DETECTING REAL-TIME CHECK-WORTHY FACTUAL CLAIMS IN TWEETS RELATED TO U.S. POLITICS

by

FATMA DOGAN

Presented to the Faculty of the Graduate School of
The University of Texas at Arlington in Partial Fulfillment
of the Requirements
for the Degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE & ENGINEERING

THE UNIVERSITY OF TEXAS AT ARLINGTON
December 2015

ACKNOWLEDGEMENTS

First and foremost, I would like to sincerely thank my advisor, Dr. Chengkai Li. Dr. Li, you are an outstanding professor and one of the highlights of my education at UTA. Without your excellent guidance, caring, patience, and constant help, this thesis would not have been possible.

I would also like to thank my committee members Dr. Bahram Khalili and Dr. Dimitrios Zikos for serving as my committee members and for their invaluable comments, suggestions, and supports.

A special thanks to all members of Dr. Li's research group, especially Naaemul (Naffi) Hassan, Minumol Joseph, and Sumesh Balalapan. To Naffi, who never hesitated in helping me out and always delivered beyond my expectations. Thank you. To Minu and Sumesh, who have always supported me with the many great ideas and encouragement. Thank you.

I would also like to express my deepest gratitude to my mother and father for everything that you have done for me. Words cannot express how grateful I am to you. Your prayer for me was what sustained me thus far.

Finally, I would like to express appreciation to my beloved husband, Hakan. You were always my support. Whenever I felt I was at a dead end, your insight and encouragement helped me push through.

November 24, 2015

ABSTRACT

DETECTING REAL-TIME CHECK-WORTHY FACTUAL CLAIMS IN TWEETS RELATED TO U.S. POLITICS

Fatma Dogan, MS
The University of Texas at Arlington, 2015

Supervising Professor: Chengkai Li

In increasing democracy and improving political discourse, political fact-checking has come to be a necessity. While politicians make claims about facts all the time, journalists and fact-checkers oftentimes reveal them as false, exaggerated, or misleading. Use of technology and social media tools such as Facebook and Twitter has rapidly increased the spread of misinformation. Thus, human fact-checkers face difficulty in keeping up with a massive amount of claims, and falsehoods frequently outpace truths. All U.S. politicians have successively adopted Twitter, and they make use of Twitter for a wide variety of purposes, a great example being making claims to enhance their popularity.

Toward the aim of helping journalists and fact-checkers, we developed a system that automatically detects check-worthy factual claims in tweets related to U.S. politics and posts them on a publicly visible Twitter account. The research consists of two processes: collecting and processing political tweets. The process for detecting check-worthy factual claims involves preprocessing collected tweets, finding the check-worthiness score of each tweet, and applying several filters to eliminate redundant and irrelevant tweets. Finally, a political classification model distinguishes tweets related to U.S. politics from other tweets and reposts them on a created Twitter account.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

Chapter 1

INTRODUCTION

Political fact-checking has come to play a significant role in increasing democracy and improving political discourse [1]. Politicians make claims about the facts all the time. As Hassan et al. stated in their paper, claims about facts made by politicians are frequently “false, exaggerated, and misleading” because of “careless mistakes and even deliberate manipulation of information” [2]. Thus, the need for fact-checking political claims has led to institutes and platforms dedicated to revealing the truth about the claims.

Use of technology and social media tools such as Facebook and Twitter has rapidly increased the spread of misinformation. Therefore, human fact-checkers have been facing the difficulty in keeping up with a massive amount of claims since fact-checking is a tedious and time-consuming task. This thesis aims to help journalists and fact-checkers by providing a publicly open Twitter account, which retweets real-time political check-worthy factual tweets. To the best of our knowledge, there is no Twitter account that automatically finds and shares check-worthy factual claims in political tweets.

Twitter, a microblogging service, is one of the most powerful and widely used social networking sites. As of November 2015, Twitter has over 320 million monthly active users [3], and 500 million tweets per day, on average, are tweeted [4]. Clearly, Twitter has a huge influence on people. According to Tumasjan et al. [5], Twitter became a legitimate communication channel in the political arena after Barack Obama successfully used Twitter for his 2008 US presidential campaign. Nowadays, almost all U.S. politicians have Twitter accounts, so that they are able to easily reach mass audiences. Furthermore, they make use of Twitter for a wide variety of purposes including: to make claims about facts to enhance their popularity; to mobilize their supporters; to convince potential electorate; to convey the messages of their political campaigns; to disseminate information about their meetings, interviews, speeches, and news published about them; and to raise funds for their political campaigns.

In this study, we developed a system that automatically detects check-worthy factual claims in tweets related to U.S. politics and posts them on a publicly visible Twitter account. We first constructed a list of political keywords and a list of political user ids as the predicate parameters in using the Twitter Public Streaming API to collect real-time political tweets. We then applied ClaimBuster [6] on the tweets. ClaimBuster is a tool for finding check-worthy factual claims in presidential debates. Furthermore, we developed several filters to eliminate redundant tweets and non-check-worthy tweets that ClaimBuster misinterprets as check-worthy due to misleading features of tweets such as overused punctuation marks and numbers. To built a classifier that differentiates tweets related to U.S. politics from other tweets, we randomly selected and labeled a dataset of tweets among all tweets that passed all the filters we created. We then trained and tested several classification models using the labeled dataset. Experiment results demonstrated the promising accuracy of the models. Furthermore, the most effective features in the models have been identified and analyzed.

Overall, the contribution of this research is as follows:

- A set of political keywords and a set of political ids were constructed.
- Several filters were either created or implemented to detect check-worthy factual claims and to eliminate irrelevant and redundant tweets.
- Classification models were trained and tested on human-labeled ground truth tweets. The most effective features in the models have been identified.
- A publicly visible Twitter account has been created to retweet detected check-worthy factual claims.

This thesis work is organized into eight chapters. Chapter 2 discusses related works. Chapter 3 gives a brief explanation of the architecture of the system. While Chapter 4 gives details about the first phase of the architecture, Data Preparation, Chapter 5 explains the approaches that have been taken to detect check-worthy tweets and eliminate falsely detected non-check-worthy and redundant tweets. Furthermore, Chapter 6 explains the approaches taken to distinguish U.S. political tweets from others. Chapter 7 discusses experiments and evaluation.

Chapter 2

RELATED WORKS

As stated before, Twitter is one of the most popular social media tools among politicians. While scholars have predominantly conducted studies on forecasting elections by mining tweets, there are studies that focus on the use of Twitter by politicians. However, most of these existing studies have motivated on the use of Twitter by members of Congress in the USA. For instance, [7] analyzed contents of more than 6000 tweets posted by Congress members. The authors indicated that Congress members primarily used Twitter to report their daily activities and to disseminate information about themselves. Gulati and Williams [8] investigated the use of social media tools in the 2010 U.S. Congressional Election. They found that most of the major party candidates for the House of Representatives had Twitter accounts, and there were no party differences for adopting Twitter. According to another study of Williams and Gulati [9], the party and campaign resources are important dynamics to lead members of Congress to use Twitter extensively. One another study, [10] found that Congress members' adoption of Twitter were related to following reasons including: if they belonged to a minority party; if their party leaders urged them to use it; if they served in the Senate; or if they were young. One another study that also focused on the adoption of Twitter claimed that Republicans were likely to adopt Twitter more than Democrats [11].

Chapter 3

SYSTEM OVERVIEW

In this chapter, we introduce the design of the system that we developed for detecting political tweets with check-worthy factual claims. The system is composed of two components; data preparation and data processing. We provide a brief overview of each component in this chapter. However, in further chapters we give detail information about each component. Figure 3-1 shows the workflow of the system of this study.

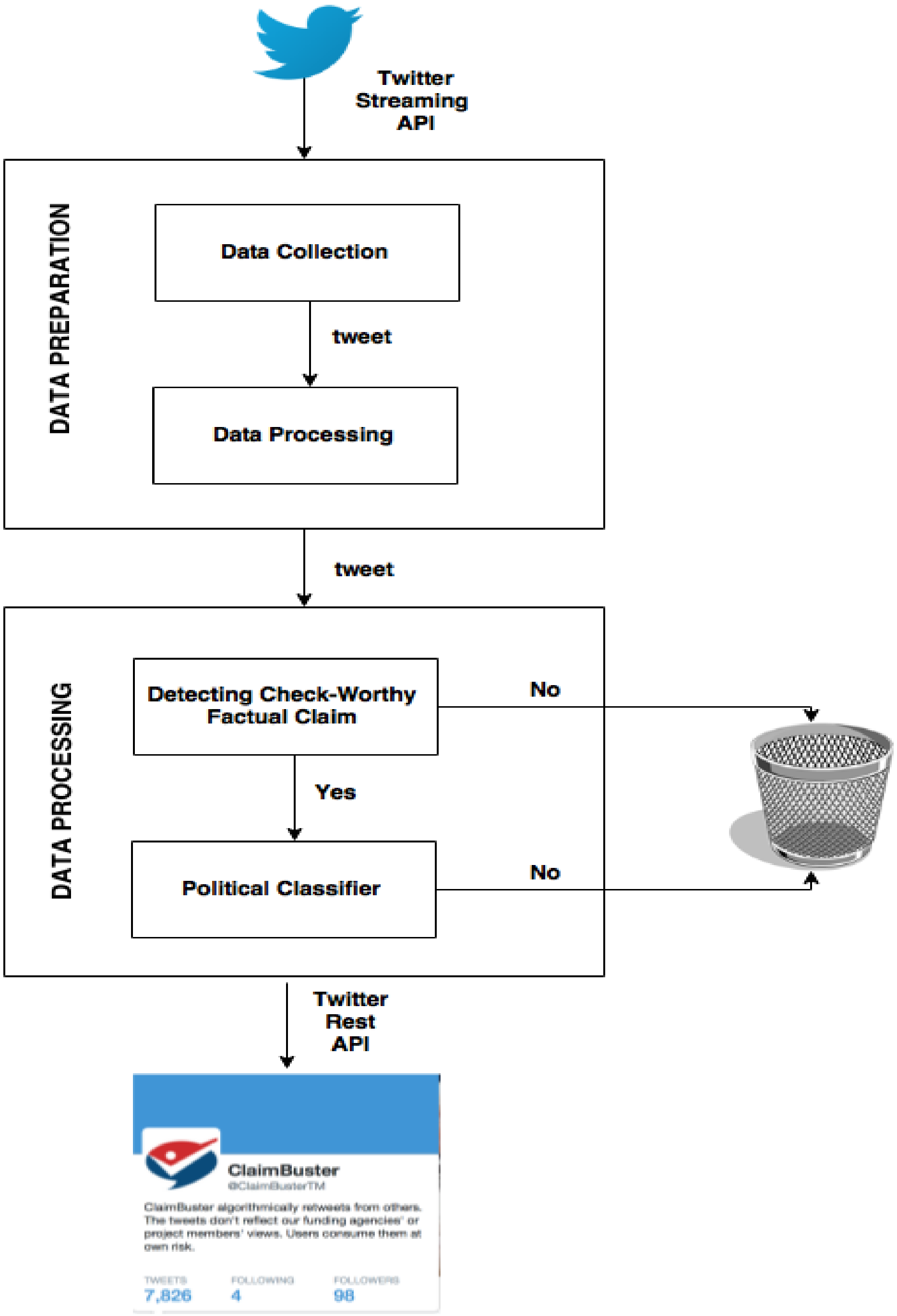
The data preparation phase of the system includes two stages; data collection and data preprocessing. In the data collection stage, we are collecting real-time tweets from the Twitter global stream by using the Twitter public Streaming API. To use the public Streaming API, at least one predicate parameter must be specified. We determined predicate parameters as political keywords and user ids for political Twitter accounts. We implemented a python program for collecting tweets. After tweet collection phase, the data preprocessing phase starts to extract the necessary entities of the tweet and prepares the text entity of the tweet for further processes.

The data processing phase contains two stages, detecting check-worthy factual claims in tweets and distinguishing tweets related to U.S. politics than other tweets. To detect check-worthy factual claims in tweets, we used ClaimBuster [6], which is a tool developed to detect important factual claims in presidential debate sentences. It calculates the check-worthiness score of the sentence ranging from 0 (extreme unimportant) to 1 (extremely important) to distinguish check-worthy factual sentences from either unimportant factual sentences or non-factual sentences. ClaimBuster provides an API to its users to find the score of given sentence or sentences. We used ClaimBuster API to get a score for each tweet. We determined a threshold as 0.75 for ClaimBuster' scores. If a tweet has scored 0.75 or higher, we consider it as a candidate tweet for further phases. Otherwise, we filtered it out.

ClaimBuster is built on presidential debate transcripts. While the language of presidential debate transcript is a formal language, the language used in tweets shows expressiveness and spontaneity of the spoken language. The difference between these two types of languages causes ClaimBuster to misinterpret not-check-worthy tweets as check-worthy. We call this kind of tweets as junk tweets. To eliminate junk tweets and redundant tweets, we created several filters. These filters are constructed based on rules that we derived from tweets.

In this study, we only interest in detecting check-worthy factual claims in tweets related to U.S. politics. However, among the collected tweets, we have tweets for other countries politics and unpolitic tweets as well. We model this problem as a classification task. We used a supervised learning approach to tackle the problem. We constructed a labeled dataset of tweets that are high-scored and passed all filters. We then trained and tested several supervised learning models by extracting many features from the labeled dataset.

Finally, we created a twitter account for this project. We used the Twitter REST API to retweet the tweet that passed all filters and classified as a U.S. political tweet by the political classifier that we developed.



ClaimBuster algorithmically retweets from others. The tweets don't reflect our funding agencies' or project members' views. Users consume them at own risk.

Figure 3-1 Workflow of the system

Chapter 4

DATA PREPARATION

4.1 Data Collection

Twitter provides APIs to researchers and practitioners to get access to its data. Twitter APIs are classified as Streaming APIs and REST APIs based on their designs and access methods. In terms of desired type and amount of information to be retrieved, both types of APIs have their limitations and capabilities. While the REST APIs use pull strategy to retrieve data, the Streaming APIs use push strategy. Unlike REST APIs, Streaming APIs provide a continuous stream of data from Twitter. Whereas REST APIs' searches go back in time to find already posted tweets, Streaming APIs' searches go forward in time to catch new tweets in current time after starting the API call as they are posted.

In the data collection phase, we used the Twitter Streaming API to collect tweets from public streaming, since it was desired to find check-worthy real-time tweets. According to Twitter documentation, when a connection to the streaming API is established, an infinite HTTP request will be created, and the response will be incrementally parsed unless an error occurs. Therefore, a user can determine the duration of keeping the connection alive based on the purpose. The response is raw tweets encoded in JSON format. Even though a tweet that a user wants to post can be only up to 140 characters of text, a raw tweet can be as big as 4 KB.

To collect streaming tweets, at least one of the predicate parameters (filters) of the Streaming API must be specified. These parameters are locations, follow, and track. The Twitter Streaming API has a rate limit to determine how many tweets it will deliver. It allows users to get at most one percentage of all created tweets during the user-determined time of request based on the established parameter or parameters.

In this study, Tweepy, a Python library for accessing Twitter APIs, is chosen to use since it is one of the most popular, widely used, and maintained wrappers. As we want to collect as many check-worthy tweets as possible, we need to find the best predicate parameter/parameters. In following sections, we explain the use of each of these parameters; locations, track and follow.

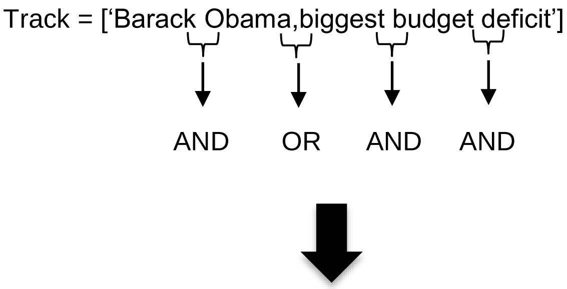
4.1.1 Location-based tweet collection

To use the locations parameter, a user should specify a comma-separated list of longitude-latitude pairs that make up a bounding box to filter tweets. The bounding box allows users to establish a 4-sided geographic area which looks like [west_longitude, south_latitude, east_longitude, north_latitude] [12]. The southwest corner of the bounding box has to come first.

Only geolocated tweets that fall within the requested region will be included in the response. A study conducted by Weidemann and Swift indicates that nearly one in five tweets are geotagging enabled [13]. In terms of our data collection purpose, it means 80 percentage of tweets will be lost. Since Twitter users post about anything, not only politics, and we want to collect only political tweets, that rate will be much higher. Moreover, not only Americans but also anyone from anywhere around the world could post about US politics using the allowance of Twitter up to 25 bounding boxes which makes the rate significantly high. In the light of this information, we decided not to use locations as a predicate parameter.

4.1.2 Track-based tweet collection

The track filter, a comma-separated list of phrases, is to determine what tweets will be delivered from the stream. A phrase may be one word or more separated by spaces. Whereas commas act like logical ORs, spaces are equivalent to logical ANDs. Therefore, delivered tweets will definitely include each word of at least one phrase regardless of word orders. As seen from the following example, a delivered tweet should contain all words in the first phrase of the list, “Barack Obama”, or all words in the second phrase, “biggest budget deficit”.



Track = [‘Barack Obama,biggest budget deficit’]

- **“Barack Obama**: "That’s not American. That’s not who we are. We don’t have a religious test for our compassion" http://politi.co/1MNy0ZK “
- “I believe it is impossible to ever get a balance **Budget** when you run the **biggest** trade **deficit** in the world.”

Table 4-1 shows more examples for user-specified parameters and matching tweets in response to these parameters [14].

Table 4-1 Phrase matching with streaming tweets.

Parameter value	Will match…	Will not match…
Twitter	TWITTER twitter “Twitter” #twitter @twitter http://twitter.com	TwitterTracker #newtwitter
twitter api,twitter streaming	The Twitter API is awesome The Twitter streaming is fast Twitter has a streaming API	I’m new to Twitter

Furthermore, the track filter can be used for searching not only the text entity of a tweet but also hashtags, user account names, and URLs as seen from examples given in Table 4-1. Twitter allows up to 400 phrases to be used, and a phrase must be at least 1 byte and up to 60 bytes.

To create the list of phrases for the track filter, all presidential debate transcripts that were taken from 30 debates for 11 elections in US history were used: 1960, 1976 – 2012. During these debates, about 30 thousand sentences were spoken. A python program was created to prepare unigrams, bigrams, and trigrams of words from these sentences with their frequencies after removing stop words. Since Twitter allows users to use up to 400 phrases, the most frequent 400 words from unigram, bigram, and trigram lists were taken. We also created another list as a mix of these three lists along with some new phrases for candidates of the coming US presidential race. To do that, we manually go over these n-gram lists and eliminate nonpolitical words, as well as words not specific enough for politics. For instance, word the “people” is the most frequent unigram word, but it is not specific enough for collecting US political tweets.

To compare results of these four phrase lists for tweet collection, we created four tweet collection programs for each list, and we executed them in parallel at different times of day for 10 minutes. Table 4-2 presents total numbers of collected tweets for different lists of phrases. While the list of unigram phrases has the highest amount of tweets delivered, trigram has the lowest number of delivered tweets. Apart from unigrams and trigrams, the list of bigrams and the mixed phrases have almost the same amount of tweets. Furthermore, the time of tweet collection has no significant effect on a delivered number of tweets for unigram phrases while the others have major changes.

Table 4-2 Tweet Collection with different lists of phrases for 10 minutes.

List of Phrases	2 am	9 am	9 pm
Unigram	37044	36142	35840
Bigram	5986	10712	11027
Trigram	130	472	165
Mix	6617	13233	11342

In regards of quality and accuracy of tweets to be related to politics, we decided to use the list of the mixed phrases, which contains 29 phrases from the unigram list, 200 phrases from the bigram list, 150 phrases from the trigram list, and 21 new phrases for names of presidential candidates. Figure 4-1 shows a cloud of words for the list of mixed phrases created based on their frequencies.

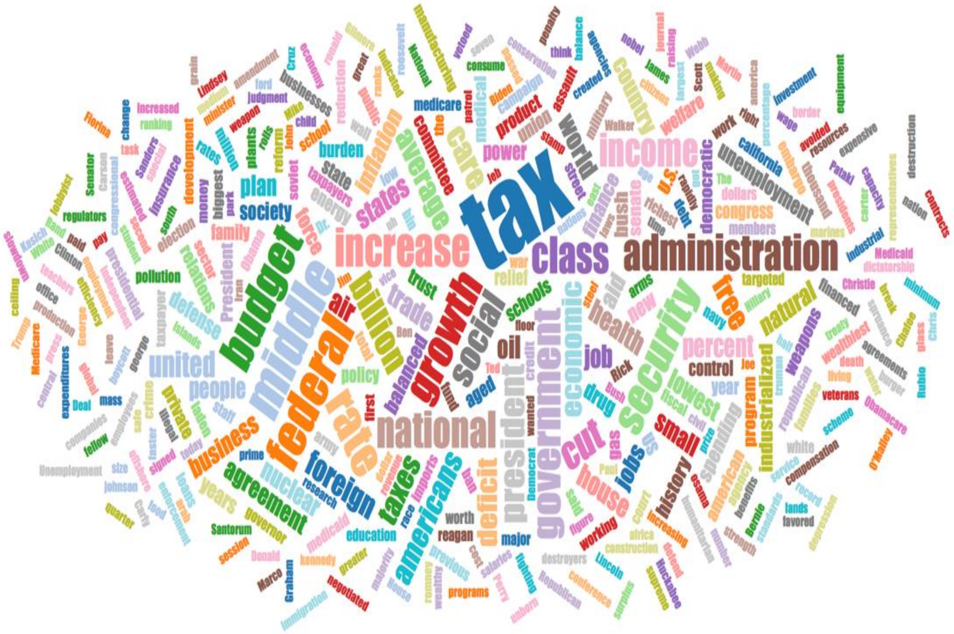


Figure 4-1 Word-cloud of keywords

4.1.3 Follow-based tweet collection

The follow filter is a comma-separated list of Twitter users' ids, and it indicates users whose tweets should be delivered from the stream. Based on the specified user, the response of the request will contain tweets created and retweeted by the user, and retweets of any tweets created by the user and replies to any tweets created by the user.

Twitter allows its users to create lists of their Twitter friends. Whereas this functionality is originally for the purpose of organizing friends so that the user can quickly look at the activities of selected friends, Twitter lists can be used for creating any lists of accounts which share the same interests. We used this functionality to collect U.S. political Twitter accounts' ids, especially for politicians' account ids. We collected ten list names created specifically for politicians, journalists, and news agencies who cover US politics, and we created a program and used Twitter REST API to collect the id of every account in these lists [15-17]. From these lists, we collected totally 2220 account ids for the follow filter. Table 4-3 shows used lists, their description, and their sample members.

Table 4-3 Twitter Lists: List Names, Descriptions, and Sample Members.

List Name	Description	Sample List Members
US Governors	Principal Accounts of State Governors in the U.S. (a mix of campaign and government accounts)	JohnKasich, BobbyJindal, GovChristie, GovWalker, FLGovScott
US Senate	Principal Accounts of Members of the U.S. Senate (a mix of campaign and government accounts)	SenTedCruz, SenSanders, SenBillNelson, marcorubio,

Table 4-3 — *Continued*

US Cabinet	Principal Accounts of U.S. Cabinet Level Federal Agencies and Executives	USTreasury, CEAChair, VP, GinaEPA, USDOT, NRCgov
US House	Principal Accounts of Members of the U.S. House of Representatives (a mix of campaign and government accounts)	RepTedLieu, RepDebDingell, RepRussell, RepStefanik, RepCurbelo, RepHastingsFL
US Election Officials	Principal Accounts for U.S. secretary of state offices and voter information.	PAStateDept, IowaSOS, ElectionsUtah, TXsecofstate
US Election 2014	Candidates and retiring incumbents for the 2014 US congressional midterm and gubernatorial elections	RepJasonSmith, RepBeatty, ChrisChristie, GovInslee, SenatorFischer
White House Accounts	A list of official White House accounts	POTUS, FLOTUS, VP, Cabinet, AmbassadorRice, TheIranDeal
USG	A list of Twitter accounts from Cabinet Secretaries, Agencies, and Departments	NASA, FBI, USArmy, fema, DHSgov, usedgov
US politics	A list of journalists, news organizations covering U.S. politics	foxnewspolitics, nprpolitics, NateSilver538, HuffPostPol, andersoncooper, NBCNews, CNNPolitics, politico
Presidential Candidates	Principal accounts of candidates for President of the United States	HillaryClinton, JebBush, realDonaldTrump, tedcruz CarlyFiorina, RealBenCarson

4.2 Data PreProcessing

In the data collection phase, all collected tweets are stored locally in a text file. Once the data collection phase is done, a data pre-processing phase automatically starts to read tweets from the stored text file. Tweets are encoded in JSON format and each tweet becomes an object after decoding JSON format. We process each tweet sequentially. Each tweet contains several entities such as id, text, user information, and so on. If the tweet is a retweeted tweet, it includes its entities along with the original tweet's entities together. Thus, when we extract the required entities of a tweet, we first check whether it is a retweeted tweet. If so, we extract the original tweet's entities instead of retweeted tweet's entities. Finally, we prepare the text entity of the tweet by removing URLs, user mentions, and so on for further processes.

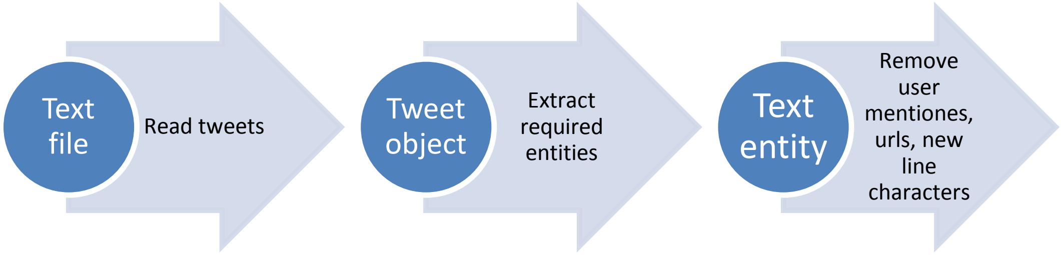

Text file
Read tweets
Tweet object
Extract required entities
Text entity
Remove user mentiones, urls, new line characters

Figure 4-2 Data Preprocessing steps.

Chapter 5

DETECTING CHECK-WORTHY FACTUAL CLAIMS

5.1 Check-worthy Factual Claim

The goal of this thesis is to detect real-time check-worthy factual claims in tweets related to U.S. politics and to share them on a publicly visible Twitter account for journalists, researchers, and citizens to fact-check. In order to find check-worthiness of the tweet, we used an existing tool called ClaimBuster [6], which specializes in detecting check-worthy factual claims in presidential debate sentences.

5.1.1 ClaimBuster

ClaimBuster is a tool that helps to find check-worthy factual claims for fact-checking. Furthermore, it determines whether truthfulness of the claim is significant to the public or not by giving every checked sentence a score ranging from 0 to 1. While 0 refers to the least likely important factual claim, 1 indicates the most likely important factual claim [2].

ClaimBuster categorizes sentences into 3 classes, check-worthy factual sentences (CFS), unimportant factual sentences (UFS), and non-factual sentences (NFS). Non-factual sentences are subjective sentences that express opinions, beliefs, and declarations. While NFS do not contain any factual claim, UFS have factual claims, but fact-checkers do not consider them important enough for checking. On the other hand, CFS contain check-worthy factual claims, and these are the kind of sentences that journalist want to fact-check their claims [2]. Furthermore, this study aims to find tweets that are CFS. Table 5-1 shows examples of CFS, UFS, and NFS categories, and all these sentences are taken from previous presidential debates.

Table 5-1 Examples of NFS, UFS, and CFS.

NFS	I think that we are showing the proper compassion and concern.
	This is a perfect example of the kind of leadership that the United States, under this administration, has taken.
	I want to give every American a $5,000 refundable tax credit.
UFS	In 1957 I was in Havana.
	My grandmother died three days before I was elected president.
	I was an altar boy.
CFS	We reduced welfare rolls by 2 million already.
	And this is the first president in 72 years to preside over an economy in America that has lost jobs, 1.6 million jobs.
	We’ve brought twice as many cases against unfair trading practices than the previous administration and we’ve won every single one that’s been decided.

5.1.2 Implementation of ClaimBuster

ClaimBuster provides an API to its users to find check-worthiness of any type of text given by user. If the given text has more than one sentence, ClaimBuster tokenizes them into individual sentences and calculate the score of each sentence separately. Thus, the API returns a list of sentences along with their scores.

We use ClaimBuster API to find the score of each tweet sequentially. If a tweet has more than one sentence, as we stated previously, the response will be a list of sentences and their scores. Therefore, we take the highest score in this list as score of the tweet because if at least one sentence of the tweet has check-worthy factual claim, it makes the tweet check-worthy. Table 5.2 demonstrates sample tweets and their ClaimBuster scores.

Table 5-2 Sample tweets with ClaimBuster scores.

Tweets	ClaimBuster Score
Federal Eye: The federal government wants you to review it on Yelp #Washington #Post #news	0.161773
The Military is not a social experiment! #GOPDebate	0.393128
U.S. Trade Gap Widens Sharply in August: The August Trade Deficit?widened to $48.33 billion. E... #breaking #news	0.598642
Since 2001, nearly 30 percent of manufacturing jobs in this country have disappeared and over 60,000 factories have been shut down.	0.778081
Hillary and Bill Clinton paid $43.9 million in federal taxes from 2007 through 2014 on adjusted gross income...	0.811472
Colorado collects $9.7 million in #marijuana tax revenue - up almost $5 million from 2014. Revenue totaled $88 million as of May 2015. Do...	0.903244

5.2 Eliminating Not Check-Worthy, Redundant, and Irrelevant Tweets

As we stated previously, ClaimBuster has been built on presidential debate sentences. The presidential debate sentences show a form of formal language. Due to Twitter's 140-character constraint policy, tweets, however, demonstrate the expressiveness and spontaneity of the spoken (informal) language. Moreover, this restriction of 140 characters induces a grammatically incorrect language that also consists of acronyms, hashtags, misspellings, and a numerous number of lexical variants created out of the human imaginary. The difference between the formal language of debate sentences and informal language of tweets leads ClaimBuster to make mistakes in determining not check-worthy factual claims in tweets as check-worthy. To detect all this kind of tweets, we retweeted high-scored tweets for a month. We then analyzed not check-worthy tweets and found some common patterns in these tweets. For every pattern, we created a filter to tackle the issue. Therefore, we created several filters to eliminate not check-worthy tweets that are determined as check-worthy.

We also created another filter to eliminate redundant tweets that are slightly different from tweets that we posted before. We named all these filters as junk filters: including similarity filter, punctuation filter, number filter, top-k filter, and blacklist filter.

5.2.1 Similarity Filter

Twitter does not allow its users to post exactly the same tweet in the scope of the user's historical tweets. On the other hand, users want to post the same tweet for reaching more followers from different time zones and helping a subject to be a trending topic. To the best of our knowledge, there are two ways of tackling this issue, including rewriting the same tweet and making slight differences to each tweet by adding slightly different links or different time stamps, or tagging different user/users. Thus, we are collecting these kinds of tweets. While tackling rewritten tweets is beyond our goal, we came up with approaches to tackle the other types of tweets. If we remove those small changes that user made on the tweets, the remain of the tweets will be exactly same. We firstly changed our data preprocessing phase to cover removing links and user-mentioned tags. For tweets with time stamps and preprocessed tweets, we created a similarity filter in order to detect similarity of a new tweet with tweets that we posted before.

In the similarity filter, we are detecting similarity between a new tweet and tweets that we posted before. If the candidate tweet has at least 0.8 similarity with a posted tweet, that candidate tweet will be filtered out. To find the similarity of two tweets, we are finding the longest contiguous matching subsequence of these two tweets.

Table 5-3 Samples of similar tweets with time stamps.

Cut Social Security, Cut Medicare, Cut Taxes on the richest. Republicans believe these moves strengthens America! August 10, 2015 at 01:30AM
Cut Social Security, Cut Medicare, Cut Taxes on the richest. Republicans believe these moves strengthens America! August 10, 2015 at 03:30PM
Cut Social Security, Cut Medicare, Cut Taxes on the richest. Republicans believe these moves strengthens America! August 10, 2015 at 04:30PM
Cut Social Security, Cut Medicare, Cut Taxes on the richest. Republicans believe these moves strengthens America! August 10, 2015 at 09:30PM

5.2.2 Punctuation Filter

Some punctuation marks such as “!”, “?”, and “$” are important features of ClaimBuster while finding a sentence’s check-worthy score. Even though a tweet does not contain any factual claim, these overused punctuation marks falsely cause ClaimBuster to give it a higher score. We created a filter to eliminate this type of tweets. If a tweet has at least three consecutive of these marks: !, ?, $, we filter out that tweet. Table 6-2 shows examples of tweets with overused punctuation marks that were filtered out by punctuation filter.

Table 5-4 Samples of Tweets with overused punctuation marks.

#JebBush LEHMAN says what? $$$$$$$$$$$$$$$$ $FOXA $$$$$$$$$ $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$ #GOPClownCar
YES!!!!!!!!!!!!!!!!!!!Emails: Hillary Clinton May Go to Prison
Toughest illegal immigration laws are in Mexico yet it allows illegal migration into US. Why? Trump knows..$$$$$$$$$$$
Did sign pledges ? $$$
OMG. Please no more Bushes!!!!!!!!!!!!!!!!we are still trying weed our way out of the the other bushes!!!

5.2.3 Number Filter

Numeric values are the most important feature of ClaimBuster for detecting a check-worthy factual claim [2]. In other words, a sentence with numeric values is more likely to be selected as check-worthy. This feature causes ClaimBuster to falsely select a tweet as check-worthy while it is not check-worthy. To tackle this issue, we created a filter to eliminate this kind of tweets. This filter removes all punctuation marks from the tweet and uses Natural language toolkit NLTK to tokenize the tweet into words. Then it checks every token to determine whether it contains numeric values. If at least 30 percent of all tokens are numerical values, the tweet will be considered as junk and filtered out. 30 percent was chosen based on observations. Table 6-3 exhibits samples of tweets that were filtered out by the number filter.

Table 5-5 Samples of Tweets with overused numbers.

US 10% sclist 15% lib 12% mod lft 30% moderate 20% mod rght 13% CON we win 52 47 in 16
I trade free follows 9x9 8x8 7x7 6x6 5x5 4x4 3x3 2x2 1x1 Tweet me
South Korea Charts: #1 Best Mistake : 29,005 (75,000) #11 Problem: 10,133 (1,225,092) #15 Bang Bang: 6,572 (357,322)
iTUNES Album: Get Weird #8 United States #15 Canada #17 Norway #63 Netherlands #84 Denmark #96 Australia #142 Italy #160 France #183 UK

5.2.4 Top-k Filter

Table 6-4 shows tweets that we call top-k list tweets. These tweets contain noun phrases separated by consecutive numbers that are in either a descending or an ascending order. This filter removes each character except numbers. It then checks whether there are at least three consecutive numbers. If so, it filters out that tweet.

Table 5-6 Samples of Top-k Tweets.

TopWords(3hrs) 1:Obama 2:death 3:climate 4:Holmes 5:James 6:penalty 7:carbon 8:podcast 9:trial 10:President http://t.co/DY5CSVXfAz
RT @user: 6) Happiness for everyone 7) Some lovely flowers 8) World peace 9) No more horridness 10) 20bn more off welfare https://t…
If you mapped the US optimism it would correlate with 1) % population immigrants 2) Rate of employment 3) GDP growth 3) Those are the same

5.2.5 Blacklist Filter

This filter is created to eliminate any tweet that contains a slang or profanity word. We crawled existed profanity and slang lists over the Internet, and combined them so that the blacklist can detect as broad a range of tweets as possible. A blacklist of around 800 phrases was created.

Chapter 6

POLITICAL CLASSIFIER

7.1 Problem Formulation

In this study, we are only interested in detecting check-worthy factual claims in tweets related to U.S. politics. However, we have tweets for other countries' politics and unpolitic tweets as well. We modeled this problem as a classification task to differentiate tweets related to U.S. politics from other tweets. Below, some examples of these categories are given.

U.S. Political Tweets:

- Twenty five of the largest corporations in America in 2010 paid their CEOs more money than they paid in taxes that year.
- Carly Fiorina outsourced (fired) 18,000 USA jobs to China jobs, bought her $1,000,000 yacht with Golden Parachute $
- In July 2014 the official unemployment rate for white Americans was 5.3 percent, blacks were 11.4 percent. That's around 4 million blacks.
- Wisconsin had 2.878 million jobs right before the recession. We now have 2.882 million. Very impressive Scott Walker. #sarcasm

Not U.S. Political Tweets:

- Australia has the highest Gross debt ever recorded at $384.7 billion dollars. All under Abbott's watch. #auspol
- In 1980 there were 700,000 small businesses in the UK. Now there are over 5.1 million. #startups #entrepreneurs
- Kenya's economy is 70% owned by foreigners i.e US$35 billion out of US$50 billion of our GDP is in the hands of foreigners.

Unpolitic Tweets:

- Manchester United Paid 55.2million dollars for a 19 year old. While I'm over here at 24 worth about $13.50. Transfers nowadays are NUTS!
- Apple, for example, paid just $80 million in tax in Australian on sales of $6 billion. We are all being ripped off. #socialmedia
- Daniel Norris is 4th starting pitcher in last 100 years to throw 5 perfect innings w/o finishing game (1st since Bob Knepper in 1986).

Uncertain Tweets:

- 165 million children under the age 5 were stunted (reduced rate of growth and development) due to chronic malnutrition.
- Total startup funding projected to reach $5.5 billion this quarter, second largest quarter in the last 5 years.
- Our economy has now added 8 million jobs over the past 3 years, a pace that hasn't been exceeded since 2000.
- August unemployment rate falls to seven year low of 5.1%, but employers added a smaller than expected 173,000 jobs

7.2 Dataset

In order to construct a dataset for training and testing our classification models to differentiate tweets related to U.S. politics from other tweets, we randomly chose 1000 tweets that we retweeted before. All these tweets were high-scored and passed all filters that we created. We labeled these tweets as US and Not-US. While the label US refers to tweets related to U.S. politics, the label Not-US denotes tweets that are not associated with U.S. politics.

7.3 Feature Extraction

We extracted multiple categories of features from the labeled tweets in order to use them for training and testing classification models. We used the following tweet to explain the features.

- For the first time in U.S. history, 90 percent of Americans are covered. 17.6 million have signed up!! #getcovered

Word (W): We used the natural language toolkit NLTK to tokenize a tweet into words. We used words in tweets to build tf-idf features. There are 5200 words in the corpus. We did not apply any stopword removal and we used all words.

Length (L): The word count of a sentence generates the length feature. We used NLTK to tokenize a tweet into words. The length of the example tweet is 19.

Parts of Speech Tag (P): The NLTK POS tagger was applied to all tweets. A feature for each tag was constructed, and there are 40 POS tags in the corpus. The count of words in a sentence that belong to a POS tag is the value of the corresponding feature. The sentence above has three words (90, 17.6, and million) with POS tag CD (Cardinal Number), three words (covered, signed, getcovered) with POS VBN (Past Participle), and one word (Americans) with POS tag NNPS (Plural Proper Noun).

Sentiment (S): We used Alchemy API to calculate a sentiment score for each tweet. The score ranges from -1 (the most negative sentiment) to 1 (the most positive sentiment). The example sentence has a sentiment score 0.667601.

Entity Type (E): We used Alchemy API to extract entities from tweets. All extracted entities belong to 25 types. The above tweet has an entity “U.S.” of type “Country” and an entity “#getcovered” of type “Hashtag”. We constructed a feature for each entity type. For a tweet, its number of entities of a particular type is the value of the corresponding feature.

Table 6-1 Extracted Feature Categories

Category	Type	# of Features	Example
Word (W)	continuous	5200	U.S., million
Length (L)	discrete	1	8, 15, 20
POS Tag (P)	Discrete	40	CD, VBN, NNP
Sentiment (S)	continuous	1	-0.5, 0, 0.5
Entity Type (ET)	discrete	20	Country, City

7.4 Classification

We performed 10-fold cross-validation using supervised learning methods, including Support Vector Classifier (SVM), Naïve Bayes Classifier (NBC), and Random Forest Classifier (RFC). We tested and evaluated multiple classification algorithms to learn a model that separates U.S. political tweets from other tweets. Table 7-3 shows these classifiers' performances in terms of precision (p), recall (r), and f-measure (f). We experimented with four combinations of features – Word (W), Word + POS Tag (W_P), Word + Entity Type (W_E), and Word + POS Tag + Entity Type (W_P_E). Sentiment and Length were included in all combinations.

Chapter 7

EXPERIMENTS AND EVALUATION

7.1 Dataset

As stated in the data collection chapter, for each data collection process we store collected tweets to text files due to fastness and easiness. To conduct experiments on the tweets that we collected for three months from August 1st to October 31st, we created two tables in MySQL database. For the first table, we stored all tweets without applying any processing method. However, for the second table, we applied same processing steps as we do in our data processing phase. Figure 7-2 shows workflow of the system that we used for tweet storing, and Figure 7-1 shows total number of tweets before and after each processing phase. For example, Table A has over 50 million of tweets, but table B has around 27 million of tweets in total because we eliminated duplicate tweets by taking original id of retweeted tweets. After eliminating tweets with score less than 0.75, we remained with around 50 thousand tweets.

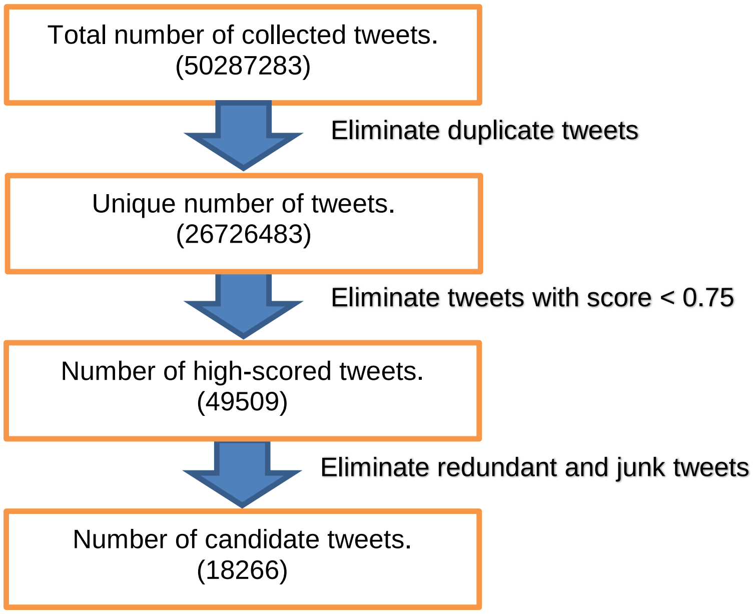

Total number of collected tweets.
(50287283)
Eliminate duplicate tweets
Unique number of tweets.
(26726483)
Eliminate tweets with score < 0.75
Number of high-scored tweets.
(49509)
Eliminate redundant and junk tweets
Number of candidate tweets.
(18266)

Figure 7-1 Total number of tweets before and after each process.

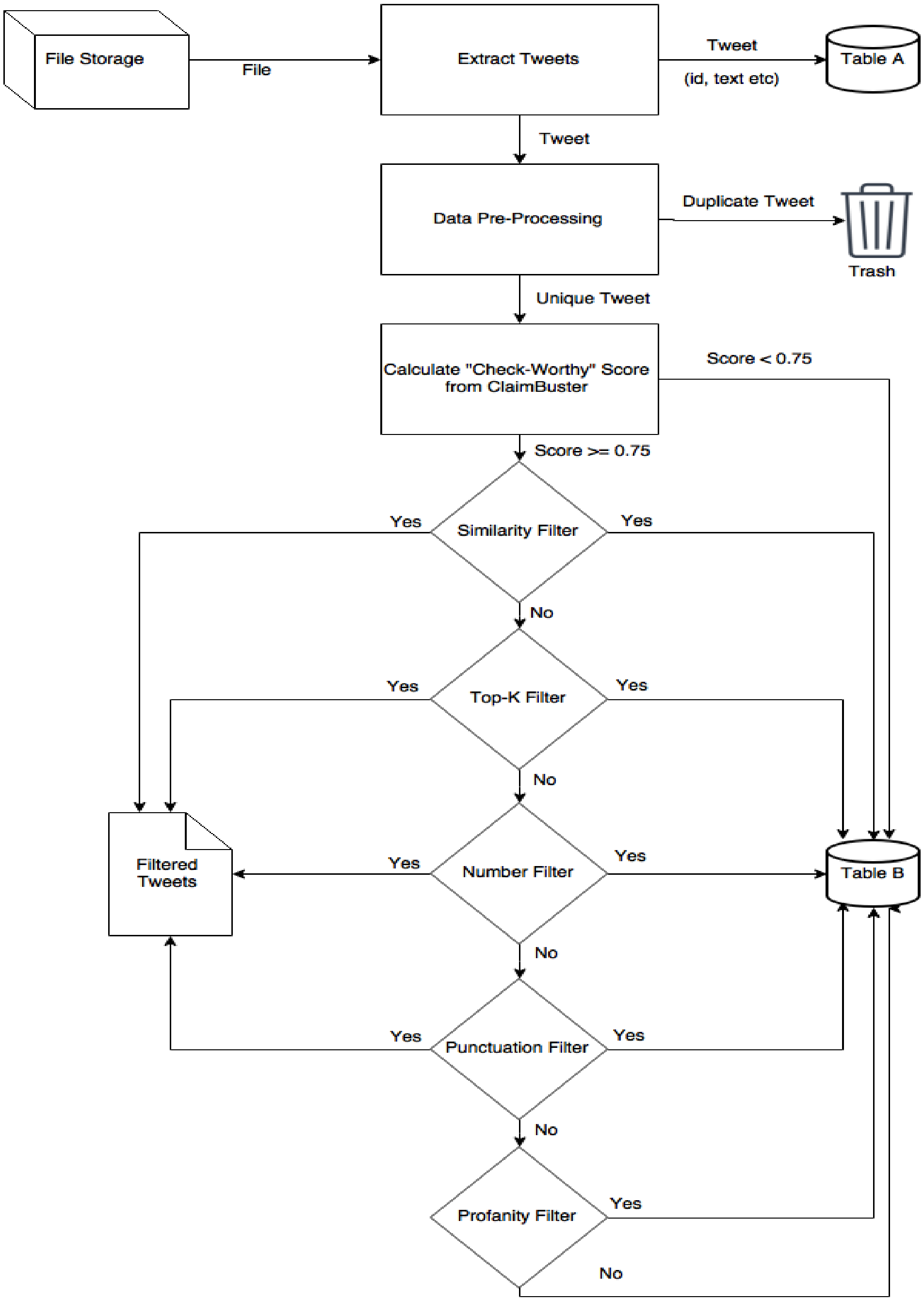
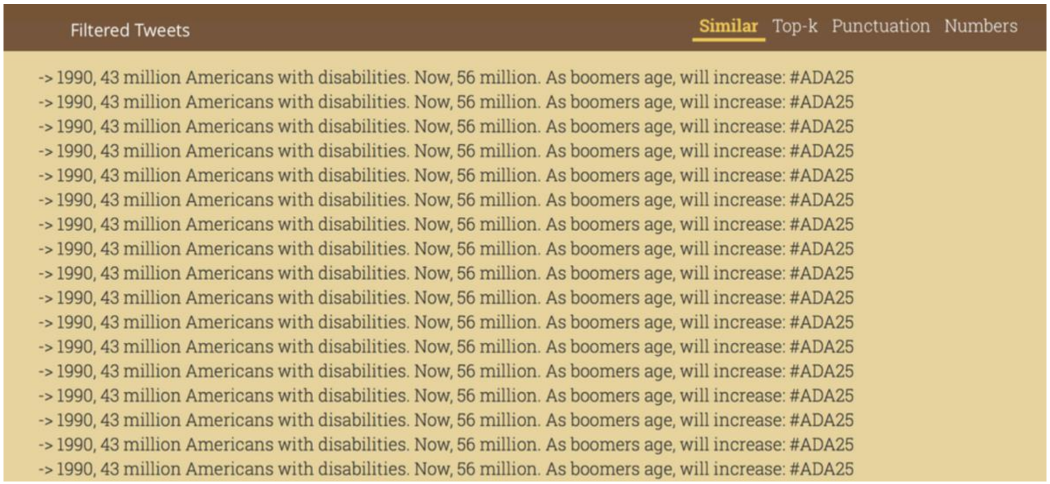


Figure 7-2 The workflow of system for storing tweets to database.

We created a web page to see each failed tweet through junk filters. Each section of the webpage is for tweets eliminated by each filter.



```
-> 1990, 43 million Americans with disabilities. Now, 56 million. As boomers age, will increase: #ADA25
-> 1990, 43 million Americans with disabilities. Now, 56 million. As boomers age, will increase: #ADA25
-> 1990, 43 million Americans with disabilities. Now, 56 million. As boomers age, will increase: #ADA25
-> 1990, 43 million Americans with disabilities. Now, 56 million. As boomers age, will increase: #ADA25
-> 1990, 43 million Americans with disabilities. Now, 56 million. As boomers age, will increase: #ADA25
-> 1990, 43 million Americans with disabilities. Now, 56 million. As boomers age, will increase: #ADA25
-> 1990, 43 million Americans with disabilities. Now, 56 million. As boomers age, will increase: #ADA25
-> 1990, 43 million Americans with disabilities. Now, 56 million. As boomers age, will increase: #ADA25
-> 1990, 43 million Americans with disabilities. Now, 56 million. As boomers age, will increase: #ADA25
-> 1990, 43 million Americans with disabilities. Now, 56 million. As boomers age, will increase: #ADA25
-> 1990, 43 million Americans with disabilities. Now, 56 million. As boomers age, will increase: #ADA25
-> 1990, 43 million Americans with disabilities. Now, 56 million. As boomers age, will increase: #ADA25
-> 1990, 43 million Americans with disabilities. Now, 56 million. As boomers age, will increase: #ADA25
-> 1990, 43 million Americans with disabilities. Now, 56 million. As boomers age, will increase: #ADA25
-> 1990, 43 million Americans with disabilities. Now, 56 million. As boomers age, will increase: #ADA25
-> 1990, 43 million Americans with disabilities. Now, 56 million. As boomers age, will increase: #ADA25
-> 1990, 43 million Americans with disabilities. Now, 56 million. As boomers age, will increase: #ADA25
```

Figure 7-3 Webpage of showing filtered tweets.

7.2 Performance Evaluation of ClaimBuster

To evaluate performance of ClaimBuster, we randomly select 1000 tweets among 26726483 (around 27 million) tweets in Table B. We labeled them as 1 (not check-worthy) and 2 (check-worthy). Only 39 tweets were labeled as check-worthy. We then applied ClaimBuster on all tweets and took their ClaimBuster score. We ranked those tweets based on their ClaimBuster scores, and we measured the accuracy of the top-k tweets by several widely-used measures, including Precision-at-k (P@k), Average Precision (AvgP), and nDCG (Normalized Discounted Cumulative Gain). Table shows these measure values for various k values.

Table 7-1 Ranking accuracy of ClaimBuster.

k	p@k	Avgk	nDCG
10	0.3	0.167	1.0
25	0.32	0.208	0.698
50	0.24	0.243	0.835
100	0.18	0.229	1.157
200	0.145	0.185	1.076
500	0.072	0.132	1.027
1000	0.039	0.091	1.012

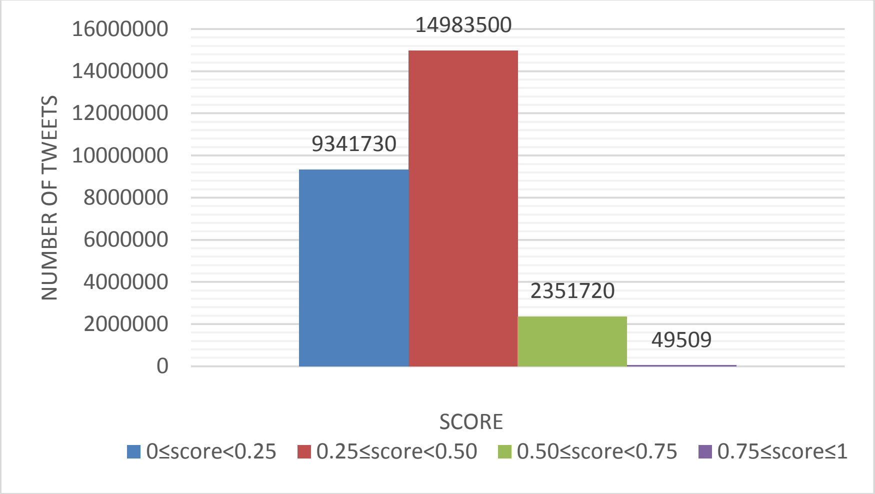


Figure 7-4 Distribution of tweets in terms of score.

7.3 Performance Evaluation of Junk Filters

To evaluate the performance of junk filters: the top-k filter, the punctuation filter, the number filter, and the blacklist filter, we randomly selected two datasets for each filter. While one dataset contains only tweets that failed the filter, the other dataset contains tweets that passed the filter. We labeled each tweet of each dataset to measure the performance of each filter. Each dataset has 1000 tweets, and we labeled 8000 tweets in total. Table shows the performance of each filter in terms of evaluation metrics: precision, recall, and F-score.

Table 7-2 Performance of Top-k, Punctuation, Number, and Profanity Filter.

Filter	Precision	Recall	F-measure
Top-k Filter	0.996	0.984	0.989
Punctuation-Filter	0.997	0.984	0.99
Number Filter	0.956	1	0.977
Profanity Filter	1	1	1

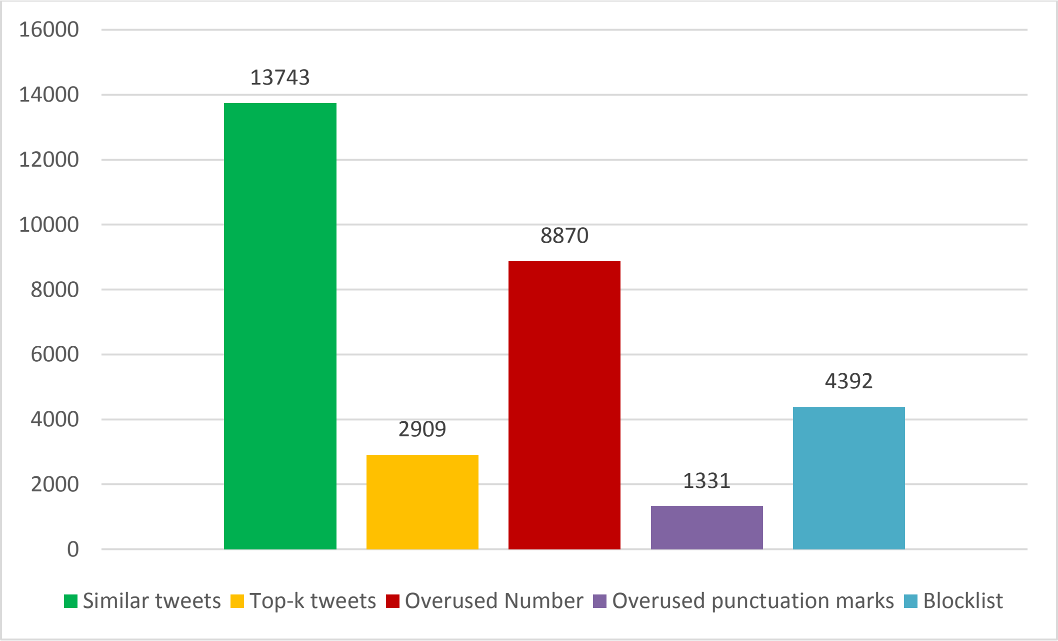


Figure 7-5 Total number of filtered tweets by junk filters.

7.4 Performance Evaluation of Political Classifier

We performed 10-fold cross-validation using supervised learning methods, including Support Vector Classifier (SVM), Naïve Bayes Classifier (NBC), and Random Forest Classifier (FRC). Table 7-3 shows these classifier's performance in terms of precision (p), recall (r), and f-measure (f). We experimented with four combinations of features – Word (W), Word + POS Tag (W_P), Word + Entity Type (W_E), and Word + POS Tag + Entity Type (W_P_E). Sentiment and Length were included in all combinations. SVM and RFC outperformed NBC in most cases. However, NBC paired with W accomplished the best performance for the precision of Not-US and the recall of US classes. On the hand, SVM paired with W attained the best performance in the precision of US and the recall of Not-US classes. Overall, SVM achieved the best performance in terms of f-measure for both classes.

Table 7-3 The performance of SVM, NBC, and RFC classifiers.

algorithm	features	p_US	p_Not-Us	r_US	r_Not-US	f_US	f_Not-US
SVM	W	0.769	0.74	0.744	0.766	0.756	0.752
NBC	W	0.68	0.78	0.818	0.633	0.742	0.698
RFC	W	0.68	0.719	0.733	0.67	0.70	0.69
SVM	W_P	0.723	0.752	0.759	0.714	0.74	0.732
NBC	W_P	0.70	0.65	0.589	0.752	0.639	0.697
RFC	W_P	0.719	0.73	0.729	0.72	0.723	0.724
SVM	W_E	0.74	0.758	0.759	0.738	0.749	0.747
NBC	W_E	0.66	0.734	0.769	0.62	0.71	0.672
RFC	W_E	0.694	0.734	0.75	0.675	0.72	0.70
SVM	W_P_E	0.737	0.754	0.755	0.736	0.745	0.744
NBC	W_P_E	0.7	0.66	0.61	0.744	0.65	0.699
RFC	W_P_E	0.717	0.736	0.739	0.714	0.727	0.724

Chapter 8

CONCLUSION AND FUTURE WORK

We constructed a list of political keywords and a list of political user ids as predicate parameters of the Twitter Public Streaming API to collect real-time political tweets. We used ClaimBuster for finding the check-worthiness score of tweets. We then created several filters to eliminate redundant, not check-worthy, and irrelevant tweets. We finally presented a supervised learning-based approach to automatically distinguish U.S. political tweets from other tweets. We labeled overall ten thousand tweets for evaluation of each component of the system. Performance evaluation of junk filters shows that all filters achieved over 95% precision, recall, and f-measure. For the political classifier, preliminary experiment results show that models achieved 76.9% precision and 74.4% recall in classifying tweets related to U.S. politics.

We plan to carry on future research along the following directions:

- We plan to label more tweets to be used as the training and test sets of the political classifier.
- We aim at improving feature extraction, feature selection, and classification methods to obtain better classification accuracy.
- We will also extract tweets' special features and use them in classification methods.

REFERENCES

1. Nyhan, B. and J. Reifler, *The Effect of Fact-Checking on Elites: A Field Experiment on US State Legislators.* American Journal of Political Science, 2014.
2. Hassan, N., C. Li, and M. Tremayne. *Detecting check-worthy factual claims in presidential debates*. in *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*. 2015. ACM.
3. Twitter. *Twitter usage / Company facts*. 2015 [cited 2015 September 12]; Available from: https://about.twitter.com/company.
4. internetlivestats.com. *Twitter Usage Statistics* 2015 [cited 2015 Sep 12]; Available from: http://www.internetlivestats.com/twitter-statistics/.
5. Tumasjan, A., et al., *Predicting Elections with Twitter: What 140 Characters Reveal about Political Sentiment.* ICWSM, 2010. **10**: p. 178-185.
6. Hassan, N. *ClaimBuster*. 2015 [cited 2015 Nov 2]; Available from: http://idir-server2.uta.edu/claimbuster/.
7. Golbeck, J., J.M. Grimes, and A. Rogers, *Twitter use by the US Congress.* Journal of the American Society for Information Science and Technology, 2010. **61**(8): p. 1612-1621.
8. Gulati, J. and C.B. Williams, *Social media in the 2010 congressional elections.* Available at SSRN 1817053, 2011.
9. Williams, C.B. and G.J. Gulati, *Communicating with Constituents in 140 Characters or Less.* 2010.
10. Lassen, D.S. and A.R. Brown, *Twitter: The electoral connection?* Social Science Computer Review, 2010: p. 0894439310382749.
11. Chi, F. and N. Yang, *Twitter Adoption in Congress: Who Tweets First?* 2010.

12. Moffitt, j. *Filtering Twitter by Location*. 2013 [cited 2015 Oct 9]; Available from: http://support.gnip.com/articles/filtering-twitter-data-by-location.html.

13. Weidemann, C., *Social media location intelligence: The next privacy battle-an arcgis add-in and analysis of geospatial data collected from twitter. com.* International Journal of Geoinformatics, 2013. **9**(2).

14. Twitter. *Streaming API request parameters*. 2015 [cited 2015 Sep 3]; Available from: https://dev.twitter.com/streaming/overview/request-parameters.

15. TwitterWhiteHouse. *Lists of official White House accounts* 2015 [cited 2015 Sep 3]; Available from: https://twitter.com/WhiteHouse/lists/.

16. TwitterGovernment. *Twitter Political Lists*. 2015 [cited 2015 Sep 3]; Available from: https://twitter.com/gov/lists/.

17. TwitterBreakingNews. *Breaking News*. 2015 [cited 2015 Sep 3]; Available from: https://twitter.com/BreakingNews/lists/.

BIOGRAPHICAL INFORMATION

Fatma Dogan received her B.S. degree in Computer Engineering from Firat University, Turkey, in 2010. After finishing her B.S., she worked for a software company for one and half years. She completed her M.S. degree at The University of Texas at Arlington in Computer Science and Engineering in Fall 2015. Her areas of interest includes data mining and computational journalism.