianu built a model that tries to predict if a certain area in Europe was happy or sad based on good or poor weather at that location. He used Apache Spark to collect and analyze data and HDFS for storing tweets. The sub-sections below give an overview of "Tweeather" implementation details and necessary configurations to run the project. The changes made to the source code of "Tweeather" for the purpose of the thesis are discussed in Chapter 6.1

5.3.1 Implementation

The "Tweeather" project consists of three sets of scripts: Sentiment140 scripts, Emo Scripts and Fire Scripts.

Sentiment140 Scripts

Sentiment140[34] is a tool to analyze the sentiment of a brand, product or topic on Twitter. This project uses a corpus of 1.6 million tweets that were collected

[30] https://zeppelin.apache.org/
[31] https://flink.apache.org/
[32] https://zeppelin.apache.org/docs/latest/manual/interpreters.html
[33] https://github.com/Aluxian/Tweeather
[34] http://www.sentiment140.com/

automatically using the Twitter Search API [61]. The project was started by three Stanford University graduate students, Alec Go, Richa Bhayani and Lei Huang after their research on Twitter sentiment classification [19]. All tweets with a positive emoticon were considered to have a positive sentiment and those with negative emoticons were considered negative. In their research Go et al. used different machine learning algorithms: Naive Bayes, Maximum Entropy and Support Vector Machine to train their dataset of tweets and found an accuracy of 80%. They implemented the Sentiment140 tool using a Maximum Entropy Classifier.

The Sentiment140 code is not open source, however, the dataset of 1.6 million tweets is freely available to download. Roşianu wrote four scripts: Sentiment140Downloader, Sentiment140Parser, Sentiment140Trainer and Sentiment140Repl to downloaded the Sentiment140 dataset, parse the dataset, train a sentiment analyzer using Naive Bayes algorithm and then test the analyzer respectively for the "Tweeather" project. The model had an accuracy of 80%.

Emo Scripts

The Emo script set consists of four scripts: TwitterEmoCollector, TwitterEmoParser, TwitterEmoTrainer and TwitterEmoRepl. The collector script used Twitter's streaming API [60] with help of the Spark streaming library "TwitterUtils[35]" to collect 100 million tweets with an average throughput of 325 tweets/second. Only English language tweets with atleast one emoticon was collected. Tweets were classified in a similar manner as the Sentiment140 project- tweets with positive emoticons was classified as positive, those with negative emoticons as negative and tweets with both types of emoticons were ignored. Roşianu configured two Twitter apps to collect the tweets. Due to multiple apps and re-tweets there were many duplicate tweets. So, after removing the duplicate tweets, there were 8.4 million tweeets, 90% of which were used for training a Naive Bayes Classifier. Finally, the remaining 10% of the collected tweets were used for testing the model resulting in an accuracy of 75%.

Fire Scripts

The Fire scripts were used to train an Artificial Neural Network (NN) [63] that will predict the sentiment of tweets given the weather at the location of the tweets. The collector script, "TwitterFireCollector.scala", collected tweets filtered by location and language, which are Europe and English respectively. The sentiment analyzer retrieved the polarity of the collected tweets. Then the parser script, "TwitterFireParser.scala", extracted three weather features: temperature, pressure and humidity, from the Global Forecast System (GFS) model[36] provided by National Oceanic and Atmospheric Administration (NOAA). In the "TwitterFireTrainer.scala" script, the Neural Network was trained using the three weather features as input and

[35] `https://spark.apache.org/docs/1.6.0/api/java/org/apache/spark/streaming/twitter/TwitterUtils.html`

[36] `https://www.ncdc.noaa.gov/data-access/model-data/model-datasets/global-forcast-system-gfs`

the predicted polarity as output. However, the prediction was not accurate and no correlation between weather conditions and Twitter sentiment was established in the project during the time of this thesis.

5.3.2 Configuration

In order to run the “Tweeather” project first two configuration files need to be copied and configured- twitter.properties and log4j.properties. The twitter.properties should contain the Twitter app credentials and log4j.properties is used for logging. The environment variable “TW_SPARK_MASTER” can be used to link to a Spark master url. If no url is provided, then the scripts will run locally. There are two system properties supported by the project- tw.streaming.timeout and tw.streaming.interval. The first property indicates the time period in seconds for the termination of streaming and is set to unlimited by default. The second property sets the value for the duration in seconds for each batch of streaming data. In the project the default value of ”tw.streaming.interval” is set to 5 minutes. The recommended RAM of the Spark cluster for running the project is 14 GB. If the Spark cluster does not have at least 14 GB, then the value of the variable “executorHighMem” in “SparkSubmit.scala” has to be changed.

6 Experiments and Results

In order to collect the data and carryout the experiments, the Emo scripts described in Section 5.3 were modified and also, a new set of scripts, the EmoCountry scripts, were added to the "Tweeather" project. The EmoCountry scripts consist of three scripts: the collector script, "TwitterEmoCountryCollector.scala" collects and stores data from Twitter, the parser script, "TwitterEmoCountryParser.scala" processes the collected data before training, the trainer script and finally, "TwitterEmoCountryTrainer.scala" trains a Naive Bayes Classifier for Sentiment Analysis and tests the accuracy of the model. Apache Zeppelin was used to analyze and visualize the data processed and labeled by the EmoCountry parser script.

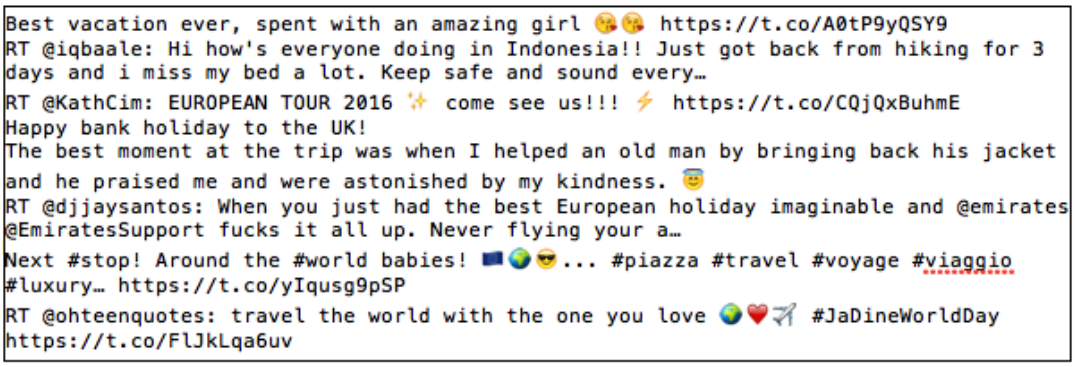
This Chapter gives a detailed explanation of the experimental procedure and results. Section 6.1 describes what kind of data was collected and how it was collected, then Section 6.2 explains how the data was processed and filtered before carrying out the experiment, Section 6.3 explains how the training classifier was implemented and executed and finally, Section 6.4 outlines and evaluates the results obtained during the experiment.

6.1 Data Collection

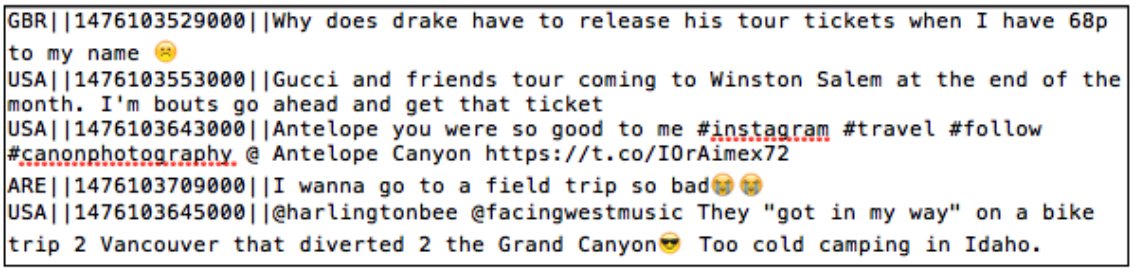
This thesis focuses on Sentiment Analysis in the context of tourism. Twitter contains a myriad of opinionated tweets from a large and diverse user space on almost every topic imaginable. The main goal for using Twitter as the data source in this thesis was to identify if Twitter data can be used to develop a tourism recommendation system. Therefore, the collected tweets were filtered according to a list of keywords (see Appendix B) that are commonly used words, phrases or hashtags in social media about tourism and travel. The list was compiled from personal experience and blog posts [33], [1]. Since the thesis focuses on single language Sentiment Analysis, only English language tweets were collected. For the purpose of the thesis, two sets of data were collected, filtered by the tourism related keywords and the English language using the Twitter Streaming API.

Both the datasets were collected in batches with a 5 minute interval and stored locally in text files. The first and second dataset will be referred to as dataset-1 and dataset-2 respectively in the rest of the thesis. Dataset-1 was collected over a period of 10 days using the "TwitterEmoCollector.scala" script and contained 7057799 (approximately 7 million) tweets. Dataset-2 was not only filtered by keywords and language, but was also geo-localized. The "TwitterEmoCountryCollector.scala" script was used for collecting dataset-2. Over a period of two weeks, 173740 tweets along with their country codes were collected from Twitter. Since, not all tweets on Twitter are geo-localized, dataset-2 was immensely smaller than dataset-1, even though it was collected over a longer time period. Dataset-1 aimed to address the first research goal, i.e. studying Sentiment Analysis in the context of tourism. Meanwhile, dataset-2 was used to answer the second research question, i.e. whether Twitter can be used as a potential data source for tourism recommendation in different countries, specifically

Bangladesh? Figure 19a and 19b show some examples from dataset-1 and dataset-2 respectively.



(a) Dataset-1 (without country code)



(b) Dataset-2 (with country code and timestamp)

Figure 19: Data Collected using Twitter Streaming API

6.2 Data Pre-Processing

After the datasets were collected, the data was labeled for training using the parser scripts: "TwitterEmoParser.scala" and 'TwitterEmoCountryParser.scala" for dataset-1 and dataset-2 respectively. In the parser scripts, first, all re-tweets were removed removed. Then, tweets with positive emoticons were labeled as positive (label = 1) and those with negative emoticons were labeled as negative (label = 0). All tweets with no emoticons or both positive and negative emoticons were discarded. A list of commonly used positive and negative emoticons in the context of tourism and travel was used to label the tweets. The list (see Appendix C) was compiled from a database[37] of emoticon unicodes used in different social media and digital platforms.

After parsing and labeling dataset-1, 324330 tweets (257704 positive and 66626 negative) were remaining for the Sentiment Analysis from the 7 million tweets. The raw tweets and their respective labels were stored as a Parquet file. For dataset-2 (approximately 174000 tweets), the unlabeled tweets were not discarded and the raw tweets, labels, timestamp and respective country codes of the tweet location was

[37] http://unicode.org/emoji/charts/full-emoji-list.html

stored in a Parquet file. For dataset-2 24233 tweets were labeled as positive and negative.

6.3 Training

The "TwitterEmoTrainer.scala" script trained a Naive Bayes Classifier on dataset-1. For training the classifier, the automatically labeled tweets were used. The dataset was highly imbalanced, i.e. the number of positive tweets (257704 tweets) outweighs the number of negative tweets (66626 tweets) by a very large degree (approximately 3×). Randomly, 90% of the labeled tweets were used for training and the remaining 10% were used for testing the accuracy of the trained model. Before the model was created, the tweet text was sanitized and stop words were removed. Then the Naive Bayes classifier was trained using term frequency of unigrams. The results obtained after running the trainer script is discussed in Section 6.4.

6.4 Results and Evaluation

This section presents and evaluates the results obtained from the experiment discussed above. Sub-sections 6.4.1 and 6.4.2 evaluate the results obtained after the experiments on dataset-1 and dataset-2 respectively.

6.4.1 Dataset-1 Experiment

Dataset-1 was used to address the first research goal, i.e. to study sentiment analysis in the context of tourism. For this purpose a Naive Bayes Classifier was trained to classify tourism related tweets into positive or negative sentiment classes. The performance of the classifier was tested on randomly selected 10% of the collected data using different evaluation metrics, such as, accuracy, area under the Receiver Operating Characteristic (ROC) curve and area under the Precision-Recall (PR) curve. In order to calculate the evaluation metrics, the value of the four possible outcomes (see Figure 20) of a binary classifier are required:

True Positive (TP): Both the label and the prediction are positive.

True Negative (TN): Both the label and the prediction are negative.

False Positive (FP): The label is negative, but the prediction is positive.

False Negative (FN): The label is positive, but the prediction is negative.

The accuracy of the test dataset was found to be approximately 86.5%. A high accuracy value was achieved using only unigrams and their term frequencies to train the classifier. There are many other features discussed in Chapter 3, such as using bigrams or trigrams, attaching negation word, POS tagging and applying domain-specific opinion rules, that can increase the dataset accuracy. The accuracy was calculated using the following equation:

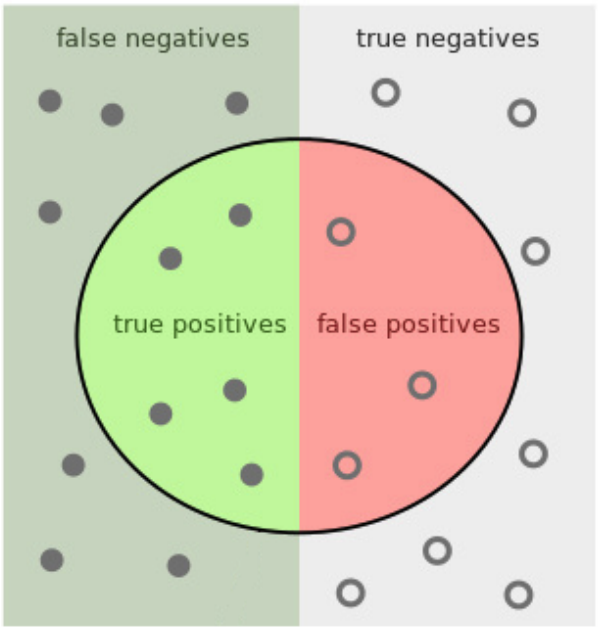


Figure 20: Possible Outcomes From A Binary Classifier (Source: `https://en.wikipedia.org/wiki/Precision_and_recall`)

$$accuracy = \frac{matches}{predicted} \times 100\%,$$

where,
$matches = \sum TP + \sum TN$ (total number of sentiment labels that matched the predicted value) and,
$predicted = \sum TP + \sum TN + \sum FP + \sum FN$ (total number of predicted sentiments used for testing the model).

Apart from the dataset accuracy, two commonly used metrics for evaluating the performance of binary classifiers, i.e. the area under the ROC curve and the area under the PR curve, were also used. If the area under the ROC or PR curve is 1 then it indicates perfect classification and if it is 0.5 then it indicates random classification. The ROC curve is a graphical plot of the True Positive Rate (TPR) against the False Positive Rate (FPR). The TPR is the fraction of correct positive predictions out of all actual positive labels and is also known as sensitivity, recall or probability of detection. The FPR is the fraction of incorrect positive predictions out of all actual negative labels and is also known as fall-out or the probability of false alarm. The TPR and FPR can be calculated using the following equations:

$$TPR = \frac{\sum TP}{\sum TP + \sum FN}, \quad FPR = \frac{\sum FP}{\sum TN + \sum FP}$$

In case of the trained Sentiment Classifier for this thesis, the area under the ROC curve

was found to be approximately 0.64 (i.e. 64%), which indicates poor performance. The PR curve is a plot of precision against recall. As mentioned above, recall is also the true positive rate and can be calculated using the TPR equation. Precision shows the fraction of correct positive predictions out of all positive predictions. It is also known as Positive Predictive Value (PPV) and can be calculated using the equation below:

$$PPV = \frac{\sum TP}{\sum TP + \sum FP}$$

The area under the PR curve for the Sentiment Classifier was found to be approximately 0.929 (i.e. approx. 93%). While the area under the ROC curve indicates poor performance of the Sentiment Classifier, the values of the dataset accuracy and the area under the PR curve indicate otherwise. This may be due to the reason that the dataset is strongly imbalanced. The area under the ROC curve may prove to be misleading for highly imbalanced datasets and therefore PR curves are considered a more reliable measure of classification performance in case of imbalanced datasets [48]. Thus, according to the accuracy value and area under the PR curve, the Naive Bayes Sentiment Classifier shows very good performance.

6.4.2 Dataset-2 Experiment

The second research objective of this thesis was to identify if Twitter could be a possible data source for sentiment-based tourism recommendation for different countries, especially Bangladesh. The initial objective for collecting dataset-2 was to analyze tourism sentiment per country, however due to a very small dataset, i.e. only 173740 tweets in total. Sentiment Analysis for dataset-2 was not possible as discussed in Chapter 4. Therefore, dataset-2 was used to find statistical information that will help address the second research goal.

Among the dataset, 19675 tweets were positive, 4558 were negative and the remaining had no emoticons and therefore couldn't be labeled. Figure 21 shows that positive tweets in dataset-2 are almost 4× that of negative tweets. This imbalance can be seen in dataset-1 as well and can be due to the "Positivity Effect" or "Pollyanna Principle" [31]. This principle is a psychological phenomena that states that human beings are biased towards remembering and sharing positive experiences. This bias is considered to be prevalent in social media interactions as well except for a few exceptional topics, such as, politics or crime. It is more likely that the "Pollyanna Principle" will be applicable in the context of tourism. Also for individual countries, the number of positive tweets outweighed that of negative tweets.

Figure 22 shows the top 5 countries that had the highest number of tweets during the two week collection period (for a complete list with ISO Alpha-3 country code[38] mappings and number of tweets per country see Appendix D). USA had the highest number of tweets i.e. 102004 in total, out of which only 9257 were positive and 2463

[38] `http://www.nationsonline.org/oneworld/country_code_list.htm`

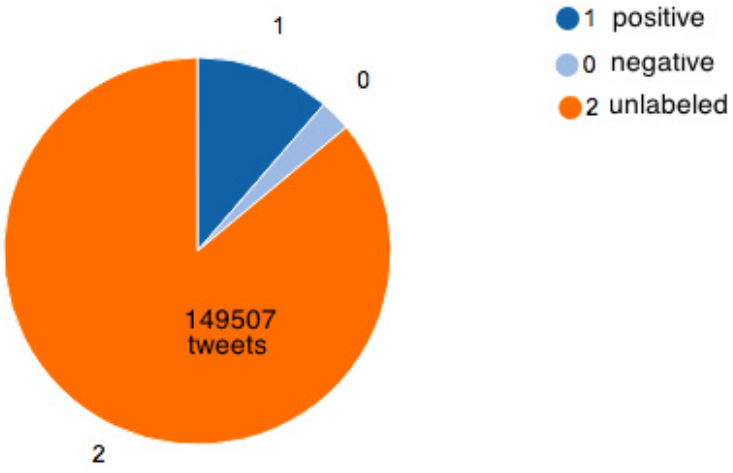


Figure 21: Proportion of positive, negative and unlabeled tweets in Dataset-2

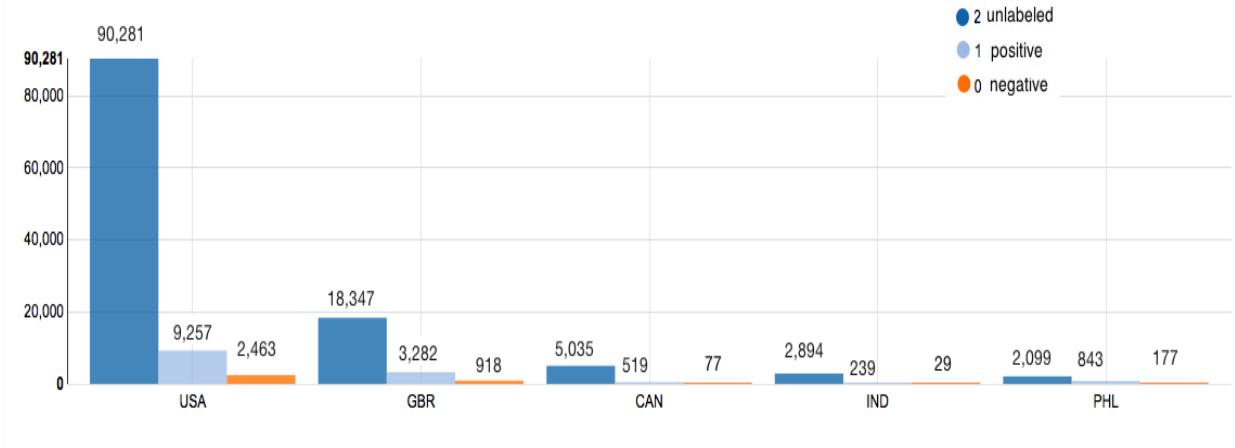


Figure 22: Countries with highest number of Tweets about Tourism

negative. Thus, only 11720 tweets were available for training a Sentiment Classifier for USA. The dataset was too small for Sentiment Analysis even for the country with the highest number of tweets. The second highest number of tweets was 22547 for the UK (country code: GBR) and this was considerably lower than that of USA.

In case of countries where Twitter is generally popular such as the USA, it might be feasible to collect tourism data over a longer period for successful sentiment based tourism recommendations. This begs the question of whether Twitter could also be a source for tourism data for countries where it is not popular in general such as Bangladesh if the data was collected over a longer period. Figure 23 shows the number of positive, negative and unlabeled tweets for Bangladesh and some neighboring countries. Only Indonesia (country code: IDN) had a little over 2000

tweets and Malaysia (country code: MYS) had over 1500 tweets, in spite of the fact that these countries are popular tourist destination in Asia. In case of Bangladesh (country code: BGD), there were only 62 tweets in two weeks, out of which 11 were positive and 1 was negative. These small amount of tourism related tweets are for the entire country and not even specific local destinations. Thus, while Twitter might provide enough data for recommendations for countries like USA, UK or Canada if the data is collected over months or years, it is unlikely to yield high enough data for Bangladesh and her neighboring countries. However, India (country code: IND) seems to be an exception and had the fourth highest number of tweets in the dataset.

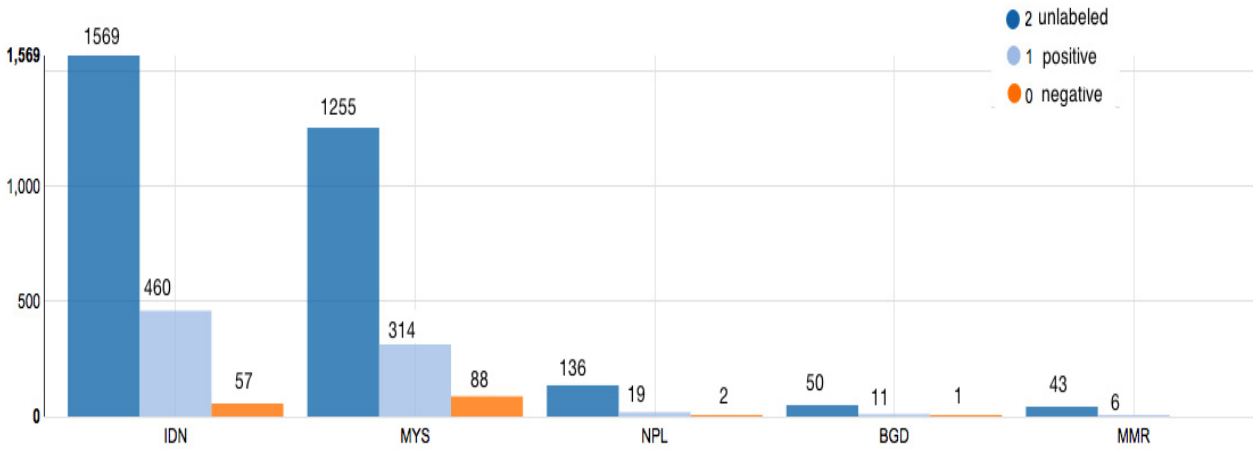


Figure 23: Number of Tweets for Bangladesh and Some Neighboring Countries

7 Conclusion

This Chapter summarizes the findings and challenges of the thesis in Section 7.1 and provides a plan for improvement and future course of this project in Section 7.2.

7.1 Discussion

Tourism is one of the fastest growing economic sectors and is the highest source of income for many countries. Just like many other domains, tourism can also benefit from the growing amount of tourism and travel related data online. Social media can be a valuable source of tourism related data as people often share their journeys, experiences and travel photographs online. Social networking platform such as Facebook, Twitter and Instagram are rife with people's opinions, reviews and pictures about tourist destinations. These opinions can be utilized to provide sentiment based recommendations for different destinations. This thesis was conducted to assess the feasibility and facilitate the business development of such a recommender system, called "JatraLog". The goal of this thesis can be divided into two parts: first, the study of Sentiment Analysis in the context of tourism and second, the evaluation of Twitter as a source for tourism recommendation for different countries, specifically Bangladesh.

The thesis was concluded in two steps: firstly, a thorough literature review was conducted to get familiarized with different Big Data technologies, Sentiment Analysis research and Sentiment Classification techniques, and secondly, experiments were conducted on Twitter data to address the aforementioned research goals. Accordingly, the thesis report can also be roughly divided in two parts: theoretical (Chapters 2, 3 and Section 4.1) and experimental (Section 4.2 and Chapters 5, 6). In the theoretical part of the thesis, first, different Big Data tools were described to establish the feasibility of the experiments. Then, different research and application domains of Sentiment Analysis was discussed to identify the scope of the research. Also, the historical evolution and current research in the field of Sentiment Analysis was studied to identify the progress and common challenges in the field. Finally, different Machine Learning and Lexicon-Based text classification techniques that are commonly used for Sentiment Classification was discussed in order to select a method for the experiment.

In the experimental part of the thesis, first, two sets of data, i.e. dataset-1 and dataset-2, was collected from Twitter using Spark and the Twitter API. Both the datasets were filtered using tourism related keywords and contained only English language tweets. Dataset-2 also contained geo-location information for each tweet. Even though dataset-2 was collected over a longer time period than dataset-1, it was smaller due being filtered by presence of geo-location data and then by keywords. Dataset-1 was used to experiment Sentiment Analysis on tourism related Twitter Data, while Dataset-2 was used to obtain statistical information in order to assess Twitter as a useful data source for recommendations. Both the datasets were strongly imbalanced, i.e. the positive tweets were far higher in number than negative tweets.

This might be due to the "Pollyanna Principle" which states that human beings are biased towards sharing positive experiences.

Dataset-1 was used to train a Naive Bayes Sentiment classifier using term frequency of unigrams. The dataset accuracy value (86.5%) and the area under the PR curve (93%) indicate a well performing classifier. Meanwhile, the area under the ROC curve (64%) indicates poor performance by the classifier. However, the area under the ROC curve might be misleading in case of imbalanced datasets and therefore in such scenarios the area under the PR curve is considered to be a better measure of performance. In case of dataset-2, after analyzing the collected data, it was found that Twitter might be used for tourism recommendation for some countries such as USA, if the data was collected over a longer time period. However, from the collected data, it can be concluded that at the moment, Twitter cannot be used as a source for tourism recommendation for Bangladesh or most other neighboring countries. Thus, Twitter cannot be utilized in the development of "JatraLog" which aims to provide recommendation for local tourism in Bangladesh.

7.2 Future Work

In this thesis, Sentiment Analysis was applied only using one feature, i.e. term frequencies of unigrams. In order to improve the accuracy of the Sentiment Classifier, term frequency of n-grams, POS tags and negation attachment can be utilized. Also, it is necessary to identify a method to separate data that do not belong to the tourism domain in spite of containing the selected keywords. In case of countries such as USA, it would be possible to collect large amount of tourism related data over a long period of time for providing recommendations. Sentiment based recommendation using Twitter might be feasible for some countries, but it is not for others such as Bangladesh. Therefore, this thesis leaves the scope for identifying other publicly available data source for providing sentiment based recommendation for "JatraLog": a tourism recommendation system for Bangladesh.

References

[1] Hashtags for tourism in Instagram, Twitter, Facebook, Tumblr, ello. `https://top-hashtags.com/hashtag/tourism/`. Accessed: 28/10/2016.

[2] Relational Database Design with ERD. `https://www.visual-paradigm.com/tutorials/databasedesign.jsp`, July 2011. Accessed: 26/07/2016.

[3] Hadoop Architecture and Deployment. `http://www.rosebt.com/blog/hadooparchitecture-and-deployment`, November 2012. Accessed: 26/07/2016.

[4] Hadoop Components and Architecture: Big Data and Hadoop Training. `https://www.dezyre.com/article/hadoop-components-and-architecture-big-data-and-hadoop-training/114`, June 2015. Accessed: 26/07/2016.

[5] L. A. Adamic, R. A. Baeza-Yates, and S. Counts, editors. *Proceedings of the Fifth International Conference on Weblogs and Social Media, Barcelona, Catalonia, Spain, July 17-21, 2011*. The AAAI Press, 2011.

[6] Apache Spark. Spark Streaming Programming Guide. `https://spark.apache.org/docs/1.2.0/streaming-programming-guide.html`. Accessed: 11/09/2016.

[7] S. Asur and B. A. Huberman. Predicting the future with social media. In J. X. Huang, I. King, V. V. Raghavan, and S. Rueger, editors, *2010 IEEE/WIC/ACM International Conference on Web Intelligence, WI 2010, Toronto, Canada, August 31 - September 3, 2010, Main Conference Proceedings*, pages 492–499. IEEE Computer Society, 2010.

[8] T. Bayes and R. Price. An essay towards solving a problem in the doctrine of chances. *Philosophical Transactions (1683-1775)*, pages 370–418, 1763.

[9] J. Bollen, H. Mao, and X. Zeng. Twitter mood predicts the stock market. *J. Comput. Science*, 2(1):1–8, 2011.

[10] W. W. Cohen and S. Gosling, editors. *Proceedings of the Fourth International Conference on Weblogs and Social Media, ICWSM 2010, Washington, DC, USA, May 23-26, 2010*. The AAAI Press, 2010.

[11] N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge University Press, 2010.

[12] S. Das and M. Chen. Yahoo! for Amazon: Extracting market sentiment from stock message boards. In *Proceedings of the Asia Pacific finance association annual conference (APFA)*, volume 35, page 43. Bangkok, Thailand, 2001.

[13] K. Dave, S. Lawrence, and D. M. Pennock. Mining the peanut gallery: Opinion Extraction and Semantic Classification of product reviews. In G. Hencsey, B. White, Y. R. Chen, L. Kovács, and S. Lawrence, editors, *Proceedings of the Twelfth International World Wide Web Conference, WWW 2003, Budapest, Hungary, May 20-24, 2003*, pages 519–528. ACM, 2003.

[14] J. Dean and S. Ghemawat. MapReduce: Simplified data processing on large clusters. *Commun. ACM*, 51(1):107–113, 2008.

[15] L. Dini and G. Mazzini. Opinion Classification through Information Extraction. In *Proceedings of the Conference on Data Mining Methods and Databases for Engineering, Finance and Other Fields (Data Mining)*, pages 299–310, 2002.

[16] E. Dumbill. Big Data Now: 2012 Edition. O'Reilly Media Inc., 2012.

[17] B. Ganter and R. Wille. *Formal Concept Analysis - Mathematical Foundations*. Springer, 1999.

[18] S. Ghemawat, H. Gobioff, and S. Leung. The Google File System. In M. L. Scott and L. L. Peterson, editors, *Proceedings of the 19th ACM Symposium on Operating Systems Principles 2003, SOSP 2003, Bolton Landing, NY, USA, October 19-22, 2003*, pages 29–43. ACM, 2003.

[19] A. Go, R. Bhayani, and L. Huang. Twitter Sentiment Classification using Distant Supervision. *CS224N Project Report, Stanford*, 1:12, 2009.

[20] G. Groh and J. Hauffa. Characterizing social relations via NLP-Based Sentiment Analysis. In Adamic et al. [5].

[21] V. Hatzivassiloglou and K. McKeown. Predicting the Semantic Orientation of Adjectives. In P. R. Cohen and W. Wahlster, editors, *35th Annual Meeting of the Association for Computational Linguistics and 8th Conference of the European Chapter of the Association for Computational Linguistics, Proceedings of the Conference, 7-12 July 1997, Universidad Nacional de Educación a Distancia (UNED), Madrid, Spain.*, pages 174–181. Morgan Kaufmann Publishers / ACL, 1997.

[22] V. Hatzivassiloglou and J. Wiebe. Effects of adjective orientation and gradability on sentence subjectivity. In *COLING 2000, 18th International Conference on Computational Linguistics, Proceedings of the Conference, 2 Volumes, July 31 - August 4, 2000, Universität des Saarlandes, Saarbrücken, Germany*, pages 299–305. Morgan Kaufmann, 2000.

[23] Y. Hong and S. Skiena. The wisdom of bookies? Sentiment Analysis versus. the NFL point spread. In Cohen and Gosling [10].

[24] M. Hu and B. Liu. Mining and summarizing customer reviews. In W. Kim, R. Kohavi, J. Gehrke, and W. DuMouchel, editors, *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data*

Mining, Seattle, Washington, USA, August 22-25, 2004, pages 168–177. ACM, 2004.

[25] L. Jiang, M. Yu, M. Zhou, X. Liu, and T. Zhao. Target-dependent Twitter Sentiment Classification. In D. Lin, Y. Matsumoto, and R. Mihalcea, editors, *The 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference, 19-24 June, 2011, Portland, Oregon, USA*, pages 151–160. The Association for Computer Linguistics, 2011.

[26] M. Joshi, D. Das, K. Gimpel, and N. A. Smith. Movie reviews and revenues: An experiment in Text Regression. In *Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, June 2-4, 2010, Los Angeles, California, USA*, pages 293–296. The Association for Computational Linguistics, 2010.

[27] H. Karau, A. Konwinski, P. Wendell, and M. Zaharia. *Learning Spark: Lightning-Fast Big Data Analytics*. O'Reilly Media, Inc., 1st edition, 2015.

[28] E. Kouloumpis, T. Wilson, and J. D. Moore. Twitter Sentiment Analysis: The good the bad and the OMG! In Adamic et al. [5].

[29] B. Liu. *Sentiment Analysis and Opinion Mining*. Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers, 2012.

[30] Y. Liu, X. Huang, A. An, and X. Yu. ARSA: A sentiment-aware model for predicting sales performance using blogs. In W. Kraaij, A. P. de Vries, C. L. A. Clarke, N. Fuhr, and N. Kando, editors, *SIGIR 2007: Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Amsterdam, The Netherlands, July 23-27, 2007*, pages 607–614. ACM, 2007.

[31] M. Matlin and D. Stang. The Pollyanna Principle. *Cambridge, MassachUsetts*, 1978.

[32] W. Medhat, A. Hassan, and H. Korashy. Sentiment Analysis algorithms and applications: A survey. *Ain Shams Engineering Journal*, 5(4):1093–1113, 2014.

[33] Michael Collins. Travel hashtags on Twitter. `https://www.linkedin.com/pulse/what-most-popular-widely-used-travel-hashtags-twitter-michael-collins`. Published on LinkedIn. Accessed: 28/10/2016.

[34] S. Mohammad. From once upon a time to happily ever after: Tracking emotions in novels and fairy tales. *CoRR*, abs/1309.5909, 2013.

[35] S. M. Mohammad and T. Yang. Tracking sentiment in mail: How genders differ on emotional axes. *CoRR*, abs/1309.6347, 2013.

[36] S. Morinaga, K. Yamanishi, K. Tateishi, and T. Fukushima. Mining product reputations on the Web. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 341–349. ACM, 2002.

[37] K. Nigam, J. Lafferty, and A. McCallum. Using Maximum Entropy for Text Classification. In *IJCAI-99 workshop on machine learning for information filtering*, volume 1, pages 61–67, 1999.

[38] B. O'Connor, R. Balasubramanyan, B. R. Routledge, and N. A. Smith. From Tweets to Polls: Linking text sentiment to public opinion time series. In Cohen and Gosling [10].

[39] OpenCV. *Introduction to Support Vector Machines*. Published in OpenCV documentation.

[40] A. Pak and P. Paroubek. Twitter as a corpus for Sentiment Analysis and Opinion Mining. In N. Calzolari, K. Choukri, B. Maegaard, J. Mariani, J. Odijk, S. Piperidis, M. Rosner, and D. Tapias, editors, *Proceedings of the International Conference on Language Resources and Evaluation, LREC 2010, 17-23 May 2010, Valletta, Malta*. European Language Resources Association, 2010.

[41] B. Pang, L. Lee, and S. Vaithyanathan. Thumbs up? Sentiment Classification using Machine Learning Techniques. *CoRR*, cs.CL/0205070, 2002.

[42] J. W. Pennebaker, C. K. Chung, M. Ireland, A. Gonzales, and R. J. Booth. *The Development and Psychometric Properties of LIWC2007*. The University of Texas at Austin and The University of Auckland, New Zealand, 2009.

[43] A. Phillips and M. Davis. *Tags for Identifying Languages*. IETF, September 2009. `https://tools.ietf.org/html/bcp47`. Accessed: 14/09/2016.

[44] D. Preoţiuc-Pietro, S. Volkova, V. Lampos, Y. Bachrach, and N. Aletras. Studying user income through language, behaviour and affect in social media. *PloS one*, 10(9):e0138717, 2015.

[45] S. Ray. Understanding Support Vector Machine algorithm from examples (along with code). `https://www.analyticsvidhya.com/blog/2015/10/understaing-support-vector-machine-example-code/`, October 2015.

[46] I. Rish. An empirical study of the Naive Bayes Classifier. In *IJCAI 2001 workshop on empirical methods in artificial intelligence*, volume 3, pages 41–46. IBM New York, 2001.

[47] A. Rosianu. Tweeather – What, Why and How? `https://blog.aluxian.com/tweeather-what-why-and-how/`, January 2016. Accessed: 23/08/2016.

[48] T. Saito and M. Rehmsmeier. The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets. *PloS one*, 10(3):e0118432, 2015.

[49] P. Sakunkoo and N. Sakunkoo. Analysis of social influence in online book reviews. In E. Adar, M. Hurst, T. Finin, N. S. Glance, N. Nicolov, and B. L. Tseng, editors, *Proceedings of the Third International Conference on Weblogs and Social Media, ICWSM 2009, San Jose, California, USA, May 17-20, 2009*. The AAAI Press, 2009.

[50] R. D. Schneider. *Hadoop for Dummies*. John Wiley & Sons Canada, Ltd., special edition, 2012.

[51] K. Shimada, S. Inoue, H. Maeda, and T. Endo. Analyzing tourism information on Twitter for a local city. In *Software and Network Engineering (SSNE), 2011 First ACIS International Symposium on*, pages 61–66. IEEE, 2011.

[52] K. Shvachko, H. Kuang, S. Radia, and R. Chansler. The Hadoop Distributed File System. In *Proceedings of the 2010 IEEE 26th Symposium on Mass Storage Systems and Technologies (MSST)*, MSST '10, pages 1–10, Washington, DC, USA, 2010. IEEE Computer Society.

[53] A. Silberschatz, H. F. Korth, S. Sudarshan, et al. *Database System Concepts*, volume 4. McGraw-Hill New York, 1997.

[54] Statista. Number of monthly active Twitter users worldwide from 1st quarter 2010 to 2nd quarter 2016 (in millions). `http://www.statista.com/statistics/282087/number-of-monthly-active-twitter-users/`. Accessed: 13/09/2016.

[55] C. A. Sutton and A. McCallum. An Introduction to Conditional Random Fields. *Foundations and Trends in Machine Learning*, 4(4):267–373, 2012.

[56] A. Tumasjan, T. O. Sprenger, P. G. Sandner, and I. M. Welpe. Predicting Elections with Twitter: What 140 Characters Reveal about Political Sentiment. In Cohen and Gosling [10].

[57] V. Turner, J. F. Gantz, D. Reinsel, and S. Minton. The Digital Universe of Opportunities: Rich Data and the Increasing Value of the Internet of Things. Technical report, International Data Corporation (IDC), April 2014.

[58] P. D. Turney. Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, July 6-12, 2002, Philadelphia, PA, USA.*, pages 417–424. ACL, 2002.

[59] Twitter Inc. REST APIs. `https://dev.twitter.com/rest/public`. Accessed: 14/09/2016.

[60] Twitter Inc. The Streaming APIs. `https://dev.twitter.com/streaming/overview`. Accessed: 24/08/2016.

[61] Twitter Inc. Twitter Search API. `https://dev.twitter.com/rest/public/search`. Accessed: 24/08/2016.

[62] G. Vaish. *Getting Started with NoSQL*. Packt Publishing, 2013.

[63] S.-C. Wang. *Artificial Neural Network*, pages 81–100. Springer US, Boston, MA, 2003.

[64] T. White. *Hadoop: The Definitive Guide*. O'Reilly Media, 3rd edition, 2012.

[65] J. Wiebe. Learning Subjective Adjectives from Corpora. In H. A. Kautz and B. W. Porter, editors, *Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on on Innovative Applications of Artificial Intelligence, July 30 - August 3, 2000, Austin, Texas, USA.*, pages 735–740. AAAI Press / The MIT Press, 2000.

[66] T. Wilson, J. Wiebe, and P. Hoffmann. Recognizing Contextual Polarity in Phrase-Level Sentiment Analysis. In *HLT/EMNLP 2005, Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference, 6-8 October 2005, Vancouver, British Columbia, Canada*. The Association for Computational Linguistics, 2005.

[67] M. Zaharia, M. Chowdhury, M. J. Franklin, S. Shenker, and I. Stoica. Spark: Cluster Computing with Working Sets. In E. M. Nahum and D. Xu, editors, *2nd USENIX Workshop on Hot Topics in Cloud Computing, HotCloud'10, Boston, MA, USA, June 22, 2010*. USENIX Association, 2010.

[68] W. Zhang and S. Skiena. Trading strategies to exploit blog and news sentiment. In Cohen and Gosling [10].

[69] S. Zheng, Y. Zhou, and T. Martin. A new method for Fuzzy Formal Concept Analysis. In *Proceedings of the 2009 IEEE/WIC/ACM International Conference on Web Intelligence and International Conference on Intelligent Agent Technology - Workshops, Milan, Italy, 15-18 September 2009*, pages 405–408. IEEE Computer Society, 2009.

A Survey Questionnaire

1. What is your nationality?
2. Do you come from a developing country?
   - 2.(a) Yes
   - 2.(b) No
3. How old are you?
4. Do you or have you at any point in life lived abroad?
   - 4.(a) Yes
   - 4.(b) No
5. How often do (or did) you travel to local tourism destinations of your country when you are (or were) living there?
   - 5.(a) Rarely or Ocassionally
   - 5.(b) Once a year
   - 5.(c) Twice a year
   - 5.(d) Very often
   - 5.(e) Never
6. When you travel to local destinations in your country where do you find information about the place? (choose all that apply)
   - 6.(a) Social Media (Facebook, Twitter, etc)
   - 6.(b) Travel Blogs
   - 6.(c) Travel App: TripAdvisor
   - 6.(d) Travel App: Foursquare
   - 6.(e) Friends
   - 6.(f) Other
7. If you have visited tourism locations abroad then what are the most common tools you use to find information? (choose all that apply)
   - 7.(a) TripAdvisor
   - 7.(b) Foursquare
   - 7.(c) Blogs
   - 7.(d) Facebook
   - 7.(e) Other Social Media
   - 7.(f) Never Been Abroad

7.(g) Friends

7.(h) Other

8. Please mention the name of any other travel app that is more popular than TripAdvisor/Foursquare in your country? (Leave blank if there is none)

9. If you answered the previous question, then why do you think the application is more popular than other internationally popular applications? (Don't answer if you left last question blank)

9.(a) Usability

9.(b) Structured Information

9.(c) More Information

9.(d) Locally Developed

9.(e) Better Performance

9.(f) Better Recommendation

9.(g) Better or More Features

9.(h) Possibility of Social Networking

9.(i) Other

10. Do you think popular tourism applications such as TripAdvisor contain inadequate information about tourism destinations in your country?

10.(a) No

10.(b) Yes

10.(c) What is TripAdvisor?

10.(d) Not enough tourism destination in my country

11. Rank the reasons that may cause difficulty/barrier in visiting a tourist site in your country (1: most important, 8: least important)

11.(a) Lack of Information

11.(b) Lack of Promotion

11.(c) Proper of Safe Transport

11.(d) Expensive

11.(e) Lack of Security

11.(f) No Suitable Accommodation

11.(g) Maintenance of Site(Cleanliness)

11.(h) Lack of Interesting Activities

12. Do you think it is possible to attract foreign tourists to your country by addressing the barriers mentioned above?

12.(a) No

12.(b) Yes

12.(c) We already have too many tourists!

What are the main reasons you might learn about or try out a new travel application in the market? (choose the best 3)

12.(a) Popularity

12.(b) Friend's Recommendation

12.(c) Advertisement

12.(d) Social Media Promotion

12.(e) Read about it on Tech magazine or blog

12.(f) Suggestion from someone I follow on Twitter

12.(g) I keep an eye out for new tech startups

12.(h) Other

13. What according to you are the most interesting/important features in a new crowdsourced travel app? (Choose no more than 3)

13.(a) Interactive Map

13.(b) Gamified (trophies etc)

13.(c) Follow friend's journey

13.(d) Sentiment-based Recommendation

13.(e) Structured Information

13.(f) Book Travel Activities

13.(g) Other

B List of Filtering Keywords

- tourism
- touristic spot
- travel
- trip
- holiday
- tour
- traveling
- vacation
- holiday view
- holiday destination
- visiting
- tourist
- traveler
- traveller
- travelblog
- sightseen
- sightseeing
- summerholiday
- winterholiday
- trekking
- hiking
- niceview
- travel photography
- ttot
- TravelTuesday
- TBEX

- MexMonday
- BeachThursday
- rtw
- travelmassive
- wanderlust

C List of Emoticons

Positive Emoticons

😀	GRINNING FACE	💛	YELLOW HEART	🌞	SUN WITH FACE	🍺	BEER MUG
😍	SMILING FACE WITH HEART-SHAPED EYES	💜	PURPLE HEART	⛅	SUN BEHIND CLOUD	☕	HOT BEVERAGE
😊	SMILING FACE WITH SMILING EYES	💝	HEART WITH RIBBON	🌈	RAINBOW	🍻	CLINKING BEER MUGS
😃	SMILING FACE WITH OPEN MOUTH	💞	REVOLVING HEARTS	⛱	UMBRELLA ON GROUND	🍷	WINE GLASS
😂	FACE WITH TEARS OF JOY	💟	HEART DECORATION	❄	SNOWFLAKES	🍸	COCKTAIL GLASS
😘	FACE THROWING A KISS	🎉	PARTY POPPER	☃	SNOWMAN	🍹	TROPICAL DRINK
😁	GRINNING FACE WITH SMILING EYES	😎	SMILING FACE WITH SUNGLASSES	⛄	SNOWMAN WITHOUT SNOW	🍩	DOUGNUT
😚	KISSING FACE WITH CLOSED EYES	😇	SMILING FACE WITH HALO	🎃	JACK-O-LANTERN	🍁	MAPLE LEAF
💕	TWO HEARTS	😈	SMILING FACE WITH HORNS	🎆	FIREWORKS	🍂	FALLEN LEAF
👌	OK HAND SIGN	😗	KISSING FACE	✨	SPARKLES	😆	SMILING FACE WITH OPEN MOUTH AND TIGHTLY-CLOSED EYES
👍	THUMBS UP SIGN	😙	KISSING FACE WITH SMILING EYES	🎿	SKIS	😄	SMILING FACE WITH OPEN MOUTH AND SMILING EYES
😹	GRINNING CAT FACE WITH SMILING EYES	😋	FACE SAVOURING DELICIOUS FOOD	😉	WINKING FACE	😅	SMILING FACE WITH OPEN MOUTH AND COLD SWEAT
😂	CAT FACE WITH TEARS OF JOY	💋	KISS MARK	😌	RELIEVED FACE	😜	FACE WITH STUCK-OUT TONGUE AND WINKING EYE
😺	SMILING CAT FACE WITH OPEN MOUTH	♥	BLACK HEART SUIT	😏	SMIRKING FACE	😛	FACE WITH STUCK-OUT TONGUE
😻	SMILING CAT FACE WITH HEART-SHAPED EYES	❤	HEAVY BLACK HEART				
😽	KISSING CAT FACE WITH CLOSED EYES	☺	WHITE SMILING FACE				
❤	BEATING HEART	🙌	RAISED HANDS IN CELEBRATION				
💖	SPARKLING HEART	👏	CLAPPING HANDS				
💗	GROWING HEART	🌴	PALM TREE				
💙	BLUE HEART	🌵	CACTUS				
💚	GREEN HEART	☀	SUN WITH RAYS				

Negative Emoticons

	UNAMUSED FACE
	LOUDLY CRYING FACE
	WEARY FACE
	PENSIVE FACE
	FLUSHED FACE
	SEE-NO-EVIL MONKEY
	FACE WITH COLD SWEAT
	CONFOUNDED FACE
	BROKEN HEART
	DISAPPOINTED FACE
	ANGRY FACE
	POUTING FACE
	CRYING FACE
	PERSEVERING FACE
	DISAPPOINTED BUT RELIEVED FACE
	FEARFUL FACE
	SLEEPY FACE
	FROWNING FACE
	TIRED FACE
	CLOUD WITH LIGHTNING AND RAIN
	UMBRELLA WITH RAIN DROPS

	FACE WITH OPEN MOUTH AND COLD SWEAT
	FACE SCREAMING IN FEAR
	ASTONISHED FACE
	DIZZY FACE
	POUTING CAT FACE
	CRYING CAT FACE
	WEARY CAT FACE
	THUMBS DOWN SIGN
	ANGRY FACE WITH HORNS
	PILE OF POO
	CONFUSED FACE
	WORRIED FACE
	ANGUISHED FACE
	FROWNING FACE WITH OPEN MOUTH
	FACE WITH OPEN MOUTH
	SNEEZING FACE
	GRIMACING FACE
	CLOUD WITH RAIN
	CLOUD WITH LIGHTNING
	UMBRELLA

D Number of Tweets and ISO Alpha-3 Country Code Per Country

The list does not contain countries that had less than 50 tweets over the two week collection period.

Country	ISO Alpha-3 Country Code	Number of Tweets
United States of America	USA	102,004
United Kingdom	GBR	22,547
Canada	CAN	5,631
India	IND	3,163
Philippines	PHL	3,119
Australia	AUS	2,482
Indonesia	IDN	2,086
Spain	ESP	1,874
South Africa	ZAF	1,786
Ireland	IRL	1,748
Italy	ITA	1,748
France	FRA	1,676
Malaysia	MYS	1,657
Netherlands	NLD	1,367
Germany	DEU	1,163
Brazil	BRA	1,097
Thailand	THA	969
Japan	JPN	924
Mexico	MEX	906
United Arab Emirates	ARE	775
Singapore	SGP	739
Nigeria	NGA	665
Greece	GRC	605
New Zealand	NZL	531
Viet Nam	VNM	490
Belgium	BEL	473
Turkey	TUR	458
Portugal	PRT	448
Kenya	KEN	418
China	CHN	383
Hong Kong	HKG	377
South Korea	KOR	342
Morocco	MAR	337
Switzerland	CHE	333
Sweden	SWE	285
Russisa	RUS	269

Pakistan	PAK	254
Argentina	ARG	248
Austria	AUT	242
Egypt	EGY	226
Norway	NOR	217
Czech Republic	CZE	213
Iceland	ISL	210
Poland	POL	210
Peru	PER	201
Denmark	DNK	188
Colombia	COL	186
Croatia	HRV	178
Cambodia	KHM	173
Ghana	GHA	161
Nepal	NPL	157
Sri Lanka	LKA	155
Israel	ISR	152
Hungary	HUN	148
Chile	CHL	145
Finland	FIN	144
Taiwan	TWN	143
Cyprus	CYP	138
Jamaica	JAM	135
Dominican Republic	DOM	130
Saudi Arabia	SAU	124
Qatar	QAT	113
Ecuador	ECU	111
Costa Rica	CRI	107
Uganda	UGA	102
Tanzania	TZA	94
Romania	ROU	89
Mauritius	MUS	86
Maldives	MDV	79
Slovenia	SVN	78
Venezuela	VEN	74
Ukraine	UKR	72
Bahamas	BHS	69
Trinidad and Tobago	TTO	63
Bangladesh	BGD	62
Bulgaria	BGR	62
Malta	MLT	57
Kuwait	KWT	56
Lebanon	LBN	54

Oman	OMN	51
Slovakia	SVK	51
Fiji	FJI	51