Queen Mary, University of London
Department of Electronic Engineering and Computer Science

Discovering Themes in Social Media

Fayimora Femi-Balogun

Supervisor: Dr. Matthew Purver

Submitted in part fulfilment of the requirements for the degree of
BSc Computer Science with Industrial Experience, April 2014

Abstract

Acknowledgements

A lot of people have contributed in different ways to the success of this project but firstly, I would like to thank my supervisor, Dr Matthew Purver who was very helpful throughout the project.

I would also like to thank the authors of all the tools used in this project. Without those tools, this project could not have been completed within the time frame.

Thanks to all the volunteers who helped with the evaluations and also those who proof read the report. Names are too numerous to mention so I say thank you all.

Finally, I would like to thank my family for their continual support to my studies.

"No amount of experimentation can ever prove me right; a single experiment can prove me wrong."

Albert Einstein

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation

Organisations today continuously search for new ways to get feedback from their clients in a bid to improve customer satisfaction. Technology firms like Apple, Samsung and Google want to know if their software/hardware products meet their consumers' needs. Merchandise retailers like Walmart and Tesco are constantly trying to make sure they are serving the right products in the right quantity and at for right price. Startups continuously evaluate their products to measure the probability of the company being successful sometime in the future. Postal services like Royal Mail are very interested in how their services are doing and what their customers despise most so they can improve.

Current ways of achieving this include **Surveys** (questionnaires or interviews) and **Focus Groups**. Surveys are very easy to create and distribute. There are also a variety of tools to help with this such as SurveyMonkey[1] and Google Docs[2]. Unfortunately, Surveys also have a few unpleasant drawbacks like time consumption and labour intensity. It can also be difficult to encourage participants to respond. Nevertheless, the main drawback to using Surveys is that some questions are left unanswered while the answers given in answered questions may not reflect the truthful sentiments of the participant. Rubin (1987) concurs with this and he goes on to discuss how this problem can be solved (to a certain extent) with imputation[3]. Hayes (2008)

[1] https://www.surveymonkey.com/

[2] https://drive.google.com

[3] Imputation is the process of inferring plausible values for missing entries

also agrees with this point of view and suggests the use of well designed leading questions to put the participant in the right frame of mind. For instance, a leading question like "*How likely will you recommend our service to friends?*" gets the participant thinking about recommendations. While the above solutions might work, they have the same drawbacks as the original problem. Imputation can be very time consuming, labour intensive and error prone while the use of leading questions fails to solve the problem of unanswered questions.

Unfortunately, interviews and focus groups also suffer from false answers due to the fact that they are not anonymous. This means that the participants, in the face of an interviewer, try to be lenient in other not to sound too negative. This could sometimes be due to the fact that participation in the interview/focus group has been incentivised with money or desirable items.

Ideally, the next question we should be asking is "*How can we get the truthful views of our clients about our products and services*"? We need to find a way to get this information without putting any pressure on our clients.

1.2 Aims and Objectives

The aim of this project is to investigate other means of getting our customer views and also, how we can make use of Machine Learning and Natural Language Processing techniques to make sense of the data.

Fortunately, the recent surge in the use of social media makes the former relatively easy. People, more often than not, tend to post their truthful feelings about services they use on social media. For instance, Person A buys an iPhone today and realises that the Wi-Fi connectivity is faulty. He/She will most likely post something like "*New iPhone wifi not working #NotCool*" on one or more of the available social networking platforms. From this statement, we can infer that Person A is talking about *the iPhone*, *Wi-Fi* and *Connectivity.* The process of discovering abstract topics in text is called **Topic Modelling** and Chapter 4 discusses how we can automate this process.

We try to answer two main research questions. They include:

- Can we use supervised techniques to accurately classify tweets into what is relevant and what is not?

- Can we detect themes/topics in our dataset? If yes, are these topics related to Apple Inc in any way?

1.3 Why Twitter?

Twitter is a social micro-blogging platform where users can share messages in 140 characters. It also allows its users to follow each other. This means, if person A follows person B, A will see public messages from B. These messages are usually referred to as tweets.

Tweets are capped to 140 characters and can contain text, links or a combination of both. They are usually related to either an event, interests or just personal opinion. Facebook posts are mostly always well thought out and each post might include multiple topics. Tweets on the other hand are usually written at the speed of thought.

According to Mashable, DOMO, a Business Intelligence company paired up with Column Five Media to create an infographic[4] about the web back in 2012. It showed that Twitter at the time received around 100,000 tweets per minute. As at 1st February 2014 Twitter claims to receive 500 million tweets a day[5]. That is roughly 350,000 tweets per minute which is over 3 times the amount 2 years before. Twitter also claims to have 241 million monthly users.

Finally, Twitter's data is open compared to other social platforms like Facebook. This means developers are free to tap into this wealth of data in almost real time and free of charge. This makes Twitter a very good source for our data.

1.4 Methodology

This study requires social data and the dataset used was gathered from Twitter between October and November 2013. Each tweet in the dataset is in someway related to Apple Inc and/or their products.

[4] See `http://mashable.com/2012/06/22/data-created-every-minute/`

[5] See https://about.twitter.com/company

We then train a classifier to help filter out as many irrelevant tweets as possible. We briefly analyse different ways to filter the dataset but eventually settle with using Naïve Bayes Classifier. We also look into different ways of analysing the classifier's performance and ways it can be improved.

Finally, we attempt to identify topics/themes in the dataset. We briefly look at Latent Semantic Indexing and why it might not be suitable for our needs. We then look into Latent Dirichlet Allocation, a common approach to topic modelling and use it to detect topics in our dataset. The evaluation of topics generated will be analysed empirically and qualitatively. This means we take a topic and make some assumptions about the semantics of the tweets belonging to that topic. We then analyse the tweets to confirm the validity of our assumption.

1.5 Statement of Originality

This report with any accompanying implementation, is submitted as part requirement for the degree of Computer Science with Industrial Experience at Queen Mary, University of London. I certify that it has not been submitted for any degree or other purposes.

I certify that the intellectual content of this report, to the best of my knowledge, is the product of my own labour except where indicated in the text.

Chapter 2

Background Theory

2.1 Introduction

Automatic Text Classification or Text Categorisation is a rapidly growing field in Machine Learning and Natural Language Processing. This is mainly due to the amount of electronic data we currently generate. The main task is to assign one or more classes to a given text document. Some applications of text classification include *Email Spam Detection* and *Language Detection*. The former involves trying to distinguish spam emails from legitimate ones while the latter involves the identification of the language a document was written in.

However, this study makes use of classification techniques for data filtration which involves removing irrelevant documents from a list of documents(similar to spam filtering) and topic modelling (extracting topics from a list of documents) and sentiment analysis (predicting the sentiment of the author of a document). This chapter explains a few background concepts and reviews some relevant research previously done in this area.

2.2 Naïve Bayes Classifier

The Naïve Bayes classifier is one of the simplest classifier that can be used and this is due to the fact that it is based on simple Bayes Theorem. It is a probabilistic classifier which assumes that all features of the documents are independent of each other. This means that if a document has features $f1$ and $f2$ (could be length of document, occurrence of words e.t.c), the existence of $f1$ has nothing to do with the existence of $f2$ and vice versa. This also means that it makes assumptions that may or may not be correct, hence the "Naïve" in its name.

Bayes theorem states that the probability of A given B is the probability of B given A times the probability of A divided by the probability of B. Mathematically, this is written as:

$$p(A|B) = \frac{p(B|A)p(A)}{p(B)} \tag{2.1}$$

Applying this logic to text classification, the probability that a document $d_i \in D$ belongs to a class c is denoted as:

$$p(c|d_i) = \frac{p(d_i|c)p(c)}{p(d_i)} \tag{2.2}$$

Although other techniques like Maximum Entropy, Random Forests or Support Vector Machines tend to perform better, a naive Bayes classifier will require less memory and CPU cycles. Furthermore, it is computationally less complex and simpler to implement. With regard to performance, Huang *et al.* (2003) showed using multiple datasets from Blake & Merz (1998) that the naive Bayes classifier in many cases performs as good as other complex classifiers and Zhang (2004) goes further to explain why it performs well. Other studies have also found Bayesian classifiers to be effective without being affected by its simple independence assumption (Langley *et al.* , 1992; Manning *et al.* , 2008).

The Naïve Bayes classifier has been used in many text classification problems but one of its common applications which is relevant to tweet classification is email spam[1] filtering. A spam filter is a system that takes in text and decides whether or not it is spam. Androutsopoulos *et al.* (2000) addressed this issue using a naive Bayes classifier. They trained the model using a predefined set of manually labelled messages. They were able to show that the naive Bayes classifier was capable of classifying messages with impressive accuracy and precision compared to the then common keyword based approach to classification. Deshpande *et al.* (2007) also carried out a similar research and the results were equally impressive and similar.

[1] irrelevant or unsolicited messages. They are typically to large numbers of users

2.3 Topic Modelling

Topic Modelling is a process by which abstract topics/themes are extracted from a collection of documents. This process is usually carried out with the aid of topic models, a suite of algorithms used for topic modelling. It has been applied in a variety of fields like Software Analysis where Linstead *et al.* (2009) used topic modelling to find topics embedded in code and Gethers & Poshyvanyk (2010) used topic modelling to capture coupling among classes. Kireyev *et al.* (2009) applied topic models on disaster related data from Twitter in an effort to determine what topics were discussed within the time span of a natural disaster. Hospedales *et al.* (2009) introduced a new topic model that can be used to analyze videos with complex and crowded scenes in other to discover regularities in the videos. A system built on such model will be able to answer a question like "What interesting events happened in the last 5 hours". Other fields include Audio Analysis (Smaragdis *et al.* , 2009), Influence modelling (Gerrish & Blei, 2009), Finance (Doyle & Elkan, 2009), Writer Identification (Bhardwaj *et al.* , 2009) and many more.

There are a number of topic models but the two main ones are ***Latent Semantic Indexing*** (LSI) and ***Latent Dirichlet Allocation*** (LDA) and we discuss them further in the following sections.

2.3.1 Latent Semantic Indexing

Latent Semantic Indexing(LSI) (Hofmann, 1999), sometimes referred to as *Latent Semantic Analysis*, is an indexing technique that leverages matrix-algebra computations[2] to identify any patterns in relationships between a collection of text documents. It works based on the assumption that words used in the same context tend to have homogeneous meanings (Deerwester *et al.* , 1990; Dumais, 2004; Landauer, 2006). LSI, has been used mostly in Information Retrieval and Search Engine Optimisation where it tries to figure out what words in a web page are relevant to the web page even though they might not be used in that page. One of the main drawbacks the LSI model suffers from is ambiguity.

Assuming we have two documents, one talking about Microsoft Office and the other talking about actual physical office space. How can the model differentiate between the two? Unfor-

[2] Specifically, it uses Singular Value Decomposition which is a factorization of a complex matrix. See `http://en.wikipedia.org/wiki/Singular_value_decomposition`

tunately, it is unable to differentiate between such topics and a significant step to solve this problem was made by Hofmann (1999) who presented the probabilistic LSI model. Blei *et al.* (2003) argues that while Hoffman's work is a very useful step towards using probabilistic models to model text, it is incomplete.

2.3.2 Latent Dirichlet Allocation

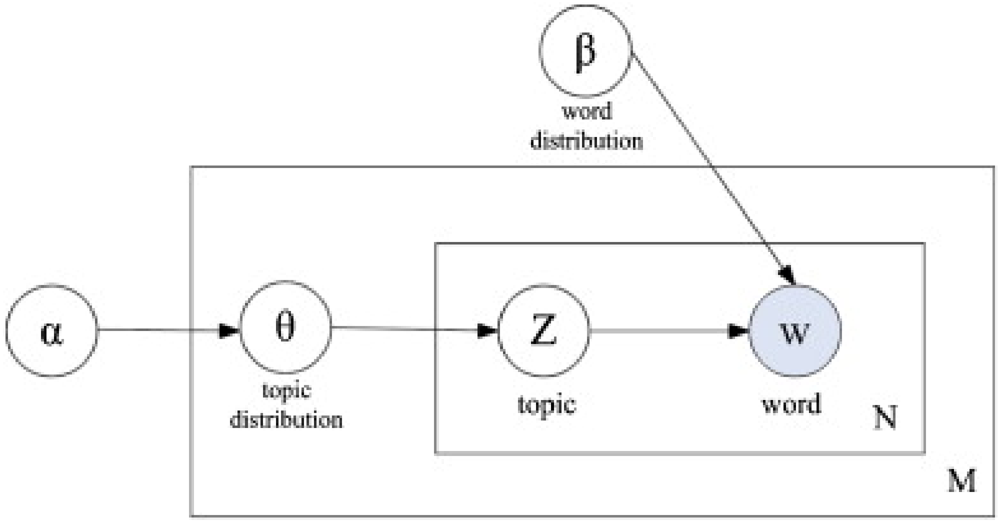


Figure 2.1: A graphical model representation of LDA (Courtesy: `http://victorfang.wordpress.com/2012/03/11/latent-dirichlet-allocation`)

Latent Dirichlet Allocation(LDA) is a generative[3] and probabilistic model that can be used to automatically group words into topics and documents into a mixture of topics (Blei *et al.* , 2003). It works based on the assumption that each document contains one or more topics. Words can also exist in multiple topics as they actually do in natural language. In other to tackle the problem of ambiguity LSI suffers from, LDA takes a combination of all topics that seem relevant to a document in a corpora[4] and compares that document to the topics in an effort to determine which topic is closer to the document. Figure 2.1 shows a graphical model representation of Latent Dirichlet Allocation. The inner boxes represents the choice of topics and words within a document while the outer box represents the actual documents.

Hoffman *et al.* (2010) developed a variant of LDA called Online LDA which uses variational Bayes as its posterior inference algorithm as opposed to Gibbs Sampling. It also allows the

[3] See `http://en.wikipedia.org/wiki/Generative_model`

[4] Corpora is simply a large collection of documents

model to be updated with more data after initial training. During initial training, the entire corpora is observed/trained in batches rather than at once. Asuncion *et al.* (2009) shows that although this model uses constant memory and it converges quicker, it still requires a full pass through the entire corpora. This makes it very slow when applied to large datasets.

An oversimplified version of the algorithm is:

```
while model is yet to converge do
    Data: $B$ = randomly selected mini-batch of documents;
    for $b \in B$ do
        Estimate approximate posterior over what topics each word in each document
        came from;
        Update posterior over topic distributions based on what words are believed to
        have come from what topics;
    end
end
```

Most of the research done on social media data, especially Twitter, has been to detect usage and communities (Java *et al.* , 2007). Nonetheless, recent research has started to look into the detection of topics in social media. Kireyev *et al.* (2009) used LDA to extract topics/themes from a collection of disaster related tweets. Zhao *et al.* (2011) used LDA to compare news related tweets on Twitter with topics in The New York Times. They were also able to show that the standard LDA might not always work well on tweets and so they proposed a new model which is a slight variant of LDA. Weng *et al.* (2010) proposed an algorithm that leverages LDA to find topic-sensitive influential twitter users. Lau *et al.* (2012) presented an LDA-based model for detecting and tracking emerging trends/events on microblogs like Twitter.

Chapter 3

Data Classification

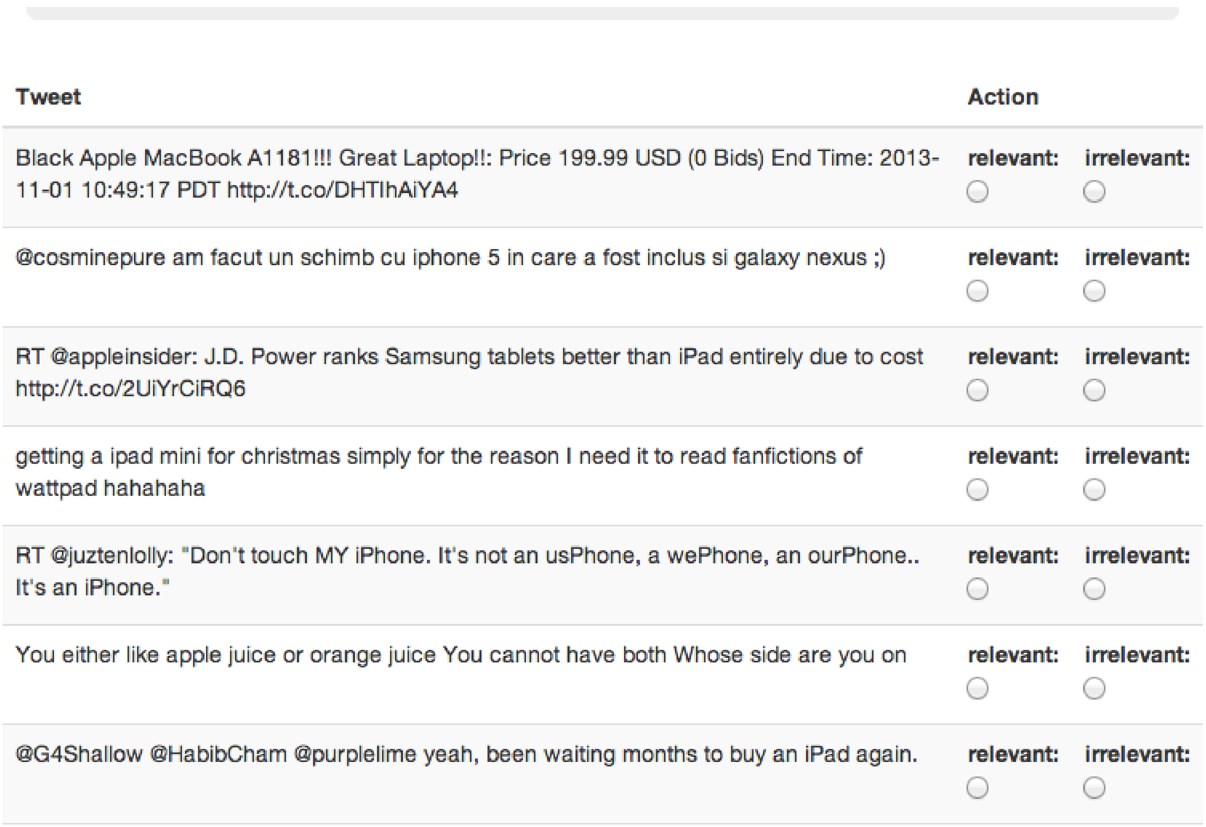
In this Chapter, we train a classifier to classify our dataset into two groups, relevant and irrelevant. Irrelevant tweets are tweets which we do not really care about. Some examples include:

- *Every day I'm levelling! And now I'm level 19 in #CSRClassics for iPhone!*
- *Yes, our apple juice and cider are both GMO-free.*
- *I just had my first carmel apple*

All three tweets could be regarded as relevant but for our use case, they are not. This is because we are only interested in tweets that contain personal opinions about Apple Incorporated. Examples of relevant tweets include: their thoughts

- *Once you get hooked to #Mac, you will definitely go back to #Windows! Lol!*
- *If Tim Cook at Apple knows anything about him, it'd be to stay away from Icahn.*

Of course we can manually classify this data but when we have millions of tweets, this becomes impracticable. This is why we employ some classification algorithms to assist us. The following sections discuss how we can achieve this.



Tweet	Action	
Black Apple MacBook A1181!!! Great Laptop!!: Price 199.99 USD (0 Bids) End Time: 2013-11-01 10:49:17 PDT http://t.co/DHTIhAiYA4	relevant: ○	irrelevant: ○
@cosminepure am facut un schimb cu iphone 5 in care a fost inclus si galaxy nexus ;)	relevant: ○	irrelevant: ○
RT @appleinsider: J.D. Power ranks Samsung tablets better than iPad entirely due to cost http://t.co/2UiYrCiRQ6	relevant: ○	irrelevant: ○
getting a ipad mini for christmas simply for the reason I need it to read fanfictions of wattpad hahahaha	relevant: ○	irrelevant: ○
RT @juztenlolly: "Don't touch MY iPhone. It's not an usPhone, a wePhone, an ourPhone.. It's an iPhone."	relevant: ○	irrelevant: ○
You either like apple juice or orange juice You cannot have both Whose side are you on	relevant: ○	irrelevant: ○
@G4Shallow @HabibCham @purplelime yeah, been waiting months to buy an iPad again.	relevant: ○	irrelevant: ○

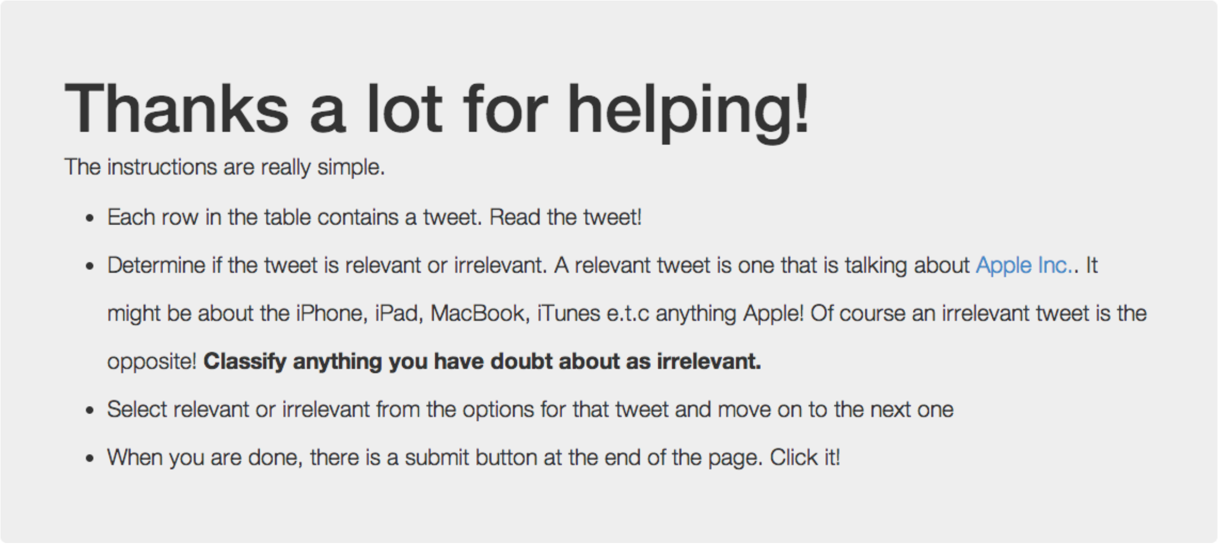
Figure 3.1: The data labelling application

3.1 Preparing train data

Train data, also known as a training set is a set of data used to train a knowledge database, in this case, a classifier. Our training set will be created by manually labelling a fraction of our dataset. People write in different ways on Twitter and trying to create a new training set to encompass all possibilities would be very time consuming and intractable. To make this process a little easier, a web application for labelling tweets was created. Figure 3.1 is a screen shot of what the application looks like.

While using the web application in Figure 3.1 makes labelling tweets easier and a little quicker, it does not change the fact that a plethora of tweets still have to be manually labelled. To speed up this process even further, the data labeller was made public and the labelling task was crowd sourced. A list of instructions (Figure 3.2) were also given to anyone who helped label the tweets.

One problem with crowd sourcing this task is that people have different opinions about what is relevant and what is not. In an attempt to solve this problem, each tweet was classified twice.



Thanks a lot for helping!

The instructions are really simple.

- Each row in the table contains a tweet. Read the tweet!
- Determine if the tweet is relevant or irrelevant. A relevant tweet is one that is talking about Apple Inc.. It might be about the iPhone, iPad, MacBook, iTunes e.t.c anything Apple! Of course an irrelevant tweet is the opposite! **Classify anything you have doubt about as irrelevant.**
- Select relevant or irrelevant from the options for that tweet and move on to the next one
- When you are done, there is a submit button at the end of the page. Click it!

Figure 3.2: Instructions on how to label the tweets

A tweet classified as relevant gets a score of 1 and an irrelevant tweet gets 0. This means that if a tweet was classified twice as relevant, it should have a score of 2 and a tweet classified as irrelevant twice should have a score of 0. Tweets that have been classified twice and have a total score of 1 are tweets that have been classified as both relevant and irrelevant. These are tweets that we have to classify ourselves. While this is not an assured way of getting the best training set, it gives us a certain level of confidence about our training set.

3.2 Training a classifier

As discussed in Section 2.2, a Naïve Bayes Classifier is a probabilistic classifier which is based on the Bayes Theorem. We will train one and use it to classify the tweets into relevant and irrelevant groups.

Unfortunately, the classifier takes as input a vector space representation of our tweets and not the actual text. This means we have to convert our tweets into a vector representation of some sort. We will be using the **bag-of-words model** in this study but before we transform the tweets, we have to pre-process the tweets.

3.2.1 Pre-processing

Pre-processing are the tasks we have to carry out before the main transformation of the tweets to a vector space model. Firstly, we will peruse through our tweets to remove new line characters, links and stop words. We then take each tweet and convert it into a list of *n-grams.*

Some tweets have special characters like new lines, excess spaces and Unicode characters and these characters are irrelevant for our use-case. Every programming language has a function to strip off new lines and whitespace and it can be easily done in one line of code. Removing the links from the text is a little more complex and the "easiest" way to do this would be to use a regular expression. Friedl (2006) in his book *Mastering Regular Expressions* describes regular expressions as a very flexible mini language that is used for text processing. The regular expression used can be found in Appendix A.

Unfortunately, all a regular expression can do is search for patterns in text and luckily, most programming languages provide support for regular expressions. All we have to do is search for the pattern in each tweet and use the language features to replace the matched pattern with nothing(an empty string).

The next step is to remove stop words in each tweet. Wilbur & Sirotkin (1992) defines a stop word as "*a word which may be identified as a word that has the same likelihood of occurring in those documents not relevant to a query as in those documents relevant to the query.*" In other words, stop words occur in every document irrespective of the document's relevance. Stop words are usually the most common word in a language, English in this case. Some examples include *and*, *or*, *the* etc. Removal of stop words usually results in better model performance as shown in Figure 3.3 on page 16.

Finally, we convert each tweet to a list of *n-grams.* An n-gram "*is a contiguous sequence of n items from a given sequence of text*"[1]. The easiest way to understand n-grams is with an example. Assuming we have a document with the text "machine learning rocks". All unigrams(n-grams where n is 1) that can be extracted from that text are *"machine"*, *"learning"* and *"rocks"*. Also, all bigrams(n-grams where n is 2) in the document are *"machine learning"* and *"learning rocks"*. In this study, we will be using a combination of unigrams and bigrams.

[1] See http://en.wikipedia.org/wiki/N-gram

	today	what	it	is	a	sunny	day
A	1	0	0	1	1	1	1
B	0	0	2	1	1	1	1
C	0	1	0	0	1	1	1

Table 3.1: A bag-of-words representation

We have discussed different preprocessing tasks that we have to apply to our documents before transforming them into the bag of words matrix representation. In the next section, we will look into how the bag of words model works and then transform our tweets into this model.

3.2.2 Transforming tweets to bag-of-words

The bag-of-words model is a common representation for text that involves representing a document as a multiset of its words. It is a very common way to represent documents and it has also been used recently in computer vision (Sivic & Zisserman, 2009). All sets are combined to form a document-term matrix of the corpora. The rows represent each document while the columns represent the occurrence/frequency of a word in that document. To show how this works, let us assume we have the following documents:

A today is a sunny day.

B it is a sunny day isn't it?

C what a sunny day!

By the above definition, Table 3.1 will be an accurate representation of our sentences using the bag-of-words model. Note that in our example, each sentence is a document and all sentences form the corpora.

Now that we have converted our corpora into a bag-of-words representation, we will now use the resulting matrix to train our classifier.

accuracy	std(σ)	AUC	std(σ)
0.9875	0.0000	0.7387	0.0000
0.9877	0.0002	0.7381	0.0000
0.9878	0.0002	0.7312	0.0090
0.9877	0.0002	0.7412	0.0190
0.9878	0.0002	0.7454	0.0190
0.9876	0.0005	0.7394	0.0220
0.9875	0.0005	0.7431	0.0220
0.9874	0.0005	0.7427	0.0200
0.9874	0.0005	**0.7455**	0.0210
0.9873	0.0005	0.7454	0.0200

(a) With stop words

accuracy	std(σ)	AUC	std(σ)
0.9908	0.0000	0.7526	0.0000
0.9904	0.0004	0.7589	0.0060
0.9903	0.0003	0.7485	0.0150
0.9901	0.0004	0.7535	0.0160
0.9901	0.0003	0.7572	0.0160
0.9901	0.0003	0.7582	0.0150
0.9902	0.0004	**0.7618**	0.0160
0.9901	0.0004	0.7587	0.0170
0.9901	0.0004	0.7592	0.0160
0.9900	0.0004	0.7572	0.0160

(b) Without stop words

Table 3.2: Accuracy and AUC for 10-fold cross validation

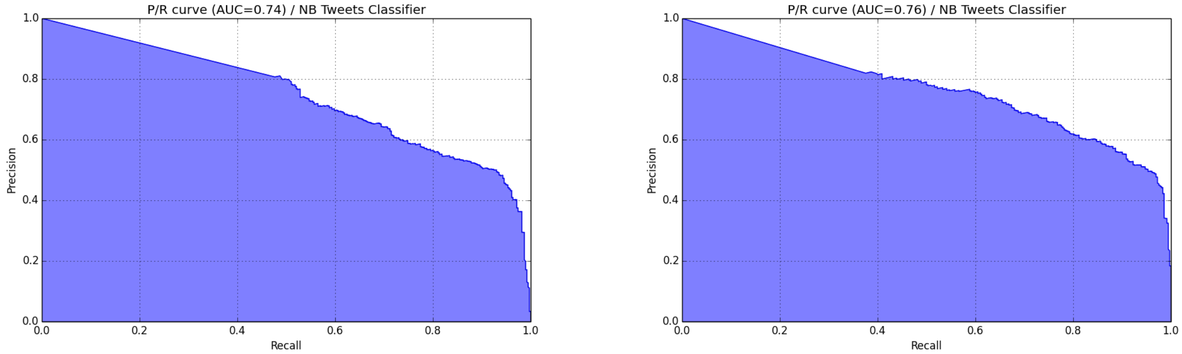
3.2.3 Training the initial classifier

In Section 3.2.1, we looked at different pre-processing tasks to be carried out, one of which was the removal of stop words. In this section, we train two classifiers, one with stop words in our data and the other with stop words removed from the data. In other to measure our classifier's performance, we use three different but complementary metrics. They include:

Accuracy: This is the degree to which our classifier is correct. For instance, if we give it 10 instances and it rightly classifies 7, then we say our classifier is 70% accurate. The problem with accuracy is that the value does not take into consideration the number of false positives or false negatives. So for instance, how many relevant samples were classified as irrelevant and vice versa. Research by Ling *et al.* (2003) shows that accuracy is also not a good measure when dealing with unbalanced classes. Our dataset is highly unbalanced so we cannot use accuracy to evaluate performance.

Receiver Operating Characteristic curve: The ROC curve was introduced in signal processing and used to evaluate the prediction power of different classification algorithms. More importantly, the area under this curve, referrred to henceforth as AUC, is the metric used to compare the performance of different classifiers (Bradley, 1997). As Ling *et al.* (2003); Huang & Ling (2005) describes, AUC is generally a better measure because it takes into consideration the precision an sensitivity(also called recall) of the classifier[2].

[2] See `http://en.wikipedia.org/wiki/Precision_and_recall` for a more detailed explanation of precision and recall



(a) AUC=74 with stop words (b) AUC=76 without stop words

Figure 3.3: AUC curves with and without stop words

The ROC curve is created by plotting the precision of a classifier against its sensitivity.

K-fold Cross Validation: Training and validating a classifier with the same dataset is ineffective because the classifier simply labels examples it has just seen. This problem is called **overfitting** and to solve it, we use the k-fold cross validation technique. The idea is to split our data into k folds[3], train the classifier on $k-1$ folds and test on the remaining fold. This process is repeated k times to ensure every fold has been used for validation.

Tables 3.2a and 3.2b contains a list of values representing the result of running 10-fold cross validation on our trained classifier with and without stop words, respectively. It also depicts the accuracy, AUC and standard deviation between folds but we shall focus more on the AUC.

We can see from the tables that the standard deviation(σ) is very low. A low standard deviation indicates that there is not much variation between folds and the values are very close to the mean value. Comparing the highest AUC in both tables(values in bold), we can see that the removal of stop words gives us approximately 2.2%(from 0.7455 to 0.7618 AUC) increase in AUC. Figures 3.3a and 3.3b gives a visual representation of the AUC curve. The value used in the plots is the average AUC over all folds. This increase proves that training a classifier without stop words in our dataset gives better performance.

While 0.76 is an acceptable AUC, it turns out we can do even better and the next section shows how we can achieve this.

[3] A fold is essentially a fraction of the dataset

accuracy	std(σ)	AUC	std(σ)
0.9939	0.0000	**0.8068**	0.0000
0.9939	0.0000	0.8006	0.0062
0.9949	0.0001	0.7893	0.0167
0.9939	0.0003	0.7960	0.0186
0.9939	0.0003	0.7892	0.0215
0.9939	0.0004	0.7920	0.0206
0.9939	0.0004	0.7902	0.0196
0.9939	0.0004	0.7900	0.0183
0.9939	0.0004	0.7890	0.0175
0.9939	0.0004	0.7876	0.0171

Table 3.3: Accuracy and AUC for tfidf weighted model

3.2.4 Improving the classifier

In the previous section we were able to get up to 0.76 AUC. In this section we explore two ways of improving this value. Firstly, we look into a new type of vector space model for representing our corpus called the tf-idf weighting scheme. We briefly look at how it is different from the bag-of-words model and what makes it better. Secondly, we train multiple classifiers with different combinations of model parameters to try to find the best possible combinations of parameters.

3.2.4.1 Using TF-IDF Weighting Scheme

TF-IDF(Term Frequency-Inverse Document Frequency) is a weighting scheme that can be used to determine the importance of a word to a document in a corpus. Up until now, we have been using the bag-of-words representation which gives us the term[4] frequency. However, this is inadequate because a document with 5 occurrences of a word is not necessarily 5 times more relevant than a document with only 1 occurrence of the same word. This is where the inverse document frequency comes in.

The inverse document frequency measures how important a word is in a document. It weighs down terms with a high frequency and scales up the terms with lower occurrences. This is useful mainly because we can get rid of stop words that do not appear in our stop words list.

[4] A term is also a word in this case

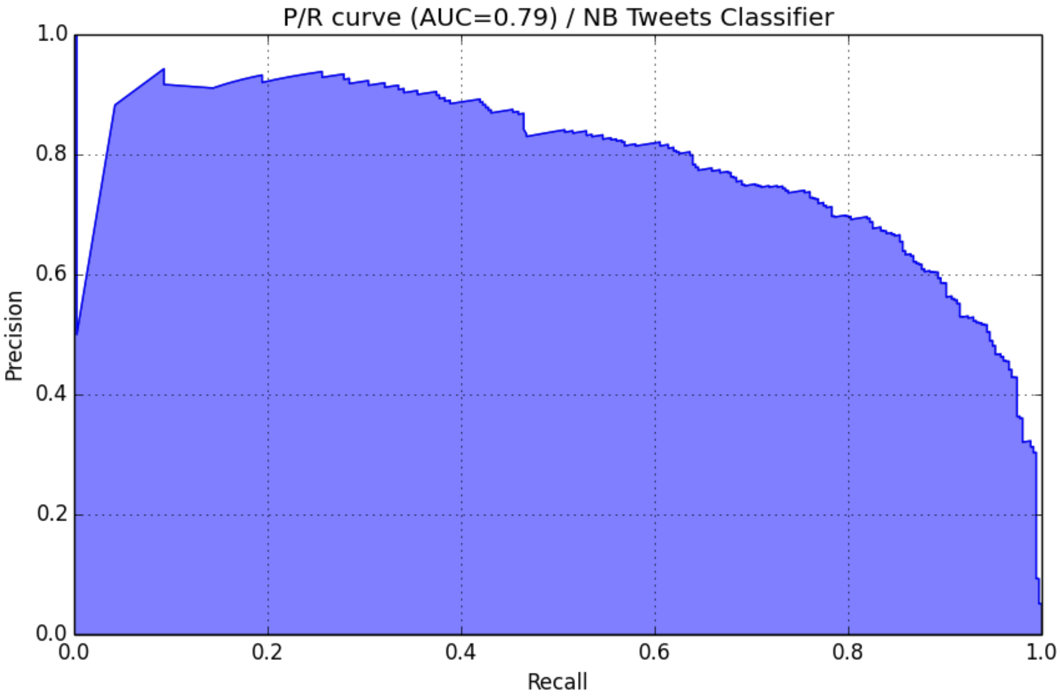


Figure 3.4: AUC curve for tf-idf weighted corpora

Table 3.3 shows the results of running 10-fold cross validation using the tf-idf scheme. We can see that our standard deviation remains low, as in the previous section which is good but more importantly, we have a increase in AUC. We achieved approximately 0.76 AUC in the previous section and our highest value after using the tf-idf weighting scheme is approximately 0.80 AUC which is a 5.7% increase in AUC. Figure 3.4 is a visual representation of the area under the ROC curve, AUC. The value used in the plot is the average AUC over all folds.

3.2.4.2 Exhaustive Grid Search for Model Selection

So far, we have improved our model by manually changing parameters. We could automate our model selection process by training multiple models with a different combination of parameters and then selecting the best model. This is usually referred to as grid search and it is a common way to optimize parameters for models. One way to achieve this is to wrap a chain of pre-processors, so it can be represented as one large pre-processor, and add the classifier to the end of the chain. We shall refer to this chain as a *pipeline*. When the pipeline is invoked, the data goes through a chain of pre-processors and finally the classifier. Our pipeline must accept all inputs for each pre-processor and the classifier. The creation of this pipeline is not necessary

accuracy	std(σ)	AUC	std(σ)
0.9961	0.0000	0.8415	0.0000
0.9960	0.0001	**0.8610**	0.0195
0.9955	0.0007	0.8445	0.0282
0.9955	0.0006	0.8491	0.0257
0.9954	0.0005	0.8441	0.0251
0.9954	0.0005	0.8447	0.0229
0.9954	0.0005	0.8465	0.0217
0.9954	0.0004	0.8470	0.0203
0.9954	0.0004	0.8468	0.0192
0.9954	0.0004	0.8487	0.0191

Table 3.4: Accuracy and AUC for best model

but it makes automating the grid search easier. Also, depending on the number of parameters and how long it takes to train a classifier and cross-validate it, this task can easily take hours to complete.

Table 3.4 shows the results for the cross-validation of the best model in our grid. Although the standard deviation across folds is slightly higher(averagely 25% higher) compared to our previous models, there is also a big improvement in AUC. Our best fold produced 0.86 AUC which is a 7.2% increase. Figure 3.5 also shows a visual representation of the AUC.

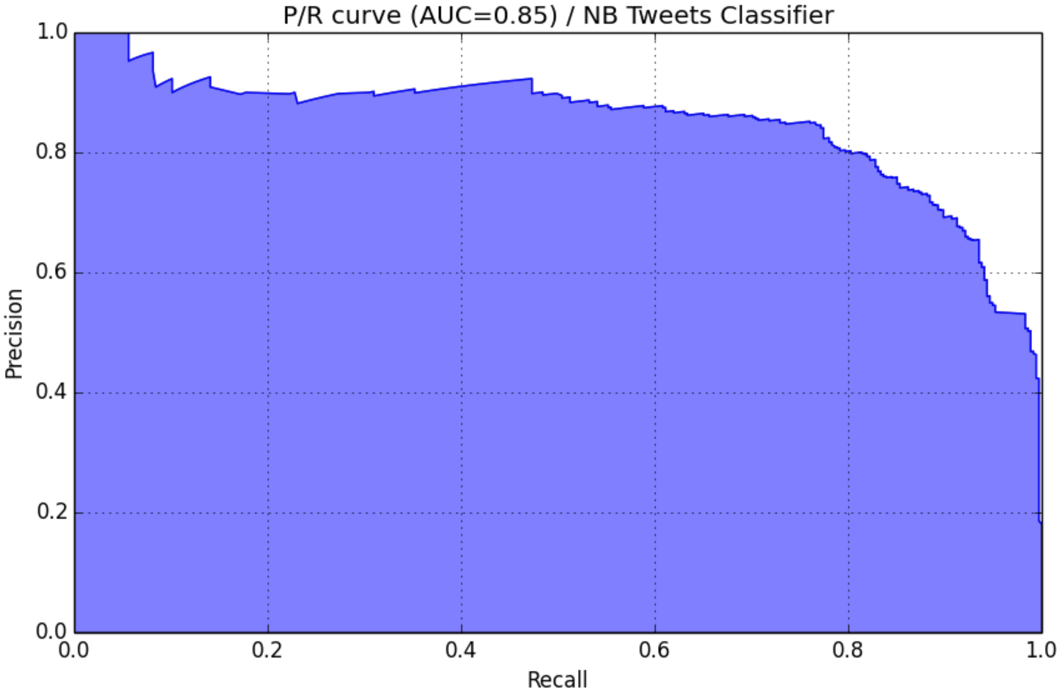


Figure 3.5: AUC curve for best found model

Chapter 4

Topic Modelling

In this chapter, we use a topic model to find themes/topics that exist in our dataset. Our input dataset is a set of relevant tweets as determined by the classifier in the previous chapter. We use Latent Dirichlet Allocation as our topic model as described in Section 2.3.2 on page 8.

Tables 4.1 and 4.11 on pages 24 and 36 respectively, show a list of 30 and 40 topics. It also contains their respective topic-tokens distribution. For the purpose of this study, a token is either a unigram or bigram. Each row comprises of a list of tokens that try to explain a topic and they are ordered by their level of influence. While it is helpful to have our tokens ordered by level of influence, the respective influence values are excluded from the table because we will not pay much attention to them during our analysis and evaluation.

4.1 Pre-processing

Before running LDA on the dataset, we have to cleanup our dataset and what we do is very similar to what we did in Section 3.2.1. In summary, we perused our tweets and removed the occurrence of special characters like new lines, all links and stop words from each tweet. Finally, we converted the tweets into features(unigrams and bigrams). For topic modelling, we perform the same operations and more. Specifically, we remove anything that does not add value to the topic model.

Firstly, we remove usernames from our data. Our topic model would try to find relationships in tweets by using the words that occur in them. Usernames are also technically words but are semantically irrelevant for our use case. Fortunately, usernames on twitter follow a specific pattern so they can be easily removed with a regular expression. The regular expression used can be found in Appendix A. Secondly, our dataset is Apple centric and as a result, words like “iphone”, “ipod” and “ipad” are bound to occur in almost every tweet. For this reason, we add these words to our stop words list so they are remove from every tweet. Appendix A has a complete list of these words.

4.2 Evaluating Topic Models

In this section, we analyse two separate models one of which comprises of 30 topics and the other, of 40 topics. They both use a mixture of unigrams and bigrams in their token distribution. This was inspired by our experiments in Section 3.2 where the classifier showed better performance when using a mixture of unigrams and bigrams. We analyse and evaluate a few topics from the 30 topics model and have a look at some of the tweets that fall under those topics. We the compare topics generated from the 30 topics model to that of the 40 topics model.

4.2.1 Evaluating 30 Topics

Table 4.1 shows a list of 30 topics and their respective token distribution. This table can be used to get an abstract view of the topics but to get an even better view of distribution across all topics, we refer to Figure 4.1 on page 22 which is a word cloud of all tokens on our table. The frequency of a word determines its size in the cloud.

We can see words like “android” and “ios” which are mobile operating systems built by Google and Apple, respectively. We can also see words like “app”, “issues”, “5s”,“mini”, “google” and “samsung” which can be in some way related to Apple. For instance, “app” might refer to applications on any of Apple’s platform while “5s” could refer to the new mobile phone released by Apple around the time our data was gathered. Google and Samsung are competitors with Apple, so these words could have been gotten from tweets that compare either their products or companies as a whole.

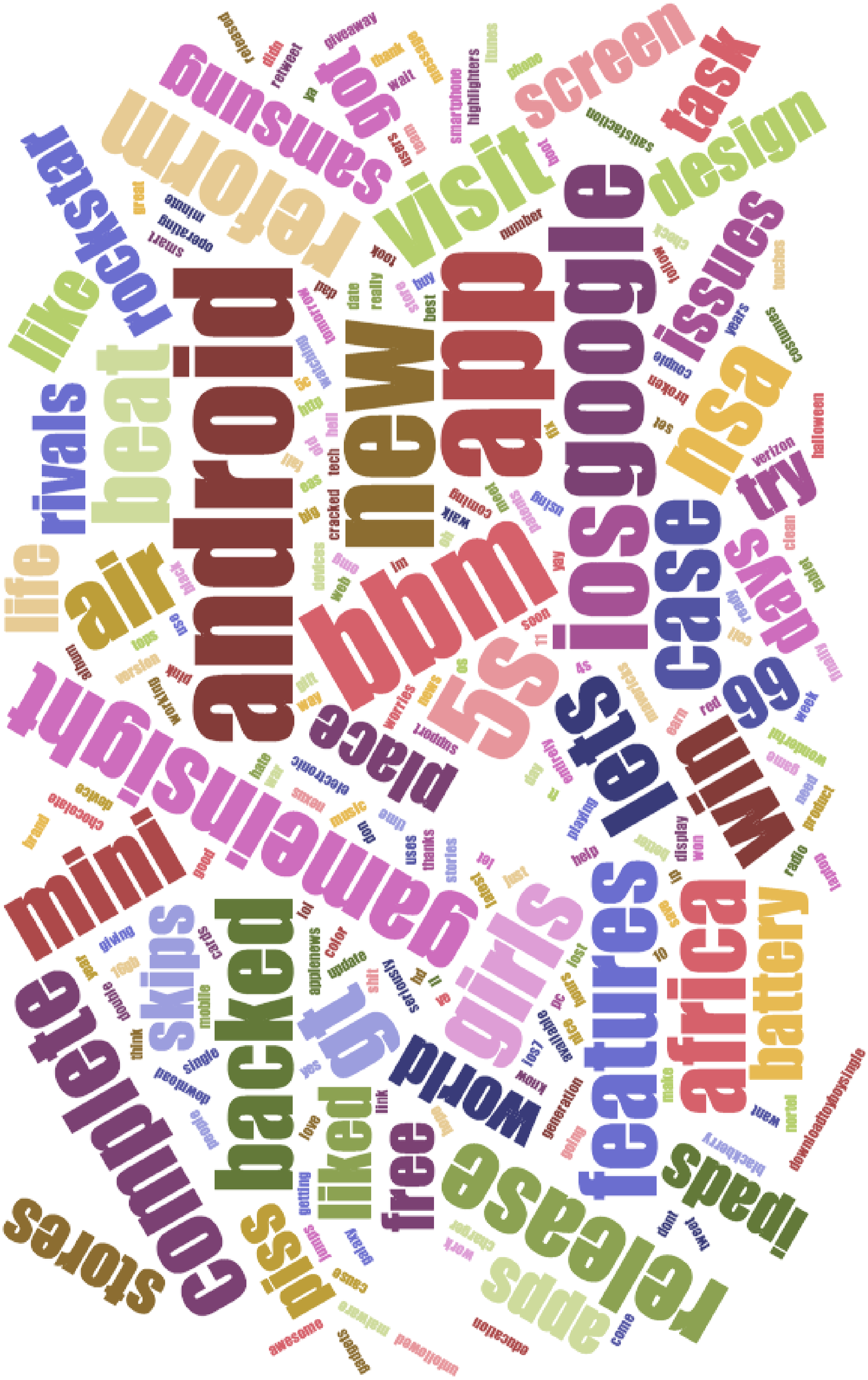


Figure 4.1: A word cloud of all tokens from all topics in our 30 topics model

To get a more detailed insight into what these words represent and what the topics represent, we analyse a few topics in detail by making a few assumptions about the topics and using the tweets to verify our assumptions.

Topic	Topic-Tokens Distribution
0	app, latest, generation, version, galaxy, won, set, oh, save, minute
1	walk, watching, unveils, today stories, beat rivals, ipads beat, missed unveils, revamped, revamped ipads, rivals
2	perfect, case, 16gb, black, gt, giving, clean, smartphone, ya, pink
3	screen, place, http, better, place visit, visit gameinsight, entirely, electronic
4	video, love, app android, yay, tomorrow, let, liked, liked video, operating, single
5	complete, follow, managed, week, having, girls, task, complete task, managed complete
6	ios, lets, bbm ios, ios lets, lets apps, features, game, tech, missed
7	google, backed, google samsung, mobile, nortel, patents, microsoft backed, rockstar, backed rockstar, uses
8	using, world, best, hate, beat, skips, skips africa, africa, africa world, release 5s
9	music, 10, mavericks, number, yes, coming, soon, earn, cards, support
10	new, app, store, available, design, playing, stargazing, stargazing app, new design
11	really, os, gift, stores, hours, news, fail, stores piss, piss, broken
12	day, battery, good, shit, display, thanks, battery life, omg, seriously, ios7
13	android, use, blackberry, update, web, 11, issues, devices, hd, fix issues
14	official, 5s, 5c, life, models, cracked, color, worries, nexus, highlighters
15	know, pc, 4s, old, does, releases, air saywhatnow, releases air, saywhatnow, did
16	time, ll, date, wonderful, charger, working, nice, message, cell, took
17	got, today, ve, going, smart, help, protector, hell, screen protector, got new
18	samsung, don, download, updated, people, radio, meet, ft, updated ios, malware jumps

19	nsa, microsoft, facebook google, google substantial, nsa surveillance, reform, reform nsa, substantial, substantial reform, surveillance
20	lol, education, chocolate, team, double, wait, boot, white girls, girls like, couple days
21	air, free, visit, power, satisfaction, 99 free, app 99, power tablet, tops, war
22	release, like, new, facebook, big, make, new 5s, gadgets, product, brand
23	just, white, app, gold, users, gold 5s, awesome, way, unfollowed, link
24	verizon, laptop, finally, say, dont, years, getting, ready, costumes, lost
25	apps, phone, check, ip, cause, thank, hope, dad, im, gt gt
26	gameinsight, halloween, try, retweet, giveaway, win, work, try gameinsight, days, tweet
27	bbm, bbm android, official release, android official, android bbm, perfect features, features bbm, want, buy, need
28	mini, win, chance, come, chance win, case mini, kickstand, kickstand case, mini models
29	itunes, released, think, year, great, touches, album, downloadtoyboysingle, didn, eas

Table 4.1: 30 topic-tokens distribution with unigrams and bigrams

4.2.1.1 Topic 6

ios, lets, bbm ios, ios lets, lets apps, features, game, tech, missed

The most common theme in the above distribution is “ios” and “lets”. “bbm” also seems to have a considerate amount of relevancy as it is the third most relevant token. The other tokens seem random but we should be able to explain them better after taking a look at some of the tweets with a fair proportion of this topic.

No	Proportion	Tweet
1	81%	Fantastical 2 for iPhone gets bold iOS 7 redesign, many new features #tech
2	75%	Limbo for iOS is now even cheaper at $0.99
3	80%	Don’t miss out on loads of great iOS game sales from…
4	70%	What new features does Apple’s iOS 7 boast?
5	63%	get the BBM on iPhone iOS, lets get the apps here now
6	89%	time guessing is hard! test yourself on ur iPhone it’s #cool tweet your results from the app #game #ios #app

Table 4.2: Tweets classified under topic 6

The tweets in Table 4.2 all seem to have at least 63% of topic 6. While they might look like carefully selected tweets, they were actually chosen at random. Our dataset contains a large fraction of the third, fourth and fifth tweet. Approximately 90% of them are retweets which explains why our topic model extracted “ios lets”, “lets apps”, “game” and “tech” as relevant tokens. The sixth tweet arguably does not really have anything to do with “iOS” but because it has been tagged with three words that our topic model finds salient, it gets tagged as having 89% proportion of topic 6.

4.2.1.2 Topic 7

google, backed, google samsung, mobile, nortel, patents, microsoft backed, rockstar, backed rockstar, uses

From a simple scan through the tokens in this topic, we can postulate that the tweets in this topic will mostly be about Google, Samsung, Nortel and the Rockstar Consortium. Nortel was a communications and networking equipment manufacturer that went bankrupt and the Rockstar Consortium was formed to negotiate licensing for patents they owned. However, we do not expect our whole dataset to be about patent war between these companies. Samsung and Google are large technology companies and we might encounter tweets that simple compare their products with that of Apple. We can make this assumption because we know our dataset is Apple-centric.

No	Proportion	Tweet
1	86%	Apple files patent for slim solar-powered technology via GigaOM
2	78%	Google, Samsung, and others sued over search patents by Apple-backed Nortel group
3	78%	Apple, Microsoft-Backed Rockstar Consortium Sues Google, Samsung Over 7
4	83%	Apple, Microsoft-backed Rockstar uses Nortel patents to sue Google, Samsung and others
5	78%	Google Fiber comes to iPhone, iPod touch with DVR functions
6	80%	Google Replacing Android ID With Advertising ID Similar To Apple's IDFA
7	68%	Nexus 4 will get the updated “in the coming weeks”. If Apple can offer to update the (almost) entire install base on day 1, why not Google?
8	88%	Google smartwatch: Will it be an “iPhone” moment for wearables?
9	84%	I wonder who's richer Google or Apple?!
10	90%	Apple earned more than Samsung, LG, Nokia, Huawei, Lenovo & Motorola's mobile shipments combined

Table 4.3: Tweets classified under topic 7

Table 4.3 is a list of 10 randomly selected tweets with a fair proportion of our topic. The first four tweets are about patents and lawsuits and our dataset contains a number of variations of those tweets, each of which have been retweeted many times. The fifth, sixth and seventh tweets all talk about products by Google while the last three tweets compare Apple's products to that of other companies.

4.2.1.3 Topic 9

music, 10, mavericks, number, yes, coming, soon, earn, cards, support

This topic, compared to previous topics, is a little tougher to analyse. "music" is th most salient token in the distribution but we also have unrelated tokens like "mavericks", "cards", "earn" and "support". While they might actually be closely related, it is not very obvious by merely looking at the topic-token distribution. To get a better understanding of this topic, we take a look at some tweets with a fair proportion of the topic.

No	Proportion	Tweet
1	63%	Just updated my mac - in loveee #Mavericks
2	83%	nahhhh, save up some cash to buy that freaking iPhone 6 thats coming out soon :)
3	77%	iPhone Battery Always Running Low? 10 Tips To Prolong The Battery Life.
4	70%	I wish I could type my mood into my iPhone and it would make a playlist for me.
5	67%	I would like to tag items on my iPhone and then be able to search tags, like in Mavericks.
6	75%	Apple needs to make a iPhone thats bigger than 64gb! My music has just about filled mine
7	70%	iPhone has to let me record videos while music playing on my phone. LET ME BE GREAT APPLE!!!
8	76%	if you have an iPhone you can block the number
9	58%	My phone just erased everybody messages number with an iPhone.
10	78%	Earn gift cards, flier miles and more with Perk - download for iphone now!

Table 4.4: Tweets classified under topic 9

Apple released a new operating system called Mavericks around the time our dataset was gathered which is what the first and fifth tweets are about in Table 4.4 and it also explains what the token "mavericks" means. The second tweet talks about the arrival of an iPhone 6 but our dataset only contains less than 10 tweets with the word "coming". Tweets 4, 6 and 7 all talk about music and our dataset contains a large amount of those type of tweets while tweets 8 and 9 talk about mobile numbers. Without having to dig deeper, it is obvious that our topic is not well formed as it is a combination of multiple topics that do not complement each other.

4.2.1.4 Topic 12

day, battery, good, shit, display, thanks, battery life, omg, seriously, ios7

The most salient token we have is “day” which does not mean much to us. Our token-distribution also seems to have a number of adverbs, adjectives and slang like “good”, “seriously” and “omg”. These all seem to represent sentiments towards a certain topic. Ignoring them, we are left with “thanks”, “battery” and “battery life”. At this point, we could postulate that most of the tweets in this category will have a “battery” theme. To confirm this, we take a look at some of the tweets with a fair proportion of the topic.

No	Proportion	Tweet
1	81%	iPhone battery just went from 23% to 3% in the space of five minutes. Thanks again, iOS7.
2	76%	iPhone battery is so crap
3	80%	iPhone battery dropping in 5% increments. Can’t be a good sign.
4	64%	Considering the amount of times I have to plug my IPhone in to charge a day it might as well be a fucking landline
5	50%	FORGOT MY IPHONE CHARGER oh shit man. My poor battery :(
6	66%	Apple was considering making an iPod for kids but apparently, the name iTouch Kids didn’t sit too well.
7	50%	so your apple store doesn’t get stock on release day?
8	50%	can I have an iphone with bbm and the battery life of a nokia please

Table 4.5: Tweets classified under topic 12

All tweets(except the 7th) in Table 4.5 refer to the iPhone battery life. A very large number of tweets that have a good proportion of this topic take in some way, the shape of tweets 1–4 in our table and fortunately, our topic model is able to find the relationship between these tweets.

The seventh tweet looks out of place as it has nothing to do with battery life but going back to our topic-token distribution, the most salient token as described by the model is “day”. Luckily, our dataset also contains a large number of that tweet and this is because that particular tweet was retweeted a lot of times.

4.2.1.5 Topic 13

android, use, blackberry, update, web, 11, issues, devices, hd, fix issues

At first glance, the main themes that stand out in the above distribution are "android", "blackberry" and "issues". Our dataset is Apple-centric so we could assume that the tweets with a large proportion of this topic will have some sort of comparison between Apple, Android and Blackberry with respect to issues that occur with their products. If this is not the case, it is possible that our topic model has merged two different topics into one topic. To verify our assumption, we analyse a few tweets that have a reasonable proportion of this topic.

No	Proportion	Tweet
1	51%	Wow hello typos... cracked iphone screen problems
2	50%	Check out WhatsApp Messenger for BlackBerry, Android, iPhone, Nokia and Windows Phone. Download it today from...
3	50%	Pandora finally comes to Chromecast via Android and iPhone apps
4	80%	I just connected with friends on #BBM. Invite your BlackBerry, Android and iPhone friends at...
5	80%	Develop iPhone, Ipad and Android Apps creatively with Mawaqaa.
6	75%	Was curious if that was the culprit. I have had tons of issues since upgrading my Apple devices to latest
7	57%	Apple testing Mail update for OS X Mavericks to fix several issues
8	78%	Manufacturing issue causing battery problems in some iPhone 5s devices

Table 4.6: Tweets classified under topic 13

Tweets 2–5 in Table 4.6 all have either an android or blackberry theme in them which is expected. They all talk about applications on all three platforms. On the other hand, tweets 1, 6, 7 and 8 all have an "issues" theme in them. Unfortunately, these two topics do not really complement each other and should arguably be splitted into two separate topics. Some of the latter mentioned tweets do also talk about applications on Apple's platforms which may be the reason why our topic model observed a relationship between these topics.

4.2.1.6 Topic 14

official, 5s, 5c, life, models, cracked, color, worries, nexus, highlighters

An initial scan of our token distribution above does not tell us that this topic might be mostly about the 5s and 5c. In September 2013, a month before our data set was gathered, Apple released two models of its mobile phone and they were called iPhone 5S and iPhone 5C. With this knowledge, we could hypothesise that tweets with a large proportion of this topic will mostly be about these new phones. We can also rely on the fact that our distribution contains tokens like "5s" and "5c". Tokens like "models", "cracked" and "worries" could also be used to describe a state of the phones.

No	Proportion	Tweet
1	88%	Cracked iPhone No worries. Color it in with highlighters!
2	68%	Apple discovers manufacturing defect causing iPhone 5S battery woes for some customers
3	88%	#Apple Admits Defect with Some #iPhone 5s Batteries
4	33%	Everyone who bought the iPhone 5S or iPhone 5C is dumb. The iPhone 6 has already been announced lol.
5	76%	iPhone 5S, 5C debut in India today - Customers get the new models in the price range of Rs 41,900 to Rs 71,500
6	88%	$AAPL Apple's yellow iPhone 5C is a lemon
7	51%	Well the battery life on the iPhone 5c is great... I need a charger in every room
8	22%	What Are The Most Popular iPhone 5s and 5c Colors? Space Gray And Blue.

Table 4.7: Tweets classified under topic 14

The first tweet in Table 4.7 does not really refer to the iphone 5S or 5C but is genrally about the iPhone which is acceptable. Tweets 2–8 however are all about either the 5S or 5C. They each also use terms like "color" and "models" to describe the phone in some way. Unfortunately, our topic model has tagged the last tweet with only 22% of our topic which is unexpected because the tweet does actually talk about both the 5S and 5C models.

4.2.1.7 Topic 22

release, like, new, facebook, big, make, new 5s, gadgets, product, brand

After a first scan, it is not very clear what this topic is really about. Our most salient token is “release” and in combination with other tokens like “new”, “new 5s” and “product”. We could assume that this topic could be mainly about the new release of devices. However, we also have tokens like “facebook”, “like”, “big” and “make” which we cannot really explain. We could have a mixture of topics or just a poorly formed one.

No	Proportion	Tweet
1	75%	Retina Display iPad mini 2 Release Date Tipped By Target’s Online Product Page
2	74%	Ubisoft Releases “Rabbids Big Bang” for iOS!
3	75%	Do you fancy a brand new #iPhone 5s? Like the #busuu Facebook page for your chance to #win!
4	67%	I’m not a big fan of the screen ratio of the iPad mini. Also the Nexus 5 choice is easy now, all gone.
5	84%	my iphone fell while I was tweeting and I stepped on it then i heard a big crack, I paused for 5 mins and prayed it was ok
6	80%	I have a major craving to make a Loki/Sigyn video. Blah, stupid dead back light on my Mac.
7	79%	like my #OpenTouch #OTC #iPhone #App from @ALUEnterprise to see the phone presence before I make a call
8	70%	iBoobies case for Apple iPhone 4 - make your phone even sexier

Table 4.8: Tweets classified under topic 22

From Table 4.8, the first two tweets do refer to the release of a product and mobile aplication, respectively. Unfortunately, other tweets have nothing to do with products/application release. The third tweet is a promotional tweet, the fourth is about devices and its features, the fifth and sixth refer to features of the iPhone and Mac(Apple’s laptop). Without going any further, it is fairly clear that these tweets are not related. It is possible that our topic model has incorrectly merged multiple topics.

4.2.1.8 Topic 27

bbm, bbm android, official release, android official, android bbm, perfect features, features bbm, want, buy, need

At first glance, we could say this topic has a lot to do with *android*, *bbm*, and *features*. The last three tokens also seem out of place. At the time of data gathering, there was a lot of chatter on social media about the BlackBerry Messenger(BBM) application coming to the iOS and android platform. To be certain of this, let's take a look at some of the tweets that have a fair proportion of this topic.

No	Proportion	Tweet
1	56%	More perfect features, BBM android, BBM iPhone
2	50%	BBM on android and iPhone, official release - get it here
3	50%	BBM Now on Android and iPhone.
4	72%	Just got bbm chat for the iPhone, feel free to add me if you want :)
5	60%	Apple should create the option of removing yourself from a group chat
6	68%	Anyone want to buy a black 64GB ipad 2 from me in excellent condition?
7	25%	someone buy me an iphone ugh
8	76%	I need a iphone 5 asap

Table 4.9: Tweets classified under topic 27

Table 4.9 is a list of tweets that fall under topic 27. We can see that the first four tweets have a lot to do with the new BBM for iOS and Android. The fifth tweet is a little tricky as it says nothing about BBM. However, it does in fact talk about a chat application which is what BBM is. While the user might not have been referring to BBM in particular, our model was able to pickup on the relationship between both subjects.

The last three tweets in our table explain the last three tokens in out topic-word distribution for topic 27. This means that topic 27 is actually a combination of two different topics.

4.2.1.9 Topic 28

mini, win, chance, come, chance win, case mini, kickstand, kickstand case, mini models

There are a lot of promotions/giveaways on Twitter that involve Apple products. Tokens like “win”, “chance” and “chance win” in our distribution tell us that this topic is about these promotions. Our dataset is Apple-centric so we could hypothesise that tweets with a large proportion of this topic might be offering users a chance to win Apple products like the iPad/Mac Mini, hence the “mini” in our distribution. We also have “case” occurring in the distribution which might refer to iPad cases up for promotion. To confirm that our hypothesis is valid, or not, we analyse some tweets with a fair proportion of this topic.

No	Proportion	Tweet
1	75%	Win an $800 Mac Mini for FREE from MacTrast the perfect addition to any home or office!
2	82%	RT to WIN! - #Win an iPad Mini to celebrate the start of #50at50
3	56%	15,000 Facebook Fan Giveaway happening now - Win a Lens, iPad Mini, $500 Amazon gift card and LOTS more! #colorvale15k
4	94%	WIN an iPad mini plus a chance to win a Williamson Tea Elephant Tea Caddie
5	67%	Win an #ipad follow and RT for a chance to win
6	66%	Targus Kickstand Case for Apple iPad Mini all models - Red
7	86%	Cooper Dynamo Apple iPad Mini Kids Play Case review
8	78%	Fab Purse Moschino IPhone Cases. Come in Lots of Colours
9	50%	Retina iPad Mini may be launched Nov. 21
10	20%	iMore – iMore show 373: iPad Air and Retina iPad mini buyers guide

Table 4.10: Tweets classified under topic 28

From Table 4.10, we can tell that Tweets 1–5 are all promotional tweets offering users a chance to win Apple products like the iPad mini. 60% of the tweets with a fair proportion of this topic are all variations of those tweets. Tweets 6–8 all refer to iPad and iPhone cases and contrary to what we previously assumed, these cases are not part of the promotion. This means that our topic model has merged two topics that do not complement each other. The last two tweets also have nothing to do with the promotions as well as the cases. These tweets are general tweets about the iPad mini. Fortunately, our topic model has tagged them with low proportions(50% and 20% respectively) which is acceptable.

4.2.2 Evaluating 40 Topics

In Section 4.2.1, we analysed a topic model that generated 30 topics from our dataset. In this section, we analyse a model that generates 40 topics from our dataset. We use the same dataset for both models so we expect a few overlapping topics. We briefly look at these topics and then we look at a few new topics.

Table 4.11 on page 36 shows a list of 40 topics and their respective token distribution. This table can be used to analyse all topics abstractly but to have a general overview of all tokens in our table, we use a word cloud as shown in Figure 4.2 on page 37. In our cloud, we see prominent words like “app”, “bbm”, “google”, “device” amongst others which also appears in our 30 topic word cloud on page 22.

Most of the tokens used in the 40 topics model also appear in the 30 topics model and as a result, it is difficult to draw a sane comparison between these models from either the word clouds or tables. For this reason, we take a more detailed look at each topic and it’s token distribution. We attempt to find similar topics in the 30 topics model and analyse some new topics.

Topic	Topic-Words associations
0	big, year, chocolate, mobile, gadgets, double, boot, girls like, white girls, touches
1	app, store, know, available, does, design, stargazing, stargazing app, ft, new design
2	gift, version, 25, liked, liked video, months, earn, watch, gift cards, knew
3	follow, ip, place, walk, laptop, electronic, web, dont, competition, costumes
4	using, hate, tweet, using app, reason, bout, thing, unfollowed using, case like, girl
5	white, education, giveaway, tablet, comes, brand, brand new, way, halloween giveaway, win plink
6	visit, place visit, visit gameinsight, ipads, place, unveils, power, rivals, stories case
7	video, download, 16gb, black, watching, nowplaying, clean, smartphone, eyes, sprint

8	great, giving, yay, miss, post, feeling, turns, maps, keyboard, canine evil
9	number, tfbjp teamfairyrose, retweet tfbjp, sougofollow, teamfairyrose, autofollow, 06, cards, 90sbabyfollowtrain, interesting
10	ios, official, 5s, like, world, game, facebook, new 5s, cool, africa
11	new, gameinsight, try, updated, try gameinsight, achievement, new achievement, jobs, achievement 10, areas
12	android, need, use, blackberry, devices, hd, problems, card, art, issue
13	bbm, bbm android, official release, android official, got, want, gold, macbook, gold 5s, smart
14	apps, lets, bbm ios, ios lets, lets apps, phone, don, best, people, meet
15	missed, having, app android, case missed, windows, 12, daily, cell, apps android, decided
16	perfect, android bbm, perfect features, features bbm, love, collection, pink, did, let, website
17	samsung, google samsung, backed, entirely, nortel, patents, share, beat, uses, really entirely
18	mini, win, chance, chance win, models red, kickstand case, mini models, targus, targus kickstand, case mini
19	air, http, better, young, news, mirror, market, mirror world, souvenirs mirror, envy complete
20	retina, device, red, electronic device, device using, ur, inch, 13, fix, wallet
21	pc, life, old, releases, saywhatnow, releases air, air saywhatnow, cases, away, battery life
22	check, music, 10, mavericks, generation, os mavericks, ready, fix issues, mail, mail update
23	case, buy, cover, 99, end, 2013, case 4s, case cover, smart cover, cover case
24	haha, isn, line, allowed, oh, solar, price, android phone, guy, basically
25	app, just, radio, updated ios, users, young hitta, yhr app, yhr, radio yhr, hitta radio
26	os, stores, playing, latest, hours, fail, piss, stores piss, issues, protector

27	release, halloween, 5c, work, models, days, online, product, color, pics
28	battery, lol, good, shit, thanks, tomorrow, ios7, minutes, minute, november
29	itunes, released, think, thank, album, downloadtoyboysingle, itunes link, downloadtoyboysingle itunes, saw, sister
31	ll, charger, didn, personal, later, nice, brother, movies, butt, butt away
32	features, tech, make, cracked, worries, color highlighters, cracked worries, highlighters, worries color, say
33	complete, 4s, , managed, week, display, girls, task, managed complete, complete task
34	free, enter, win, 99 free, app 99, iphone5s, navigation, comp, fucking, network
35	really, update, broken, damn, 11, family, seriously, point, message, took
36	time, retweet, date, wonderful, hell, edtech, learn, screwed, winners chosen, chosen tonight
36	today, ve, going, verizon, couple, couple days, team couple, dressed, came, online store
37	screen, finally, look, years, getting, weekend, remote, siri, finally got, ad redesigned
38	google, nsa, microsoft, reform, reform nsa, substantial, substantial reform, surveillance, facebook google, nsa surveillance
39	day, easy, link, older, candy, browser, os browser, otterbox, defender, otterbox defender

Table 4.11: 40 word-topic distributions with unigrams and bigrams

Figure 4.2: A word cloud of all tokens from all topics in our 40 topics model

No	30 topics id	40 topics id
1	6	14
2	7	17
3	9	22
4	12	28
5	19	38
6	27	13
7	28	18

Table 4.12: List of similar topics from our 30 topics and 40 topics model

4.2.2.1 Analysing similar topics

As previously mentioned, we expect our 40 topics model to have some similar topics with the 30 topics model. Fortunately, there are a few similar topics and Table 4.12 is a list of such topics. Each row contains a topic id from our 30 topics model and its corresponding similar topic from our 40 topics model. The ids correspond to the "Topic" attribute on Tables 4.1 and 4.11. Both tables can also be found on Pages 24 and 36 respectively.

In other to confirm that these topics are actually similar, we analyse the token distribution for some of the topics and also compare tweets with a fair proportion of the respective topics. Specifically, we analyse rows 2 and 6 from Table 4.12

Row 2

1 → *google, backed, google samsung, mobile, nortel, patents, microsoft backed, rockstar, backed rockstar, uses*
2 → *samsung, google samsung, backed, entirely, nortel, patents, share, beat, uses, really entirely*

We can infer that both distributions above are trying to explain the same topic because they both use tokens like "google", "samsung", "patents", and "nortel" to explain their topics. With this knowledge, we can also expect our 40 topics model should have tagged the same tweets with this topic.

No	Proportion	Tweet
1	76%	Apple Continues To Lose Tablet Market Share But Should Rebound With The Air
2	72%	iPad market share dips to less than 30% while tablet market grew 7% overall
3	57%	would you suggest the IPad or galaxy note 10? What do you like best?
4	77%	Apple, Microsoft-backed Rockstar uses Nortel patents to sue Google, Samsung and others
5	77%	Google, Samsung, and others sued over search patents by Apple-backed Nortel group
6	80%	Apple, Microsoft-Backed Rockstar Consortium Sues Google, Samsung Over 7

Table 4.13: Tweets classified under topic 17(40 topics model)

Comparing Table 4.13 above and Table 4.3 on page 26, we can see that both models have very similar tweets. The former seems to have a combination of two topics and so does the latter. Section 4.2.1.2 gives a more detailed explanation of what the common tweets represent.

Row 6

1 → *bbm, bbm android, official release, android official, android bbm, perfect features, features bbm, want, buy, need*

2 → *bbm, bbm android, official release, android official, got, want, gold, macbook, gold 5s, smart*

After an initial scan, we could infer that the above distributions are both trying to explain the same topic. They both have their first four salient topics in common, in the same order. This increases the chances of our assumption being valid. To confirm the validity of our assumption, we analyse the tweets categorised under this topic by our 40 topics model.

No	Proportion	Tweet
1	21%	More perfect features, BBM android, BBM iPhone
2	50%	BBM on android and iPhone, official release - get it here
3	65%	I just connected with friends on #BBM. Invite your BlackBerry, Android and iPhone friends
4	47%	BBM. Now on Android and iPhone.
5	82%	Just got bbm chat for the iPhone, feel free to add me if you want
6	70%	i want an iPhone 5
7	32%	white girls be like: i want a iPhone 5c with THAT case!!
8	60%	I want the gold iPhone 5s so bad

Table 4.14: Tweets classified under topic 13(40 topics model)

Comparing Table 4.14 above and Table 4.9 on page 32, we can see that both models have very similar tweets. It also tells us that the 40 topics model merged two topics as our 30 topics model did. Unfortunately, the second topics are not similar but this is acceptable because both models are slightly different which makes a 100% similarity highly unlikely. Section 4.2.1.8 gives a more detailed explanation of what the common tweets represent.

Chapter 5

Conclusion

5.1 Summary of Report Achievements

In Chapter 3, we were able to create a model to classify tweets into relevant and non relevant groups. We learnt that accuracy is not a good measure for a classifier's performance. As a result we decided to use the Area Under the Receiver Operating Characteristic Curve(AUC) as an evaluation metric. We also used k-fold cross validation to evaluate the performance of our classifier on an unseen dataset. The average AUC of our initial(Section 3.2.3) and best model(Section 3.2.4.2) were 0.74 and 0.85, respectively. This means we achieved a 13.8% increase in AUC.

In other to create our initial training set, we used a web application which displayed a number of tweets and options to classify them into relevant and irrelevant groups(Figure 3.1). The application was created for this project but can be used in other similar projects. The only requirement is that tweets are stored in a MongoDB database. It can also be extended easily to support extra features if required.

In Chapter 4, we created two topic models, one with 30 topics and the other with 40 topics. We empirically analysed and evaluated some of the topics generated by the 30 topics model. We discovered that the topic model was able to correctly place the right tweets under the right topics. We also learnt that LDA can sometimes merge two or three topics into one(Section 4.2.1.3, Section 4.2.1.5, Section 4.2.1.8).

Rather than re-evaluating the topics generated by the 40 topics model, we compared them to that generated by the 30 topics model. This is in a effort to find out if there will be overlapping topics considering the same dataset was used. Fortunately, we were able to find a few overlapping topics(Table 4.12).

5.2 Applications

As discussed in Chapter 1, companies who are looking to get customers' feedback without putting any pressure on the customers can use our research. They could gather data from multiple social platforms, train a classifier to filter out irrelevant tweets and then run topic modelling on the data. This project itself uses one of these companies as a case study.

Aside from companies looking to get feedback, sporting bodies like the Football Association in England or the International Olympics Committee(IOC) could also benefit from this research. For instance, during the London 2012 Summer Olympics, Twitter claimed that 9.66 million tweets were sent during the opening ceremony[1]. This is a lot of data that could be useful in planning similar future events. These tweets could be passed through a topic modelling system and the main themes discussed during the ceremony will be detected. The output can then be used to improve the next Olympics.

5.3 Future Work

We chose the naive Bayes classifier for this project mainly because of its simplicity. It would be interesting to find out how a Support Vector Machine will perform on our dataset compared to our Bayes classifier.

Usually, detecting topics in a dataset is only a step to achieving other goals. One interesting experiment would be to find a way of knowing the overall sentiments of a topic. Analysing the sentiment of a single tweet might be easy but analysing the sentiments of a topic might prove more difficult. This is because our topic model does not categorize a tweet under just one topic but rather, each tweet is made up of varying proportions of different topics.

[1] `https://blog.twitter.com/en-gb/2012/today-on-twitter-14`

Appendices

Appendix A

Model Pre-processing

This appendix contains more detailed information about the pre-processing tasks done in this study.

Regular Expressions

The regular expression used to find links in our text is:

```
{(https?:\/\/)?([\da-z\.-]+)\.([a-z\.]{2,6})([\/\w \.-]*)*\/?}
```

The regular expression for finding twitter user names is:

```
@[a-zA-Z0-9]+ ?
```

Stop Words

A complete list of stopwords used can be found at `http://ir.dcs.gla.ac.uk/resources/linguistic_utils/stop_words`

Extra Stop Words

Extra stop words used for topic modelling pre-processing are: *iphone*, *ipod*, *ipad*, *mac*, *imac*, *apple* and *rt*

Appendix B

Topics Evaluation

The tables in this appendix contain the re-evaluation of Section 4.2.1.1 to Section 4.2.1.9 by volunteers. They were asked to read the token distribution for each of the topics discussed in those sections and rate, on a scale of 1 to 10, how well the distribution explains the tweets in the topic's category.

Topic	Rating
6	5
7	3
9	8
12	10
13	5
14	5
22	10
27	10
28	10

Topic	Rating
6	4
7	4
9	8
12	10
13	5
14	6
22	10
27	10
28	10

Topic	Rating
6	6
7	8
9	3
12	4
13	4
14	8
22	3
27	6
28	10

Topic	Rating
6	6
7	5
9	7
12	9
13	7
14	7
22	8
27	10
28	10

Appendix C

Similar Topics Evaluation

The tables in this appendix contains ratings for the similar topics analysed in Section 4.2.2.1. They were asked to rate, on a scale of 1 to 10, how similar the topics in Table 4.12 are.

30 topics id	40 topics id	Rating
6	14	
7	17	
9	22	
12	28	
19	38	
27	13	
28	18	

30 topics id	40 topics id	Rating
6	14	
7	17	
9	22	
12	28	
19	38	
27	13	
28	18	

30 topics id	40 topics id	Rating
6	14	
7	17	
9	22	
12	28	
19	38	
27	13	
28	18	

30 topics id	40 topics id	Rating
6	14	
7	17	
9	22	
12	28	
19	38	
27	13	
28	18	

Appendix D

Tools and Implementation

The dataset was gathered with a **Node.js** script[1]. The script watched Twitter's public stream and saved tweets matching a defined pattern to a **MongoDB** database[2]. The format of tweet objects gotten from Twitter is the same for a document in MongoDB. MongoDB has also been proven to be fast and reliable which is why it was selected as our database of choice.

The data labelling application was built with **Express.js**. Express is a web application framework built on Node.js. It provides a robust set of features for building web applications.

For data classification, we used scikit-learn[3] (Pedregosa *et al.* , 2011). scikit-learn is a machine learning library built in the Python programming language. It provides algorithms for Classification, Regression, Clustering Dimensionality reduction and Model Selection. We used its naive Bayes classifier to filter out irrelevant tweets. We also used its Model Selection module to run grid search in Section 3.2.4.2.

For topic modelling, we used gensim[4] (Řehůřek & Sojka, 2010). From its website, gensim is "topic modelling for humans". It provides modules for Latent Dirichlet Allocation(LDA) and others[5]. Our initial attempt at detecting topic with was done with MALLET[6] (McCallum, 2002). The main reasons for selecting gensim over MALLET are:

[1] Node.js is a platform used for building fast and scalable network applications

[2] MongoDB is a NoSQL database with a document-oriented architecture

[3] `http://scikit-learn.org/stable/index.html`

[4] `http://radimrehurek.com/gensim/`

[5] `http://radimrehurek.com/gensim/apiref.html`

[6] a Machine Learning for Language Toolkit. See `http://mallet.cs.umass.edu/`

1. It provides support for online LDA(Section 2.3.2)
2. It is very flexible as opposed to MALLET which is highly optimised.
3. It provides a python API which is simpler than the Java API provided by MALLET.

The plots in Chapter 3 were drawn with matplotlib[7], a 2D plotting library (Hunter, 2007) and the word clouds in Chapter 4 were generated by Jason Davies's word cloud generator[8].

[7] `http://matplotlib.org/`

[8] `https://www.jasondavies.com/wordcloud`

References

Androutsopoulos, Ion, Paliouras, Georgios, Karkaletsis, Vangelis, Sakkis, Georgios, Spyropoulos, Constantine D, & Stamatopoulos, Panagiotis. 2000. Learning to filter spam e-mail: A comparison of a naive bayesian and a memory-based approach. *arXiv preprint cs/0009009*.

Asuncion, Arthur, Welling, Max, Smyth, Padhraic, & Teh, Yee Whye. 2009. On smoothing and inference for topic models. *Pages 27–34 of: Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*. AUAI Press.

Bhardwaj, Anurag, Malgireddy, Manavender, Setlur, Srirangaraj, Govindaraju, Venu, & Ramachandrula, S. 2009. Writer identification in offline handwriting using topic models. *In: Proceedings of the NIPS 2009 Workshop on Applications of Topic Models: Text and Beyond.*

Blake, Catherine L, & Merz, Christopher J. 1998. UCI Repository of machine learning databases [http://www.ics.uci.edu/˜mlearn/MLRepository.html]. Irvine, CA: University of California. *Department of Information and Computer Science*, **460**.

Blei, David M, Ng, Andrew Y, & Jordan, Michael I. 2003. Latent dirichlet allocation. *the Journal of machine Learning research*, **3**, 993–1022.

Bradley, Andrew P. 1997. The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern recognition*, **30**(7), 1145–1159.

Deerwester, Scott C., Dumais, Susan T, Landauer, Thomas K., Furnas, George W., & Harshman, Richard A. 1990. Indexing by latent semantic analysis. *JASIS*, **41**(6), 391–407.

Deshpande, Vikas P, Erbacher, Robert F, & Harris, Chris. 2007. An evaluation of Naive Bayesian anti-spam filtering techniques. *Pages 333–340 of: Proceedings of the 2007 IEEE Workshop on Information Assurance United States Military Academy, West Point, NY.*

Doyle, Gabriel, & Elkan, Charles. 2009. Financial topic models. *In: NIPS 2009 Workshop on Applications of Topic Models: Text and Beyond.*

Dumais, Susan T. 2004. Latent semantic analysis. *Annual review of information science and technology*, **38**(1), 188–230.

Friedl, Jeffrey. 2006. *Mastering regular expressions.* O'Reilly Media, Inc.

Gerrish, Sean, & Blei, David. 2009. Modeling Influence in Text Corpora.

Gethers, Malcom, & Poshyvanyk, Denys. 2010. Using relational topic models to capture coupling among classes in object-oriented software systems. *Pages 1–10 of: Software Maintenance (ICSM), 2010 IEEE International Conference on.* IEEE.

Hayes, Bob E. 2008. *Measuring Customer Satisfaction and Loyalty: Survey Design, use and Statistical analysis Methods.* Third edn. American Society for Quality Press.

Hoffman, Matthew D, Blei, David M, & Bach, Francis R. 2010. Online Learning for Latent Dirichlet Allocation. *Page 5 of: NIPS*, vol. 2.

Hofmann, Thomas. 1999. Probabilistic latent semantic indexing. *Pages 50–57 of: Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval.* ACM.

Hospedales, Timothy, Gong, Shaogang, & Xiang, Tao. 2009. A markov clustering topic model for mining behaviour in video. *Pages 1165–1172 of: Computer Vision, 2009 IEEE 12th International Conference on.* IEEE.

Huang, Jin, & Ling, Charles X. 2005. Using AUC and accuracy in evaluating learning algorithms. *Knowledge and Data Engineering, IEEE Transactions on*, **17**(3), 299–310.

Huang, Jin, Lu, Jingjing, & Ling, Charles X. 2003. Comparing naive Bayes, decision trees, and SVM with AUC and accuracy. *Pages 553–556 of: Data Mining, 2003. ICDM 2003. Third IEEE International Conference on.* IEEE.

Hunter, J. D. 2007. Matplotlib: A 2D graphics environment. *Computing In Science & Engineering*, **9**(3), 90–95. `http://matplotlib.org/`.

Java, Akshay, Song, Xiaodan, Finin, Tim, & Tseng, Belle. 2007. Why we twitter: understanding microblogging usage and communities. *Pages 56–65 of: Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 workshop on Web mining and social network analysis*. ACM.

Kireyev, Kirill, Palen, Leysia, & Anderson, K. 2009. Applications of topics models to analysis of disaster-related twitter data. *In: NIPS Workshop on Applications for Topic Models: Text and Beyond*, vol. 1.

Landauer, Thomas K. 2006. Latent semantic analysis. *Encyclopedia of Cognitive Science*.

Langley, Pat, Iba, Wayne, & Thompson, Kevin. 1992. An analysis of Bayesian classifiers. *Pages 223–228 of: AAAI*, vol. 90.

Lau, Jey Han, Collier, Nigel, & Baldwin, Timothy. 2012. On-line Trend Analysis with Topic Models:\# twitter Trends Detection Topic Model Online.

Ling, Charles X, Huang, Jin, & Zhang, Harry. 2003. AUC: a better measure than accuracy in comparing learning algorithms. *Advances in Artificial Intelligence*, 329–341.

Linstead, Erik, Hughes, Lindsey, Lopes, Cristina, & Baldi, Pierre. 2009. Software analysis with unsupervised topic models. *Page 52 of: NIPS Workshop on Application of Topic Models: Text and Beyond*, vol. 50.

Manning, Christopher D, Raghavan, Prabhakar, & Schütze, Hinrich. 2008. *Introduction to information retrieval*. Vol. 1. Cambridge University Press Cambridge.

McCallum, Andrew Kachites. 2002. *MALLET: A Machine Learning for Language Toolkit*. http://mallet.cs.umass.edu.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, **12**, 2825–2830.

Řehůřek, Radim, & Sojka, Petr. 2010. Software Framework for Topic Modelling with Large Corpora. *Pages 45–50 of: Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*. Valletta, Malta: ELRA. http://is.muni.cz/publication/884893/en.

Rubin, Donald B. 1987. *Multiple Imputation for Nonresponse in Surveys*. John Wiley & Sons.

Sivic, Josef, & Zisserman, Andrew. 2009. Efficient visual search of videos cast as text retrieval. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, **31**(4), 591–606.

Smaragdis, Paris, Shashanka, Madhusudana, & Raj, Bhiksha. 2009. Topic Models for Audio Mixture Analysis. *Applications for Topic Models: Text and Beyond, Whistler.*

Weng, Jianshu, Lim, Ee-Peng, Jiang, Jing, & He, Qi. 2010. Twitterrank: finding topic-sensitive influential twitterers. *Pages 261–270 of: Proceedings of the third ACM international conference on Web search and data mining.* ACM.

Wilbur, W John, & Sirotkin, Karl. 1992. The automatic identification of stop words. *Journal of information science*, **18**(1), 45–55.

Zhang, Harry. 2004. The optimality of naive Bayes. *A A*, **1**(2), 3.

Zhao, Wayne Xin, Jiang, Jing, Weng, Jianshu, He, Jing, Lim, Ee-Peng, Yan, Hongfei, & Li, Xiaoming. 2011. Comparing twitter and traditional media using topic models. *Pages 338–349 of: Advances in Information Retrieval.* Springer.