Classification of Twitter Trends using Feature ranking and Feature Selection

---

A Thesis

presented to

the Faculty of the Graduate School

at the University of Missouri

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

by

Abhishek Shah

Dr. Wenjun Zeng, Thesis Supervisor

December 2015

The undersigned, appointed by the dean of the Graduate School, have examined the thesis entitled

Classification of Twitter Trends using Feature Ranking and Feature Selection

Presented by Abhishek Shah,

A candidate for the degree of Master of Science,

And hereby certify that, in their opinion is worthy of acceptance.

Professor Wenjun Zeng

Professor Toni Kazic

Professor Mike McKean

Acknowledgments

I would like to thank Dr. Wenjun Zeng for his tremendous support and guidance. He not only provided excellent guidance, but also asked tough questions and pushed for excellence. He took extra time out of his current job to provide guidance and assistance whenever necessary. Without his knowledge and guidance, this work would not have been possible.

I would also like to thank Dr. Suman Deb Roy, my mentor, who brought this problem to my attention in the first place. He was the one who showed the pathway to this research. His support and guidance has also been very useful.

I would like to acknowledge A. Zubiaga, D. Spina, V. Fresno and R. Martinez, who provided the dataset and whose results were used as a benchmark for our research.

Last but not the least, I would like to thank my parents, my brother and Nikita Bhatia for providing unconditional support and affection throughout my academic life.

Table of Contents

List of Figures

List of Tables

Abstract

Twitter scales 500 million tweets per day and has 316 million monthly active users. The majority of tweets are in the form of natural language. Using natural language makes it difficult to understand Twitter's data programmatically. In our research, we attempt to solve this challenge using various machine learning techniques.

This thesis includes a new approach for classifying Twitter trends by adding a layer of feature selection and feature ranking. A variety of feature ranking algorithms, such as TF-IDF and bag-of-words, are used to facilitate the feature selection process. This helps in surfacing the important features, while reducing the feature space and making the classification process more efficient. Four Naïve Bayes text classifiers (one for each class), backed by these sophisticated feature ranking and feature selection techniques, are used to successfully categorize Twitter trends. Using the bag-of-words and TF-IDF rankings, our research provides an average class precision improvement, over the current methodologies, of 33.14% and 28.67% correspondingly.

Chapter 1: Introduction

In recent years, with the sudden increase in popularity of various social networks, the way we produce and consume information has changed dramatically. There is a massive amount of information flowing through these social networks. It has forced news organizations, journalists, marketing companies, business organizations, musicians, actors, bloggers, programmers and almost all businesses and communities to change their approach to branding, marketing and networking. Twitter is one of the most popular of these social networks and it has been at the center of most of the discussion going around the world. It has 316 million active users and 500 million tweets are sent per day[1]. From music awards to presidential elections to weather disasters, everything gets discussed and debated on twitter. It is one micro blogging site that everyone uses to, either produce or consume information. Its members include a handsome list of politicians, celebrities, researches, and you name it. With such huge user base and the plethora of information flowing through it every day, Twitter has become an obvious choice for researchers around the world to find interesting information in real time.

The huge amount of data flowing through twitter can provide interesting insights like location based topics, breaking news, ongoing events, new product launches, interesting activities, etc. Information could spread via this medium and

[1] https://about.twitter.com/company

one would like to know where it all began and how it all unfolded and the sentiment of the people. The problem here is that this kind of data isn't readily available on twitter. Twitter has personal tweets and anecdotes amongst which this kind of information is hidden. This provides an amazing opportunity to researchers to find context amongst such complicated, yet interesting data. For example, a company might be interested in the discussion that is going around its newly launched product, people might be discussing about a live event that is going on around their neighborhood or government might be wanting to know about user sentiment for its newly introduced policies. All this information is flowing through twitter and it is vital that it gets organized. The first step to organize this information is to categorize them. This research is an attempt to take this first step.

In this research, we show that feature selection is extremely important for successfully classifying twitter trends and that it not only provides higher class precisions but can also help identify an appropriate number of features for each class. In order to prove this hypothesis, we use supervised machine learning to train a Naïve Bayes Classifier to classify twitter's trending topics. The dataset is provided by A. Zubiaga, D. Spina, V. Fresno and R. Martinez [1][2]. We classify the trends into four categories, news, meme, ongoing events and commemorative. We mine the textual data in the tweets (associated with each trend) to train our classifier. The experimental setup involves three major steps: 1.) Cleaning and preparing the data in the right format, 2.) Feature selection

[2] http://nlp.uned.es/~damiano/datasets/TT-classification.html

using two different techniques: i.) Bag-of-words and ii.) TF-IDF (Term Frequency-Inverse Document Frequency),

3.) Training and testing the classifier to successfully classify the trends into the four categories with the selected features.

The rest of the thesis is organized as follows. We start by giving some basic information about twitter and its terminology that we have been using until now, like tweets, trends, twitter, etc. Then we will discuss some related work that has been done in this area. Explaining the dataset, its source and how it is organized will follow this. Next, we will explain the experimental setup, feature selection and feature ranking techniques that we have used. Finally, we discuss and analyze the results, thereby concluding the thesis.

1.1 Background

Twitter

Twitter is a social media giant that was founded in March of 2006 in San Francisco, California. It has seen massive success and has a huge user base (316 Million active users) [3] with a huge amount of data being generated every day via the 140 character messages; they call tweets (500 Million tweets a day)[3]. There are a few reasons behind twitter's success; It forces the users to be creative with 140 characters tweet limit, information can spread very quickly with

[3] https://about.twitter.com/company

the concept of followers and the ease with each one can grow their connections.

Tweets

Tweets are 140 characters short messages that any users can post which is visible to their followers who can either favorite it or retweet it after which it becomes visible to their followers and they can do the same thing and this chain can go on. One can see how fast information can spread via this chain of retweets.

Followers

Followers are the users who follow other users on twitter. This concept has been a key part of twitter's success. Any user can follow anyone who has an account on twitter including all the celebrities. While one can follow anyone, the user being followed doesn't get any tweets from the followers unless he/she follows them back. For example, if a celebrity has a million followers but only follows 10 users, then that celebrity will only see tweets from those 10 users while the million followers will see the tweets from the celebrity.

Hashtags

Hashtags are the words that normally hint at the topic of the tweet. It can be thought of as giving some sort of the context to the tweets. Hashtags have the potential to become trending topics.

Trending Topics

Trending Topics are hashtags that a lot of people are talking about on twitter. In other words, these topics are being widely discussed on twitter and are very popular. Twitter updates the list of its ten currently trending topics momentarily. Twitter has not disclosed the method by which they determine these trending topics. Marketing companies, politicians, presidential candidates, journalists and almost all major businesses constantly monitor these trending topics. Finding context among these trending topics can provide some interesting and fascinating insights into the sentiments and thoughts of the people around the twitter sphere. In order to get this context, one has to go through the tweets that contain these trending topics and try and find meaning out of it. For example, whether this topic is related to an ongoing event or news or politics, etc. But what if we can automate this process and classify these trends into their respective categories in real time as they appear on twitter? This thesis attempts to answer this question using machine learning and natural language processing.

Table 1 shows some of trending topics and a tweet related to it. It is worth noting the natural language in the tweets. Finding context among such noisy data can be really difficult.

Trending Topic	Sample Tweet
#LovatoIsMyStrength	LovatoIsMyStrength, because the lyrics of her songs are inspiring & full of strength + haters i will punch you in the face w/ that strength
#FACH	Toshiba Camileo SX900 Full-HD Camcorder (14 Megapixel, 9-fach opt. Zoom, 6,9 cm (2,7 Zoll) Display, Bildstabilisator) silber - Preissucher O
#HowardDavies	Howard Davies has resigned as director of LSE because feels he gave the college the wrong advice on taking Gaddafi money.
#mileyonsnl	c'mon we make it Trending ! #MileyonSNL #MileyonSNL #MileyonSNL #MileyonSNL #MileyonSNL #MileyonSNL #MileyonSNL #MileyonSNL #MileyonSNL

Table 1: Trending Topic Examples with a Sample Tweet

1.2 Related Work

Text mining has always been of interest to researchers. It particularly got some more traction with the introduction of social networks. However, the textual data on these social networks is in a form of natural language, which includes a lot of urban slangs and abbreviations. Therefore, understanding them became more challenging and fascinating.

This thesis draws its main inspiration from the work of A. Zubiaga, D. Spina, V. Fresno, and R. Martinez [1] where they also classify the twitter trends in real time. However, our approach differs in many different ways: i.) We incorporate rigorous data preprocessing and clean up. ii.) We use feature selection facilitated by two feature weighting techniques, TF-IDF and bag-of-

words. iii.) We use a simple Naïve Bayes Classifier as oppose to Support Vector Machines. Juan Ramos [4] discusses some interesting results about feature weighting for text documents using tf-idf score. However, their research mainly deals with a more formal language. In our dataset, the tweets are filled with special characters, grammatical errors and urban slangs. We will demonstrate, certain limitations and roadblocks that TF-IDF gets into when data is filled with such noise, when we discuss the top words for each category of trends. Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts [5] use word vectors (vector space model) for sentimental analysis on IMDB movie review data. We have organized our data into a similar model, which is eventually used to train and test the classifier. Walter Daelemans and Ve´ronique Hoste [3] explain the forward selection feature selection technique for classifying sparse data. This thesis uses forward feature selection to demonstrate the effectiveness and accuracy of our approach.

Chapter 2: Classification System and Dataset

2.1 System Overview

Figure 1 is the end-to-end design of our classification system. We begin with a dataset consisting of tweets related to a trending topic and their labels (categories). The dataset is sent through a rigorous pre-processing stage where the punctuations, hyperlinks, emoticons and stop words are removed from the tweets. Clean data is then sent to the feature ranking stage where they are ranked with two different feature ranking techniques bag-of-words and TF-IDF. Once the features are ranked top k features are selected and the dataset is prepared in appropriate format to train and test the classifier. Once the classifier is trained, test data is passed through it and a predicted label for a trend is generated.

The necessity of such rigorous data preprocessing in our system (figure 1) arises out the fact that twitter's data isn't like other textual data, like news articles, reviews, etc. It's an informal text filled with hyperlinks, emoticons, abbreviations, urban slang, numbers, etc. In order to find context amongst such noisy dataset, one cannot resort to traditional natural language processing concepts, of lemmatization and part-of-speech tagging because they don't follow any grammar rules, nor do they have any standard abbreviations.

TF-IDF and Bag-of-Words are very popular feature ranking techniques. However, they are mostly useful in a less noisy dataset. We believe that

surfacing important features, is extremely important in such a noisy dataset and these ranking techniques can prove to be very important. In order to do that, we use feature-ranking techniques. Once, the features are ranked, feature selection becomes the natural next step.

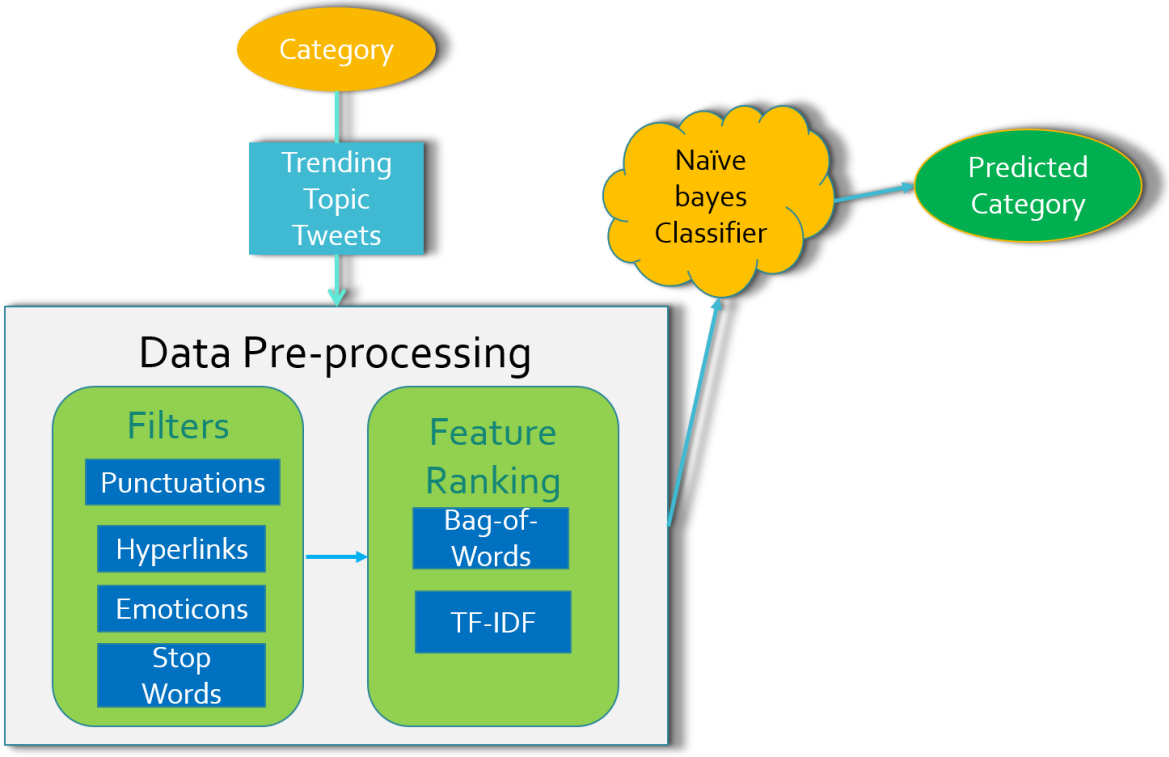

Figure 1: An overview of the end-to-end classification system

The dataset used for this research has been provided by, A. Zubiaga, D. Spina, V. Fresno and R. Martinez [1]. The dataset contains 1036 unique trending topics and a total of 567,452 tweets. All these tweets are written in 28 different languages with majority of them in English. An average of 548 tweets are associated with each of these trending topics.

2.2 Data Collection

The data was gathered using twitter's publicly available API. Twitter momentarily updates its top ten trending topic list. There is no information as to how a topic gets chosen to appear in this list or how often this list gets updated. However, one can request up to 1500 tweets for a given trending topic. The authors (A. Zubiaga, D. Spina, V. Fresno and R. Martinez [1]) had two processes running to collect this data. One process requested a list of trending topics from twitter every 30 seconds and maintained a unique list. Whenever there was a new trending topic detected, the other process requested a list of related tweets from twitter using its search API.

After the data was collected, the trending topics were manually annotated into the following four categories:

1. News
2. Meme
3. Ongoing Event
4. Commemorative

Three annotators were used to annotate the trending topics. Each one of them looked at the tweets related to the trending topics to assign a suitable category. If the tweets were in the language that the annotators didn't understand, Google Translate[4] was used to convert them in the language of their understanding. In order to measure the inter-annotator agreement, the authors of the dataset computed the Fleiss' Kappa coefficient [2].

[4] https://translate.google.com

Above annotation process yielded the following distribution of 1,036 trending topics categories:

- 616 ongoing events
- 251 memes
- 142 news
- 27 commemorative

Let's look at a brief description for each of these categories.

2.3 Trending Topic Categories

News

As discussed above, in recent years Twitter has become the center of discussion, debates, opinions, sharing and marketing interesting things. This ability of Twitter's to spread information with its incredibly simple concept of retweeting has made it very attractive to the news organizations. Almost all major news outlets are active members of Twitter and their ultimate goal is to spread their news articles or news in general via tweeting. This automatically makes this category of trending topic one of the most common and attractive amongst others. Sometimes a normal user on twitter notices something happening before the news organizations and this gives rise to the possibility of news breaking on twitter before the professional news organizations can get to it. Furthermore, the news organizations can get to the breaking news from twitter.

Meme

These topics can be thought of as some sort of catchy taglines that are intended to resonate with the users to support a movement, product, presidential candidate, celebrity or just about anything that has been a topic of discussion around the world. Here are a few examples of a meme:

- A trending topic called “DeflateGate” dominated the twitter space shortly after 2015 Super Bowl. This trending topic was about the controversy that the super bowl winning team, New England Patriots, led by Quarterback Tom Brady, allegedly cheated by deflating the footballs so that the wide receivers can catch the ball relatively easily.
- Another trending topic called “Kony2012” became very famous because of its extremely noble cause. It was a campaign to capture a warlord, Joseph Kony, in Uganda by the end of 2012.

The world of twitter is filled with such interesting memes and hence, this is one of the categories in the dataset.

Ongoing Event

These kind of trending topics are discussions going around live events like concerts, sports, elections, etc. Normally, these topics appear when a big sports game is going on or something out of the ordinary happens during a popular event.

Commemorative

These topics are understandably the least frequent since they are commemorating a certain event or person in the history. Nevertheless, they are popular enough to be in a separate category of their own.

Trending Topic	Category	Description
Angry Birds Game Coming	News	Angry birds is a very popular game whose CEO announced that it is coming to Facebook
WeAdoreLovato	Meme	This topic was trending when actor/singer Demi Lovato announced her new album
Mileyonsnl	Ongoing-event	Miley Cyrus was performing on the Saturday Night Live show and twitter space exploded because of it.
#happybirthdayjbj	Commemorative	It was singer-songwriter Jon Bon Jovi's Birthday and his fans on twitter were congratulating him for the same

Table 2: Example and description of each category of trending topic

2.4 Data Pre Processing

The goal is to train a text classifier to classify twitter trending topics into four different categories; News, Meme, Commemorative and Ongoing-event

However, to get to that step a lot of preprocessing is required since the data is filled with noise of special characters, emoticons, hyperlinks, punctuations etc. Here are the steps we took to prepare the data for classification:

1. Data collection from the source[5] and organization
2. Data clean up
3. Feature Selection
    a. Feature weighting

2.5 Data organization

The data extraction from the source was a two-step process. First, the trending topic file was provided with trending topics, their manually annotated label and an md5 hash ID to identify tweets associated with the topic. Second, one needed to programmatically access the tweets associated with these trending topics using the md5 hash ID. Using this method we organized the data into four comma separated files (one for each category of trend topics). The next step was to clean up the data and organize the data into a Vector Space Model, i.e, create a vector of word features for each trending topic.

2.6 Data Cleanup

Since we are dealing with natural language it is very important to find out the keywords in the tweets amongst the noise of stop words, punctuations, urban slangs, hyperlinks, emoticons, numbers and twitter specific vocabulary. So once

[5] http://nlp.uned.es/~damiano/datasets/TT-classification.html

the data was organized, all the tweets were broken down into words and sent through a rigorous multi-step filtering process:

Step 1: Stop Words Filter

All the English stop words like the, is, that, was, etc were removed. The stop words list was provided the NLTK python library[6] to which twitter specific words like; RT (Retweet), etc. were removed. Because of majority of data being in English and other languages not explicitly laid out, we stuck to removing only English stop words.

Step 2: Hyperlinks Filter

A lot of tweets include hyperlinks that the user wishes to share. For our work such hyperlinks could really cause a problem when we start weighing our features for feature selection and that could lead to incorrect classification. So filtering them was critical

Step 3: Emoticons and integers Filter

Since there is a 140 characters limit, users tend to use emoticons and numbers to express their thoughts concisely. The emoticons are Unicode characters and the numbers aren't helpful as far as features are concerned. So removing them made sense.

[6] http://www.nltk.org

Step 4: Punctuation Filter

Tweets being in natural language, punctuations are a natural part of it. Since we are only interested in actual words as features removing them was also important. It is worth noting that traditional NLP techniques are lemmatization, lexical annotations and parts of speech tagging are ineffective in our case since the users are not expected to follow any grammatical rules especially because of the character limit. For Twitter users, getting the message across is more important.

Chapter 3: Feature Selection

In order to perform feature selection, we first needed to get insights about the features. We need to find the outliers amongst our feature space, the ones that can hint at a class more than other features can. In order to do that we need to weigh these features using certain specific criteria. We use two feature-weighting methods to do this: bag-of-words and TF-IDF (Term Frequency Inverse Document Frequency).

3.1 Feature Ranking

1. **Bag-of-Words**

   This is one of the very popular, common and simple, yet an extremely effective way of weighing features. The idea is to simply count the number of times each word has appeared in the tweets of a particular trend and then sort them in descending order. This method is particularly effective in a less noisy dataset. This is another reason why we have made a conscious effort to perform a rigorous data cleanup.

   Table 3 summarizes some of the top features that the bag-of-words feature ranking method produces.

News	Meme	Ongoing-event	Commemorative
google	making	alex	pie
howard	winning	viktor	yearoftheboss
sirah	deano	hatch	hongstarday
mclobster	miller	drucker	zico
angry	dhsmemories	panel	bon
ubuntu	damnitstrue	haley	rodney
facebook	winner	baby	jensen
phil	drinkmytearstonight	catherine	choi
eldar	gucci	hee	complimentendag
prince	unlimited	smokie	serge

Table 3: Top 10 features using bag-of-words approach for each category

2. TF-IDF

Term Frequency – Inverse Document Frequency (TF-IDF) is an extremely powerful measure to rank features. A TF-IDF score finds outliers amongst a huge feature space. The idea behind TF-IDF score is that term frequency or frequency count by itself isn't a good measure to weight a particular feature; It is equally important to know how many times the same feature appears in other classes. This additional information will help in identifying the features that exclusively identify a class. Because of this promising motivation, TF-IDF ranking technique is one of most widely used ranking techniques in text mining and information retrieval. Juan Ramos [4] shows how one can use TF-IDF ranking to perform effective query retrieval.

In our work, we use TF-IDF to rank the words for each category of trends. However, we had to group the words for each category first and then treat that as a document. So basically we had four documents, one for each category of trends.

Term Frequency (tf)

Term frequency is a measure of how many times a term appears in a particular document.

Equation (1) explains how we calculate the term frequency for the words.

$$\boldsymbol{tf(t,d) = \frac{f(t,d)}{total\ words\ in\ (d)}} \quad \textbf{(1)}$$

Where,

f(t,d) = frequency of term t in document d.

Inverse Document Frequency (idf)

$$\boldsymbol{idf(t,D) = log\ \frac{|D|}{1 + |\{d \in D : t \in d\}|}} \quad \textbf{(2)}$$

[7] Equation provided by https://en.wikipedia.org/wiki/Tf–idf#cite_note-4

Where,

|D| = Total number of documents

|{d ε D : t ε d}| = Number of documents (d) in which term (t) appears.

We add plus 1 to the denominator to make sure that we don't get division by zero error if the term doesn't appear in any of the documents.

Term Frequency-Inverse Document Frequency

$$\boldsymbol{TF - IDF = tf(t,d) \times idf(t,D)} \quad \textbf{(3)}$$

The above equation finally gives us the tf-idf weight for a term t. The higher tf-idf score is achieved with a higher term frequency (tf) and a lower document frequency. This tells us the uniqueness of a term for a particular document.

Table 4 shows the top ten terms tf-idf terms for each category. A lot of words are similar to the ones that were found by the bag-of-words approach, however there are some words that reflect the noise among the data. For example, words like "digital\u306b\u58f2\u5374" in the news category and "lv42040645" in ongoing-event category. These examples show the drawback of TF-IDF. Because of the inverse document frequency, these noisy words become a part of the top features list. This limitation will become more and more evident as we decipher the classification results. Such a limitation isn't part of the bag-of-words approach since it only considers the frequency count of the words.

News	Meme	Ongoing-event	Commemorative
murphy	tigerbloodintern	baby	pie
tarapaca	drinkmytearstonight	viktor	complimentendag
gwent	miller	lv42040645	serge
supreme	ninewest	Indiana	helau
digital\u306b\u58f2 \u5374	damnitstrue	haley	yearoftheboss
alprazolam	thataintwinning	eugene	bootyappreciationday
google	5bestsoundtracks	maca\xe9	fach
facebook	lovatoismystrength	berbatov	mamonas
jimmy	zach	rachel	zx81
mclobster	draft	derek	hongstarday

Table 4: Top 10 features using TF-IDF for each category

3.2 Forward Selection

Now that we have feature weights, feature selection becomes a natural next step. We use forward selection as used by Walter Daelemans and Veronique Hoste [3] in their work. This feature selection method is very simple yet very effective. You start with a small set of features and keep adding more features until the classifier accuracy plateaus or starts declining.

Without feature selection, the feature space becomes very large. For this data it can be as large as 512,943 features as reported by A. Zubiaga, D. Spina, V. Fresno and R. Martinez [1]. One can comfortably say that a lot of these features are not required for classification. In fact they can prove to be nothing

more than noise, as we will show in our results. Moreover, such a huge feature space can affect the computation time heavily, thereby rendering the system inefficient.

Chapter 4: Classification

Until now we have described, the organization of the data, it's preprocessing, feature weighting and feature selection, now we will discuss the classification methodology.

4.1 N-different classifiers

Similar to A. Zubiaga, D. Spina and V. Fresno, R. Martinez [1] we use one-against-all binary combination method (Rifkin, R. and Klautau, A. [7]) to classify our data. In other words, instead of using a single multi-class classifier to classify the trends into n different classes, we use n different classifiers, one for each class. Therefore, in the training phase each of the n classifiers learn from a model that helps them distinguish a class from the rest of the k-1 classifiers. For our dataset, this approach resulted in the following four classifiers:

1. news vs not news (this includes meme, ongoing-event, commemorative)
2. meme vs not meme (this includes news, ongoing-event, commemorative)
3. ongoing-event vs not ongoing-event (this includes meme, news, commemorative)
4. Commemorative vs not commemorative (this includes news, meme, ongoing-event)

4.2 Training and Testing Dataset

1/4th of the data used for testing and the rest for training. We will explain in detail how each of these sets were created.

For each of the classifiers, the training set was created by choosing 3/4th of the feature-selected data for the category in question and 1/4th from each of the other categories. For example, if we are training the classifier for news category then 3/4th of the labeled news trends' tweets were used for training and 1/4th were used for testing. Similarly, 3/4th of the non-news trends' tweets (meme, commemorative and ongoing-event) were used for training and 1/4th for testing.

Once the training and test sets were created, the classifier was trained with the training set and then test set was passed through to classifier to get the accuracy measure of the classifier. The test data set was prepared in the exact similar way. The feature ranking and feature selection techniques were also applied to the test dataset. So once a new trending topic appears all its tweets are gathered from twitter and all those tweets go through the same data filtering and feature ranking stage and then is passed to the classifier for classification.

4.3 Naïve Bayes Classifier

Unlike the work of A. Zubiaga, D. Spina and V. Fresno and R. Martinez [1] who uses Support Vector Machines (SVM) as their classifier, we chose to use to Naïve bayes classifier, primarily because we believed that having a cleaner and more organized data was more important than the classifier itself. Naïve bayes proved to be a better choice when it comes to text classification as shown by

Sundus Hassan, Muhammad Rafi and Muhammad Shahid Shaikh [6]. Moreover, since we treat the features as independent of each other, Naïve Bayes becomes an obvious choice. Next we describe the inner workings of the text classifier.

The Naïve Bayes classifier is a Bayesian classifier that has its roots in Bayes theorem (see the equation below)

$$\boldsymbol{P(c \mid t) = \frac{P(t \mid c)P(c)}{P(t)}} \quad \textbf{(4)}$$

Where,

P (c | t) = Probability of trend t belonging to class c (Posterior)

P (t | c) = Likelihood of generating trend t given class c

P (c) = Probability of occurrence of class c

In our case since we have word vectors as our features, the above equation changes as follows:

$$\boldsymbol{P\,(c \mid t) = \frac{P(word1 \mid c)\ \times\ P(word2 \mid c)\ \times \ldots \times\ P(wordN \mid c)\ P(c)}{P\,(t)}} \quad \textbf{(5)}$$

Using the above information, the naïve bayes classifier reaches a decision in the following way:

$$C_{MAP} = argmax_{c \in C} P(t \mid c)\ P(c) \quad \textbf{(6)}$$

$$= argmax_{c \in C} P(word1, word2, \ldots, wordN \mid c) P(c)$$

Where,

MAP = maximum a posteriori, i.e the most likely class.

4.4 Bayesian Network

Given, the above information, one can easily visualize a simple Bayesian network as shown in figure 1. As one can see that while each word is related to the trending topic category, they are independent of each other. In other words, the words (features) are not related to each other. This feature independence is at the core of every Bayesian Network.

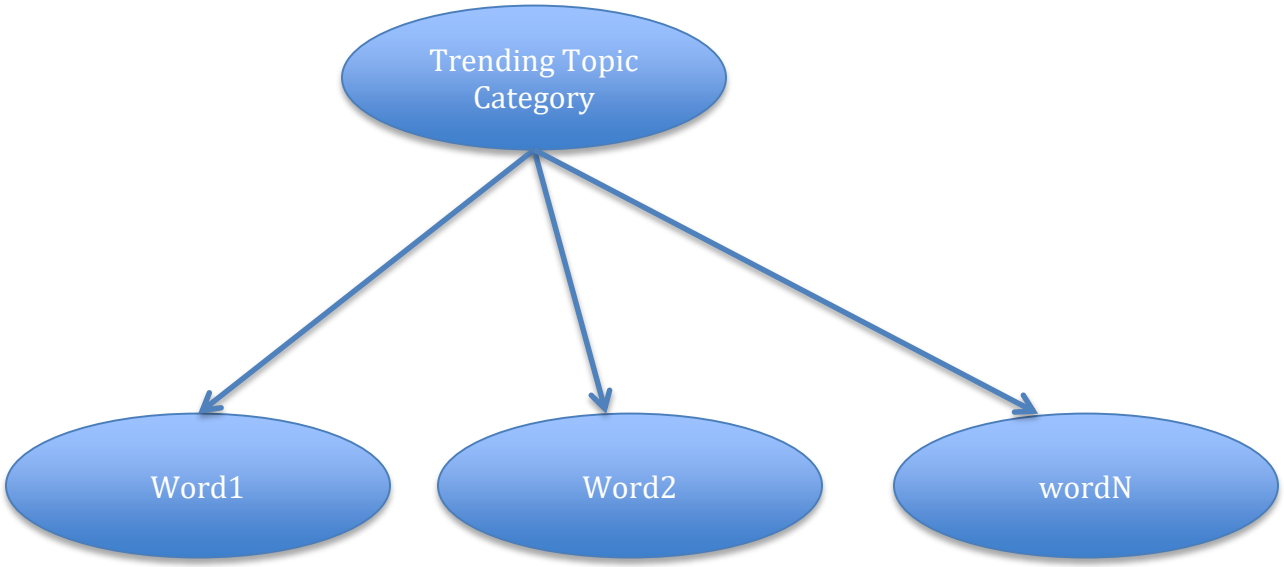

Figure 2: Bayesian Network for trending topic classification

We have used python's TextBlob[8] library's Naïve Bayes Classifier. In addition to the classifier, the library provides some useful functionality for data cleanup and pre-processing in general.

[8] https://textblob.readthedocs.org/en/latest/

Chapter 5: Results

We will now look at our results and compare them with some of the existing work of A. Zubiaga, D. Spina, V. Fresno and R. Martinez [1].

5.1 Bag-of-Words Ranking Analysis

Table 5 shows class precisions for different number of features selected after being ranked using frequency count. The class precision is quite satisfactory for each class. The commemorative class has the highest frequency, however, it is important to note that there are only 27 examples of commemorative class, which means there is high chance of overfitting and not very reliable. Ongoing-event class has 616 examples and provides a very good insight into the performance of our approach. There are two notable patterns in these results: 1.) The precision of most of these classes slowly increases as we add more features, plateaus or drops after a certain number of features. 2.) The peak precision value for most of these classes is reached somewhere between 20 and 30 features (exception of news), adding any more features shows either negative or no effect on the precision value. Both of these patterns support our hypothesis that feature selection is extremely important for text classification and adding more features after a certain number will only add noise to the classifier, which will have none to a negative effect on the results.

# Of Features	News	Meme	Ongoing-event	Commemorative
2	80.67	79.90	65.80	97.39
5	80.67	80.20	73.20	97.39
10	81.78	81.04	74.74	97.39
20	81.41	81.04	79.92	97.39
30	82.15	80.66	80.29	97.39
40	82.15	80.66	77.69	97.39
50	82.89	80.66	78.81	97.39

Table 5: Class Precision (%) for features selected using Frequency Count

Now, if we compare our best results to similar research done by A. Zubiaga, D. Spina, V. Fresno, R. Martinez [1] in table 6, we see how our results almost always gives better results.

Results	News	Meme	Ongoing-Event	Commemorative
Without Feature Selection [1]	67.3	63.6	78.3	54.8
Feature Selection with FC	82.89	81.04	80.29	97.39

Table 6: Class precision comparison. Feature Selection (bag-of-words) vs No Feature Selection

Following is the percentage improvement in class precision based on the based on the numbers in table 6:

- News = 23.2
- Meme = 27.4

- Ongoing-Event = 2.5
- Commemorative = 77.7

Figure 6 summarizes the comparison between results with and without feature selection. It provides a much clear picture of the comparison of class precisions between feature selection, done using bag-of-words feature ranking and non-feature selection.

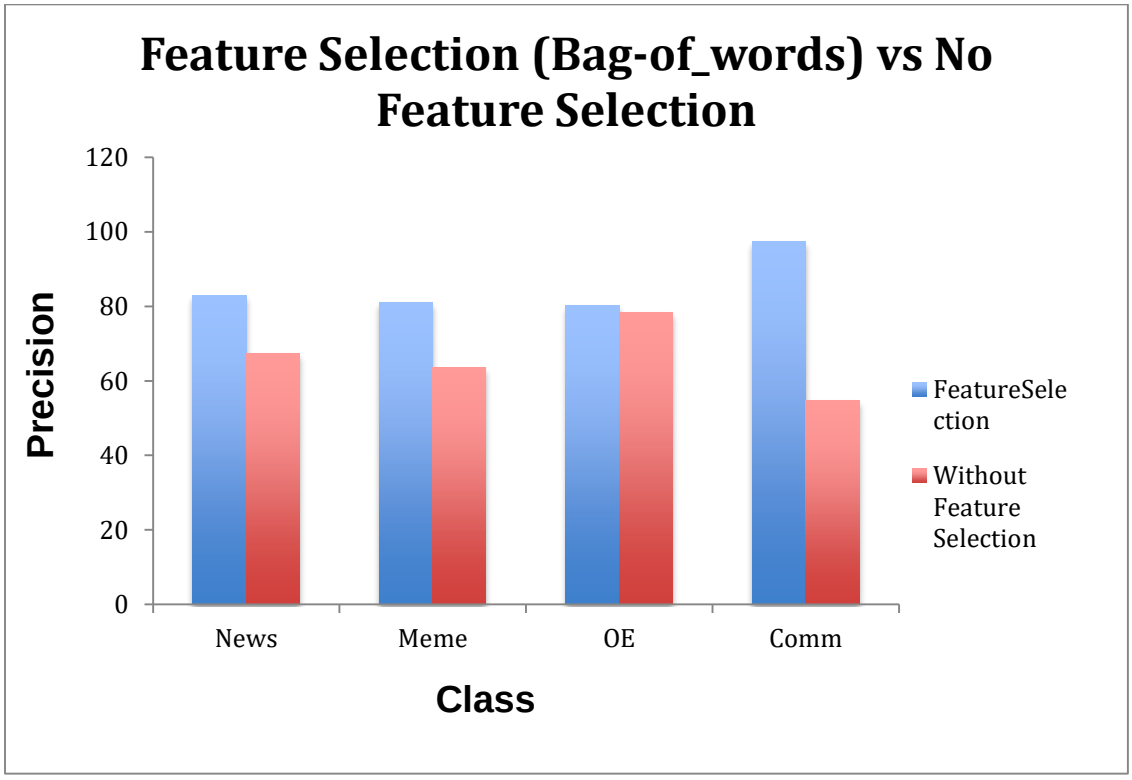

Figure 3: Feature Selection (Bag-of-words) vs. No Feature Selection

5.2 TF-IDF Ranking Analysis

Table 7 summarizes the results of the class precision gathered using TF-IDF ranking for feature selection. Just like with bag-of-words approach, the class precision is quite high for most classes with commemorative class having the

highest amongst all and Ongoing-event being the lowest. This further strengthens the argument about overfitting the data in the case of the commemorative class.

TF-IDF ranking results have lower class precision than bag-of-words approach. This is the effect one of the limitation of TF-IDF that we discussed in chapter 4. We can see words like “digital\u306b\u58f2\u5374” and “maca\xe9”, popping up in the top words list. In spite of rigorous data cleaning process, some of these words sneak pass the filters and due to the inverse document frequency of TF-IDF, they become unique to classes. These words introduce noise into the dataset, thereby affecting the precision values for each class.

# Of Features	News	Meme	Ongoing-event	Commemorative
2	80.67	79.92	67.60	97.39
5	80.67	79.92	68.40	97.39
10	81.04	80.29	70.26	97.39
20	81.04	80.29	68.77	97.39
30	81.41	80.29	70.26	97.39
40	81.41	80.29	68.77	97.39
50	81.04	79.92	68.40	97.39

Table 7: Class precision (%) for features selected using TF-IDF ranking

Now, if we compare our results to similar research done by A. Zubiaga, D. Spina, V. Fresno, R. Martínez [1] in table 8, we see how our results almost always gives better results. The only case where we get a less precision is in the ongoing-event class.

Results	News	Meme	Ongoing-Event	Commemorative
Without Feature Selection [1]	67.3	63.6	78.3	54.8
Feature Selection with TF-IDF	81.41	80.29	70.26	97.39

Table 8: Feature Selection (TF-IDF) vs No Feature Selection

Following is the percentage improvement in class precision based on the numbers in table 6:

- News = 21
- Meme = 26.2
- Ongoing-Event = -10.3
- Commemorative = 77.7

Figure 7 summarizes the comparison between the results with and without feature selection. It provides a much clear picture of the comparison of class precisions between feature selection, done using TF-IDF feature ranking and non-feature selection.

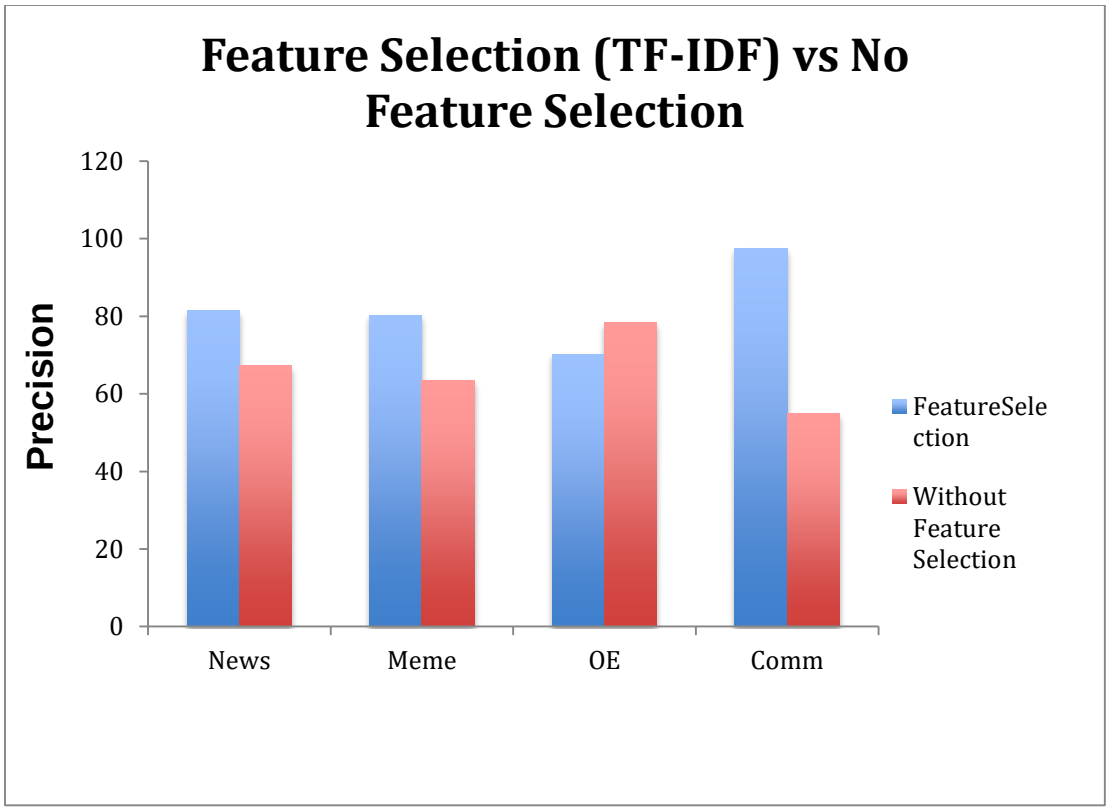

Figure 4: Feature Selection (TF-IDF) vs. No Feature Selection

5.3 Bag-of-Words vs TF-IDF

Figures 8-11 compare the performance of Bag-of-Words vs TF-IDF feature ranking methods and provide insight about the trend of precision change with different number of features.

In all the cases Bag-of-Words perform better than or as good as TF-IDF. In fact the following figures summarizes the percentage improvements for each class on an average:

- News: 1.1
- Meme: 0.7
- Ongoing-event: 14.6

- Commemorative: 0

TF-IDF can’t equate the word with its plural [2]. This can be a significant limitation for a large dataset. We discussed another limitation of TF-IDF in chapter 4 where certain unfiltered words cropped up in the top words list due to the inverse document frequency measure of TF-IDF. The effect of these limitations is evident when we compare TF-IDF with Bag-of-Words and particularly when we look at the results of ongoing event class where bag-of-words performs significantly better than TF-IDF.

5.4 Class Precision Analysis

Looking at the trend of precision change, for both Bag-of-Words and TF-IDF as we increase the number of features, it becomes quite evident that adding more features only increases the performance up to a certain point. Adding more features after a certain point adds nothing but noise to the dataset. For example, for Ongoing-event class precision in figure 10, we can see when we start with 2 features the precision is about 65.8%, then with 5 it becomes 73.20%, 74.72% with 10, peaks at 20 features with precision value of 79.92% and starts dropping from 30. Similar trend is observed in almost all the classes, which confirms our initial argument that without feature selection, most of the features that will be used for classification will provide no additional information that a few meticulously ranked and selected will. In fact, these results prove that they will have a negative effect more often than not.

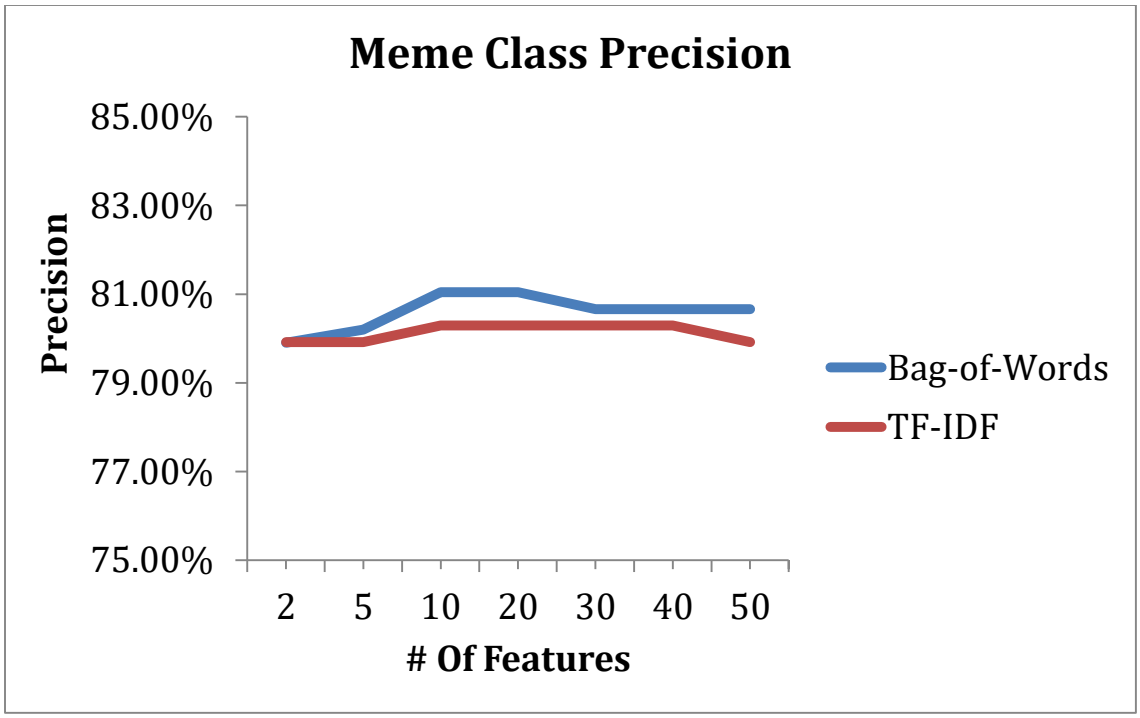

Figure 5: Meme class precision

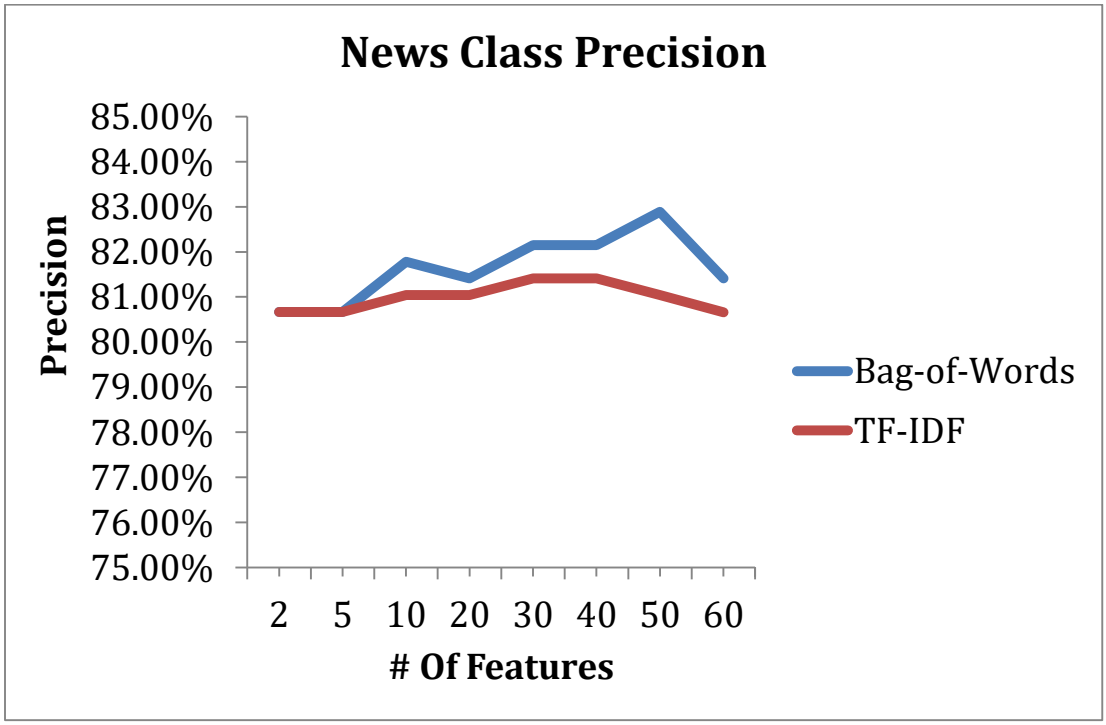

Figure 6: News class precision

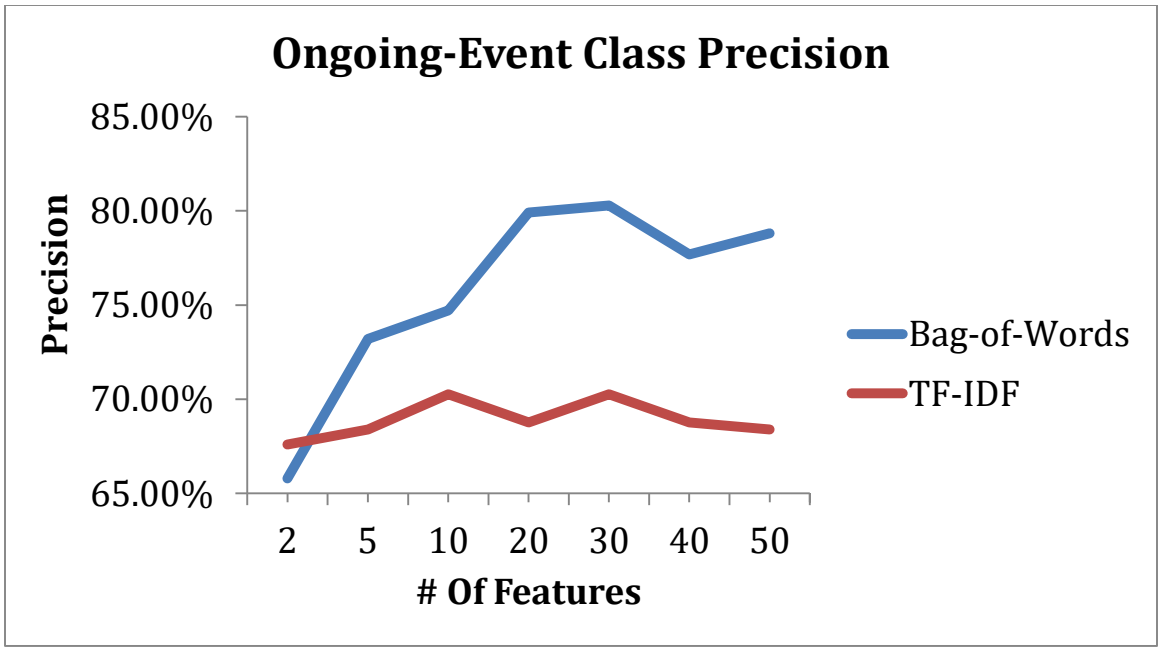

Figure 7: Ongoing-event class precision

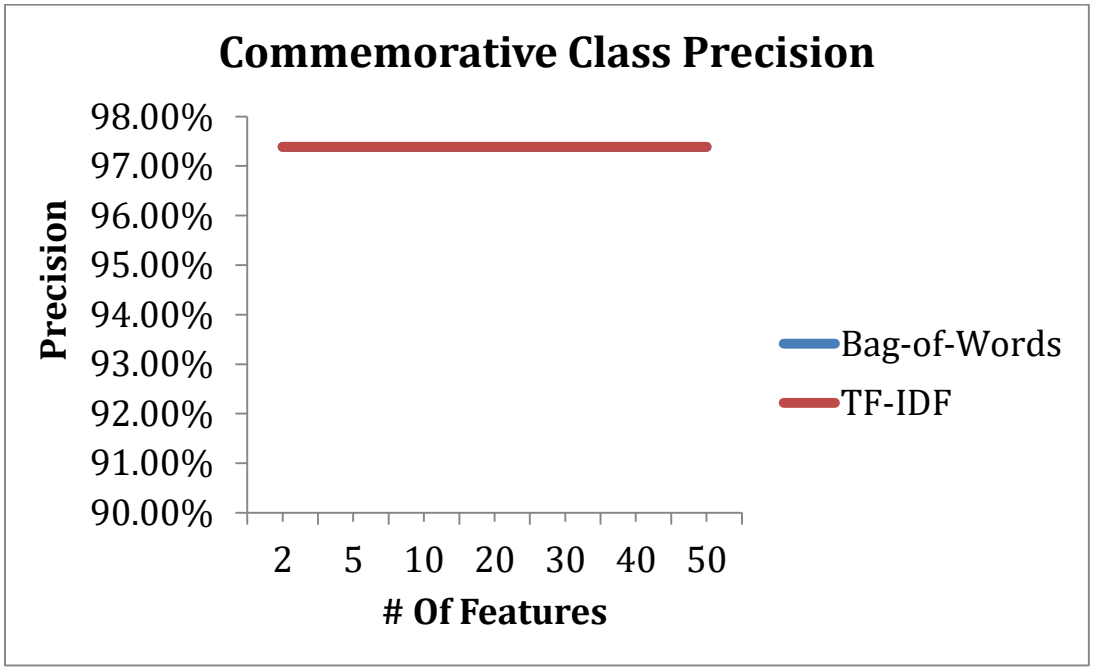

Figure 8: Commemorative class precision

Chapter 6: Discussion and Future Work

6.1 Discussion

Our research has tried to improve upon the work of A. Zubiaga, D. Spina, V. Fresno, and R. Martinez [1] by adding a layer of feature selection and feature ranking. This area of study is not only interesting for the computer science and Natural Language Processing community but is also appealing to news media organizations, marketing companies, startups, etc. The four classes that we discussed in our work can provide some fascinating information to these groups. Breaking news can be captured early on if we can start categorizing the trending topics as they appear on twitter. This could provide a great platform for the news media organizations. An ongoing-event can be captured and people can join one of their interests if it is going on around their area. It can also provide live information about people's opinion about the event. Commemorative trends can give a look into people's sentiments about a particular person or event. All these analytics can be captured and discussed live with real time classification of twitter trends. With twitter increasingly becoming "the" social media platform for information sharing and discussions such classification systems can prove to be extremely valuable.

Over and above just classification, this system can also work as a filtering system that can filter out the noise amongst the large amount of data flowing through twitter. Using such a classification system, one can surface the important

trends and filter out the unwanted ones. For example, catching breaking news is more important than some Internet meme that has gone viral. A news organization can do that by looking at the just the trending topics for news, which such a classification will automatically surface as it happens on Twitter.

Impact on News Media

Classification systems like ours can also help in presenting the news to the audience in a better way. Surfacing important and relevant news to an individual or an organization can be facilitated by a system like ours if one can monitor and mine data like, user interests, time spent on reading news related to a particular category of trends, author bias, etc. Obviously there are security and privacy measures that needs to be addressed in such a system. This can change the way the news media organization produces news. This can also help surface news from one's own archives and see if there were any relevant stories published for a particular category of trending topics in the past that can provide some context to the present trends.

6.2 Recognition

In 2014, we used our feature selection technique in University Missouri's RJI (Reynolds Journalism Institute) Tech Competition. We used the top words (from tweets) generated by our feature selection technique as queries to retrieve news articles, audios and videos from a publicly available public media API called PMP[9]. We then ranked those articles against the trending topics using a

[9] https://support.pmp.io/docs

ranking algorithm we developed to provide the most relevant articles to our users.

Our work won the “Technical Innovation Award” and won the grand prize of a fully expense paid trip to Washington D.C and an Apple watch. This work can be applied to a lot of similar applications. The keywords generated by our technique can be a great first step to get context amongst a topic.

6.3 Future Work

Above results have opened up some very interesting opportunities. While it proves that feature selection is extremely essential in a text based classification system, more research is required in finding the optimal number of features. This is particularly tricky when we are dealing with natural language. Each platform has a vocabulary that is quite different than others, for example twitter’s vocabulary of retweets (RT), user mentions (@username), and favorites, etc does not exists in other social networks like Facebook. If we move away from social networks and a pick a news organization then the vocabulary completely changes. Given these challenges, it would be really interesting to know if there is an optimal number of features that a text based system can use that probabilistically guarantees a higher classification accuracy.

Feature selection is incomplete without feature ranking, it would be interesting to know if there is a particular feature selection method other than Bag-of-Words and TF-IDF that outperforms all other feature selections methods. We mentioned earlier that TF-IDF couldn’t differentiate between a word and it’s plural and also surfaces certain junk words due to its Inverse Document

Frequency Measure. These limitations open up some more opportunities in terms of data pre-processing stage where one can add some additional features. One can also add some filters after the feature ranking and analyze the results.

Chapter 7: Conclusion

In this work, we have touched variety of topics like natural Language Processing, Text Classification, Feature selection, Feature ranking, etc. Each one of these topics was used to leverage the massive information flowing through twitter. Understanding twitter was as important as knowing the topics in question. After all the background knowledge was acquired, we used the dataset provided by A. Zubiaga, D. Spina, V. Fresno, and R. Martinez ([1]). After cleaning up the data and making it go through a rigorous filtering process of emoticons, punctuations, and stop words filter, we ranked them using two different ranking methods: (i) Bag-of-Words and (ii) TF-IDF and selected top k features. We fed these features and the class associated with them to four Naïve Bayes Classifiers (one for each class).

The results of the previous experiments, led us to the conclusion that feature selection is an absolutely necessity in a text classification system. This was proved when we compared our results with a system that uses the exact same dataset without feature selection. We were able to achieve 33.14% and 28.67% improvement with bag-of-words and tf-idf scoring techniques correspondingly. Our feature selection technique utilized forward selection (Walter Daelemans, Ve´ronique Hoste. [3]) on features ranked using two feature ranking techniques: (i) Bag-of-Words and (ii) TF-IDF.

Bag-of-Words is quite straightforward where one just needs to count the number of times a word appears in a particular document. TF-IDF, on the other

hand, is a little more involved. It is the measure of how unique a word is to a particular document. In other words, at a very high level, TF-IDF score is a result of frequency count of a word in a particular document divided by the number of times it appears in other documents. We also compared the performance of these two feature-ranking techniques and it turns out that Bag-of-Words performs slightly well than TF-IDF. However, feature selection provided better class precision than no feature selection.

We also mentioned recognition and some opportunities that our work provides in the fields of news media, marketing and businesses in general. We hope that our work can provide a good foundation to the future of text classification in social media and to the opportunities that comes with it.

References

[1] A. Zubiaga, D. Spina, V. Fresno, R. Martinez.
Real-Time Classification of Twitter Trends
Journal of the American Society for Information Science and Technology (JASIST). In Press.

[2] Fleiss, J. L.
Measuring nominal scale agreement among many raters.
Psychological bulletin, 76(5):378, 1971.

[3] Walter Daelemans, Ve´ronique Hoste.
Evaluation of Machine Learning Methods for Natural Language Processing Tasks.
CNTS Language Technology Group, University of Antwerp UIA, Universiteitsplein 1 (bldng A), B-2610 Antwerpen, Belgium.

[4] Juan Ramos.
Using TF-IDF to Determine Word Relevance in Document Queries.
Department of Computer Science, Rutgers University, 23515 BPO Way, Piscataway, NJ, 08855.

[5] Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts.
Learning Word Vectors for Sentiment Analysis.
Stanford University, Stanford, CA, 94305

[6] Sundus Hassan, Muhammad Rafi, Muhammad Shahid Shaikh.
Comparing SVM and Naïve Bayes Classifiers for Text Categorization with Wikitology as knowledge enrichment.
NUCES-FAST, Karachi Campus.

[7] Joachims, T.
Text categorization with support vector machines: learning with many relevant features. In *Proceedings of ECML-98, 10th European Conference on Machine Learning*, pages 137–142, Chemnitz, DE, 1998. Springer Verlag, Heidelberg, DE.

[8] Rifkin, R. and Klautau, A.
In defense of one-vs-all classification.
The Journal of Machine Learning Research, 5: 101–141, December 2004.

[9] Vivek Narayanan, Ishan Arora, Arjun Bhatia.
Fast and accurate sentiment classification using an enhanced Naïve Bayes Model.
Department of Electronics Engineering, Indian Institute of Technology (BHU), Varanasi, India