SHORT TEXT CLASSIFICATION IN TWITTER TO IMPROVE INFORMATION FILTERING

THESIS

Presented in Partial Fulfillment of the Requirements for the Degree Master of Science in the Graduate School of The Ohio State University

By

Bharath Sriram

Graduate Program in Computer Science and Engineering

The Ohio State University

2010

Master's Examination Committee:

Dr. Hakan Ferhatosmanoglu, Advisor

Dr. Gagan Agrawal

Abstract

In micro-blogging services such as Twitter, the users may get overwhelmed by the raw data. One solution to this problem is the classification of Twitter messages (tweets). As short texts like tweets do not provide sufficient word occurrences, classification methods that use traditional approaches such as "Bag-Of-Words" have limitations. To address this problem, we propose to use a small set of domain-specific features extracted from the author's profile and text. The proposed approach effectively classifies the text to a predefined set of generic classes such as News, Events, Opinions, Deals, and Private Messages.

Existing works on classification of short text messages integrate every message with meta-information from external information sources such as Wikipedia and WordNet. Automatic text classification and hidden topic extraction approaches perform well when there is meta-information or the context of the short text is extended with knowledge extracted using large collections. But these approaches require online querying which is time consuming and unfit for real time applications. When external features from the world knowledge is used to enhance the feature set, complex algorithms are required to carefully prune overzealous features. These approaches eliminate the

problem of data sparseness but create a new problem of the curse of dimensionality [38]. Hence efficient ways are required to improve the accuracy of classification by using minimal set of features to represent the short text.

We propose an intuitive approach to determine the class labels and the set of features with a focus on user intentions on Twitter such as daily chatter, conversations, sharing information/URLs. We classify incoming tweets into five generic categories – news, opinions, deals, events and private messages. We believe that these categories are diverse and cover most of the topics that people usually tweet about. Experimental results using our proposed technique outperform the baseline Bag-Of-Words model in terms of accuracy and speed.

Next, we allow users to add new categories based on their interest. Since the accuracy is bound to deteriorate with increase in number of classes, we also allow users to add new features corresponding to the new classes. Our system takes in sampled tweets from the user and diagnoses the feature. Experimental results show that with our small feature set and the user-defined features, the classification accuracy is better than the Bag-Of-Words model.

This work is dedicated to Appa and Amma, for their unconditional support

Acknowledgments

In the journey that led to the creation of the work presented in this thesis, I have been guided, helped and supported by many. This is my opportunity to thank them for everything they have done for me.

First and foremost, I am deeply indebted to my advisor Prof. Hakan Ferhatosmanoglu, for teaching me the principles of good research. His dedication and enthusiasm towards research is truly inspirational. He has been an excellent mentor and guide throughout my Master's program. I would also like to thank Prof. Gagan Agrawal for agreeing to be on my Master's Thesis Committee and for all the valuable suggestions he has made.

Special thanks to Dr Engin Demir and David Fuhry for helping me through all stages of my research. I am forever obliged for their immense patience and valuable insights. I will always look back on our fierce brain-storming sessions with great pride.

Thanks to the entire team at the Database Research lab, Amit, Onur, Rohit, Shakil and previous members Dr Ahmet Sacan, Dr Guadalupe Canahuate and Yogesh for their constant support and making my stay a memorable experience. This acknowledgement would be incomplete without thanking my friends Ashwin, Arun, Archana, Ajay, Harsha, Reetam, Shilpa, Kavitha, Kedar and Vishnu. To my cousins Cheeni, Shashi, Shanthi, Deepak, Varsha, Lakshmi, Karthik, Neeta, Aditya, Raghu, Abhejit, Abhishek, Naveen, Achyuth and all my family members, I am forever grateful.

I cannot put into words the gratitude I owe to my parents. They have constantly encouraged me to excel in whatever I do and always had staunch faith in me. All that I am, I owe to my parents.

Special thanks to my grandmother, Alamelu athe, Nalini, Prabha aunty, Giri mama, Vijay Mandayam, N.K. Srinath for all their support and encouragement.

To my motherland, India and her countrymen, I am thankful for providing a solid foundation and teaching me the true spirit of "Unity in Diversity" which helped me adapt to a foreign culture with ease. I would like to quote a popular Sanskrit verse "जननी जन्म भूमिश्च स्वर्गादपि गरीयसी" (*Janani Janma-bhoomi-scha Swargadapi Gariyasi*) which translates to "Mother and Motherland are superior to the Heaven".

Finally, I thank the almighty for giving me the courage and patience to come this far and complete the thesis successfully.

Vita

May 31, 1985……..Born – Bangalore, India

2003 – 2007…….B.E. Information Science,
Rashtreeya Vidyalaya College of Engineering,
Bangalore, India

2007 – 2008……Software Engineer, IBM India Software Labs

2008 - 2009 ……Graduate Administrative Associate,
Office of Financial Services,
The Ohio State University

2009 - 2010 ……Graduate Research Associate,
Dept of Computer Science,
The Ohio State University

Fields of Study

Major Field: Computer Science and Engineering

Table of Contents

List of Figures

Chapter 1: Introduction to Text Classification

1.1 Classification

Classification is a supervised data mining technique that involves assigning a label to a set of unlabeled input objects. Based on the number of classes present, there are two types of classification:

- Binary classification – classify input objects into one of the *two* classes.
- Multi-class classification – classify input objects into one of the *multiple* classes.

Unlike a better understood problem of binary classification, which requires discerning between the two given classes, the multiclass classification is a more complex and less researched problem [18].

1.2 Text Classification

Text classification is an area where classification algorithms are applied on documents of text. The task is to assign a document into one (or more) classes, based on its content. Typically, these classes are handpicked by humans. For example, consider the task to classify set of documents (say, each 1 page long) as good or bad. In this case, categories

(or labels) “good” and “bad” represent the classes. The input objects are the 1-page long documents.

Some of the popular areas where text classification is applied are as follows [33]:

- Classify news as Politics, Sports, World, Business, Lifestyle
- Classify email as Spam, Other.
- Classify Research papers by conference type.
- Classify movie reviews as good, bad and neutral.
- Classify jokes as Funny, Not Funny.

For a classifier to learn how to classify the documents, it needs some kind of ground truth. For this purpose, the input objects are divided into training and testing data. Training data sets are those where the documents are already labeled. Testing data sets are those where the documents are unlabeled. The goal is to learn the knowledge from the already labeled training data and apply this on the testing data and predict the class label for the test data set accurately. Hence, the classifier is built of a *Learner* and an actual *Classifier*. The learner is responsible for learning a classification function (F) that maps the documents (d) to the classes (C), i.e:

$$F: d \rightarrow C$$

The classifier then uses this classification function to classify the unlabeled set of documents. This type of learning is called *supervised learning* because a supervisor (the

human who defines the classes and labels training documents) serves as a teacher directing the learning process [19].

The choice of the size of the training and testing data set is very important. If the classifier is fed with a small number of documents to train from, it may not acquire substantial knowledge to classify the test data correctly. On the other hand, if the training data is too large compared to the test data, it leads to a problem called "Overfitting". In that case, the document is too finely tuned with respect to the training data, so much so that its performance degrades on the unseen test data.

1.2.1 Text Representation

For the learner to compute a classification function, it needs to understand the document. For the learner, the document is merely a string of text. Hence, there is a need to represent the document text in a structured manner. The most common technique to represent text is the Bag-Of-Words (BOW) model. In this technique, the text is broken down into words. Each word represents a feature. This process is also referred to as "Tokenization" since the document is broken down into tokens (individual words). A group of features extracted thus forms a feature vector for the document. Note that in such a model, the exact order of word occurrence is ignored. Since this vector becomes too large, there are several ways to prune this vector. Techniques like stop word removal and stemming are commonly applied. Stop word removal involves removing words which add no significant value to the document. For example, words like "a, an, the, if,

for" can be removed from the vector. Stemming is the process for reducing inflected (or sometimes derived) words to their stem, base or root form – generally a written word form [20]. For example, ran, running, runs are all derived from the word "run". A commonly used stemming algorithm for English language is the "Porter's Algorithm" [21].

Alternate techniques include weighing the features by using a TF-IDF model [19]. 'TF' refers to the Term Frequency of a word, i.e. the total count of the number of occurrences of a particular word in a document. Higher the value of TF, higher the weight for the feature. But TF by itself has some short comings. For example, if the documents were all about "Google search algorithm", the term "Google" is very likely to occur multiple times. The emphasis of the document was not about the company Google but the search algorithm they employ. Hence, to reduce the effect of the word "google", we make use of IDF (Inverse Document Frequency). Document frequency (DF) refers to number of documents in the collection that contain a specific word. Higher the value of DF, lower the importance of the feature. IDF for a feature is calculated as follows:

$$IDF = \log (N/DF)$$

Here, 'N' refers to the total number of documents in the corpus. Finally, the TF-IDF score for a feature is computed as:

$$TD\text{-}IDF = TF * IDF$$

After representing the document, various classifier algorithms can be applied. Some of the popular ones include [33]:

- Naïve Bayes Classifier
- Support Vector Models
- Decision Trees
- Voted Perception

1.3 Short Text Classification

Earlier sections dealt with classification of text messages. By “text”, we referred to documents housing the text. These documents are typically large and are rich with content. Traditional techniques like Bag-Of-Words work well with such data sets since the word occurrence is high and though the order is lost, word frequency is enough to capture the semantics of the document. Alternate approaches like TF-IDF help to counter some loop holes in the Bag-Of-Words approach by weighing the terms.

With the increase in popularity of online communication like chat messages, rich information can be mined from concise conversation between groups of people. Some of the other types of short text messages that are interesting to mine are shown below [26, 27]:

- SMS messages
- Image captions
- Code snippets
- Forum posts
- Product descriptions

- Reviews about various products
- Blog and news feeds (RSS)
- Twitter messages

However, when dealing with shorter text messages, traditional techniques will not perform as well as they would have performed on larger texts. This matches our intuition since these techniques rely on word frequency. Since the word occurrence is too small, they offer no sufficient knowledge about the text itself.

1.3.1 Related Work

Most of the related work focuses on trying to eliminate the problem of data sparseness. One intuitive method to do this is to inflate the short text with additional information to make it appear like a large document of text. Then, traditional classification or clustering algorithms can be applied to it. Some of the previous work [22, 23, and 24] primarily focuses on integrating short text messages with Web search engines like Google, Bing to extract more information about the short text. For each pair of short text, they retrieve statistics on the engine results to determine the similarity score. However, these techniques require additional entity disambiguation approaches. For example, “jaguar” and “cars” are highly related. But, when thesaurus search or web search is performed, many hits may be related to the animal “jaguar” than the car. Hence, there is a need to get explicit feedback from the user to direct the searching and text inflation process. It is also not feasible to perform semantic similarity search on every

pair of short text messages since it is time consuming and not suitable for real-time applications. Although, these techniques do identify pre-dominant terms between messages, there is a need to also compute similar words that are very likely to occur in the same context. The advantage however of using web search as opposed to a thesaurus search (example, WordNet) is that the method does not require pre-existing taxonomies. Therefore, such methods can be applied in many tasks where such taxonomies do not exist or are not up-to-date.

More recent works [26, 27, 28] do away with web searches and instead utilize data repositories. One of the richest data source of information is the Wikipedia. By integrating knowledge available within the Wikipedia, short text messages can be enhanced with more semantic knowledge. [28, 29] make use of the user-defined categories and concepts extracted from Wikipedia and experimental results show significant improvement in accuracy. Titles of selected Wikipedia articles were used to augment the text quality. Banerjee et al [28] used a snap shot of the Wikipedia database and all queries were directed to this database upon which the Lucene (http://lucene.apache.org/java/docs/) index was built. This approach however will not capture the up-to-date information and is especially unsuitable when the input data is highly volatile in its theme like news feeds. On the other hand, online querying of Wikipedia and parsing concepts are not suitable for real-time applications because of the time constraints. Banerjee et al [28] also mentions that usage of additional Wikipedia

concepts (other than titles) did not offer any significant improvements in performance of various clustering algorithms.

Phan et al [26] not only uses the explicit user defined categories in Wikipedia but extracts hidden topic of Wikipedia articles to gain more knowledge. Although it eliminates the problem of data sparseness, it is very time consuming and there is a need to understand what concepts of Wikipedia are useful to extract as mentioned in [28]. Also, there is a need to analyze the final features extracted and eliminate redundant and useless features from classification. This may again require querying the web to select only the necessary features.

In general, approaches that use integration of external knowledge have the following framework as illustrated in Figure 1.1:

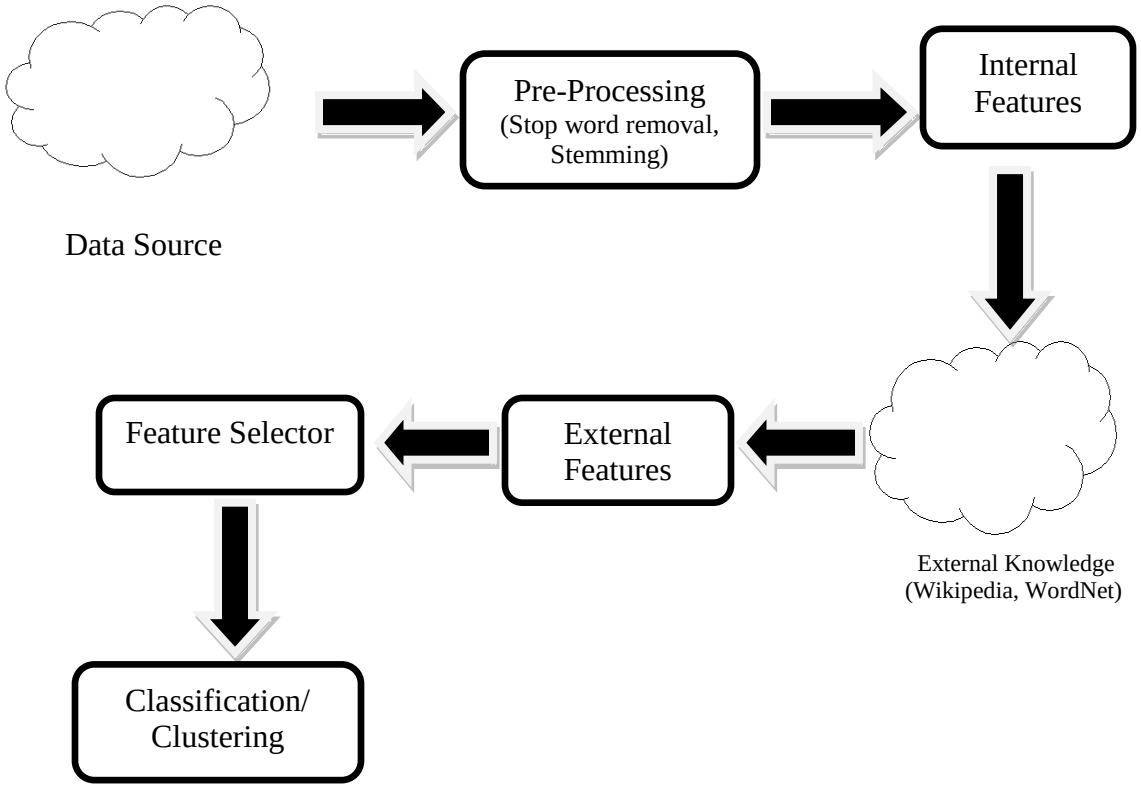

Figure 1.1: General framework of existing techniques to classify short text

Hu et al [27] enhanced existing features by using both WordNet and Wikipedia. WordNet was used for key words whereas Wikipedia was used for concepts. Experimental results showed that the accuracy improves when the knowledge from both the sources are extracted as opposed to using only one of them. Additionally, they also proposed a novel hierarchical resolution phase to parse through the short text and categorize them into segments, phrases and words. From this pool, seed phrases were carefully selected which formed the queries to Wikipedia and WordNet. Techniques from

Natural Language Processing (NLP) were used to perform this task. Then a feature generator module was used to extract external features.

One of the biggest challenges in almost all the related work so far is that by enhancing the feature set by using external knowledge; a new problem creeps in – the Curse of Dimensionality [38]. When the feature set becomes too large, data becomes difficult to visualize and the basis for classification or clustering is lost. Hence, there is a need to effectively prune features and reduce the feature size to an optimal value. Also, there might be several overzealous, unimportant external features that degrade the performance of a classifier. For example, consider the sample message, "Federer wins Australian Open 2010".

If the internal features extracted after pre-processing were Federer and Australia, snap-shot of Wikipedia categories for query "Federer" are shown in Figure 1.2.

Figure 1.2: Wikipedia Categories for search term "Roger Federer"

As you can see from Figure 1.2, the underscored categories offer no significant value to the actual text message. Further, although Australia was one of the features extracted, the emphasis was not on the continent but the tennis tournament held there. Hence, there is a

need to carefully pick seed phrases to query external data. If the seed itself is of little importance, the entire feature set can be useless. When working with Bag-Of-Words, another important issue is that everything in a text is broken down into words. There are situations where a word itself offers no value to the semantics of a message but a key phrase which is a group of two or more words adds sense to the message. For example, “grand slam” in tennis context should be preserved as it is rather than splitting it into two words “grand” and “slam”. Once split, both words offer a completely different connotation to the text message which is incorrect.

In conclusion, related works on short text messages in recent times have primarily focused on eliminating the problem of data sparseness by using external sources like Wikipedia, WordNet etc. Querying such sources online poses the problem of longer time whereas using a snap shot of such data sources has the problem of out dated information. Although these techniques have shown improvement in accuracies, they still rely on the traditional Bag-Of-Words models to represent the features. Increasing the feature set leads to a new problem of curse of dimensionality [38]. Smart algorithms are required to analyze and prune the feature set. Increase in number of features also results in higher model building time and also makes the classification or clustering slower.

In this work, we propose the use of a small feature set to classify Twitter messages which are short text messages of 140 characters in length. Initially, we classify the Twitter messages into diverse pre-chosen classes like New, Opinions, Deals, Events

and Private messages. We also allow users to add new classes based on their interest and also add new features corresponding to the new classes. Experimental results show that the proposed work outperform the traditional Bag-Of-Words techniques.

The rest of the thesis is organized as follows. In Chapter 2, we provide an overview of Twitter and its concepts and also describe why it is necessary to mine Twitter. In Chapter 3, we discuss our proposed work in detail. Here, we discuss how a small set of hand-picked features outperforms baseline techniques for short text classification. In Chapter 4, we discuss about incremental addition of new classes and new features by the user. Chapter 5 contains details about the experimental results and comparison of the proposed work with baseline algorithms. In Chapter 6, we conclude with the future work.

Chapter 2: Overview of Twitter

"*Everyone will be tuned into everything that's happening all the time! No one will be left out*". These were the lines published by Robert Dennis Crumb, an American artist and an illustrator in the 1960's [6]. Although this was published in a cartoon, this is no longer a vision of the future. Thanks to Jack Dorsey, Twitter originated in 2006 making these cartoon lines a reality.

2.1 Introduction to Twitter

Twitter [5] is a social networking application which allows people to micro-blog about a broad range of topics. Micro-blogging is defined as "*a form of blogging that lets you write brief text updates (usually less than 200 characters) about your life on the go and send them to friends and interested observers via text messaging, instant messaging (IM), email or the web."*[7]. Twitter helps users to connect with other Twitter users around the globe. The messages exchanged via Twitter are referred to as micro-blogs because there is a 140 character limit imposed by Twitter for every tweet. This lets the users present any information with only a few words, optionally followed with a link to a more detailed source of information. Therefore, Twitter messages, called as "*tweets*" are

usually focused. In this regard, Twitter is very similar to SMS (Short Message Service) messages exchanged via mobile phones and other hand held devices. In fact, the 140-character limit on message length was initially set for compatibility with SMS messaging, and has brought to the web the kind of shorthand notation and slang commonly used in SMS messages. The 140 character limit has also spurred the usage of URL shortening services such as bit.ly, goo.gl, and tr.im, and content hosting services to accommodate multimedia content and text longer than 140 characters [11]. Several other social networking sites like Facebook [8], Orkut [9] introduced the concept of "Status" messages, some much before Twitter originated. But it was Twitter that went a step ahead and made these "statuses" be sharable between people through mobile phones since its creation.

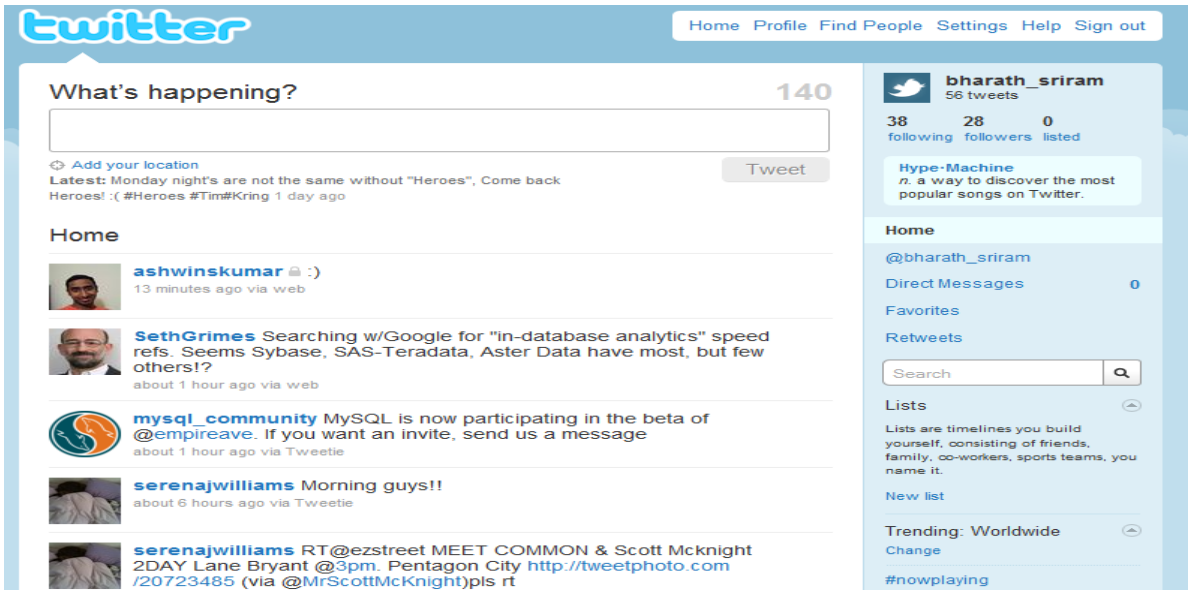

Figure 2.1: Home Page of a Twitter user

Some bloggers criticize the usage of the term “micro-blogging” in Twitter. Their argument is that blogging requires good writing skills, large content to portray one’s thoughts. Since Twitter does not require sound grammar knowledge or long thoughts on a topic, almost everyone post small messages. Some bloggers even suggest that “The idea that someone can send a 140 character twitpitch or let the world know where about in some city street they are is considered to be blogging is stupid and devalues the hard work that most bloggers do every day.” [13]

Although, Twitter is termed as a “social networking” website, it has the flavor of a personal diary rather than a platform to interact with people. The very question Twitter originally asked was “*What are you doing*?” rather than “What do you know?” or “What do you think?” With the increase in popularity of Twitter, it changed its focus and changed its question to “*Discover what’s happening right now, anywhere in the world*”. Slowly, the need to use Twitter changed from using it to record one’s own thoughts to a broader domain. Users now became news reporters and news followers. Sankaranarayanan et al [10] primary research uses this aspect of Twitter and focuses on mobilizing the users of Twitter to be our eyes and ears in this world.

2.2 Architecture of Twitter

We will briefly discuss about how Twitter works before delving into the concepts and entities of Twitter. The Twitter API (Application Programming Interface) is based on the REST (Representational State Transfer) architecture introduced and defined in 2000

by Roy Fielding. [16] REST-style architectures consist of clients and servers. Clients initiate requests to servers; servers process requests and return appropriate responses. Requests and responses are built around the transfer of "representations" of "resources". A resource can be essentially any coherent and meaningful concept that may be addressed. A representation of a resource is typically a document that captures the current or intended state of a resource.

At any particular time, a client can either be transitioning between application states or "at rest". A client in a rest state is able to interact with its user, but creates no load and consumes no per-client storage on the set of servers or on the network. An important concept in REST is the existence of resources (sources of specific information), each of which is referenced with a global identifier (example,, a URI in HTTP). In order to manipulate these resources, *components* of the network (user agents and origin servers) communicate via a standardized interface (example,, HTTP) and exchange *representations* of these resources (the actual documents conveying the information).

The Twitter API consists of three parts: two REST APIs and a Streaming API [17]. The two distinct REST APIs are entirely due to history. The Streaming API is distinct from the two REST APIs as Streaming supports long-lived connections on a different architecture. The Twitter REST API methods allow developers to access core Twitter data. This includes update timelines, status data, and user information. The Search API methods give developers methods to interact with Twitter Search and trends data. The

concern for developers given this separation is the effects on rate limiting and output format. The Streaming API provides near real-time high-volume access to Tweets in sampled and filtered form.

2.3 Concepts in Twitter

2.3.1 User

A Twitter user '*A*' is a person or a system who publish tweets. These tweets are by default public to any user of the system unless the author specifically sets it to be private. All users of a system are identified by a unique user name and user id. When '*A'* initially registers to Twitter, he has no tweets or no followers or friends (concepts explained later) to begin with. '*A*' can start posting tweets but these tweets are not read by other users since the user does not have any followers yet. Once *'A'* identifies another user (say *'B'*) to follow, his tweets are visible to *'B'*. Consequently, '*A*' becomes a follower of '*B*'. Thus, '*B*' becomes a friend of '*A*'. Note that, however, friendship need not be two-way. It is possible that '*A*' is not a friend of '*B*' if '*B*' does not follow the tweets of '*A*'. Thus there exists an asymmetrical relationship between users of Twitter. Twitter also imposes a limit of 2000 friends for a particular user but there are no restrictions on the number of followers since users do not have any control of the number of followers they have.

The following information is also optionally stored for each user:

- Language of tweets of the user
- Time zone of the user's location
- Tweet location – the location from which the tweet was tweeted

- User’s profile picture
- User’s location
- User’s web page
- Short biography of the user
- Favorite links

2.3.2 Tweet

A tweet is a Twitter message. It is short message since it is restricted to be within 140 characters by Twitter. This restriction enforces the users to be concise in what they have to say. This is also the reason why users tend to use word shortenings (*Eg:* “fr”-for, “cud” – could) and abbreviations. Interestingly enough, there is a rich and well understood set of abbreviations which is surprisingly consistent across user groups, and even across other electronic mediums such as SMS and chat rooms [10]. Since users want to convey all they have to say within 140 characters, they could also make spelling mistakes and tweets can be prone to syntactic errors. This makes Twitter a challenging medium to work with. Most of the times, users usually provide links to external resources when they cannot convey the complete information within 140 characters. These URL links to text, audio or video files are referred to as “*Artifacts*”.

2.3.2.1 Special Characteristics of Tweets:

Reference to another user

To refer to another user within a tweet, the '@' symbol is used followed by the intended user name. When a user refers to another user at the beginning of a tweet, the tweet becomes a "Direct Message" (DM). Direct messages are those tweets that are public yet intended as a correspondence between exactly two users of the system. Twitter provides a provision to view only direct messages intended to the user. This ensures that these messages which usually have a higher priority to the intended user do not get lost by the overwhelming stream of other tweets in the user space. When the reference to another user does not occur at the beginning of the tweet, it does not qualify for a direct message but merely serves as a reference point.

Eg: *Bob: @Alice*, How was the Biology test? (Direct Message)

Trudy: I really had loads of fun at the party. @Bob made it extra special with his cookies. (Reference to another user)

Re-tweets (RT)

If a tweet is compelling and interesting enough, users might republish that tweet – commonly known as re-tweeting. A re-tweet is similar to forwarding an e-mail. When a user re-tweets some content, the user is effectively endorsing and sharing the content with their followers [12].

Twitter earlier lacked a specific structure for re-tweets by merely providing a convention on how to re-tweet. Several forms of re-tweets were used, some of the most

common being "*retweet @username*", "*RT @username*" or "*via @username*", before or after the re-tweeted message. But the current version has an option called "Retweet" right next to each tweet.

Hash Tags

Twitter allows users to tag their tweets with the help of "Hash tags". Hash tags are of the form *'#<tag-name>'*. Users can thus convey what their tweet is primarily about by using keywords that best represent the content or the genre of the tweet. Hash tagged tweets help Twitter to group similar tweets together that have the same hash tags. This makes search on Twitter easier and faster. Hence, with this provision users can follow a specific topic of interest. Most of the Twitter search tools [14, 15] use hash tags to enhance search quality. Note that the hash tag itself adds to the character count of the tweet. Shown is Figure 2.2 are hash tags associated with the Iranian elections from June 2009 [10].

Figure 2.2: Hash tags associated with Iranian Elections from June 2009 [10].

2.4 Why mine Twitter?

Twitter serves as a rich source of information. Unlike other information sources, Twitter is up-to-date and reflects the current news and events happening around the world. Conventional news agencies often employ reporters and journalists to gather news. The quality and content is constrained by the number and the type of journalists. Also, once news is published, Web spiders must be updated to crawl for the latest information. On the other hand, Twitter provides information by having millions of users serve as reporters. News or Events here could be global, which means messages that can be understood by a large group of audience or it could be local, which is understood by a small group of people or even one specific individual. Sankaranarayanan et al [10] refers to this principle as “push-pull” where on Twitter, information is “pushed” automatically to the users rather than the users “pulling” the necessary information from the web.

Another important feature about Twitter is that there is minimal time lag between the time of occurrence of an event and the tweet publication time. Hence, information is conveyed rapidly to users. A good example for this is provided in [10] which is shown in Figure 2.3.

Figure 2.3: Tweet/hour relating to Michael Jackson's death [10].

The first tweet regarding the death was reported 20 minutes after the 911 call, which was almost an hour before conventional news media reported the death [10].

Tweets originating from users not only talk about news but a wide range of topics. Java et al [1] mentions that the user intentions on Twitter include daily chatter, conversations, sharing information/URLs, and reporting news. Since, there is a lot of rich information being transmitted across the globe at a rapid rate, there is a need to mine

knowledge from it and provide users with a system that helps them to understand and comprehend the information as easily and quickly as possible.

Chapter 3: Improved Information Filtering using 8F

As already mentioned in the Background, classification of short text messages is a hard task due to lack of content and context. As techniques that use word occurrence and its variations as features do not perform as well as it does on larger corpus of text, there is a need to research beyond using words as features. We should also ensure that the feature set does not become too large and eventually suffer from the "curse of dimensionality".

3.1 Tweet Classification

On Twitter, tweets are presented to the user in a chronological order. This format of presentation is useful to the user since the latest tweets from the user's followers are rich on recent news which is generally more interesting than tweets about an event that occurred long time back. But the major drawback of this approach is that tweets arrive at a furious rate. Merely, presenting tweets in a chronological order may be too overwhelming to the user. Also, if the user has many friends out of whom, few tweet at a rapid rate compared to other friends, the dominant friend takes a lot of the user's space. Hence, tweets from the lesser dominant friends may be lost in the overwhelming tweet

stream. Due to these issues, there is a need to separate the tweets into different categories and then present the categories to the user.

To begin with, we identified seven generic categories (classes) that the users may be interested in. We choose these categories to be as diverse as possible and ideally hope that almost all the tweets can be classified into one of seven chosen categories. Therefore, based on the user intentions on Twitter [1] such as daily chatter, conversations, sharing information, URLs, and reporting news, we come up with the following seven classes:

- Neutral News
- Personal News
- Opinionated News
- Opinions
- Deals
- Events
- Private Messages

We define each of these classes in detail as follows.

3.1.1 Description of pre-determined classes

Neutral News

News tweets are those that can be understood by a larger group of audience and hence generic. News tweets are generally neutral in nature, i.e. they are not highly opinionated on a particular topic but merely present the facts. They usually originate from

corporate tweeters (Eg, CNN, NY times) although there have been cases where they originated from personal tweeters [2, 3]. They usually convey the main summary information and provide a link to an external detailed resource. They tend to be very structured and are seldom highlighted by typos and word shortenings.

Eg: *nytimes:* Bob wins Australian Open 2010. Read more at www.nytimes.com/

Personal News

Personal News tweets are those that generally highlight an individual user's train of thoughts or a description of his/her situation. It is generally significant to a smaller group of users and does not have global importance. Like neutral news, they tend to be non-opinionated. They usually originate from personal tweeters. They are to the point and seldom convey information via an artifact. They are not necessarily structured and may contain shortening of words to convey the thoughts within the 140 character limit.

Eg: *Bob:* I am having severe headache…Shud call the doc later tonight!

Opinionated News

Opinionated News tweets are those which describe some kind of a positive or negative opinion expressed by the subject of the news. News coming from news agencies is not opinionated, i.e. they are not biased by the sentiments of the journalist or the agency but may still talk about news which by itself is opinionated.

Eg: *nytimes:* Mayor Alice claims that the new policy is ineffective and in short sucks!

As shown in the example, the tweet does not convey any sentiment of the author (nytimes) but still indicates that Mayor Alice feels that the new policy is not good. This tweet highlights the opinion of Alice. Opinionated news can be significantly important either to a local crowd or to a global crowd. It is difficult to make any claim about the author of the tweet. They are different than personal opinions since they do not capture the opinion of the author of the tweet.

Opinions

Opinion tweets are similar to the opinionated news but convey the opinions of the author of the tweet. They usually originate from personal tweeters who talk about their take on diverse entities. They either convey a positive or a negative sentiment through the tweet. They are to the point. They may contain shortening of words and may additionally use emphasis on words to convey stronger opinions.

Eg: *Bob:* I think the new movie just rocks!

In this work, we do not differentiate between positive and negative opinions. We merely identify opinionated tweets. Opinionated words are emphasized in two ways:

- Using Upper case letters (Eg: LOVE it!!)
- Repeating a character multiple times (Eg: loveee it!)

Also, opinions are expressed via smiley faces in tweets to convey the sentiment in fewer characters.

Deals

Deal tweets are about offers on various products or services. They often contain artifacts to a more detailed description. They tend to be very structured and are seldom highlighted by typos and word shortenings. They are characterized by a small dictionary of terms that are frequently used like “deal”, “free” etc. There might also be deal tweets that are just spam. They usually originate from corporate tweeters and rarely from personal tweeters.

Events

Event tweets are those that specify an “event”. Dictionary.com (from Ask.com) [4] defines it as “*something that occurs in a certain place during a particular interval of time*”. Events are characterized by participant information, location information and time information. But event tweets may not be structured perfectly to suit the definition of an event due to limitation of character length. In some cases, the participant information could be implicit and assumed. Similarly, location information may not be a well-defined geographic location but merely a hang-out place. Some event tweets may not even contain time information. Therefore, event tweets may contain all the necessary information that makes an event or just a subset of them. However, event tweets from corporate users usually tend to be well defined and include almost all the information or provide an artifact with the actual details.

Eg: - *Alice:* Going to @Bob’s place tonight for watching the game.

IBM_Informix: 2nd IIUG conference at Atlanta, Georgia from 12th May 2010 to 15th May 2010. Registration window opens today!!

Private Messages

Private message tweets are those intended to a specific user of Twitter. These messages include those that are addressed to the user of the system and those that are exchanged between two other friends of the system. These messages originate from personal tweeters and rarely from corporate tweeters. This is attributed to the fact that the target audience for corporate tweeters is large whereas personal tweeters can send tweets to a specific friend since the context of the tweet may be understood only by the user and his friend. Private messages are characterized by the presence of '@' followed by the name of the friend at the beginning of the tweet. Although this tweet can be seen by all other followers of the user (unless the tweet is made private), it is only addressed to that particular friend and other followers may not be interested or understand the tweet.
Eg: *Bob*: *@Trudy*, Had so much fun at ur place yest, thanks for having me :)

It is important to note that it is not mandated by Twitter to have the @<*username*> to denote messages to a specific person. Bob could have as well said "Trudy, Had so much fun at ur place yest, thanks for having me :)". Another important point to note is that the name after '@' is the Twitter "username" of the friend. When Bob does actually follow the @<*username*> format, the friend with <*username*> can check the tweet by clicking the circled option as shown in Figure 3.1.

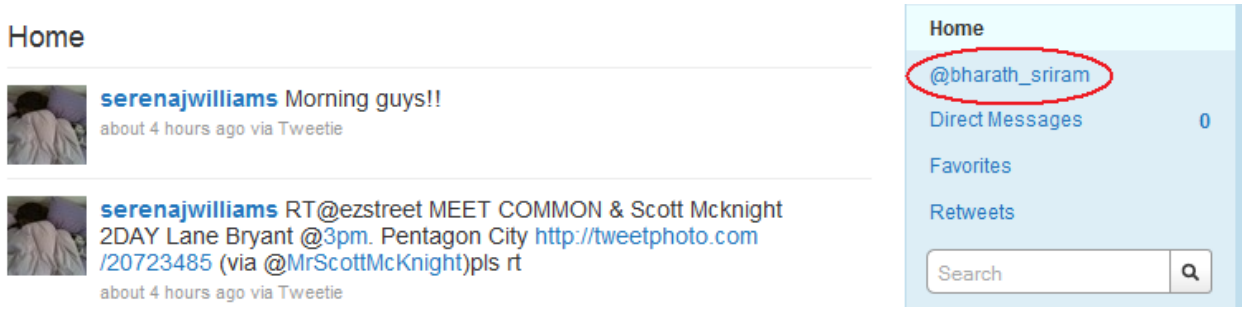

Figure 3.1: Checking private messages addressed to user

Private messages are usually not well structured and may contain typos and/or shortening of words. They may additionally contain opinions or event information addressed to the specific friend.

On closer analysis, we feel that the classes' opinions and opinionated news are highly similar and so are neutral news and personal news. Therefore, we merge opinions and opinionated news into a single class called "Opinions" and neutral and personal news into a single class called "News". Experimental results are therefore shown for five classes, namely news, opinions, deals, events and private messages.

3.2 Feature Selection

Selecting a subset of relevant features for building robust learning models is another research problem. Hence we used a greedy strategy to select the feature set, which generally follows the definitions of classes. We already mentioned in chapter 1 and

earlier in this chapter that there is a need to use a minimal set of discriminating features to represent the short text messages. Clearly, this rules out Bag-Of-Words as a representation since the dimensionality increases exponentially with increase in short text messages. Also, tweets are prone to typos and short forms which make the choice of Bag-Of-Word less popular. Based on our five classes, we define eight features that best represent the text message.

These eight features are defined as authorship information (Nominal) and the presence of:

- Shortening of words and slangs (Binary)
- Time-event information (Binary)
- Opinions (Binary)
- Emphasis on words (Binary)
- Currency, statistical information (Binary)
- Reference to another user at beginning of tweet (Binary)
- Reference to another user within tweet (Binary)

We observe that the authorship information is very important to classification. Hence, we choose the authorship information as our primary feature. We observe that authors generally adhere to a specific tweeting pattern i.e., a majority of tweets from the same author tend to be within a limited set of categories. Studies [31] also indicate that 10% of the most active users contribute up to 90% of the tweets. Experimental results show a significant improvement in accuracy when the authorship information was

included in classification. Detailed analysis is presented in the experimental results chapter. The following section discusses how the features represent our pre-determined classes.

3.3 Feature Extraction

3.3.1 News

Categorization of tweets into the selected classes requires the knowledge of the source of information. Hence, we selected the authorship information as our primary feature. Corporate tweeters generally have different motivations than personal tweeters. While the former generally publish news in a clear form, the latter instead frequently express themselves by using slang words, shortenings and emotions. Thus, a feature for discriminating news may be the absence of shortenings, emotions, and slang words. This feature can be further used to differentiate the personal tweeters from corporate tweeters.

3.3.2 Events

If we define an event as “something that happens at a given place and time”, the presence of participant, place, and time information could determine the existence of an event in the text. Hence, we extracted the date/time information and time-event phrases which are collected from a set of tweets based on general observation of users and set the presence of them as a feature. Participant information is also captured via the presence of the ‘@’ character followed by a username within tweets.

3.3.3 Opinions

Presence of opinions is determined by a lookup in a wordlist which consist of about 3000 opinionated words obtained from the Web. Usage of pronouns has a powerful insight if it is a personal opinion or an opinion highlighted from a different source. We also capture the emphasis on words based on the usage with uppercase letters. Another way to detect the emphasis is the usage of repeating characters in a word (example,, "veery").

3.3.4 Deals

The keyword "deal" and special characters within the text such as currency and percentage signs are good features to capture the context of deals.

3.3.5 Private messages

Twitter lets the users send private messages to other users by using the „@" character followed by the username at beginning of tweet. Hence, private messages are captured by the usage "@username" within tweets.

Once these features are extracted from the input data set, they are fed to a classifier for measuring the system performance. The selection of features may seem to be ad hoc at first glance but the features were chosen such that it is tuned for the five classes. To build a complete system however, users may be interested in adding new

classes of their choice and experimenting with addition of new features to represent these classes. With the frame work proposed in this chapter, this is not possible but the next chapter deals with incremental addition of new classes and new features by the user.

The 8F feature set could also be enhanced to include more of the author profile information like the author's interests, location, language, tweet publication time etc. In the next chapter, we include more features extracted from the author profile information and include them for classification.

A very attentive reader might notice that tweets may demonstrate flavors of multiple classes. Example, a tweet could be an opinion and a private message. In such cases, the classifier is forced to choose only one class which has the highest probability to house the tweet. Under those circumstances, careful analyses is required and assign the tweet to multiple classes simultaneously in case the probability of a tweet being assigned to a specific class is below a certain threshold 'Θ'.

Chapter 4: Addition of User Defined Classes and User Defined Features

In the previous chapter, we explained how only eight features were used to classify tweets. Tweets were classified into five classes, namely, news, opinions, deals, events and private messages. Experimental results (next chapter) show that with only a small set of features, the classifier achieves a significant improvement in accuracy when compared to the traditional Bag-Of-Words technique. However, one could argue about the choice of the selection of the features and choice of selection of the classes. Although selection of the classes is based on the general interest of the public in using Twitter, there could be users interested in classifying tweets into a class of their choice. For example, a user could be interested in creating a new class (category) called “Ipod” to keep track of all tweets related to Ipod. Note that however, with the previous approach, tweets about Ipod may be distributed within news, deals and opinions. Rather than searching for Ipod tweets every time, creating a separate class for Ipods help user to get tweets related to Ipod automatically in real time. Hence, there is a need to allow the user

to create new categories based on his interest. In this chapter, we present an approach that extends our original framework to add new classes and consequently add new features. This research is still preliminary and experimental results are based on the initial algorithm.

4.1 Addition of new classes

Apart from the pre-existing five classes of news, events, deals, opinions and private messages, we allow the users to create their own class based on their interest. We also suggest potential class categories to users to begin with. Therefore, a new class is created by:

- Suggesting global trends in Twitter
- Suggesting local trends in Twitter
- Suggesting trends in user space
- Allowing user to create a new class of his interest

Twitter maintains a list of global trending topics which refer to the latest news occurring around the globe. These topics could serve as candidates for new classes for the user. Alternately, one could also suggest trending topics that are local to the user's location. Twitter also provides such an option to view trending topics at a specific location. A user is more likely to be interested in knowing more about the local trends and news than the global trends. A snap shot of Twitter showing global and local trends are shown in the Figures 4.1 and 4.2 respectively**.** We also observe the user space to find

trending patterns within the user space. For this, we merely count the occurrence of words within categories. This also shows the dominating terms in the user space. For example, if dominating terms within the news category in the user space was “Obama”, it is more beneficial to create a separate class called “Obama” rather than have many Obama related tweets dominate the news category. By doing so, non-Obama related news get more visibility in the News category. A snap shot of dominating terms within the News and Deals category is shown in Figure 4.3.

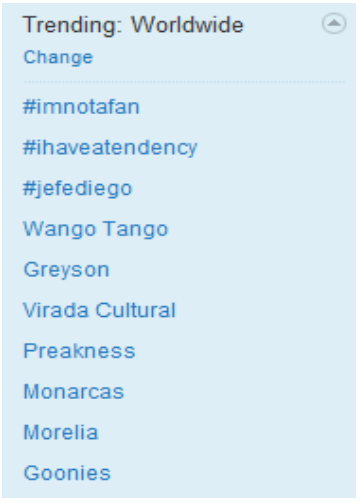

Figure 4.1: Global trends

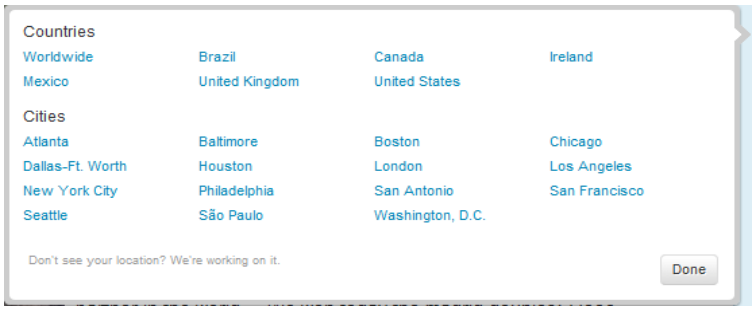

Figure 4.2: Local trends on Twitter

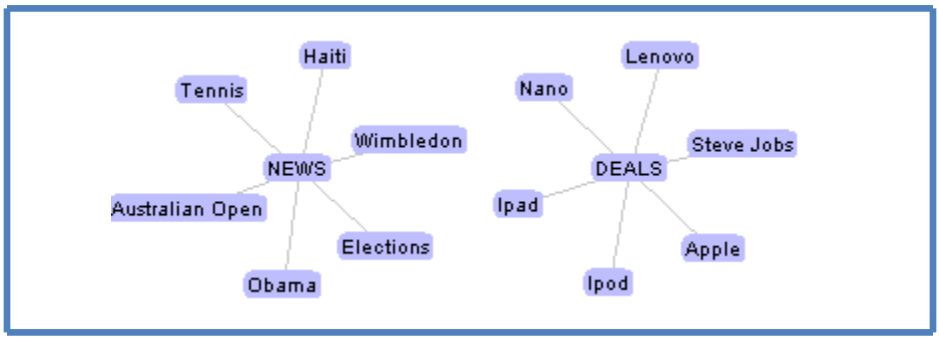

Figure 4.3: Trends in user space

Prefuse (http://prefuse.org/) was used to visualize the trends within the user space. We could also use smarter techniques to compute trends within user space by weighing terms that are re-tweeted more than the other tweets.

Finally, we let users add classes based on their interest. For this purpose, the users need to enter few tags that define the new class. For example, for a new class dealing with IBM tweets, potential tags the user could enter could be DB2, Informix, Thinkpad, AIX, Rational, Tivoli etc. The more tags the user provides, the better is the coverage of obtaining tweets related to IBM.

Once a new class is created, the training data needs to be updated to reflect the new class. Hence, addition of new classes requires updating the training data completely. Once the training data set is updated, the classifier needs to be re-trained on the new

training data. Since, the training data set is relatively smaller compared to the new data (test data) which arrives at a rapid rate; this should not be a major overhead.

4.2 Addition of new features

In the previous section we talked about letting the users add new classes to the existing system. Consequently, there is a need to update the feature set as well since 8F feature set is designed to work well with the five pre-determined classes. If new classes are added and the corresponding features are not added, the accuracy of the system may degrade. Users can also experiment by adding new features without adding new classes and measure accuracy. New features can be deleted incase, the performance of the system goes down by the addition of these features. Intuitively, a feature corresponds to a class, i.e. there is a need for a presence of at least one discriminating feature per class. The quality of this feature is also very critical to the performance of the system. Note that, addition of multiple ineffective features will bring down the quality of the accuracy. The need for new features to classify new classes is shown below. Consider a situation where the five pre-determined classes are in place along with a small feature set for classification. Say, the user adds a new class “tennis” that houses all tennis related tweets. Say, for simplicity that there are only 5 binary features (5F) used for classification purpose. Each feature merely represents the presence (or absence) of the class label in tweet. The problem now is illustrated in Figure 4.4.

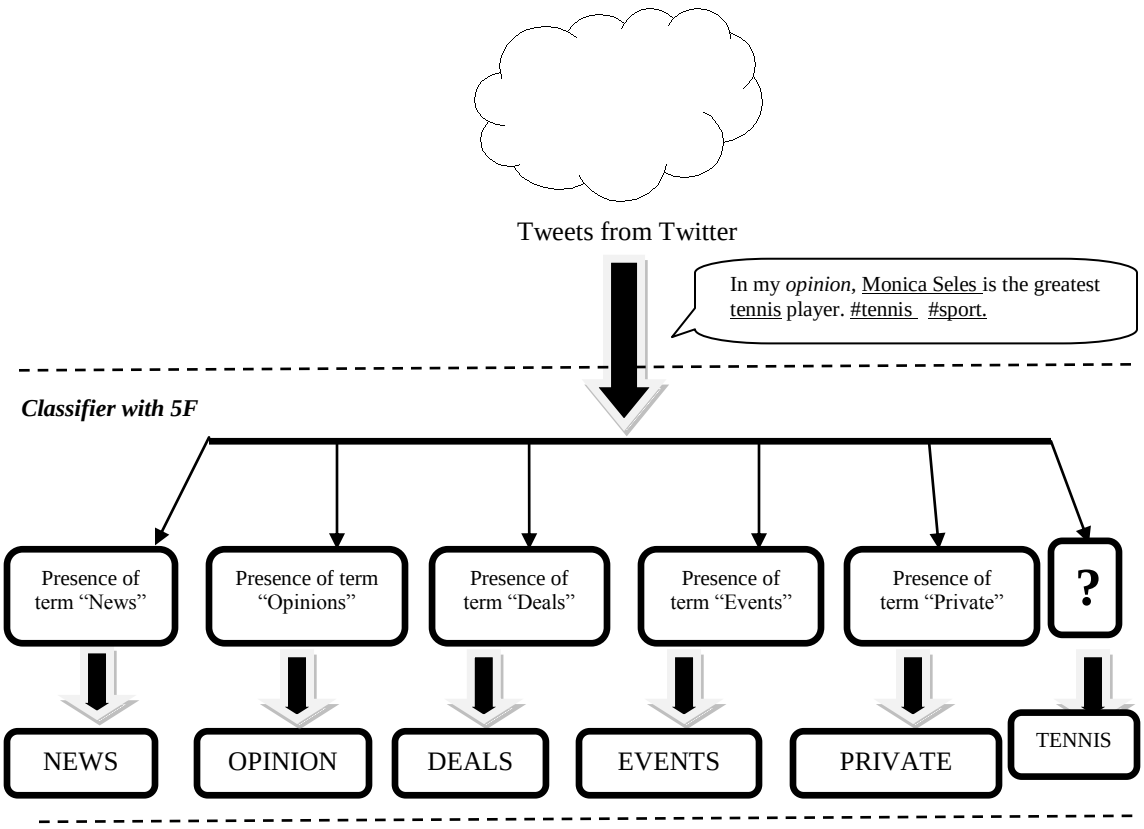

Figure 4.4: Illustration of need of new features

As shown in Figure 4.4, when a new tweet comes in (shown in oval callout), it is very likely to be classified as opinion rather than tennis because there exists a feature that is set because the tweet has the word "opinion" in it. But the tweet is highly related to tennis since it contains several words related to the sport (underscored words in tweet). This is because there was no discriminating feature for the newly added "Tennis" class. Hence, there is a need for the users to add at least one new feature when they add a new class. On adding several new classes without adding corresponding features, the system

performance is bound to degrade. Experimental results prove this intuition when six new classes were added to the pre-existing classes without adding any additional features. Detailed values are presented in the next chapter.

Apart from the additional features that the user adds, we further enhance 8F feature set to include more author profile information. For now, we include the location and tweet publication time into the feature set. The reason why we do this is that tweeting patterns can also tend to be local. For example, a local art festival in Columbus, Ohio could generate many tweets about the art festival event. Tweet publication time was chosen to be one of the features since we believe that there might be interesting patterns among the pre-determined or newer classes with respect to time of publication. For example, news tweets are likely to happen around the clock whereas private messages may be tweeted only during the day.

To add a new feature, user provides sample tweets for the system to learn about the new feature. Note that the user does not point out explicitly what the new feature is but merely feeds tweets to the system to learn. The more samples the user provides, the better understanding the system gets about the new feature. To compute the new feature, we look at several aspects of a tweet. Apart from the already pre-computed 8F features, we look at common words between sampled tweets, presence of URL, common special characters between tweets, common authors. Although this analysis is useful, there could be sampled tweets which have none of these features in common but are still perceived to

be similar by the user. This could be because the sampled tweets may revolve around a common theme which is not captured by merely looking at common words between them. For example, consider a set of four sampled tweets below:

Tweet 1: *Bob:* I love coffee so much, cud never live without it

Tweet 2: *Alice: @Bob*, I agree completely, u shud chk out the new Café coffee day on 7th Ave

Tweet 3: *Trudy:* Nothing like a good espresso to make ur day vibrant :)

Tweet 4: *Alice:* The new coffee machine at my office is ROCKING!

As one can see, the common theme between these tweets is "coffee". Although some tweets do not have the word "coffee" itself, it mentions several types of coffee or words that are synonymous to coffee. Capturing such themes is very important.

To capture common themes between tweets, we do the following:

4.2.1 Key Term Identification

To first identify the central theme, we need a seed term that represents this theme. By taking two tweets as inputs to analyze, we first pre-process the tweet. This is done by removing stop words and opinionated words from the tweets. We also clean the tweets from any special characters. We discard words that begin with '@' since this often refers to another user on Twitter. Next, we consider every pair of words between tweets and query Bing (http://www.bing.com/) search engine to get the total hit count for the query made of the two words. This is done to find out similar words between tweets. Higher the

count, higher do they co-occur in documents and are very likely to represent a common theme. For example, the page count of the query “apple" AND “computer" in Bing is 86,600,000, whereas the same for “banana" AND “computer" is 13,800,000. This indicates that apple is more semantically similar to computer than a banana. We take into consideration only top ‘n’ (‘n’ is usually small ranging from 1-5) hit pairs from these results. Alternately, Bollegala et al [22] argues that page counts alone would not suffice to conclude similarity between words and proposes several page-count based similarity scores to reduce the effect of false positives. One of them is the WebJaccard co-efficient. If ‘P’ and ‘Q’ are the query words individually and “P AND Q” is the query to Bing, similarity score between these pair of words is computed as:

$$\text{WebJaccard}(P, Q) = \begin{cases} 0 & \text{if } H(P \text{ AND } Q) <= \theta \\ H(P \text{ AND } Q) \,/\, \{H(P) + H(Q) - H(P \text{ AND } Q)\} & \text{otherwise} \end{cases}$$

where H(P) represents the total hit count for query ‘P’ and ‘θ’ represents a threshold value set by user.

However, since what constitutes as a false positive is not certain in our system, we let the user decide if the system analyzed the theme correctly by providing diagnostic messages. Here, the user can explicitly discard irrelevant themes the system captured from sampled tweets. Such themes will not be again re-computed by the system.

4.2.2 Querying Microsoft Word Thesaurus

After identifying key terms from tweets, we take a step further and compute other similar words that represent the central theme. This is done to ensure that tweets that do not have the exact same words as ones present in the sampled tweets but yet represent the same theme are captured in the feature. To do this, we query MS-Word thesaurus by providing the hit terms previously identified as a query to the thesaurus. This helps generate more words that enhance the quality of the captured theme. Note that however, it is very necessary to get the user feedback early during diagnosis of sampled tweets. If irrelevant themes creep in early, they are further enhanced by finding other similar irrelevant words from the thesaurus. Alternately, one could use a more comprehensive dictionary like the WordNet to achieve this task.

4.2.3 Using Google Sets

Apart from computing synonymous words, it is also necessary to compute similar terms that are very likely occur with the captured theme. For this we make use of the Google Sets API from Google Labs [30]. Google sets identify groups of related items on the web and use that information to predict relationships between items.

A sample diagnosis of tweets revolving around the theme “coffee” is shown. The words represents words captured from tweets, those synonymous to captured words, and words

related to the captured words through Google sets. Underscored words are the captured words from sampled tweets.

For example, Coffee, café, chocolate, bar, espresso, kaffee, food, eating, wifi, coffeehouse starbucks

Once the new feature is diagnosed correctly, we update the training data to include the new feature and re-train the classifier on the training data. A summary of the new feature addition process is shown in Figure 4.5.

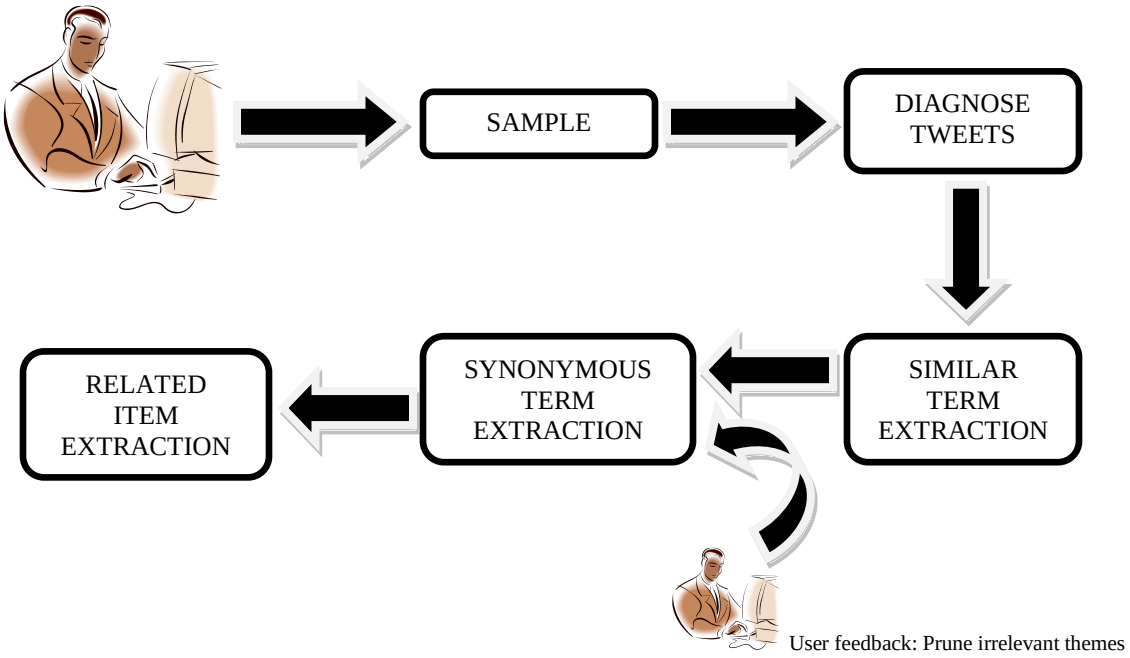

Figure 4.5: Summary of feature addition process

When the sampled tweets from the user becomes greater than a threshold 'α', we weigh extracted features like common special characters, common authors and common words more than the theme. If more than 'α' tweets have such patterns in common, it is less likely to be a co-incidence and very likely that the sampled tweets represented the presence of an explicit word or a character rather than a theme. In such cases, we only set the feature if such patterns are found in other tweets irrespective of their theme. Note that in this approach, unlike online querying for every message, only sampled tweets from users are enriched with meta-information.

Chapter 5: Experimental Results

In this chapter, we present the experimental results for techniques described in chapter 3 and 4. The data sets used in these experiments are tweets from Twitter. The first set of experiments was run on tweets collected from seed users. We created a mock user on Twitter called “osu_user” and followed people of diverse fields ranging from sports, arts, computer science, corporate organizations, book reviews, product dealers etc. We also identified seed users who are known to publish tweets about a specific subject. Example, tweets from CNN are assumed to be all news. Finally to perform the first set of experiments, we came up with 5407 tweets collected from 684 followers and seed users. These tweets were manually labeled as belonging into one of the five classes namely news, opinions, deals, events and private messages. In case the tweet exhibits flavors of multiple classes, the best possible class is chosen as the label. We pre-process the tweets and eliminate tweets that:

- Are not in English,
- Have too few words (threshold set as three),
- Have too few words apart from greeting words,
- Have just a URL and

- Have too few words apart from URL.

5.1 Experimental Results for Original Framework

We also consider only those tweets that can be labeled into one of the five classes. We believe that the choice of our classes is very generic and diverse enough to cover almost all tweets in a user space. Alternately, we could also include a "Miscellaneous" class to house tweets that cannot be labeled into one of the five pre-determined categories. In case we are dealing with a very noisy data set, efficient noise removal techniques have to be employed to clean tweets before classification. Techniques similar to [32] can be used for this purpose.

The distribution of tweets per class is shown in Figure 5.1.

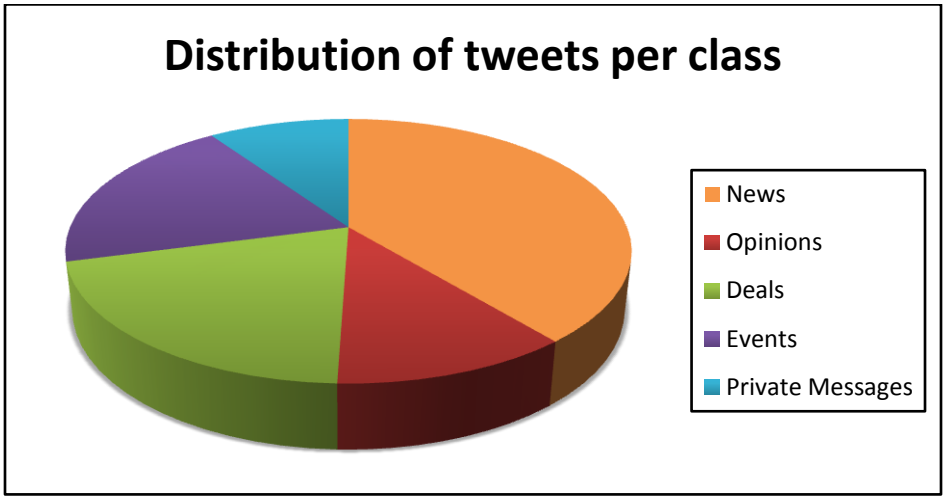

Figure 5.1: Distribution of tweets per class

All experiments were run using available implementation of Weka [34]. Three classification algorithms, namely Naïve Bayes, C4.5 decision tree and Sequential Minimal Optimization (SMO) were used on the training data. Experimental results are based on 5-fold cross validation of the data.

The features used for each of these experiments are as follows:

- 8F: The eight feature set mentioned in chapter 3
- BOW: Bag-Of-Words is chosen as the baseline
- 8F + BOW: Combination of 8F with Bag-Of-Words
- 7F + BOW: Combination of 8F with Bag-Of-Words without the authorship feature
- BOW – A: Bag-Of-Words with authorship feature

We choose the BOW model as our baseline since it is popularly used for traditional text classification purposes. We mentioned in chapter 3 that the authorship information plays a vital role in classification. We demonstrate this by updating BOW with authorship information. Figures 5.2, 5.3 and 5.4 show the total accuracies of the classifier using different type of feature sets for Naïve Bayes, C4.5 and SMO algorithm respectively.

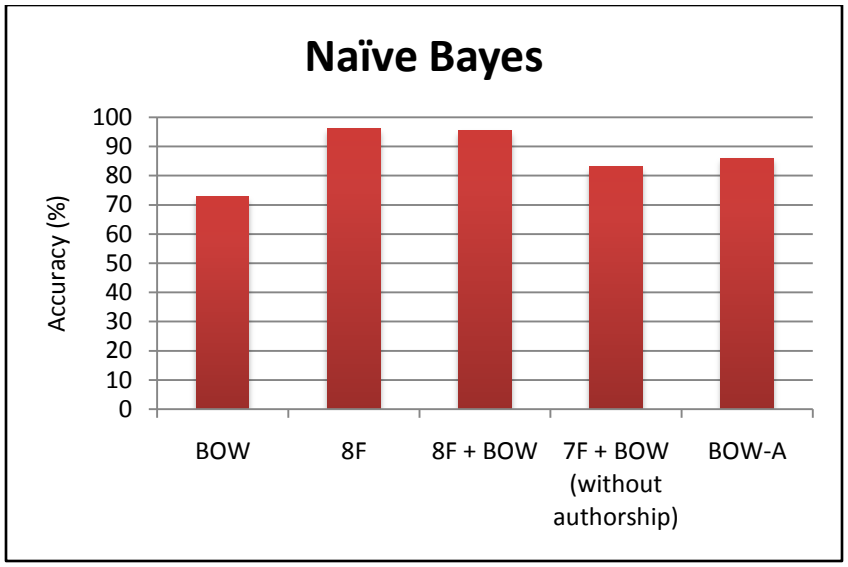

Figure 5.2: Overall accuracies using Naïve Bayes Algorithm

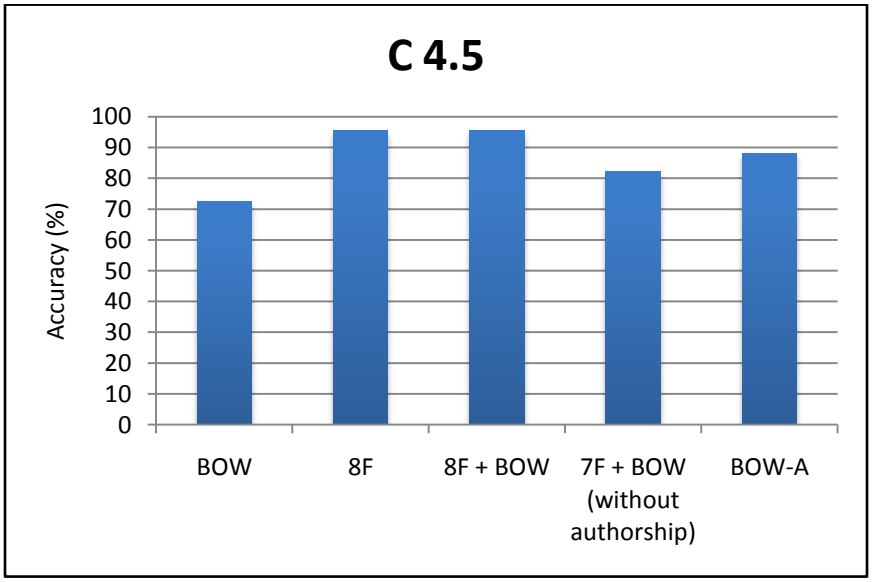

Figure 5.3: Overall accuracies using C4.5 Decision Tree Algorithm

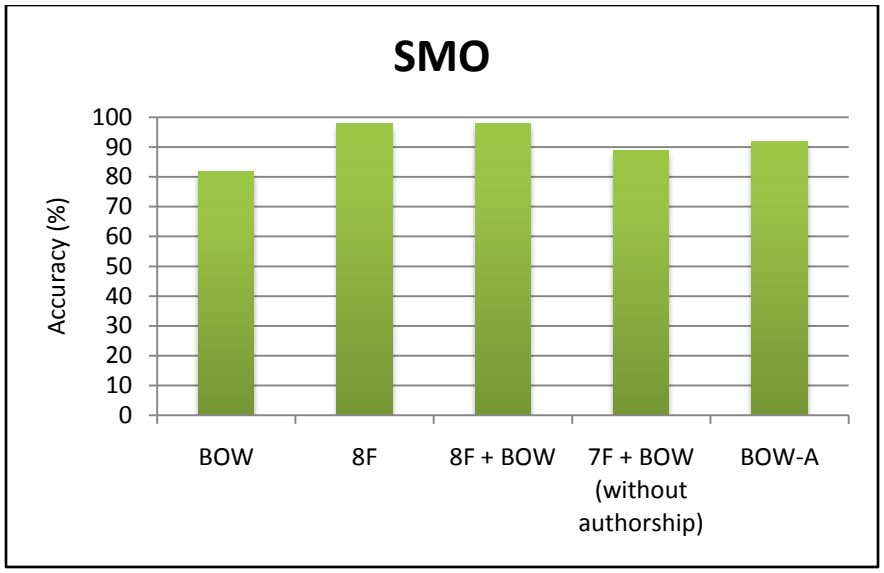

Figure 5.4: Overall accuracies using SMO Algorithm

As seen in the above Figures, 8F performs the best amongst any other chosen feature set for all the three algorithms. BOW has the least accuracy among all other feature sets. By combining 8F with BOW, the accuracy came close to 8F and sometimes on par with 8F. By just including the authorship information with BOW, there is 18.3% improvement accuracy over BOW. Figure 5.5 shows the improvement in percentage of 8F over BOW for the three algorithms.

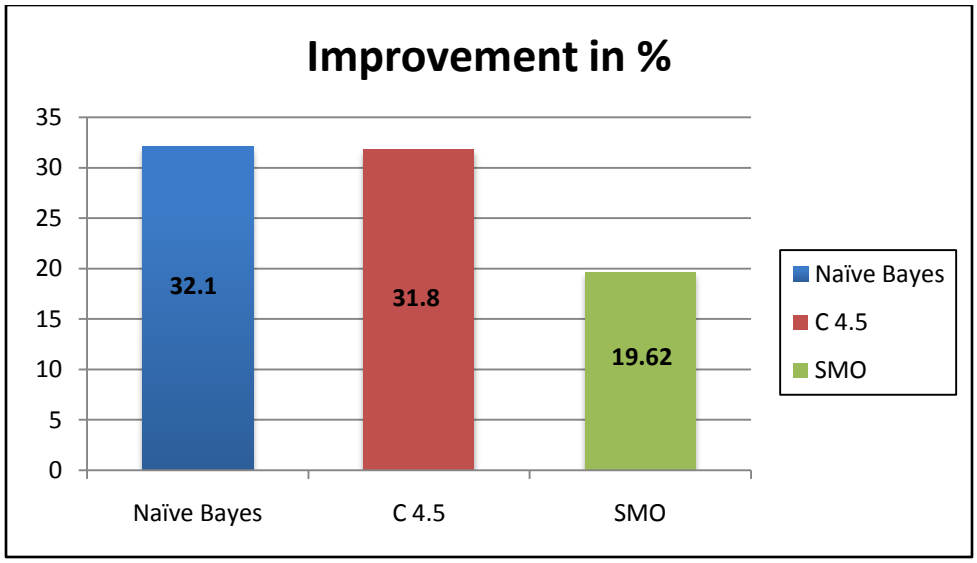

Figure 5.5: Percentage improvement of 8F over BOW

From Figure 5.5, it is clear that 8F performs significantly better than BOW for all three algorithms. The choice of the algorithm depends on the application. Previous research has shown that the newer algorithms like support vector machine classifier algorithms tend to perform better on text. However, a deeper analysis is required to understand the pros and cons of different classifier algorithms when applied to shorter, noisier text. In our experimental results, we observe that Naïve Bayes and SMO performs almost the same although SMO has a slightly higher accuracy when using 8F but the percentage improvement of 8F over BOW is higher in Naïve Bayes than in SMO.

The individual accuracy per class for the Naïve Bayes algorithm is shown in Figure 5.6.

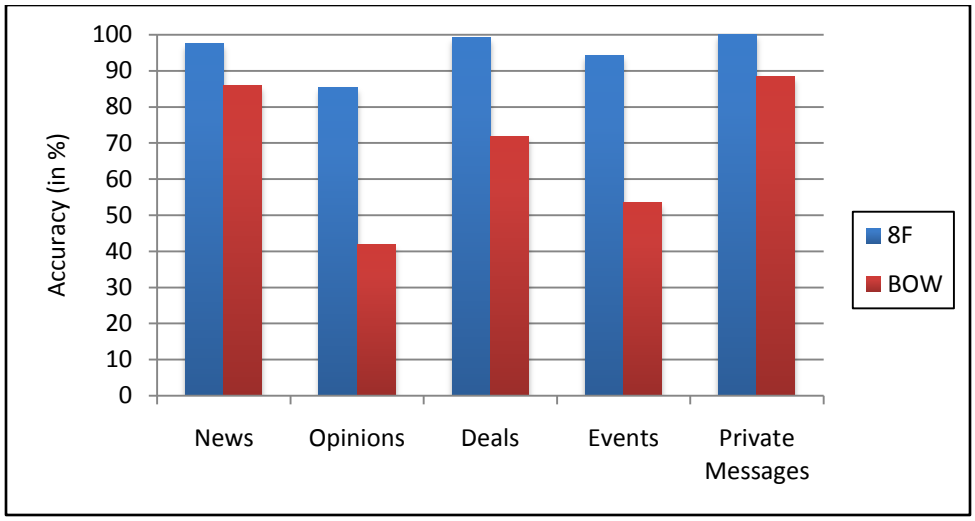

Figure 5.6: Accuracies per class using Naives Bayes

Figure 5.6 re-asserts that 8F performs consistently better than BOW for all five classes.

The model building time is a critical factor for any classification system. The time tends to be higher with higher dimensionality and this poses a problem for text classification since standard techniques like BOW result in a very high dimensionality. Figure 5.7 shows the model building time for 8F and BOW for various algorithms.

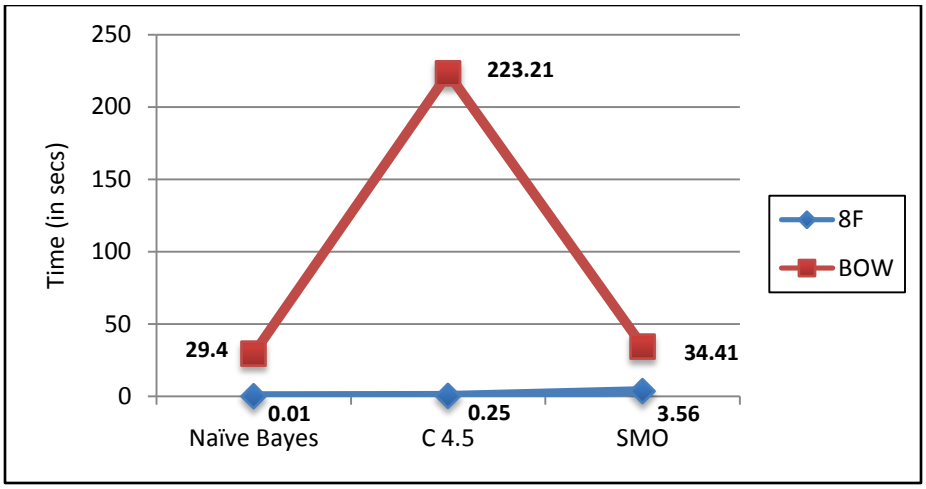

Figure 5.7: Model building time

From the above Figure, it is clear that BOW is not a popular choice for classification for large sets of documents. The number of words that were used in BOW was 6747 after removing stop words. With increase in number of tweets and hence increase in number of words (features), these times are bound to go up. 8F on the other hand has a fixed set of dimensions. Hence, 8F outperforms BOW by a very high margin with respect to model building time.

In BOW, misclassified tweets are mainly between news and private messages (383), news and opinions (407), whereas in 8F, they are mainly between news and opinions (104). We attribute this to the fact that tweets in news may also be opinionated. We believe that multi-label classification would resolve this issue to a certain extent.

5.2 Experimental Results for Extended Framework

The second set of results is based on addition of user-defined classes and user-defined features. For this purpose we collected 5292 tweets related to news, opinions, deals, events and private messages (henceforth referred to as Category 1 tweets). These tweets however were collected from the public time line and not from handpicked seed users. The distribution of tweets per category is shown in Figure 5.8. Apart from tweets belonging to the pre-determined classes, we chose six random user-defined categories. They were coffee, pizza, ipad, fitness, paintings and laptops. A total of 6537 tweets were collected comprising of these six categories (henceforth referred to as Category 2 tweets).

The distribution of tweets per the new category is show in Figure 5.9. We also add a new feature to the 8F feature set which is the tweet publication time information. Based on the time, we divide the tweet publication time into the following granularities – morning, afternoon, evening, night and mid-night. Hence for the next set of experimental results, we will refer to our proposed feature set as 9F. Experimental results are shown using the Naïve bayes classifier with 5-fold cross validation.

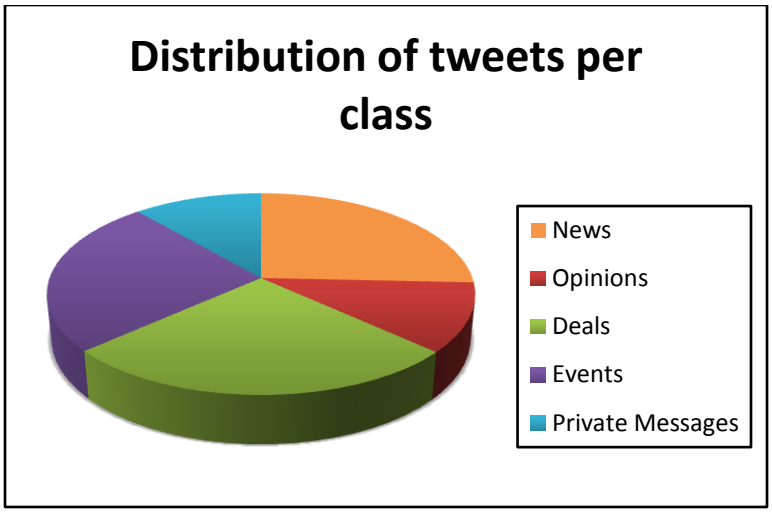

Figure 5.8: Distribution of tweets per class

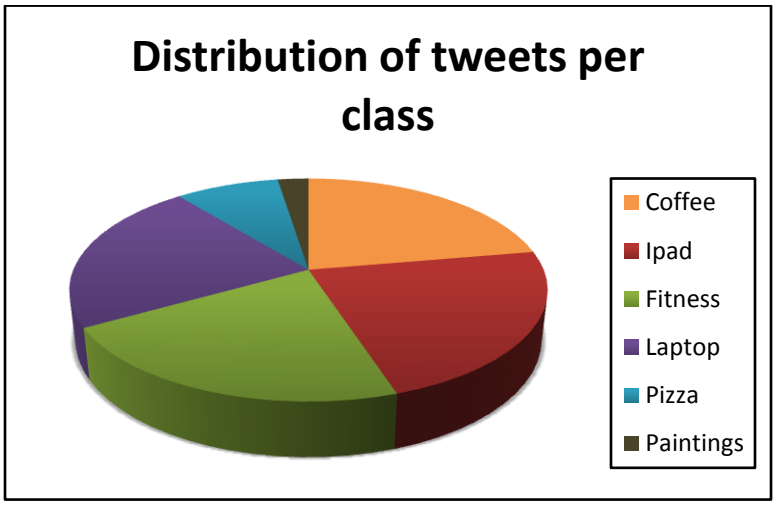

Figure 5.9: Distribution of tweets per user-defined class

Chapter 4 discussed the effect of the addition of new classes without adding the corresponding new features. Here, we run the classifier using the existing 9F feature set

on category 1 tweets. Since the tweets are collected from the public time line, we do not expect the same accuracy as what we observed in the first set of experiments. The reason being that the author feature is now sparse and author profile information contributes very little to the classifier performance. We also employ the baseline BOW strategy on the category 1 tweets. The respective accuracies are shown in Figure 5.10. The individual accuracies per class are shown in Figure 5.11.

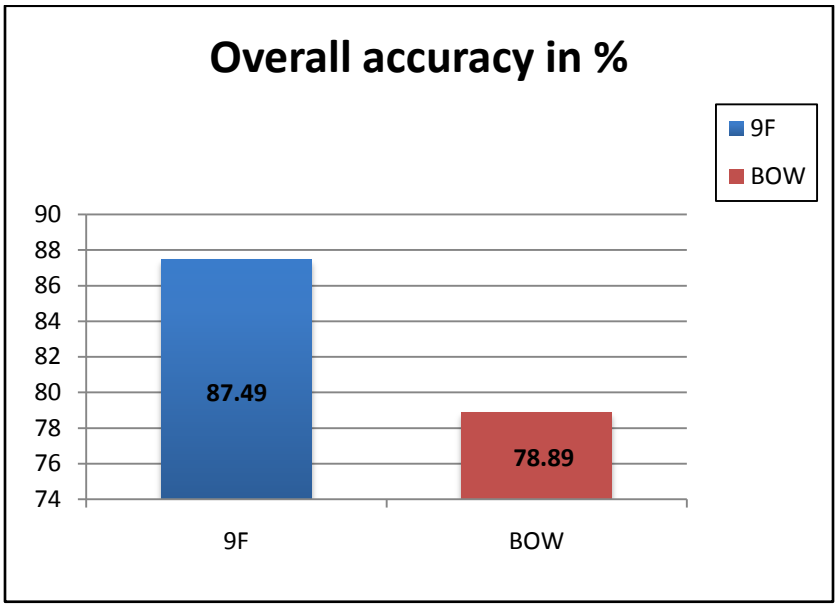

Figure 5.10: Overall accuracy for Category 1 tweets

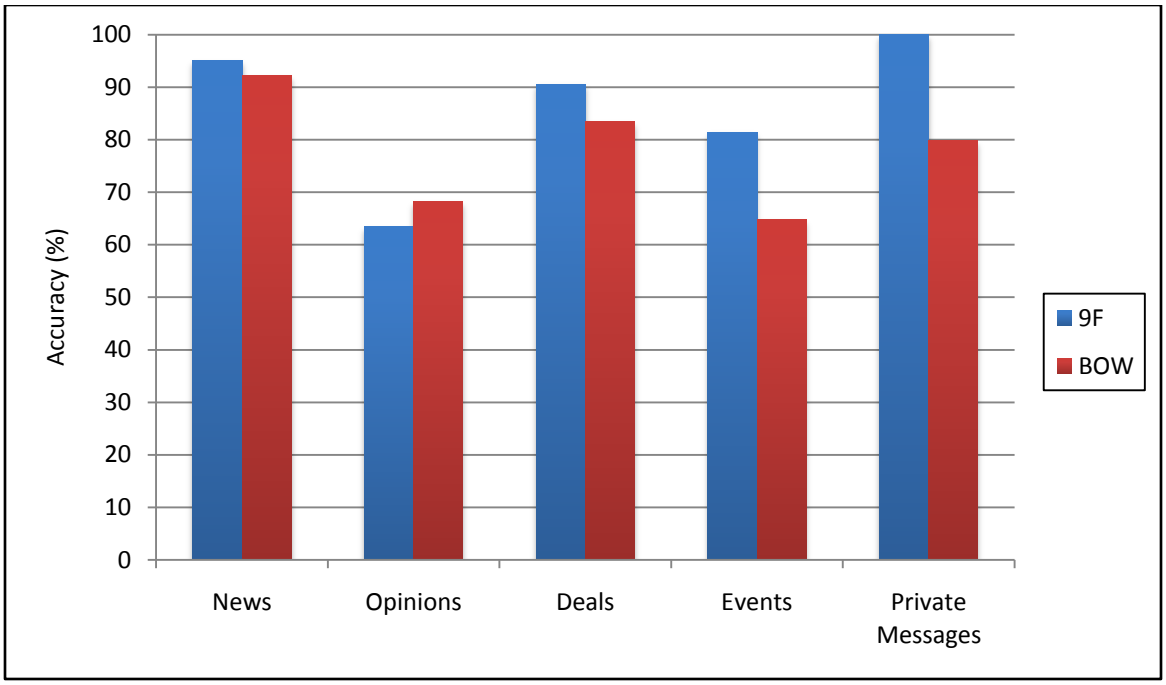

Figure 5.11: Individual accuracy per class for Category 1 tweets

For category 2 tweets, we ran the classifier using 9F and measured the accuracy. We performed the same experiment but with BOW as the feature set. We expect that the BOW strategy performs better than 9F in this case since 9F is tailored specifically to the pre-determined categories of news, opinions, deals, events and private messages. Experimental results confirmed our belief. The overall accuracies with the individual class accuracies are shown in Figure 5.12 and 5.13 respectively. We stated in the previous chapter that addition of new classes required addition of new features to discriminate the newly added class. Hence, we fed the system with 4-5 "good" sampled to diagnose the new feature. By "good", we created synthetic tweets that best define the feature corresponding to the new category. We again ran the classifier after adding six new features to 9F feature set. Each of these six features corresponded to the six new

classes added. Chapter 4 already discussed about how the new features were diagnosed by the system. The accuracies after integrating the new features is depicted in Figure 5.12. From the Figure, we observe that after the addition of the new features corresponding to the new classes, the accuracy improved. An important point to note here is that the performance of the system depends heavily on the samples the user provides as an input. Therefore, the quantity and quality of the samples is critical to the performance of the system. In our case, we observe that relatively few (4-5) good samples sufficed to achieve better accuracy than BOW model. Figure 5.12**,** we can also deduce that 9F with integrated features outperforms BOW by 7.66%.

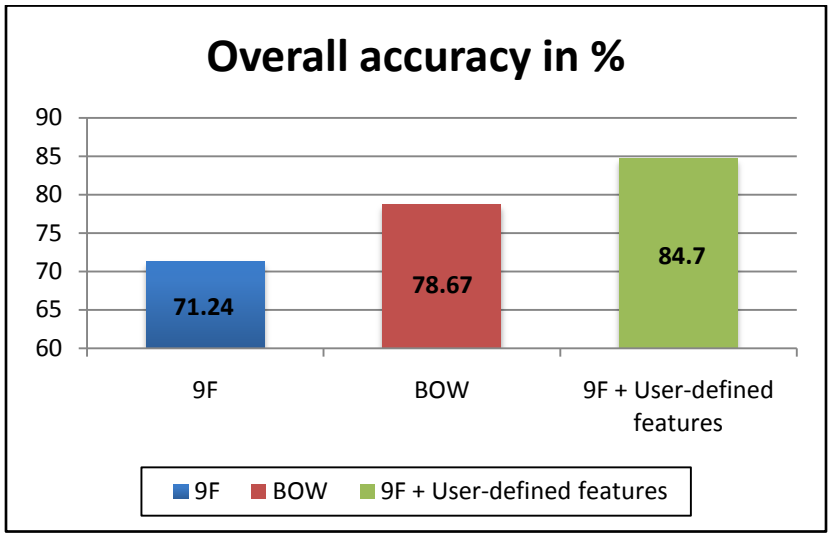

Figure 5.12: Overall accuracy for Category 2 tweets

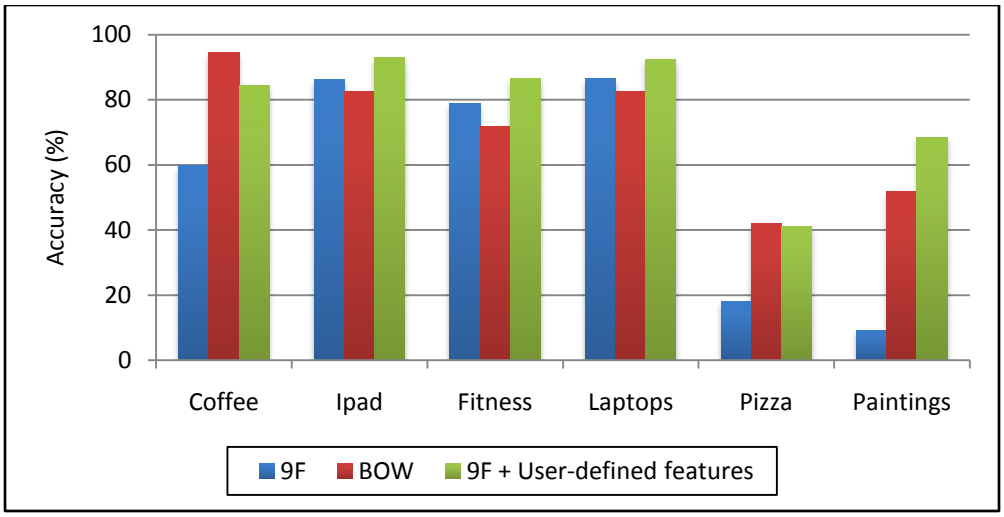

Figure 5.13: Individual accuracy per class for Category 2 tweets

Finally, we integrate category 1 and category 2 tweets collected into a single group and run the classifier using 9F, BOW and 9F integrated with user-defined features. The overall and individual accuracies are shown in Figure 5.14, 5.15 and 5.16.

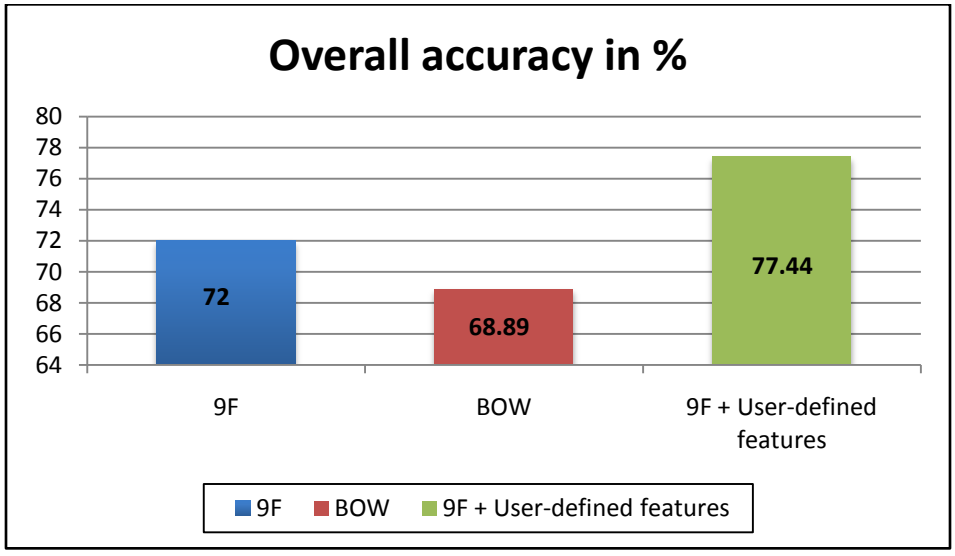

Figure 5.14: Overall accuracy for Category 1 and Category 2 tweets

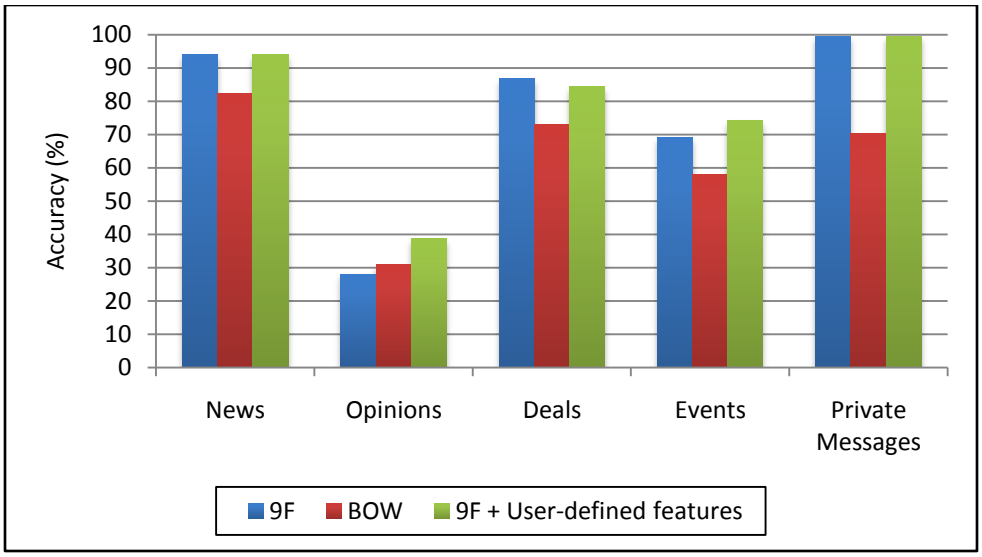

Figure 5.15: Individual accuracy per class for Category 1 & Category 2 tweets

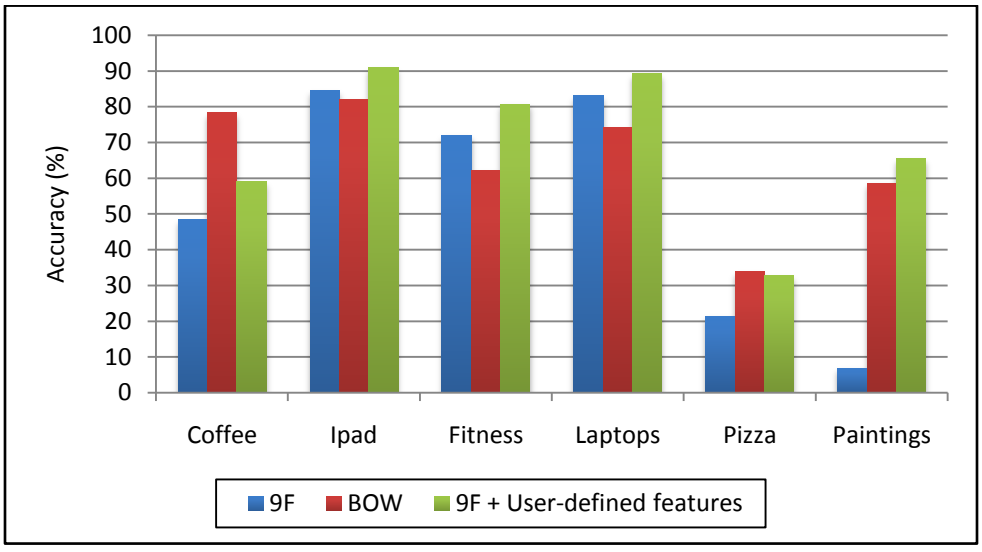

Figure 5.16: Individual accuracy per class for Category 1 & Category 2 tweets

As we can observe in Figures 5.15 and 5.16, 9F performs well on pre-determined classes but required the integration of user-defined features to outperform BOW model.

Based on out experimental results, we can conclude that instead of using the entire BOW model, one can use only those words that define the new classes and integrate it as a single feature with the existing 9F framework. We observe that this not only improves the accuracy but also reduces the model building time significantly.

Chapter 7: Conclusions and Future Work

The work described in this thesis is a step towards efficient classification of short text messages. Short text messages are harder to classify than larger corpus of text. This is primarily because there are few word occurrences and hence it is difficult to capture the semantics of such messages. Hence, traditional approaches like "Bag-Of-Words" when applied to classify short texts do not perform as well as expected.

Existing works on classification of short text messages integrate messages with meta-information from other information sources such as Wikipedia and WordNet. Automatic text classification and hidden topic extraction approaches perform well when there is meta-information or when the context of the short text is extended with knowledge extracted using large collections. But these approaches require online querying which is very time-consuming and unfit for real time applications. When external features from the world knowledge is used to enhance the feature set, complex algorithms are required to carefully prune overzealous features. These approaches eliminate the problem of data sparseness but create a new problem of the "curse of dimensionality". Hence efficient ways are required to improve the accuracy of classification by using minimal set of features to represent the short text.

We have proposed a framework to classify Twitter messages which serve as an excellent candidate for short text messages because of their 140 character limit. In this framework, we have used a small set of features, namely the 8F feature set to classify incoming tweets into five generic categories – news, opinions, deals, events and private messages. We have also extended this framework to allow users to define new classes based on their interest and experiment with new features to improve the performance of our system.

Here is a brief overview of how we have presented our arguments towards achieving our goal in this thesis:

- In Chapter 1, we defined the problem of text classification in general and specifically discussed the issues with short text classification.
- In Chapter 2, we briefly provided an overview of Twitter and explained the various concepts in Twitter. We also provided illustrations highlighting the need to mine Twitter's rich source of information.
- In Chapter 3, we provided our framework to classify incoming tweets into five generic classes, namely, news, opinions, deals, events and private messages by using only a small set of features captures from tweets (8F)
- In Chapter 4, we extended our framework to facilitate addition of new user-defined classes and user-defined features to the system.

- In Chapter 5, we analyzed the experimental results and compared the performance of our proposed approach with the baseline algorithm for both the initial and extended framework.

A “Perfect classifier” does not exist. It is always a compromise between several factors that are application dependent. However, the underlying goals of all classifiers are the same, higher accuracy and better speed. In this thesis, we have tried to achieve both the goals but there is scope for a lot of improvements.

We intend to use our approach with a multi-label classifier effectively. Initial results with multi-label classifiers look promising when we analyzed the probability distribution of misclassified tweets. Further analysis is required to effectively integrate a multi-label classifier with our system online.

Although the thesis mentions “short text”, we experiment with Twitter messages only. Our feature set is tailored towards various characteristics of tweets like presence of @, shortening of words etc. There is a need to adapt this approach to work well on other short text messages. We hope to come up with a generic framework that can perform consistently well on different types of short text messages.

There is a lot of scope to process tweets to capture better information; Crawling the tiny URL’s is one such approach. We currently do not crawl URL but discard the information.

We also plan to enhance our 9F feature set with more user profile information like location. It would be interesting to experiment with different granularities of tweet publication time, for example, month-wise, year wise, quarterly etc and analyze the accuracy. Another addition to the system could be to analyze the sentiment of tweets to differentiate positive and negative opinions. Although the current system is not online, there are several issues to consider about an online system. For example, when to re-train the classifier model is an important question to address. Re-training for every incoming tweet is inefficient. Alternately, we could consider re-training when a "Concept Drift" [37] is detected in the incoming data. Our vision is to build an online classifier to classify tweets robustly with high speed and accuracy with minimal set of features.

References

[1] A. Java X. Song, T. Finin, and B. Tseng, 2007. Why we twitter: understanding microblogging usage and communities. In *Procs* WebKDD/SNA-KDD '07 (San Jose, California, August, 2007), 56-65.

[2] N. Cohen. Twitter on the barricades: Six lessons learned. http://www.nytimes.com/2009/06/21/weekinreview/21cohenweb.html, Pub. June 20, 2009.

[3] M. Milian. Twitter sees earth-shaking activity during SoCal quake. http://latimesblogs.latimes.com/technology/2008/07/twitter-earthqu.html, Pub. July 30, 2008.

[4] http://dictionary.reference.com/browse/event

[5] http://www.twitter.com

[6] http://www.time.com/time/magazine/article/0,9171,1044658,00.html

[7] http://en.wikipedia.org/wiki/Micro-blogging

[8] http://www.facebook.com/

[9] http://www.orkut.com

[10] J. Sankaranarayanan, H. Samet, B. E. Teitler, M.D. Lieberman, J. Sperling, TwitterStand: News in Tweets. In Proc. ACM GIS'09 (Seattle, Washington, Nov. 2009), 42-51.

[11] http://en.wikipedia.org/wiki/Twitter

[12] "What's a Retweet?" Jill Kurtz, Socialmediatoday, http://www.socialmediatoday.com/SMC/96585

[13] “Twitter is Not a Micro-Blogging Tool” Steven Hodson, http://mashable.com/2008/07/18/twitter-not-a-microblogging-tool/

[14] Twitter Search. http://search.twitter.com/. Retr. July 1, 2009.

[15] HashTags. http://hashtags.org. Retr. July 1, 2009.

[16] Representational State Transfer http://en.wikipedia.org/wiki/Representational_State_Transfer

[17] Twitter API Wiki http://apiwiki.twitter.com/API-Overview

[18] Sariel Har-Peled, Dan Roth, Dav Zimak , Constraint Classification for Multiclass Classification and Ranking, Advances in Neural Information Processing Systems (NIPS) 2002,

[19] Christopher D. Manning, Prabhakar Raghavan & Hinrich Schütz, Introduction to Information Retrieval, e, 2008

[20] “Stemming”, http://en.wikipedia.org/wiki/Stemming

[21] M.F.Porter, An algorithm for suffix stripping, Computer Laboratory, Cambridge.

[22] D. Bollegala, Y. Matsuo, and M. Ishizuka, Measuring semantic similarity between words using Web search engines, , Proc. WWW, 2007

[23] M. Sahami and T. Heilman , A Webɲbased kernel function for measuring the similarity of short text snippets, , Proc.WWW, 2006

[24] “Improving similarity measures for short segments of text”. W. Yih and C. Meek, Proc. AAAI, 2007

[25] “WordNet – A Lexical Database for English”, http://wordnet.princeton.edu

[26] Phan, X.-H., Nguyen, L.-M., and Horiguchi, S , Learning to classify short and sparse text & web with hidden topics from large-scale data collections, In Proc. WWW (Beijing, China, Apr. 2008), 91-100.

[27] Hu, X., Sun, N., Zhang, C., and Chua, T.-S, Exploiting internal and external semantics for the clustering of short texts using world knowledge, In Proc. CIKM (Hong Kong, China, Nov. 2009), 919-928

[28] S., Ramanthan, K., and Gupta , Clustering short text using Wikipedia, Banerjee, A.. In Proc. SIGIR (Amsterdam, The Netherlands, July 2007), 787-788.

[29] P. Schonhofen , Identifying document topics using the Wikipedia category network, Proc. the IEEE/WIC/ACM International Conference on Web Intelligence, 2006.

[30] Google Sets API, Google Labs, http://labs.google.com/sets

[31] B. Heil and M. Piskorski, New Twitter research: Men follow men and nobody tweets, http://blogs.harvardbusiness.org/cs/2009/06/new_twitter_research_men_follo.html Pub. June 1, 2009.

[32] Gordon V. Cormack, José María Gómez Hidalgo, Enrique Puertas Sánz , Spam Filtering for Short Messages, CIKM 2007

[33] William Cohen, Text Classification, Tutorial, CMU, CALD Summer Course.

[34] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, Ian H. Witten, The WEKA Data Mining Software: An Update, (2009), SIGKDD Explorations, Volume 11, Issue 1.

[35] D. Metzler, S. Dumais, and C. Meek, Similarity measures for short segments of text. Lecture Notes in Computer Science, 4425:16, 2007.

[36] S. Osinski, J. Stefanowski, and D. Weiss. Lingo, Search results clustering algorithm based on singular value decomposition, In Proceedings of the IIS: IIPWM'04 Conference, page 359, 2004.

[37] "Concept Drift", http://en.wikipedia.org/wiki/Concept_drift

[38] "Curse of Dimensionality", http://en.wikipedia.org/wiki/Curse_of_dimensionality