Automatic Identification of Representative Content on Twitter

by

Prashanth Vijayaraghavan

B.E, Anna University (2012)

Submitted to the Program in Media Arts and Sciences
in partial fulfillment of the requirements for the degree of

Master of Science in Media Arts and Sciences

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2016

Author .....
Program in Media Arts and Sciences
May 11, 2016

Certified by .....
Deb Roy
Associate Professor
Program in Media Arts and Sciences
Thesis Supervisor

Accepted by .....
Pattie Maes
Academic Head
Program in Media Arts and Sciences

Automatic Identification of Representative Content on Twitter

by

Prashanth Vijayaraghavan

Submitted to the Program in Media Arts and Sciences
on May 11, 2016, in partial fulfillment of the
requirements for the degree of
Master of Science in Media Arts and Sciences

Abstract

Microblogging services, most notably Twitter, have become popular avenues to voice opinions and be active participants of discourse on a wide range of topics. As a consequence, Twitter has become an important part of the political battleground that journalists and political analysts can harness to analyze and understand the narratives that organically form, spread and decline among the public in a political campaign. A challenge with social media is that important discussions around certain issues can be overpowered by majoritarian or controversial topics that provoke strong reactions and attract large audiences.

In this thesis we develop a method to identify the specific ideas and sentiments that represent the overall conversation surrounding a topic or event as reflected in collections of tweets. We have developed this method in the context of the 2016 US presidential elections. We present and evaluate a large scale data analytics framework, based on recent advances in deep neural networks, for identifying and analyzing election- related conversation on Twitter on a continuous, longitudinal basis in order to identify representative tweets across prominent election issues. The framework consists of two main components, (1) a dynamic topic model that identifies all tweets related to election issues using knowledge from news stories and continuous learning of Twitter's evolving vocabulary, (2) a semantic model of tweets called *Tweet2vec* that generates general purpose tweet embeddings used for identifying representative tweets by robust semantic clustering.

The topic model performed with an average F-1 score of 0.90 across 22 different election topics on a manually annotated dataset. *Tweet2Vec* outperformed state-of-the-art algorithms on widely used semantic relatedness and sentiment classification evaluation tasks. To demonstrate the value of the framework, we analyzed tweets leading up to a primary debate and contrasted the automatically identified representative tweets with those that were actually used in the debate. The system was able to identify tweets that represented more semantically diverse conversations around each of the major election issues, in comparison to those that were presented during the debate. This framework may have a broad range of applications, from enabling exemplar-based methods for understanding the gist of large collections of tweets, ex-

tensible perhaps to other forms of short text documents, to providing an input for new forms of data-grounded journalism and debate.

Thesis Supervisor: Deb Roy
Title: Associate Professor
Program in Media Arts and Sciences

Acknowledgments

First and foremost, I would like to express my sincere gratitude to my advisor Prof. Deb Roy, whose expertise, understanding, and patience, added greatly to my graduate experience. I sincerely thank him for all his guidance during the time of research and writing of this thesis.

I would like to thank Prof. César Hidalgo, Prof. Sepandar Kamvar, and Dr. Allen Gorin, for their guidance, insightful comments, and feedback throughout this whole process.

It has been a wonderful experience to be a part of the Laboratory of Social Machines group and a great opportunity to work with all my amazing friends and colleagues. A big thank you goes to all my friends at Laboratory of Social Machines and MIT Media Lab for the stimulating discussions and for all the fun we have had together in the lab. Here is a list of my colleagues in random order: Preeta Bansal, Sophie Chou, James Kondo, Perng-hwa "Pau" Kung, Soroush Vosoughi, Eric Chu, Ivan Sysoev, Lisa Conn, Amber Franey, Anneli Hershman, Andrew Heyward, Mike Koehrson, Raphael Schaad, Mina Soltangheis, Luke Guolong Wang, Neo Mohsenvand, Heather Pierce, William Powers, Martin Saveski, Russell Stevens, Philip Decamp, Eric Pennington and Misha Sra.

Very special thanks to Dr. Soroush Vosoughi from the Laboratory for Social Machines whose friendship and guidance have helped me push my boundaries in achieving our small yet fruitful professional goals.

Last, but certainly not least, I would like to thank my dad, mom, sister, brother-in-law, cousins and friends for their continued support and encouragement through my years at MIT.

Automatic Identification of Representative Content on Twitter

by

Prashanth Vijayaraghavan

The following people served as readers for this thesis:

Thesis Reader .....

Sepandar Kamvar
Associate Professor
MIT Media Lab

Thesis Reader .....

César Hidalgo
Associate Professor
MIT Media Lab

Contents

List of Figures

List of Tables

Chapter 1

Introduction

A democratic election empowers the citizens to vote for a vision that is based on ideas, policies and principles. Therefore, it is essential for elections to play out as the celebrated Competition of Ideas (established by Milton, Jefferson, Mill, Holmes, as essential to liberty and democracy)[9] rather than a mere race between personalities. The desirability of enhancing this competition involves factoring in public opinion and people's perception about these ideas. The public's interests have traditionally been captured and analyzed via polling, surveys, interviews, etc. With the advent and rise in popularity of social media platforms, people have a venue to directly partake in the conversation around various topics. Recent studies have shown that people are readily taking advantage of this opportunity and are using public social media, most notably Twitter, as a political outlet to talk about the issues that they care about [74]. Though campaigns cannot completely bypass the traditional media, social media is driving a fair share of political engagement beyond horse-race discussions among people due to its horizontal nature of communication process. According to a Pew report, 44% of American adults learned something new about the election in the past week from social media [37].

The importance of social media in elections has been studied extensively. A study [13] in 2012 found that 41% of young people between the ages of 15 and 25 had participated in some kind of political discussion or activity online. Another study [2] concluded that the engagement of registered voters with candidates for office,

political parties or elected officials on social media increased by 10% since 2010. This was attributed to not just the young voters but also those between the ages of 30 and 49. Despite the inherent demographic bias on Twitter [19], the ability to swing conversations, trends, beliefs in certain directions can be a game changer and a phenomenon to ponder upon [25, 18]. Furthermore, the public nature, and the sheer number of active users on Twitter, make it ideal for in vivo observations of what the public care about in the context of an election as opposed to in vitro studies done by polling agencies. To do that, we need to trace the election narratives as they form, spread, morph and decline among the Twitter public. Due to the nature and scale of tweets, analyzing the highly decentralized and fragmented discourse on Twitter can be challenging yet rewarding.

1.1 Motivations

The ability to capture all election narratives requires a system that not only detects and characterizes election related tweets but also represents the diverse conversational spheres within various topics in the election discourse on Twitter. However, the volume of user-generated content on Twitter is so enormous that tweets around certain topics get suppressed under the blanket of topics with controversial or majoritarian views. Also, the tweets under these dominating topics may not best represent the overarching election conversation around various other issues. Since the task of extracting representative tweets is difficult or perhaps even impossible to do manually, we define the main motivation of this thesis as follows:

Given access to Twitter's entire database (growing by an estimated 500 million tweets per day) and a context, is it possible to have a tool that can provide a realistic representation of tweet landscapes across various topics in that context? If so, what are the representative tweets and how representative are they with respect to the entire context-specific tweets?

Answering these questions in the context of US presidential elections (illustrated in Figure1-1) is the focus of this thesis and in order to achieve this goal, we identify

the following challenges that need to be addressed.

Noisy Nature of Tweets

Tweets are short (140 character limit) and the language is very informal, with unique spelling and punctuation, misspellings, new words, URLs, and Twitter specific terminology like DM ("Direct Message") and #hashtags for tagging. Such noisy and idiosyncratic nature of tweets make standard information retrieval and data mining methods ill-suited to Twitter. Consequently, there has been an ever-growing body of literature [52, 57] focusing on Twitter. However, most of these works employ extensive feature engineering to create task-specific models. Therefore it is necessary to have robust models that do not require extensive feature engineering instead have the ability to learn the features automatically.

Less Context & Open Domain

Tweets have much less context and contain far more diversity in style and content compared to standard text documents. Tweets can cover almost any topic as they are user-generated. Since our focus is on elections, tracking election specific tweets from the voluminous Twitter database containing tweets about various topics is not a trivial task. This demands the system to adapt to constantly changing vocabulary of election conversations on Twitter.

Disproportionate Representation of Topics

Generally, some of the election topics dominate the discourse on Twitter. However, the brute majority that some of these topics enjoy on Twitter need not always align with the high priority issues that influence people's votes. A report on The Washington Post [12] showed clear differences in poll data about public's high priority issues and the amount of discussions that each of these issues evoke on Twitter. Among the most striking differences, economy related tweets contributed 8% of the issues-specific discussions, which was far less than the share who said it was their top voting issue (28%). Similarly, foreign policy played a far bigger role on Twitter driving 36% of all issues-specific conver-

sations, while only 12 percent of Americans said foreign policy was their top issue in the Post-ABC poll. So, it is evident that we cannot undermine the importance of those election issues merely based on the minimal attention they draw on Twitter and vice-versa. Therefore, this calls for enforcing a top down structure to identify and represent these weaker Twitter signals, which we know from the poll data that they matter disproportionately to people.

The focus of this thesis is a framework that generates representative tweets from various election narratives on Twitter taking into consideration the challenges enlisted above. The framework consists of components that can (a) robustly track the shifting conversations around elections on Twitter, (b) efficiently categorize tweets into various election topics, (c) identify representative tweets that best typify the interactions under the umbrella of topics.

Representative tweets can shed light on the diverse questions and opinions that arise in the context of the elections and can act as a building block of a responsive election analysis platform for news consumers and journalists. The potential of such a platform paves way to design and implement generic systems that use similar short text analysis in various other domains beyond elections.

1.2 Key Contributions

The major contributions of this thesis are:

- Development of robust tweet aggregation system that ingests dynamically changing election-related conversations using query expansion methods and knowledge augmentation from traditional media sources. We also filter spam tweets using character-level deep learning models to address the noisy nature of tweets.

- Vector-based representation of tweets based on a character-level Encoder-Decoder model. Tweets that share semantic properties are mapped to similar vector representations. These representations are general-purpose vectors that can be used for any classification task.

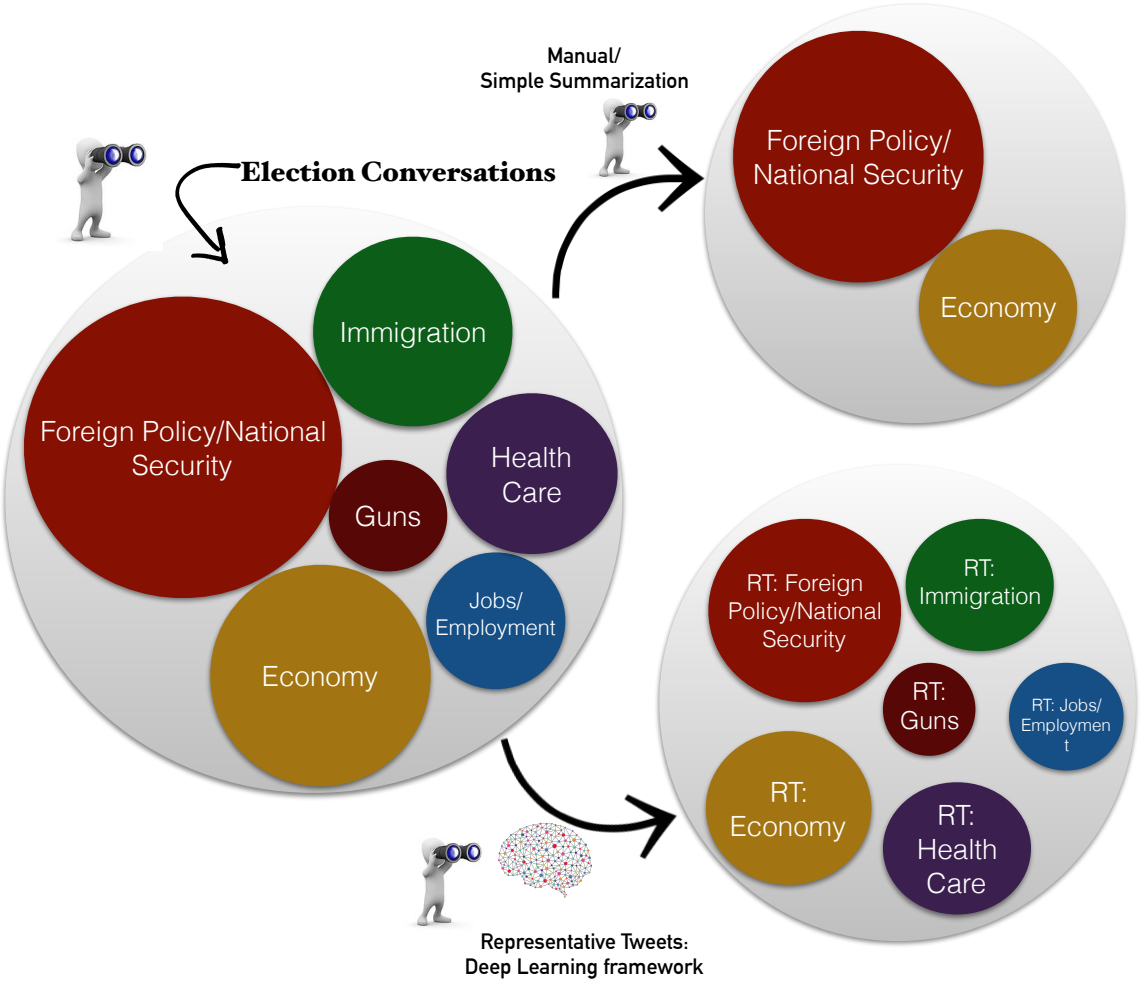


Figure 1-1: Illustration of the goal of the thesis: Provide a representation of election conversations (represents all major election issues; RT: Representative Tweets in the election conversation). Majority topics (Foreign Policy/National Security and Economy) dominate the conversation when filtered through manual selection/simple summarization techniques.

- Bottom-Up analysis of election related tweets to encapsulate the election-specific conversations into few representative tweets using the general-purpose vectors.

- Development of a complete natural language processing framework that is easily adaptable to new domains.

1.3 System Overview

Figure 1-2 shows the general pipeline of our framework. As can be seen, the framework consists of three main components - Twitter Data Pipeline, Twitter Analysis Pipeline and Representative Tweet Extractor. Twitter data pipeline collects and stores the election related tweets keeping track of the constantly changing Twitter election vocabulary. This is accomplished by augmenting the Twitter data pipeline with all election specific terms populated using a semi-automated approach. Twitter analysis pipeline filters all the non election spam tweets and categorizes the tweets under various election topics. Representative Tweet Extractor selects the representative tweets that encompass all the different perspectives under each election category. Each component of our system consists of modular sub-components i.e., they can be modified without affecting the internal working of rest of the system.

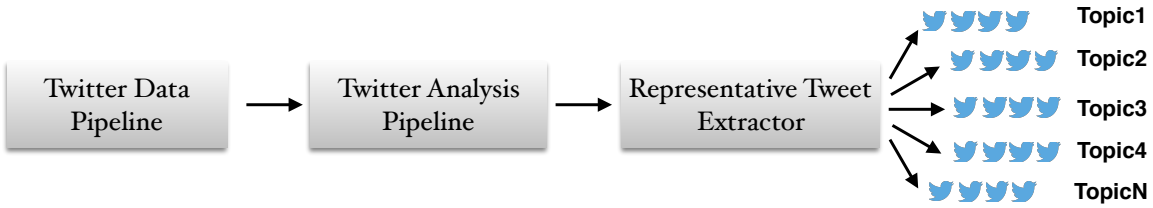


Figure 1-2: Illustration of System Pipeline

1.4 Outline of the Thesis

The current chapter is an introduction to the project and a brief overview of the motivations and contributions of this thesis. The remaining chapters are arranged as follows:

- Chapter 2 explains in detail the overall system architecture and design.
- Chapter 3 discusses the various models used in our system and their evaluation in detail.
- Chapter 4 showcases the overall performance of the system with a real-world example and the evaluation of parts of the system.

- Chapter 5 describes in detail the existing works and some of the prominent deep learning approaches.
- Chapter 6 highlights the results of our system and concludes with ongoing and future work and contributions.

Chapter 2

System Architecture Overview

The overarching goal of this thesis is to build a tool that can extract representative tweets from the election narratives on Twitter. In this chapter, we discuss the flow of data through various components of our system as shown in Figure 2-1.

2.1 Twitter Data Pipeline

The primary task of Twitter Data Pipeline is to gather all the election-related tweets from Twitter. We describe the details of this component of the system in the following sections.

2.1.1 Election Tweet Aggregator

With access to more than half a billion tweets available through GNIP [1] (Twitter Firehose), we use a Boolean query containing one or more 'clauses' to quickly narrow down the large search space to a less precise election-related data. A clause is a keyword, exact phrase, or one of the many operators that GNIP API supports. An

[1] https://gnip.com/sources/twitter/

example of a simple Boolean query is given below:

$$\#election2016 \quad OR \quad \#DemDebate \quad OR \quad \#GOPDebate \quad OR \quad @realDonaldTrump \quad OR \quad @HillaryClinton$$

Building a Boolean query with terms that are essentially a part of Twitter's election vocabulary is critical to our system so that we will ideally be able to capture all the conversation around elections on Twitter. This vocabulary evolves continuously either due to Twitter's informal nature with new hashtags or due to external events that drive the election conversation on Twitter. It is challenging to adapt to these shifts and keep track of varying election landscapes. We address this challenge using a series of following steps:

1. Manual selection of high-precision election terms. (e.g., commonly used hashtags like #election2016, #2016ers, #DemDebate, #GOPDebate etc., candidate names and their Twitter handles like Hillary Clinton, @realDonaldTrump, etc.)

2. Continuous learning from the existing tweets to account for the natural shifts in Twitter's election vocabulary. (e.g., #hillary4potus, Hitlery)

3. Knowledge extraction from other election sources, especially election news data. This will help the system to adapt to changes driven by election events.

Manually selected high-precision terms are the Twitter seed election terms. We use them to initially collect election related tweets. Continuous learning is established by training a distributed word representation model (explained in detail in Section 3.1.1.1) on these collected tweets and updating regularly with tweets from previous week. As a consequence, we expand the list of query terms by including those that are semantically similar to the seed election terms. The final step of extracting knowledge from election news data is accomplished by a component called media knowledge miner. Since mainstream media reports on a wide range of election events, it is highly likely that the information from the data generated by the media organizations can be useful to capture election relevant tweets. Media knowledge miner processes the

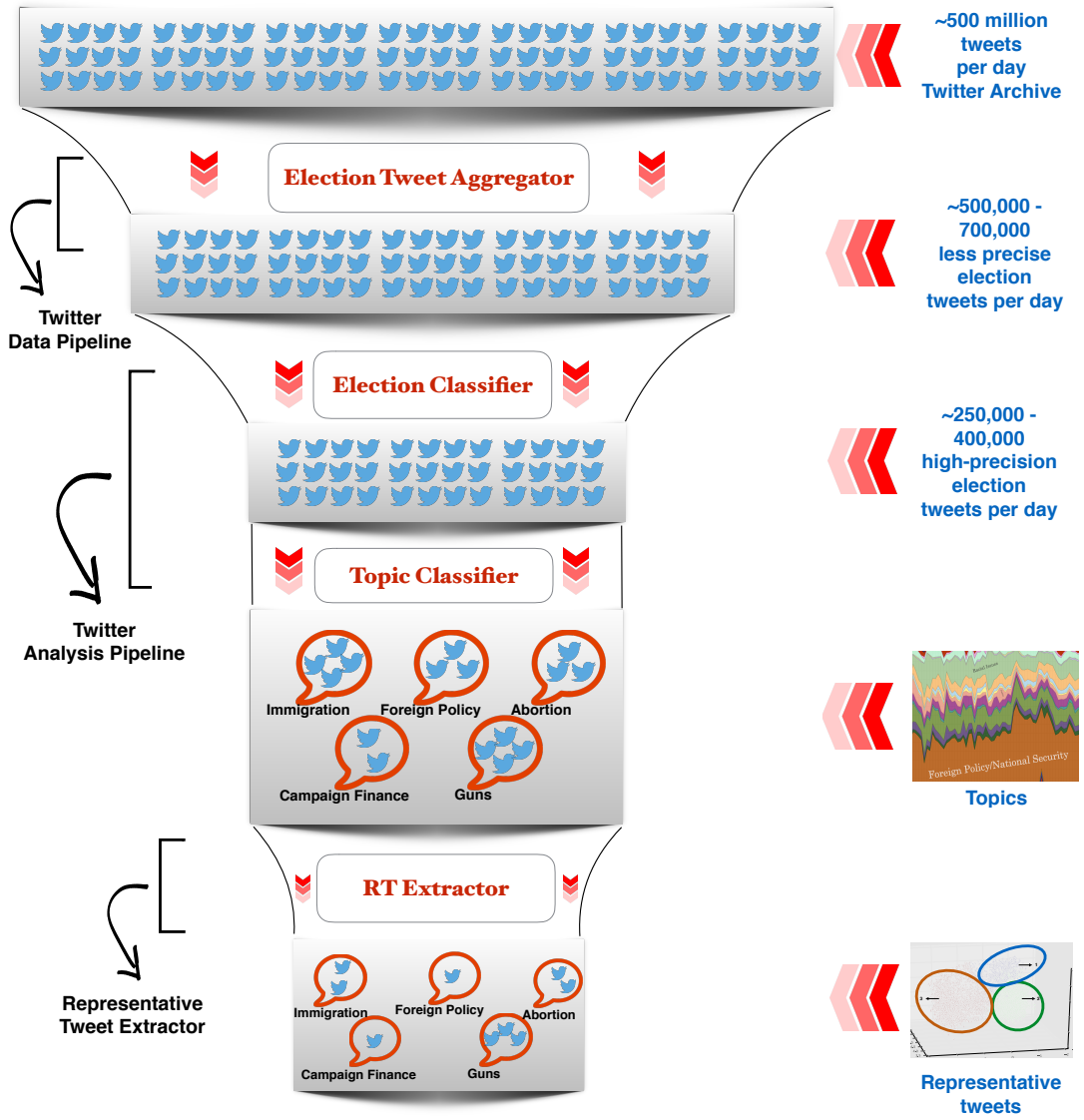


Figure 2-1: Illustration of flow of data through components: Twitter Data Pipeline, Twitter Analysis Pipeline, Representative Tweet Extractor.

election news stories and identifies influential personality names with their probable twitter handles, thereby helping unearth more election related tweets. The following list summarizes the tasks (described in Section 3.1.2) involved in this component of the system.

- Collect the digital news stories from the websites of various news organizations using Data Collection Engine (Section 3.1.2.1).

- Eliminate non-election news stories with the help of Election news classifier (Section 3.1.2.2).
- Identify mentions of personality names in the election news stories using Entity recognizer (Section 3.1.2.3).
- Select the personalities with some political affiliations by categorizing them using Person categorizer (Section 3.1.2.4).

Election tweet aggregator takes the terms accumulated from the seed election terms and continuous learning, combines it with the terms imbibed from media knowledge miner and formulates a Boolean query which is finally used to retrieve tweets from the Twitter fire hose. The ensuing raw tweets are then stored in our in-house election database. Thus, the number of tweets that needs to be analyzed is reduced from half a billion tweets per day to nearly half a million tweets per day.

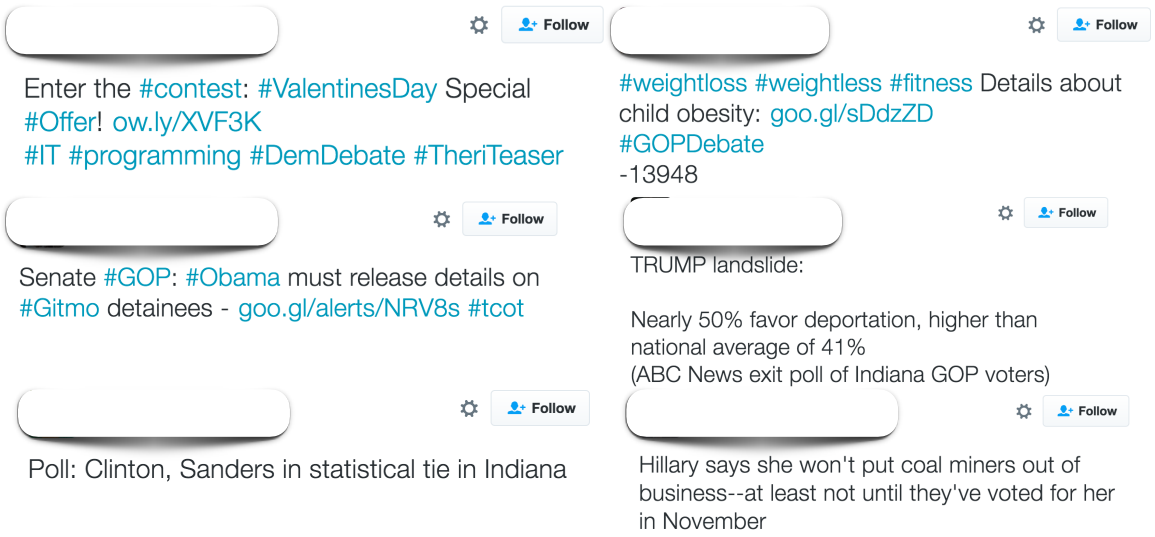


Figure 2-2: Sample of raw tweets gathered by election tweet aggregator.

Figure 2-2 shows a sample of raw tweets that are stored in our database. Though these tweets have been aggregated based on election-related terms, they contain a number of non-election contents and spam tweets. As seen in the figure, presence of #GOPDebate or #DemDebate doesn't necessarily make it germane to elections. Thus, all these tweets need to be processed further to eliminate such inconsistencies.

2.2 Twitter Analysis Pipeline

Twitter analysis pipeline processes the raw tweets from the Twitter data pipeline and cleans all the spam and non-election tweets using an election classifier. Since there is a striking difference between Twitter public and the general public on the issues that they associate with, it is important to have a representation from all major election issues and not just the central issue talked on Twitter. This is because such a reductionist approach will constrict a sizable amount of general public who might have divergent interests compared to Twitter public and vice-versa. So, we need to have a top-down structure in place to capture those divergent views. This is achieved by categorizing the spam filtered tweets among major election issues using a topic classifier. We describe how data flows through these sub-components of the system with few examples.

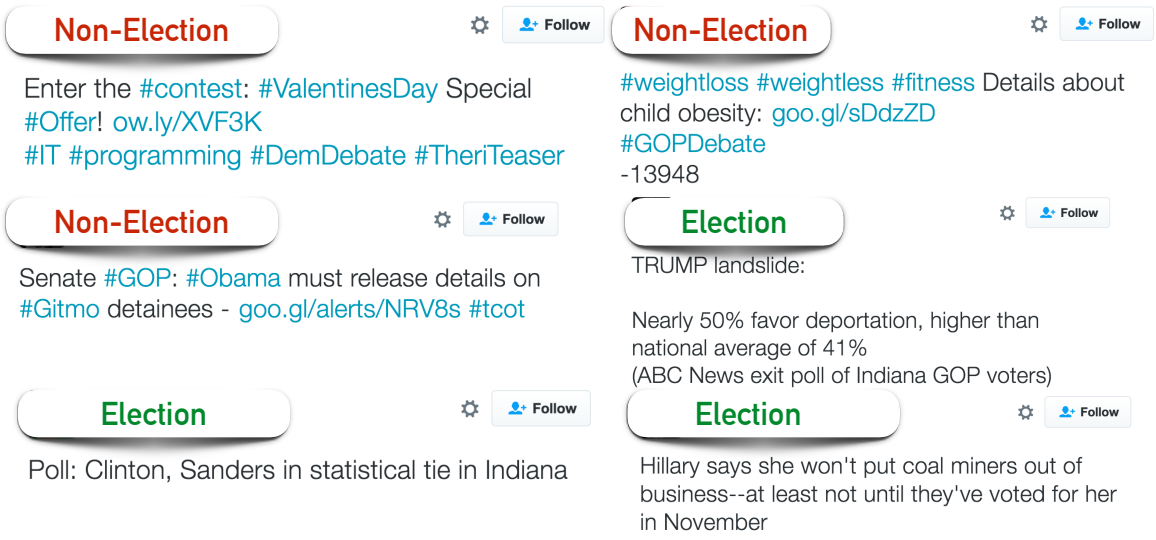


Figure 2-3: Sample output of the election classifier on the tweets in Figure 2-2.

2.2.1 Election Classifier

Since the aggregated tweets contain some noise in the form of spams and non-election contents (as seen in Figure 2-2), we introduce an election classifier that filters such discrepancies and creates a data dump of high-precision election-specific tweets. Fig-

ure 2-3 shows the tweets in Figure 2-2 validated for their relevance in election context using the election classifier. This further reduces the number of tweets to analyze from half a million to nearly quarter million tweets per day. Figure 2-4 shows the daily counts of number of high-precision election-specific tweets stored in our system between February 2015 and May 2016. In many cases, spikes reflect the days when Republican or Democratic debates were held.

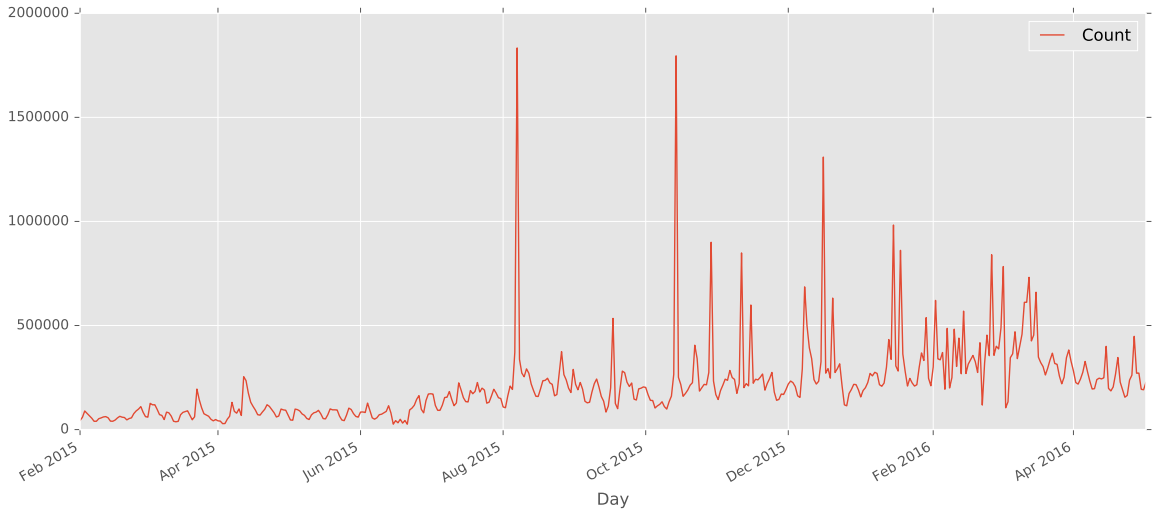


Figure 2-4: Number of high-precision election-specific tweets on a daily basis from February 2015 to May 2016.

2.2.2 Topic Classifier

These high-precision election-specific tweets represent the overall election conversation that our system finds worthy of analysis based on our aggregation and classification. In order to have a good representation of all the major election issues that might get mired in this pool of election content, it is important to categorize the tweets among those issues. The ultimate goal of extracting representative tweets will be based on these categories.

With the help of an annotator with political expertise, we identified 22 election-related topics that capture the majority of the issue-based conversation around the election. Table 2.1 shows all these 22 topics and a sample of a classified tweet. We employ a supervised topic categorization mechanism that allows us to characterize

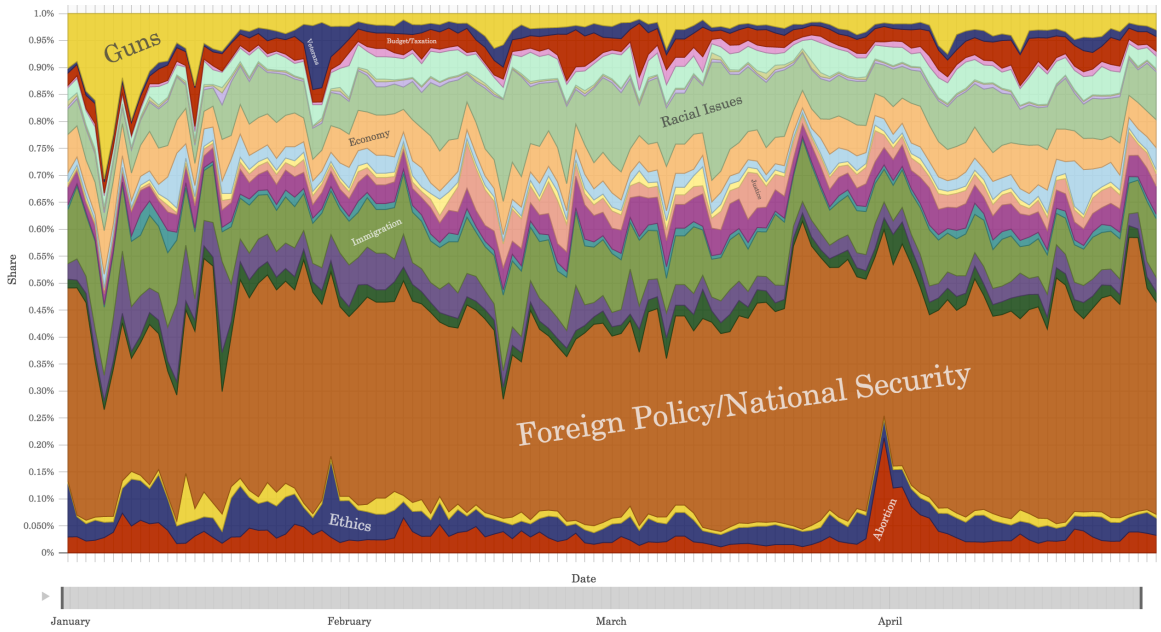


Figure 2-5: Share of conversation for topics on Twitter from January 2016 to May 2016

tweets under each of these different topics. We often expand our topic list and retrain our classifiers every month to keep our systems updated for new topics and vocabulary. Figure 2-5 shows the share of conversation for topics on Twitter from January 2016 to May 2016.

2.3 Representative Tweet Extractor

The final and crucial component of our system is representative tweet extractor. The election tweets distributed among various topics are given as input to this segment and the output is a list of representative tweets for each of the election topics. These tweets give a summary of diverse conversational spheres within each topic. Given a topic and their corresponding tweets, we perform the following steps:

- Convert tweet into a semantic vector based representation using a encoder-decoder model called Tweet2Vec. (Section 3.3.1)

- Cluster these semantic tweet vectors. (Section 3.3.8)

	Topics
1	Income Inequality
2	Environment/Energy
3	Jobs/Employment
4	Guns
5	Racial Issues
6	Foreign Policy/National Security
7	LGBT Issues
8	Ethics
9	Education
10	Financial Regulation
11	Budget/Taxation
12	Veterans
13	Campaign Finance
14	Surveillance/Privacy
15	Drugs
16	Justice
17	Abortion
18	Immigration
19	Trade
20	Health Care
21	Economy
22	Other

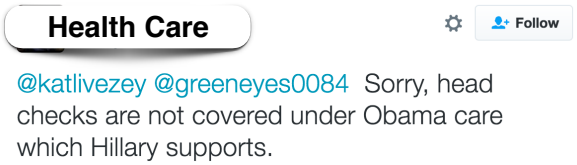


Table 2.1: Left: List of Election Topics; Right: Sample tweet with its corresponding topic.

- Rank the tweets within each cluster. (Section 3.3.9)

The highly ranked tweets from all the different clusters within that topic eventually form the representative tweets for that topic. It is straightforward to select representative tweets from the biggest cluster in that topic, but we select highly ranked tweets from all the clusters. This is based on our initial argument that the size of the discussion alone is not a measure of their importance and The Washington

Clusters	*Top*-3 *Tweets*
Cluster 1	Donald Trump, Abortion Foe, Eyes 'Punishment' for Women, Then Recants - New York Times https://t.co/RTzlwEZhPR
	Trump actually did state the punishment for getting an abortion: "...they would perhaps to illegal places..." https://t.co/9LZ7pfNwPK
	Trump issues statement that Trump disagrees with Trump's "punish women" abortion idea: https://t.co/ZSPE93ptBz https://t.co/eS2BIZoz2J
Cluster 2	Donald Trump's 3 positions on abortion in 3 hours https://t.co/CDxt24vT87
	Scarborough: Trump Flip from Pro-Partial Birth to Pro-Life 'Impossible' https://t.co/w92xm9gh2A
	#RT #Follow #TopStories Donald Trump's Evolving Stance on Abortion - ABC News https://t.co/iMJcTW9u8F https://t.co/yandvxns0D
Cluster 3	Hillary Clinton Knocks Bernie Sanders Over Response to Donald Trump's Abortion Comments https://t.co/WjfJjmRQ3S #ImWithHer #ShesWith
	Bernie doesnt think Trump saying women should be punished for getting an abortion is something that should be covered on the news #ImWithHer
	Boom. @berniesanders said Punishing women for abortion was not an important topic. Let's move on. Ladies!!!. #NoWomem4Sanders
Cluster 4	What are your thoughts on @realDonaldTrump's comments about abortion yesterday? Do you think it was a mistake that will impact his campaign?
	@KStreetHipster i did :-) i think trump's is the consistent position unless you deny women agency in the decision for the abortion.
	IWhat, did you think @realDonaldTrump was ever planning to lock up a single woman for an abortion, or saying respect the Law of the Land?

Table 2.2: Representative Tweets for Abortion on March 31, 2016

Post's report [12] on how some issues, though are crucial for general public, are not discussed in the same scale on Twitter. We remain true to the original definition

of being "representative", hence, we select the top ranked tweets from every cluster inside an election issue. These might represent different sub-topics within that topic.

From the Figure 2-5, we find that there is a peak in the “Abortion” topic at the end of March 2016. We can investigate the reason by finding the representative tweets under topic: “Abortion”. Table 2.2 shows the representative tweets for the topic “Abortion”. It is evident from all the clusters that it is Donald Trump’s opinion on abortion that caused the sudden increase in people talking about it on Twitter. It is interesting to note that $Clusters$ 1 & 2 are the largest clusters and they refer to the comments that Trump made on abortion and his changing stance on the same issue respectively. The other clusters refer to democratic candidates’ view about the whole issue and the questions being asked about this issue. Since the main reason behind the spike is Trump’s opinion on abortion, we don’t see divergent sub-topics emerging from this topic. However it is interesting to see the nature of ensuing clusters. In Chapter 3 and 4 , we describe the models used for this purpose and evaluate the representative tweets extracted from tweets collected during one of the Democratic debates.

Chapter 3

Architecture & Model Description

In this chapter, we will take a deeper dive into various components of our system. The architecture of the complete system, henceforth referred to as Tweet Exemplifier, is shown in Figure 3-1. As seen in that figure, the system consists of four main components - Twitter Data Pipeline, Twitter Analysis Pipeline [67] and Representative Tweet Extractor. The models used in each and every phase of these components of the system are elucidated with their corresponding evaluations. The Tweet Exemplifier framework utilizes the recent advances in natural language processing and deep learning techniques to model various stages of our system pipeline.

3.1 Twitter Data Pipeline

3.1.1 Election Tweet Aggregator

Election tweet aggregator is an important component of the system that keeps track of the dynamically changing election conversation due to external events or Twitter's intrinsic vocabulary shifts driven by user-generated content. Besides the manually curated election seed terms, we use "Media Knowledge Miner" for capturing knowledge from news stories and "Continuous Learning" for adapting to the Twitter's intrinsic vocabulary shifts. Election tweet aggregator takes newly extracted terms from these system components and creates a Boolean query to gather data from Twitter firehose.

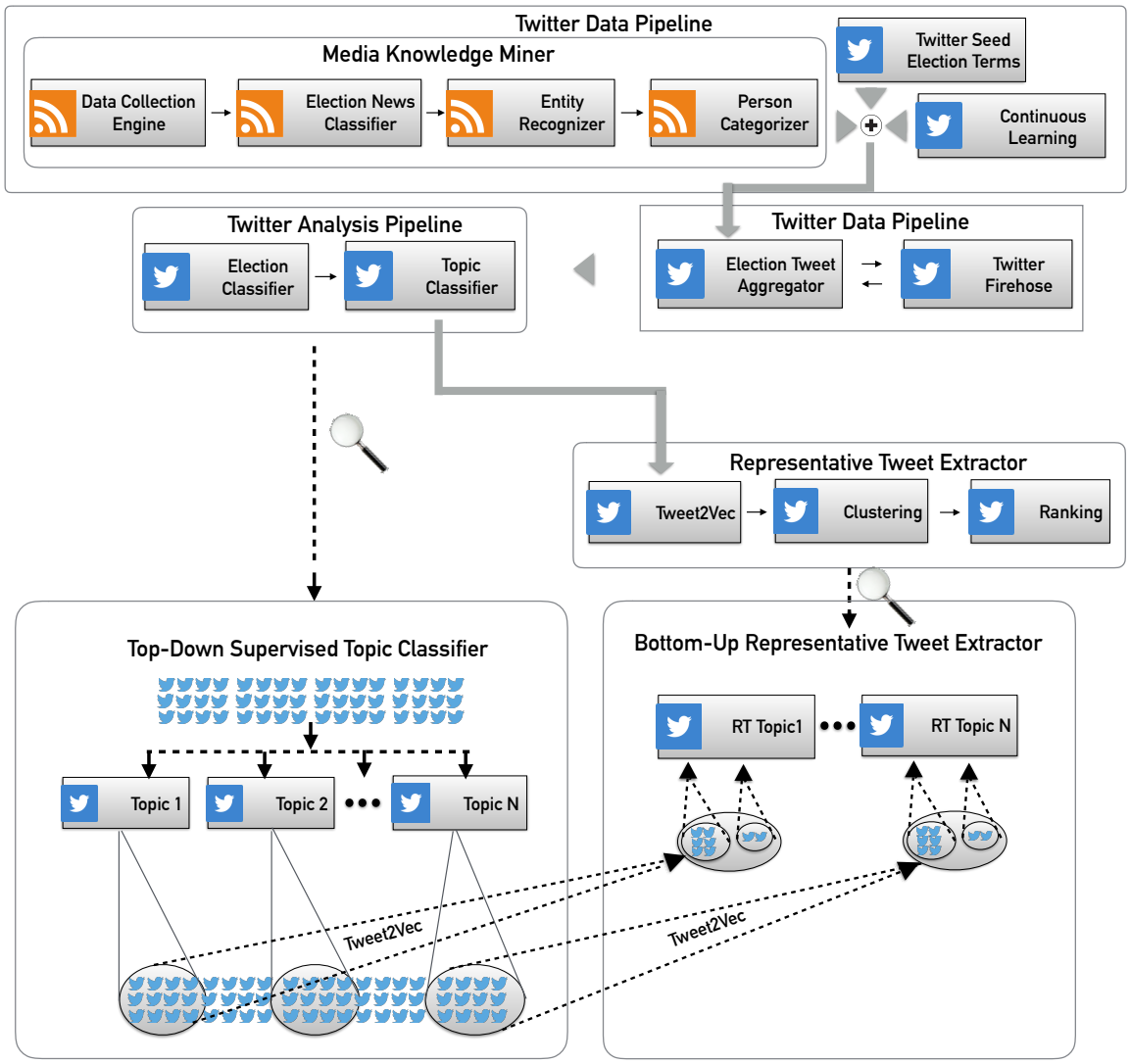


Figure 3-1: Overall System architecture comprising of four main components - Domain Knowledge Extractor, Twitter Data Pipeline, Twitter Analysis Pipeline and Representative Tweet(RT) Extractor.

These components increase the number of tweets extracted by almost 70%. As we show later in the thesis, this corresponds to an increase in the recall (capturing more election-related tweets), and a decrease in the precision. Finally, in order to increase the precision, the tweets extracted using the expanded query are sent through a tweet election classifier that makes the final decision as to whether a tweet is about the election or not. This reduces the number of tweets from the last stage by about 41%, on average. In the following sections we describe these methods in greater detail.

3.1.1.1 Continuous Learning

On a weekly basis, we use the Twitter historical API (recall that we have access to the full archive) to capture all English-language tweets that contain one or more of our predefined seed terms. For the query expansion, we employ a continuous distributed vector representation of words using the continuous Skip-gram model (also known as $Word2Vec$), introduced by Mikolov et al. [46]. The model is trained on the tweets containing the predefined seed terms to capture the context surrounding each term in the tweets. (Figure 3-2 illustrates the Skip-gram model). This is accomplished by maximizing the objective function:

$$\frac{1}{|V|}\sum_{n=1}^{|V|}\sum_{-c\leq j\leq c, j\neq 0}\log p(w_{n+j}|w_n) \tag{3.1}$$

where $|V|$ is the size of the vocabulary in the training set and c is the size of context window. The probability $p(w_{n+j}|w_n)$ is approximated using the hierarchical softmax introduced and evaluated in a paper by Morin and Bengio [47]. The resultant vector representation captures the semantic and syntactic information of the all the terms in the tweets which, in turn, can be used to calculate similarity between terms.

As we train the model, the noun phrases are also extracted from the tweets. This is done using a simple state machine. First, each term in a tweet is assigned a part-of-speech (POS) tag using the Twitter POS tagger [24]. These are given to a state machine based on the following grammar to filter out noun phrases:

$$NP \mapsto Noun|AdjectiveNP$$

Given the vector representations for the terms, we calculate the similarity scores between pairs of terms in our vocabulary using cosine similarity. For a term to be shortlisted as a possible extension to our query, it needs to be mutually similar to one of the seed terms (i.e., the term needs to be in the top 10 similar terms of a seed term and vice versa). The top 10 mutually similar terms along with noun phrases containing those terms are added to the set of probable election query terms. We combine

Query	*Expanded Terms*
LGBT Issues	marriage equality, lgbtq, gay, marriage law, servicemembers, #lgbtrights, #gayrights, sex marriage, equality, gun rights, #lgbt, #transgender, discrimination, #marriageequality, #lgbtnews, lgbt rights
Racial Issues	racial divide, #christopherlloyddresses, hurls, slurs, #restorethevra, rupert murdoch, #equality, undocumented immigrants, #sandrabland, blackness, #sayhername, injustice
Justice	criminal justice, scotia, system, supreme court, obstruction, obstructing, #privateprisons, hillaryforprison 2016, irreparable, irrevocable, #prison, criminal
Budget/Taxation	tax dollars, debt ceiling, income inequality, self funding, discretionary, entitlements, #cair, debt limit, austerity, #debtceiling, aca, tax cuts, tax plan, enrollment, budgets, socsec, revenues
Ethics	#hillaryclintonemails, benghazi probe, complaint, disciplined, bidders, saddam gaddafi, hillarys emails, violations, #clintonemail, saddam hussein, email mistake, predicated, #servergate, broken, clintons war, email server, clintons server data, email woes, email scandal, fbi probe
Immigration	#builthewall, #syrian, immig, immigration plan, jimmy kimmel, immigration protester, deporting, illegals,#gangof8, immigration policies, immigratio, jimmy fallon, immigration reform, #immigrant, gun policy, undocumented immigrants, #amnesty, #greencard, immigration policy, ann coulter, syrian refugees, #h1b, #refugees
Mike Huckabee	ckabee, labrador, fuckabee, backwoods, hucka, georgehenryw, mpp, wannabee, huckabe
Bernie Sanders	reaganism, idear, #feelthebernbernie, #feelthebern
Carly Fiorina	failorina, #fiorinas, scoots, dhimmi, fiorin, backchannel, carlie
Hillary Clinton	hillaryclinton, #hillaryforpotus, clintons, hitlery, hellary, mmurraypolitics, hrc
Donald Trump	trumpdonald, #trumptheloser, #trumpthefascist

Table 3.1: Examples of expanded terms for a few candidates and topics.

this list of terms with the names of all new politicians (Government/Ex-Government Officials and Candidate/Party Officials), obtained from the media knowledge miner described in the Section 3.1.2.3 & 3.1.2.4, thus ensuring that our Twitter data pipeline

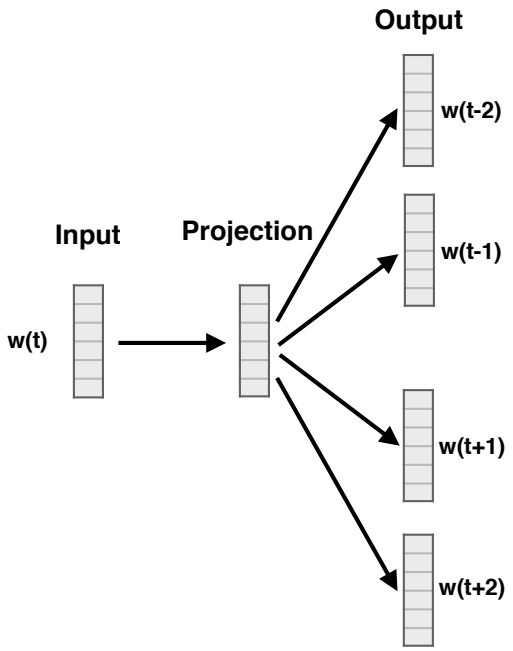


Figure 3-2: Skip gram model- Window size=5. The context around $w(t)$, here the two words before after, is captured.

does not miss any new politicians that join the election dialogue in the media.

However, there could potentially be voluminous conversations around each of the expanded terms with only a small subset of the conversation being significant in the context of the election. Therefore, the expanded terms are further refined by extracting millions of tweets containing the expanded terms, and measuring their election significance metric ρ. This is the ratio of number of election-related tweets (measured based on the presence of the seed terms in the tweets) to the total number of retrieved tweets. For our system, the terms with $\rho \geq 0.3$ form the final set of expanded query terms. The ρ cut-off can be set based on the need for precision or recall. We set the cut-off to be relatively low because, as we discuss in the Section 3.2.1, we have another level of filtering (an election classifier) that further increases the precision of our system.

The final list of terms generated usually includes many diverse terms, such as terms related to the candidates, their various Twitter handles, domain knowledge terms, names and handles of influential people and prominent issue-related terms and noun phrases (e.g., immigration plan). As mentioned earlier, the query is ex-

panded automatically on a weekly basis to capture all the new terms that enter the conversation around the election (for example the hashtag, #chaospresident, was introduced to the Twitter conversation after the December 15, 2015 republican debate). Table 3.1 shows a few examples of expanded terms for some of the candidates.

3.1.2 Media Knowledge Miner

Media knowledge miner is employed to analyze traditional media data to choose political personalities associated with election events and enhance the overall recall (capturing more election-related tweet) of the system. As described earlier in Section 2.1.1, it involves four major sub-components: Data collection engine, Election news classifier, Entity recognizer and Person categorizer. The output of the person categorizer is a list of political personalities with their probable twitter handles which is combined with terms extracted from continuous learning. The functions of these subcomponents are delineated in the following sections.

3.1.2.1 Data Collection Engine

Considering the increasing digital presence of news organizations, the day to day news stories are easily accessible through RSS feeds. The data collection engine agglomerates news stories every hour from the feeds of popular news organizations and political blogs. The complete list of news organizations is provided in Table 3.2. These outlets are selected to represent a balanced collection of outlets: politically (i.e., liberal and conservative), new and old (e.g.,Buzzfeed and NYT), public and private (e.g., NPR and Fox News), for-profit and non-profit (e.g., CNN and ProPublica), wire services [1] (e.g., Reuters and AP), and to include some smaller but influential outlets (e.g., The McClatchy). The timeline of the articles trace back to February 2015. RSS feeds provide hyperlinks to the article and we use a simple crawling mechanism to download the HTML document containing the article.

The HTML Document Object Model (DOM) is extracted from the feeds and

[1] A wire service is a news agency that supplies syndicated news to other outlets.

	Orgs & Blogs
1	CNN
2	Fox News
3	The Wall Street Journal
4	ProPublica
5	Politico
6	The McClatchy
7	The Washington Post
8	BuzzFeed
9	National Public Radio
10	The Huffington Post
11	Associated Press
12	Reuters
13	The New York Times
14	The L.A. Times

Table 3.2: News Organizations and Political Blogs

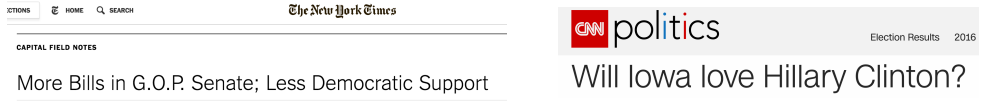
passed to a structural parser. The parser uses Beautiful Soup [2], which is a python package for parsing HTML to extract the headline, body, date-of-publication, and authors of each article and stores it in a database. At this stage, data deduplication is performed to ensure that only one copy of an article is in the database. This is necessary since articles from wire services like the AP and Reuters sometimes end up in the feeds of other news outlets. On average 2,000 articles are ingested daily from the 14 media outlets. Next, all unique articles are passed to election news classifier.

3.1.2.2 Election News Classifier

The data collected using RSS feeds includes news about Sports, Politics, Entertainment, etc. All the news stories, including those that belong to the political category, may not necessarily fit in the election scenario. Therefore, it is necessary to have a classifier which predicts if the particular news story is election related or not. Figure 3-3 shows an example of election and non-election news article. We eventually

[2]https://www.crummy.com/software/BeautifulSoup/

segregate election specific news stories from the whole data collection for further processing.



Non-Election News **Election News**

Figure 3-3: Example of election (right) and non-election news (left)

Election News Classifier is a binary classifier which takes a news article as input and determines whether it is about the 2016 US election or not. Since news articles usually contain clean and structured language, they can easily be classified as election-related using Bag-of-Word (BoWs) features. We use a chi-square test for feature selection technique. Chi-square measures the lack of independence between a term in an article and a class (in this case the election). High scores on chi-square indicate that the null hypothesis of independence should be rejected and that the occurrence of the term and class are dependent. The features are ranked based on their scores and the top 20,000 features form the vocabulary for the binary classifier. Next, using scikit-learn [53] - a Python machine learning library - a binary Maximum Entropy (MaxEnt) text classifier [49] is trained on a balanced dataset of 1,000 manually labeled news articles.

3.1.2.2.1 Evaluation The classifier was evaluated on a separate balanced test set of 300 articles, with the precision and recall of the election-related articles being 0.93 and 0.86 respectively. Table 3.3 shows the performance of MaxEnt classifier in comparison to other models like SVM, Naive Bayes, etc.

Methods	*Precision*	*Recall*	$F1 - Score$
MaxEnt Classifier	**0.93**	0.86	**0.90**
Naive Bayes	0.88	0.79	0.83
SVM	0.87	**0.89**	0.88
Random Forest	0.95	0.70	0.81

Table 3.3: Performance of Election News Classifier

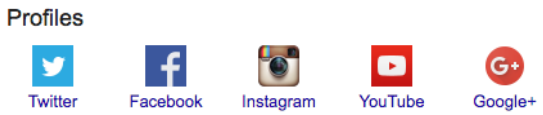
3.1.2.3 Entity Recognizer

The purpose of entity recognizer is to identify personality names mentioned in news articles in the context of the election. The personalities might include campaign managers, pollsters, analysts, etc. We take advantage of an off-the-shelf Stanford Named Entity Recognition (NER) [43] tool for identifying people in the articles. There are possibilities of false positives and typographical errors. To address this, each new person that is detected is automatically searched on Google. When available, we use that to extract features for a people validity classifier which determines whether a newly extracted person is indeed a person. The features used for this classifier are as follows:

- Google Spelling Suggestions

- Wikipedia infobox properties and description

- Existence of verified social profiles



- Search term part (partial or full match) of URL (e.g., search term: Bill Nelson, last token of the URL partially matches the search term)

- Penultimate tag before the partial/fully matched part of URL. (e.g., search term: Patrick Caddell, penultimate tag: author)

- Social media profiles in search results
- Tokens before and after the search term in the search result titles.

3.1.2.3.1 Evaluation The people validity classifier is a binary classifier that uses the above features to verify the validity of the personality name. We used a balanced dataset of 500 different names to train and test a MaxEnt classifier [49]. The accuracy of the system is 95%. The search results for every person are stored in our database for future analysis.

Methods	*Precision*	*Recall*	$F1 - Score$
MaxEnt Classifier	**0.95**	**0.95**	**0.95**
Naive Bayes	0.84	0.85	0.84
SVM	0.91	0.89	0.90
Random Forest	0.92	0.80	0.87

Table 3.4: Performance of People Validity Classifier

3.1.2.4 Person Categorizer

The personalities obtained from the recognition stage are categorized into 8 different segments (Table 3.5 shows all the different categories). Person Categorizer enables us to filter non-political (Sports, Entertainment, etc.) personalities from being mapped on to twitter or be used for election tweets look up. The Twitter handles of specific personalities are obtained from Google search and Wikipedia data. The list of

personality names that fall under Government/Ex-Government officials and Candidate/Party categories is the output of the media knowledge miner.

	Categories
1	Pollster/Analyst
2	Business/Academic
3	Cand/Party
4	Govt Officials/Ex-Officials
5	Sports/Entertainment
6	International
7	Interest Group/Religion
8	Media

Table 3.5: Different classes of Person Categorizer

To achieve this, we apply a semi-supervised approach. First, we model the semantic context in which these people are mentioned in an unsupervised manner. We use the context from the corresponding news articles and augment them with context from the Google search results and Wikipedia infoboxes properties of the people when available. The context from the news articles is obtained by continuous distributed vector representations using the Skip-gram model (see Section 3.1.1.1 for more details). We can do this since we have tens of thousands of articles to capture the context in which people are mentioned.

However, for the Google and Wikipedia results, we only have a limited dataset; therefore the context from the Google and Wikipedia results is captured by applying unsupervised feature learning using denoising autoencoders (DA). BoWs style to extract the context around the search results and the Wikipedia descriptions suffer from their inherent over-sparsity and fail to capture world-level synonymy and polysemy. The drawbacks lead to the requirement of a trained classifier being exposed to a very large set of labeled examples, in order to gain the sufficient predictive power for new examples. It becomes more problematic when the amount of labeled data is limited and the number of classes is large, which is the nature of our dataset. Hence, DA are used to extract the interesting features from the dataset. We, therefore, employed a

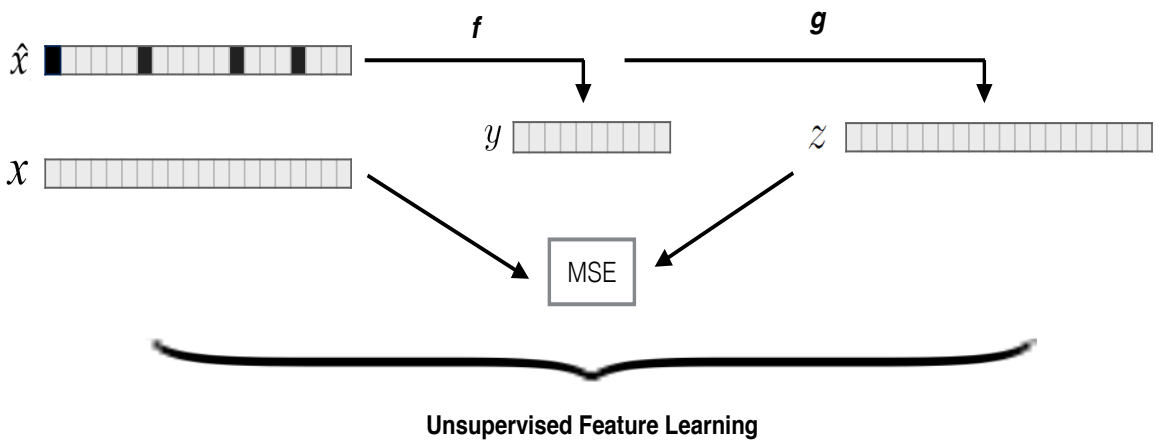


Figure 3-4: Domain Adaptive Topic Classification - Denoising Autoencoder (f is the encoder function, g performs decoding function, p is the proportion of corruption, $\hat{x}$ is the noisy data corrupted with proportion p from the input x, y is the encoder output,z is the decoder output)

denoising autoencoders [68] to first learn salient features in an unsupervised fashion on the search results and infobox dataset and then the extracted features are trained using the labeled dataset. Below, we explain how this is achieved.

Let V denote the vocabulary of the dataset. Each search result, along with their description and Wikipedia infobox properties are represented by a vector $x_i \in \mathbb{R}^{|V|}$. The aim of using denoising autoencoders is to translate the BoWs features into abstract feature representation that can capture the useful structures in the text and overcome the drawbacks mentioned above. A typical autoencoder is comprised of an encoder-decoder function where the encoder transforms the input BoW features to abstract feature representation and the decoder reconstructs the abstract features back to the $|V|$ dimensional vector space. A good representation is one that can perform the denoising task well by extracting useful structures in the input distribution rather than replicating the input representation.

A denoising autoencoder is therefore trained to reconstruct a clean input from a corrupted version of it. This is done by first adding some noise (with proportion p) to the initial input x_i into $\hat{x}_i$. Noisy input $\hat{x}_i$ is then mapped, as with the basic

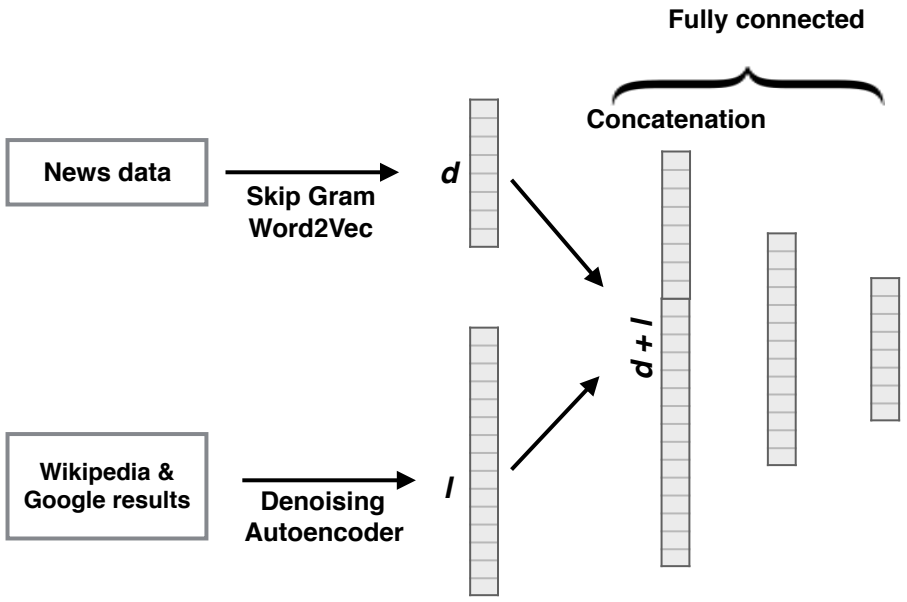


Figure 3-5: Personality Classifier - Skip-gram model for semantic context from news articles, Denoising Autoencoder for context from Wikipedia and Google results. ($d = 256$, $l = 1024$)

autoencoder, to a abstract representation:

$$y = f(\hat{x}_i) = s(W\hat{x}_i + b) \tag{3.2}$$

We use y to reconstruct the input by:

$$z = g(y) = s(W'y + b') \tag{3.3}$$

Parameters are trained to minimize the average reconstruction error over a training set, i.e., to have z as close as possible to the original input x_i. The denoising autoencoders are minimizing the same reconstruction loss between the original input and its reconstruction from the abstract features. So this still amounts to maximizing a lower bound on the mutual information between clean input x_i and representation y. Adding noise forces the learning model to extract clever features from the data rather than identity. Once we train the DA, the encoder performs a non-linear transformation of the BoW features of size r.

The output representation from the encoder is concatenated with the skip-gram entity representation and given as input to a softmax layer (See Figure 3-5) with a dropout on the penultimate layer ($\rho = 0.5$). The output of the softmax layer is the probability distribution over the eight different categories which is obtained by minimizing the cross-entropy loss as in Equation 3.12. Adam algorithm is utilized for the purpose of optimization. As mentioned earlier, the personalities classified as Government/Ex-Government officials and Candidate/Party are added to the search space of election tweet aggregator explained in Section 3.1.1.1.

3.1.2.4.1 Evaluation We evaluated our classifier on an independent dataset of 400 manually labeled people, with the average precision and recall being 0.88 and 0.79 respectively (weighted F-score of 0.83). Table 3.6 shows the precision, recall and F1-score for each category. Since Government/Ex-Government officials and Candidate/Party categories have high precision, they can be used for our analysis. The F1 score of traditional approaches like MaxEnt Classifier & SVM on the same data was 0.78 and 0.75 respectively.

Categories	*Precision*	*Recall*	$F1 - Score$
Pollster/Analyst	0.75	0.65	0.69
Business/Academic	0.80	0.80	0.80
Cand/Party	0.93	0.78	0.85
Govt Officials/Ex-Officials	0.88	0.91	0.89
Sports/Entertainment	0.75	0.82	0.78
International	0.67	0.33	0.44
Interest Group/Religion	0.60	0.50	0.55
Media	0.80	0.79	0.82
Avg	0.88	0.79	0.83

Table 3.6: Performance of Person Categorizer

3.2 Twitter Analysis Pipeline

3.2.1 Election Classifier

The tweets captured using the expanded query method include a number of spam and non election-related tweets. In most cases, these tweets contain election-related hashtags or terms that have been maliciously put in non-election related tweets in order to increase their viewership. The election classifier acts as a content-aware filter that removes non-election and spam tweets from the data captured by the expanded query.

Because of the noisy and unstructured nature of tweets, we use a deep character-level election classifier. Character-level models are great for noisy and unstructured text since they are robust to errors and misspellings in the text. Our classifier models tweets from character level input and automatically learns their abstract textual concepts. For example, our character-level classifier would closely associate the words"no" and "noooo" (both common on twitter), while a word-level model would have difficulties relating the two words.

The model architecture, illustrated in Figure 3-6), is a slight variant of the deep character level convolutional neural network introduced by Zhang et al [77]. We adapted their model to work with short text with a predefined number of characters, such as tweets with their 140 character limit. The character set considered for our classification includes the English alphabets, numbers, special characters and unknown character. There are 70 characters in total, shown below.

```
abcdefghijklmnopqrstuvwxyz0123456789
-,;.!?:'"/\|_#$%&^*~‘+-=<>()[]{}
```

Each character in the tweet can be encoded using one-hot vector $x_i \in \{0,1\}^{70}$. Hence, a tweet is represented as a binary matrix $x_{1..150} \in \{0,1\}^{150x70}$ with padding wherever necessary, where 150 is the maximum number of characters in a tweet plus padding and 70 is the size of the character set.

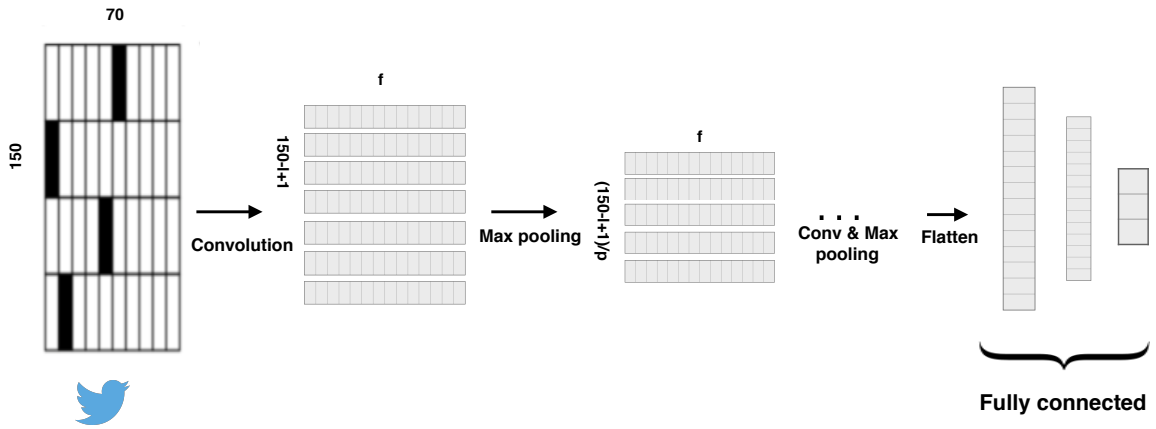


Figure 3-6: Election Classifier (f is the encoder function, g performs decoding function, p is the proportion of corruption, $\hat{x}$ is the noisy data corrupted with proportion p from the input x, y is the encoder output,z is the decoder output)

Each character in the tweet can be encoded using one-hot vector $x_i \in \{0,1\}^{70}$. Hence, the tweet is represented as a binary matrix $x_{1..150} \in \{0,1\}^{150x70}$ with padding wherever necessary, where 150 is the maximum number of characters in a tweet (140 tweet characters and padding) and 70 is the size of the character set shown above.

Each tweet, in the form of a matrix, is now fed into a deep model consisting of four 1-d convolutional layers. A convolution operation employs a filter w, to extract l-gram character feature from a sliding window of l characters at the first layer and learns abstract textual features in the subsequent layers. This filter w is applied across all possible windows of size l to produce a feature map. A sufficient number (f) of such filters are used to model the rich structures in the composition of characters. Generally, with tweet s, each element $c_i^{(h,F)}(s)$ of a feature map F at the layer h is generated by:

$$c_i^{(h,F)}(s) = g(w^{(h,F)} \odot \hat{c}_i^{(h-1)}(s) + b^{(h,F)}) \tag{3.4}$$

where $w^{(h,F)}$ is the filter associated with feature map F at layer h; $\hat{c}_i^{(h-1)}$ denotes the segment of output of layer $h-1$ for convolution at location i (where $\hat{c}_i^{(0)} = x_{i...i+l-1}$ — one-hot vectors of l characters from tweet s); $b^{(h,F)}$ is the bias associated with that filter at layer h; g is a rectified linear unit and $\odot$ is element-wise multiplication. The

output of the convolutional layer $c^h(s)$ is a matrix, the columns of which are feature maps $c^{(h,F_k)}(s)|k \in 1..f$.

The output of the convolutional layer is followed by a 1-d max-overtime pooling operation [14] over the feature map and selects the maximum value as the prominent feature from the current filter. Pooling size may vary at each layer (given by $p^{(h)}$ at layer h). The pooling operation shrinks the size of the feature representation and filters out trivial features like unnecessary combination of characters (in the initial layer). The window length l, number of filters f, pooling size p at each layer are given in Table 3.7.

Layer (h)	*Window Size* (l)	*Filters* (f)	*Pool Size* (p)
1	7	256	3
2	7	256	3
3	3	256	N/A
4	3	256	N/A
5	3	256	N/A

Table 3.7: Convolutional Layers with non overlapping pooling layers used for election classifier.

The output from the last convolutional layer is flattened. The input to the first fully connected layer is of size 2048 (8×256). This is further reduced to vector of sizes 1024 and 512 with a single output unit where we apply the sigmoid function (since this is a binary classification problem). For regularization we applied a dropout mechanism after the first fully connected layer. This prevents co-adaptation of hidden units by randomly setting a proportion ρ of the hidden units to zero (for our case, we set $\rho = 0.5$). We have the binary cross-entropy loss as the objective:

$$BCE(t, o) = -t \log(o) - (1 - t) \log(1 - o) \tag{3.5}$$

where t is the target and o is the predicted output. The *Adam Optimization algorithm* [35] is used for learning the parameters of our model.

The model was trained and tested on a dataset containing roughly 1 million election-related tweets and 1 million non-election related tweets. These tweets were collected using distant supervision. The "high precision"" seeds terms explained in the previous section were used to collect the 1 million election-related tweets and an inverse of the terms was used to collect 1 million non-election-related tweets. The noise in the dataset from the imperfect collection method is offset by the sheer number of examples. Ten percent of the dataset was set aside as a test set. The performance of our model is shown in Table 3.8.

Methods	*Precision*	*Recall*	$F1 - Score$
Our Model	**0.99**	**0.99**	**0.99**
Logistic Regression	0.93	0.93	0.93
Naive Bayes	0.90	0.87	0.88

Table 3.8: Performance of Election Classifier on the test data collected using distant supervision

3.2.1.1 Evaluation

We evaluated the full Twitter ingest engine on a balanced dataset of 1,000 manually annotated tweets. In order to reduce potential bias, the tweets were selected and labeled by an annotator who was familiar with the US political landscape and the upcoming Presidential election but did not have any knowledge of our system. The full ingest engine had an F-score of 0.92, with the precision and recall for the election-related tweets being 0.91 and 0.94 respectively. Note that the evaluation of the election classifier reported in the last section is different since it was on a dataset that was collected using the election related seed terms, while this evaluation was done on tweets manually selected and annotated by an unbiased annotator. Character-level model learns the words and phrases at various stages of the hierarchy. For the purpose of visualization, we reduce the dimensionality of an intermediate fully connected layer of size 512 to 2-dimensions using truncated-SVD.

In Figure 3-7, the probability that a word, phrase or tweet is related to elections

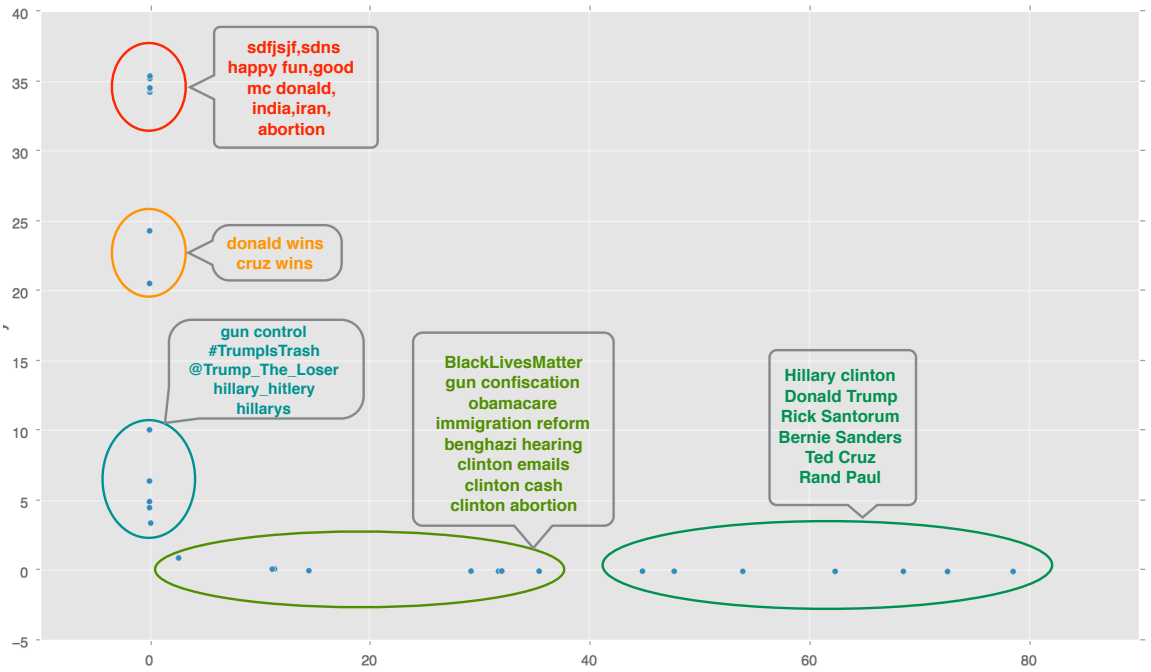


Figure 3-7: Visualizing intermediate layer using truncated-SVD to reduce dimensionality. The model is able to learn the words and phrases. The word "Abortion" is found on the top corner of the graph referring to low probability score due to its generality. However, the phrase "clinton abortion" has higher probability of being related to election and hence, it is clear from its position on the graph.

increase from top to bottom. It is interesting to see the clear demarcation between the election and non election terms. Table 3.9 shows the examples of tweets classified by our model.

3.2.2 Topic Classifier

The next stage of our Twitter analysis pipeline involves topic classification. With the help of an annotator with political expertise, we identified 22 election-related topics that capture the majority of the issue-based conversation around the election. These topics are listed in Table 3.10. We use a convolutional word embedding model to classify the tweets into these 22 different topics.

The convolutional embedding model (see Figure 3-8) assigns a d dimensional vector to each of the n words of an input tweet resulting in a matrix of size $n \times d$. Each of these vectors are initialized with uniformly distributed random numbers i.e. $x_i \in \mathbb{R}^d$.

Election Tweets	*Non − Election Tweets*
GOP presidential contender Carson makes South Carolina swing https://t.co/M08ez6TpEK #scnews https://t.co/p6VwPKCGre	Tried to watch Donald in NH on YouTube, froze the page had to reboot. Tried to watch another one. Ads galore Wonder who is doing it.
Paid? Like Trump needs the money. Although, I do think Trump would be soft on Hillary https://t.co/qyPbaJDF6m	This is War #war #gun #rifle #closer #kill #killme #soldier https://t.co/0xuQA7BH3
Hillary is candidate 'most likely to cut you off in traffic'	Tips for Your Oral Health Care Plan #beachbraces #teachingtuesday https://t.co/QHnFSqwMx6 #GOPDebate
@FordFlatheadV8: How would @hillaryclinton respond to Syrian refugees? Check out her foreign policy record here: https://t.co/vPFttGQceq	@trump_world Supply demand Friends, if I have a wedding and invite you DONOT bring a black date to traditional white vows, OK? I trust you!

Table 3.9: Sample Results of the Election Classifier

The model, though randomly initialized, will eventually learn a look-up matrix $\mathbb{R}^{|V|\times d}$ where $|V|$ is the vocabulary size, which represents the word embedding for the words in the vocabulary.

A convolution layer is then applied to the $n \times d$ input tweet matrix, which takes into consideration all the successive windows of size l, sliding over the entire tweet. A filter $w \in \mathbb{R}^{h\times d}$ operates on the tweet to give a feature map $c \in \mathbb{R}^{n-l+1}$. We apply a max-pooling function [14] of size $p = (n - l + 1)$ shrinking the size of the resultant matrix by p. In this model, we do not have several hierarchical convolutional layers - instead we apply convolution and max-pooling operations with f filters on the input tweet matrix for different window sizes (l).

The vector representations derived from various window sizes can be interpreted as prominent n-gram word features for the tweets. These features are concatenated to give a vector of size $f \times L$, where L is the number of different l values which is further compressed to a size k before passing it to a fully connected softmax layer whose output is the probability distribution over topic/sentiment labels. Two dropout layers

Tweet	*Topic*
@tedcruz You say a lot of but you don't back it up with facts so what you spout is merely unadulterated bullshit.	Other
Hillary Clinton's Email: the Definitive Timeline #HillaryClinton https://t.co/wFkWYgBWcb	Ethics
Don't Let Wall Street Get Away With Dodd-Frank Reform Rollbacks #tcot #gop #democrats #oligarchy https://t.co/1PbFGrRnUA	Financial Regulation
@MartinOMalley @NRA California has some of the strongest gun laws in the country, far stronger than anything proposed by Feds. Nice try. -	Guns

Table 3.10: Left: Election Topics, Right:Sample Results from Tweet Topic Classifier

Hyperparameters	**Values**
Embedding Size (d)	300
Window Sizes (l)	$2, 3, 4$
Penultimate Layer size (k)	256
Filters (f)	200

Table 3.11: Hyperparameters based on cross-validation for topic convolutional model

are introduced, one on the feature concatenation layer and other on the penultimate layer for regularization ($\rho = 0.5$). The hyperparameters used for this model are given in Table 3.11.

To learn the parameters of the model we minimize the cross-entropy loss as the training objective. It is given by

$$CrossEnt(p, q) = -\sum p(x) \log(q(x)) \tag{3.6}$$

where p is the true distribution (1-of-C representation of ground truth) and q is the output of the softmax. This, in turn, corresponds to computing the negative log-

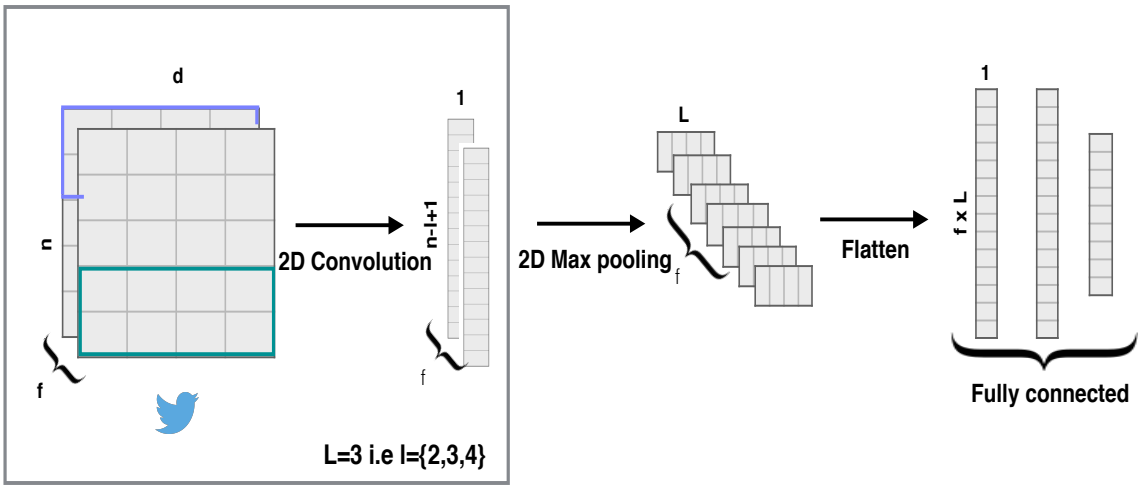


Figure 3-8: Tweet Topic Convolutional Model (n=50 is maximum number of words in a tweet, d is the word embedding dimension, L is the number of n-grams, f is the number of filters, $(n - ngram + 1 \times 1)$ is the pooling size)

probability of the true class. We resort to Adam optimization algorithm [35] here as well.

3.2.2.1 Training & Evaluation

Distance supervision was used to collect the dataset for the model. We used 85% of the collected data for training and 15% for testing. The same annotator that identified the 22 election-related topics also created a list of "high precision" terms and hashtags for each of the topics. These terms were expanded using the same technique as was used for the ingest engine. The expanded terms were used to collect a large number of example tweets (tens of thousands) for each of the 22 topics. Table 3.13 shows the precision, recall and F-1 score of our model and other traditional methods. The F-1 score at per-category level remained above 94%. As mentioned earlier in the election classifier, though distance supervision is noisy, the sheer number of training examples make the benefits outweigh the costs associate with the noise, especially when using the data for training deep neural networks. We evaluated the topic and sentiment convolutional models on a set of 1,000 election-related tweets which were manually annotated. The topic classifier had an average (averaged across all classes) precision

Topics	*Top Terms*
Health care	medicaid, obamacare, autism, health care, medicare, vaccination, vaccine, vaccine damaged, pro obamacare, alzheimer, anti vaccination, care you, anti obamacare, markets healthcare, signups, repealandreplace, obamacare burying, deductibles, obamacare defend, ddiamond, obergefell, obamacare lie
Racial Issues	blacklivesmatter, racism, ferguson, affirmative, racist, nonracist, quotas, outracist, antiracist, civil rights, bland, racisms, racistremarks, jorgeramoslink, mccarranferguson, aracist, skolnik, antiracism,, anti blacklivesmatter, freddie, unrest, tea party racist, racist undertones
Foreign Policy/National Security	unrights, diplomatic, gadhafi, irritant, marines, pro palestinian, diplomacy, untrump, benghazi, paris, confidant, anti terrorism, isil, isis,lybia, kunduz, putin, shia, cuba, ambassador, khamenei, unpc, antiisrael, jongun, ethopians,iran
Guns	concealedcarry, gunssavelives, lapierre, sandy hook, pro gun, gun rights, pro 2nd amendment, nra, 2nd amendment, nraila, gun violence, mass shooting, amend, unsung, anti 2nd amendment, peeping, progun, shooting, restated, uncanny, gun, murfreesboro, antigunrights, license
Immigration	securetheborder, noamnesty, antiimmigrant, antiimmigration, norefugees, immigrationreform, onimmigration, deport, amnesty, immigrationcomments, deportillegals, proimmigrant, egan, immigration, proillegalalien, proimmigration, undocumented, self deport, sellamnesty, immigrants, immigrant bashing, shadylady, immigrant, patrol
Jobs/Employment	equalwork, employment, jobs freedom prosperity, unemployment, unemployed, jobs, familyleave, makeamericgreatagain, pay equity, minimum wage, anti labor, fullemployment, jobsmighty, powerjobs, paidleave, preemployment, projobs, jobscreating, laborforce, labor, jobsrecord, shutupaboutyourjobs, employment stats

Table 3.12: Top-ranked terms from vocabulary for each topic

and recall of 0.91 and 0.89 respectively, with a weighted F-score of 0.90. Table 3.12 gives us the slice of top-ranked terms for each topic from the vocabulary. Table 3.10 also shows the sample results from our model.

Methods	*Precision*	*Recall*	$F1 - Score$
Our Model	**0.98**	**0.97**	**0.97**
Logistic Regression	0.91	0.86	0.88
Naive Bayes	0.86	0.79	0.82

Table 3.13: Performance of Topic Classifier on the test data collected using distant supervision

3.3 Representative Tweet Extractor

Representative Tweet Extractor consists of three stages of analysis. Given a list of tweets under each topic in a particular domain, we apply the following steps to generate representative tweets:

Similarity Measure

We need to calculate similarities between all pair of tweets within each topic. It is very important to have a sophisticated method to calculate similarity between tweets as it is difficult to map the semantic and syntactic properties of the unstructured tweets into a vector representation. Most Commonly used approaches include converting tweets into vectors using TF-IDF, distributed word vectors [46] and calculating the distance using metrics like cosine, euclidean, etc. However, these word-level approaches have their inherent limitations in the context of tweets. Therefore, we implement Tweet2Vec, a character-level CNN-LSTM encoder-decoder approach, to learn general purpose vector representation of tweets. We also evaluate Tweet2Vec using two classification tasks. The evaluations demonstrate the power of the tweet embeddings generated by our model for tasks that involve categorization and semantic relatedness.

Clustering

Next, we cluster the topic-centric tweets based on the calculated similarity measure. There are many techniques for clustering, such as hierarchical clustering, that form groups of objects using agglomerative or divisive approach [42]. In our work, we build a semantic network with topic-centric tweets represented as nodes. The edges between any pair of nodes are weighted by the similarity between the tweets associated with those nodes. We explore the Louvain method [8] to extract communities from large networks. It is a greedy optimization method that appears to run in time $O(nlogn)$.

Ranking

Finally, we rank these tweets within each of the generated communities or clusters. There are a number of network analysis metrics like Degree centrality, Betweenness centrality, etc. But we apply a modified version of PageRank [45] to handle weighted edges as our ranking metric. PageRank is a way of measuring the importance of a node in a network. The number and quality of edges that are incident on the node decides the importance of a node in the network. Similarly, tweet nodes that are connected to a large number of other tweet nodes through high similarity (indicative of quality) edges, are the most representative ones within that network.

The following sections explain each of these stages of the representative tweet extractor in detail.

3.3.1 Tweet2Vec

Tweet2Vec [72] is a method for generating general-purpose vector representation of tweets. Tweet2Vec removes the need for expansive feature engineering and can be used to train any standard off-the-shelf classifier (e.g., logistic regression, svm, etc). Tweet2Vec uses a CNN-LSTM encoder-decoder model that operates at the character level to learn and generate vector representation of tweets. Our character-level model can deal with the noise and idiosyncrasies in tweets. The tweet embeddings generated

from this model can help improve the performance of complex tasks that involve tweets like Stance detection [66], Speech act classification[71] and Rumor detection & verification (specially the linguistic features) [69].

3.3.2 CNN-LSTM Encoder-Decoder

In this section, we describe the CNN - LSTM encoder-decoder model that operates at the character level and generates a vector representation of the tweets. Encoder consists of convolutional layers to extract features from the characters and an LSTM layer to encode the sequence of features to a vector representation, while the decoder consists of two LSTM layers which predict the character at each time step from the output of encoder.

3.3.3 Character-Level CNN Tweet Model

Character-level CNN (CharCNN) is similar to the model described in Section 3.2.1. But there are slight variations to the model. Since we intend to have a general purpose tweet representation, it is important to have some extra symbols in the character set like B,E,S,H,U (upper-case characters). The new symbols in the characters indicate the following.

- B - Begin Symbol
- E - End Symbol
- S - Sad emoticon [e.g., :-(, :((]
- H - Happy emoticon [e.g., :-), :))]
- U - URL (e.g., http://t.co/jandsjk213)

Hence, the new character set includes English alphabets, numbers, special characters, the above new symbols and unknown character. There are 75 characters in total, given below:

```
abcdefghijklmnopqrstuvwxyz0123456789
-,;.!?:'"/\|_#$%&^*~‘+-=<>()[]{}BESHU
```

As explained in Section 3.2.1, each character in the tweet is encoded using one-hot vector $x_i \in \{0,1\}^{75}$. Hence, the tweet is represented as a binary matrix $x_{1..150} \in \{0,1\}^{150x75}$ with padding wherever necessary, where 150 is the maximum number of characters in a tweet (140 tweet characters and padding) and 75 is the size of the character set. The rest of the model is same, except that we have a few changes in the hyperparameters. The window length l, number of filters f, pooling size p at each layer are given in Table 3.14.

Layer (h)	*Window Size* (l)	*Filters* (f)	*Pool Size* (p)
1	7	512	3
2	7	512	3
3	3	512	N/A
4	3	512	N/A

Table 3.14: Layer Parameters of CharCNN

We define $CharCNN(T)$ to denote the character-level CNN operation on input tweet matrix T. The output from the last convolutional layer of CharCNN(T) (size-10×512) is subsequently given as input to the LSTM layer which works on sequences (explained in Section 3.3.4 and 3.3.5) and hence, pooling operation is restricted to initial layers of the model.

3.3.4 Long-Short Term Memory (LSTM)

In this section we briefly describe the LSTM model [27]. Given an input sequence $X = (x_1, x_2, ..., x_N)$, LSTM computes the hidden vector sequence $h = (h_1, h_2, ..., h_N)$ and output vector sequence $Y = (y_1, y_2, ..., y_N)$. At each time step, the output of the module is controlled by a set of gates as a function of the previous hidden state h_{t1}

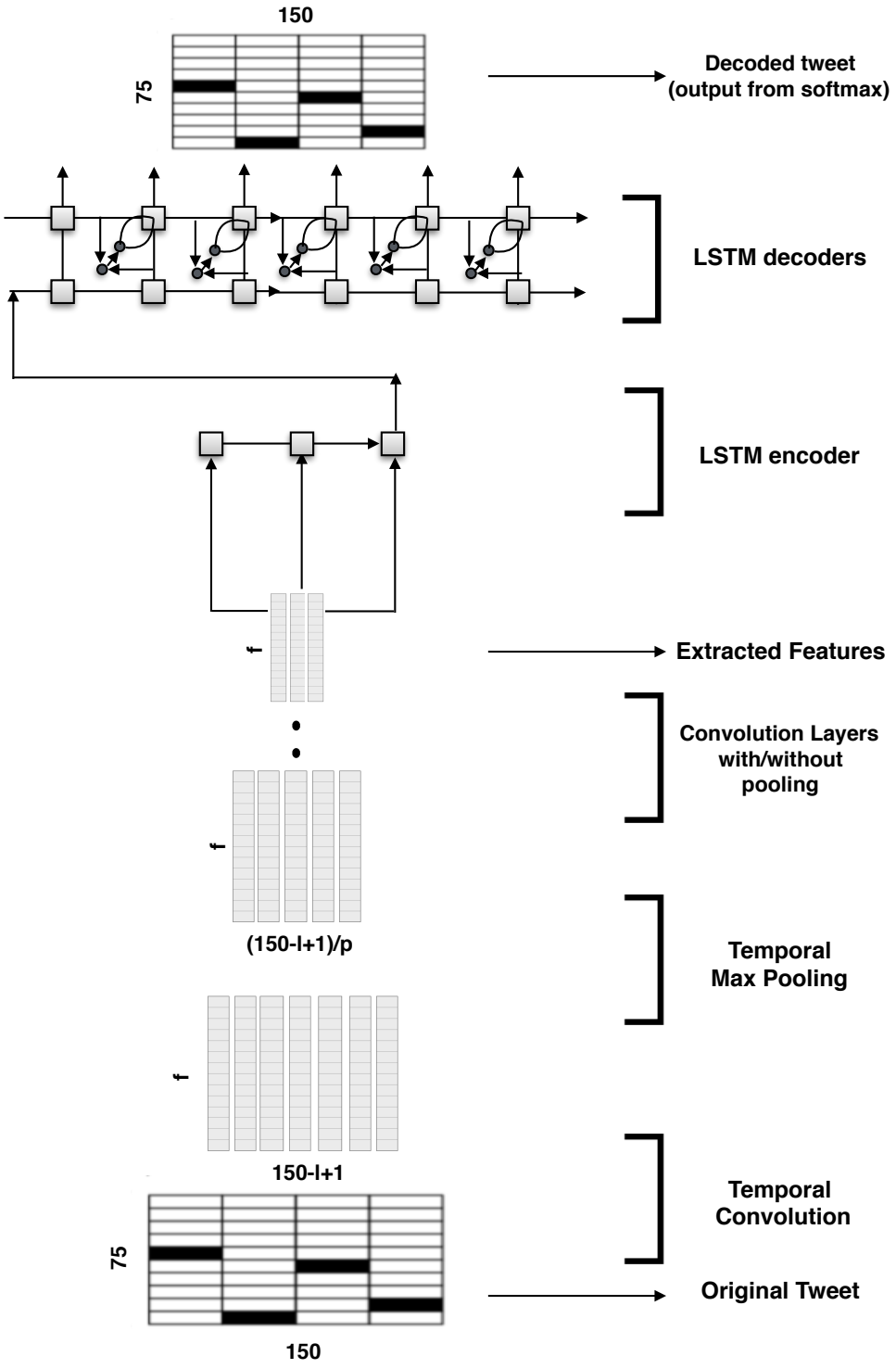


Figure 3-9: Illustration of CNN-LSTM Encoder-Decoder Model

and the input at the current time step x_t, the forget gate f_t, the input gate i_t, and the output gate o_t. These gates collectively decide the transitions of the current memory cell c_t and the current hidden state h_t. The LSTM transition functions are defined as follows:

$$\begin{aligned} i_t &= \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \\ f_t &= \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \\ l_t &= tanh(W_l \cdot [h_{t-1}, x_t] + b_l) \\ o_t &= \sigma(W_o[h_{t-1}, x_t] + b_o) \\ c_t &= f_t \odot c_{t-1} + i_t \odot l_t \\ h_t &= o_t \odot tanh(c_t) \end{aligned} \tag{3.7}$$

Here, σ is the *sigmoid* function that has an output in [0, 1], *tanh* denotes the hyperbolic tangent function that has an output in $[-1, 1]$, and $\odot$ denotes the component-wise multiplication. The extent to which the information in the old memory cell is discarded is controlled by f_t, while i_t controls the extent to which new information is stored in the current memory cell, and o_t is the output based on the memory cell c_t. LSTM is explicitly designed for learning long-term dependencies, and therefore we choose LSTM after the convolution layer to learn dependencies in the sequence of extracted features. In sequence-to-sequence generation tasks, an LSTM defines a distribution over outputs and sequentially predicts tokens using a softmax function.

$$P(Y|X) = \prod_{t \in [1,N]} \frac{exp(g(h_{t-1}, y_t))}{\sum_{y'} exp(g(h_{t1}, y'_t)} \tag{3.8}$$

where g is the activation function. For simplicity, we define $LSTM(x_t, h_{t-1})$ to denote the LSTM operation on input x at time-step t and the previous hidden state h_{t-1}.

3.3.5 The Combined Model

The CNN-LSTM encoder-decoder model draws on the intuition that the sequence of features (e.g. n-gram character and word) extracted from CNN can be encoded

into a vector representation using LSTM that can embed the meaning of the whole tweet. Fig.3-9 represents the complete encoder-decoder model. The input and output to the model is the tweet represented as a matrix where each row is the one-hot vector representation of the characters. The procedure for encoding and decoding is explained in the following section.

3.3.5.1 Encoder

Given a tweet in the matrix form T (size: 150×75), the CNN (Section 3.3.3) extracts the features from the character representation. The one-dimensional convolution involves a filter vector sliding over a sequence and detecting features at different positions. The new successive higher-order window representations then are fed into LSTM (Section 3.3.4). Since LSTM extracts representation from sequence input, we will not apply pooling after convolution at the higher layers of Character-level CNN model.

$$H^{conv} = CharCNN(T) \tag{3.9}$$

$$h_t = LSTM(g_t, h_{t-1}) \tag{3.10}$$

where $g = H^{conv}$ is an extracted feature matrix where each row can be considered as a time-step for the LSTM. LSTM operates on each row of the H^{conv} along with the hidden vectors from previous time-step to produce embedding for the subsequent time-steps. The vector output at the final time-step, enc_N, is used to represent the entire tweet. In our case, the size of the enc_N is 256.

3.3.5.2 Decoder

The decoder operates on the encoded representation with two layers of LSTMs. In the initial time step, the end-to-end output from the encoding procedure is used as the original input into first LSTM layer. The last LSTM decoder generates each character, C, sequentially and combines it with previously generated hidden vectors of size 128, h_{t-1}, for the next time-step prediction. The prediction of character at each time step is given by:

$$P(C_t|\cdot) = softmax(T_t, h_{t1}) \tag{3.11}$$

where C_t refers to the character at time-step t,T_t represents the one-hot vector of the character at time-step t. The result from the softmax is a decoded tweet matrix T^{dec}, which is eventually compared with the actual tweet or a synonym-replaced version of the tweet (explained in Section 3.3.6) for learning the parameters of the model.

3.3.6 Data Augmentation & Training

We trained the CNN-LSTM encoder-decoder model on 3 million randomly selected English-language tweets populated using data augmentation techniques, which are useful for controlling generalization error for deep learning models. Data augmentation, in our context, refers to replicating tweet and replacing some of the words in the replicated tweets with their synonyms. These synonyms are obtained from WordNet [22] which contains words grouped together on the basis of their meanings. This involves selection of replaceable words (example of non-replaceable words are stopwords, user names, hash tags) from the tweet and the number of words n to be replaced. The probability of number n is given by a geometric distribution with parameter p in which $P[n] \sim p^n$. The index q of the synonym given a word is also determined by a another geometric distribution in which $P[s] \sim r^q$. In our encoder-decoder model, we decode the encoded representation to the actual tweet or a synonym-replaced version of the tweet from the augmented data. We used $p = 0.5$, $r = 0.5$ for our training. We also make sure that the POS tags of the replaced word are not completely different from the actual word. For regularization, we apply a dropout mechanism after the penultimate layer. This prevents co-adaptation of hidden units by randomly setting a proportion ρ of the hidden units to zero (for our case, we set $\rho = 0.5$). To learn the model parameters, we minimize the cross-entropy loss as the training objective using Adam Optimization algorithm [35]. It is given by

$$CrossEnt(p, q) = -\sum p(x) \log(q(x)) \tag{3.12}$$

where p is the true distribution (one-hot vector representing characters in the tweet) and q is the output of the softmax. This, in turn, corresponds to computing the negative log-probability of the true class. Table 3.15 shows the results of the nearest neighbour scored based on the cosine similarity for query tweets selected from a sample of political tweets and data from SemEval 2015: Task 1 Paraphrase and Semantic Similarity in Twitter semantic relatedness [75].

Query Tweet and nearest Tweet
@BernieSanders Happy New Year Bernie!! This is the year!
@BernieSanders You're so hip Bernie. Welcome the New Year!
Trump rings in New Years for the Fox News audience
Trump tells Fox about his New Year resolution. #MakeAmericaGreatAgain.
Kids he is a pitcher plays for the cubs and hit a grand slam
Grand Slam pitcher Travis Wood hits a grand slam today
BBM in iOS and Android apparently
iOS BBM coming to iOS Android
Jason Kidd and grant hill retire from nba
Jason Kidd finally retiring
Donald Trump on Fox News just now: "My New Year's resolution is to make America great again!
Trump tells Fox about his New Year resolution. #MakeAmericaGreatAgain.
The best way to ruin the ball drop is with @realDonaldTrump #FoxNews2016
THE BALL IS DROPPING! ! Let's hope it lands on Trump. #2016

Table 3.15: In each instance, the first tweet is the sample tweet while the second sentence is its nearest neighbour. Nearest neighbours were scored by cosine similarity.

3.3.7 Evaluation

We evaluate our model on two classification tasks: (1) Semantic Relatedness, and (2) Sentiment classification. Our experimental setup involves :

- Extraction of the vector representation of tweets using the learned encoder.
- Computing element-wise features for pairs of sentences for Task 1.

- Train a linear classifier from the vector representation of the tweet with no additional back propagation through the trained model for Task 2.

3.3.7.1 Semantic Relatedness

The first task is based on the SemEval 2015: Task 1 Paraphrase and Semantic Similarity in Twitter semantic relatedness [75]. Given a pair of tweets, the goal is to predict their semantic equivalence (i.e., if they express the same or very similar meaning), through a binary yes/no judgment. The dataset provided for this task contains 18K tweet pairs for training and 1K pairs for testing, with 35% of these pairs being paraphrases, and 65% non-paraphrases. We first extract the vector representation of all the tweets in the dataset using our Tweet2Vec model. We use two features to represent a tweet pair. Given two tweet vectors r and s, we compute their element-wise product $r \cdot s$ and their absolute difference $|r - s|$ and concatenate them together (Similar to [36]). We then train a logistic regression model on these features using the dataset. Cross-validation is used for tuning the threshold for classification. In contrast to our model, most of the methods used for this task were largely based on extensive use of feature engineering, or a combination of feature engineering with semantic spaces. Table 3.16 shows the performance of our model compared to the top four models in the SemEval 2015 competition. We also compare our results with paragraph vectors [38] that generates distributed representations for sentences. As can be seen, our model (Tweet2Vec) outperforms all the top models, without resorting to extensive task-specific feature engineering.

Methods	*Precision*	*Recall*	$F1 - Score$
svckernel	0.680	0.669	0.674
ikr	0.569	**0.806**	0.667
nnfeats	**0.767**	0.583	0.662
Tweet2Vec	0.679	0.686	**0.677**
ParagraphVec	0.57	0.68	0.620

Table 3.16: Results of Paraphrase and Semantic Similarity in Twitter

3.3.7.2 Sentiment classification

The second evaluation is based on the SemEval 2015-Task 10B: Twitter Message Polarity Classification [58]. Given a tweet, the task is to classify it as either positive, negative or neutral in sentiment. The size of the training and test sets were 9,520 tweets and 2,380 tweets respectively (38% positive, 15% negative, and 47% neutral). As with the last task, we first extract the vector representation of all the tweets in the dataset using Tweet2Vec and use that to train a logistic regression classifier using the vector representations. Even though there are three classes, the SemEval task is a binary task. The performance is measured as the average F1-score of the positive and the negative class. Table 3.17 shows the performance of our model compared to the top four models in the SemEval 2015 competition (note that only the F1-score is reported by SemEval for this task). As can be seen, our model outperforms all the top models, again without resorting to any feature engineering. One of the methods (INESC-ID) employs word embeddings, similar to Word2Vec. It is worth noting that our character-level tweet embeddings outperformed these methods including representations generated using paragraph vectors. This is a small but noteworthy illustration of why our tweet embeddings are best-suited to deal with the noise and idiosyncrasies of tweets.

Methods	*Precision*	*Recall*	$F1 - Score$
INESC-ID	N/A	N/A	0.642
lsislif	N/A	N/A	0.643
unitn	N/A	N/A	0.646
Webis	N/A	N/A	0.648
Tweet2Vec	0.675	0.719	**0.656**
ParagraphVec	0.60	0.68	0.637

Table 3.17: Results of Paraphrase and Semantic Similarity in Twitter

3.3.8 Louvain method

The Louvain method [8] is an algorithm that is able to find high modularity partitions in large networks in a short time. Modularity measures how well a partition separates communities in a network [48]. Modularity is a scale value between -1 and 1 that measures the density of edges inside communities to edges outside communities. Optimizing this value theoretically results in the best possible grouping of the nodes of a given network, however going through all possible iterations of the nodes into groups is impractical so heuristic algorithms are used. For a weighted graph, modularity is defined as:

$$\Delta Q = \frac{1}{2m} \sum_{ij} [A_{ij} - \frac{k_i k_j}{2m}] \delta(c_i, c_j) \tag{3.13}$$

where A_{ij} represents the edge weight between nodes i and j. k_i and k_j are the sum of the weights of the edges attached to nodes i and j respectively. m is half the sum of all edge weights in the graph. c_i and c_j are the communities of the nodes, and δ is a simple delta function.

Now, for the Louvain method, it works by iteratively running two phases: the first phase tries to find a partition of the network that maximizes the modularity, and the second phase unfolds each partition found into a node and connects these new nodes with weights representing the connectivity of the partitions. The iteration stops when it is not possible to improve the modularity. This methodology gives us access to different resolutions of community detection (each iteration would correspond to a new resolution level). Furthermore, it is possible to efficiently find high modularity partitions for bigger networks than previously possible. Contrary to all the other community detection algorithms, the network size limit when using the Louvain method is limited by storage capacity rather than by computation time. Assume that we start with a weighted network of N nodes. First, we assign each node to a different community that results in a partition with N communities. Then, for each node i, we evaluate the gain of modularity that would result from replacing the community of i by the community of one of the neighbors of i. The community

that yields the maximum gain of modularity is the chosen one, and i is moved to that community (in case of a tie, a breaking rule is used), but we only change the community of i if the modularity gain is positive, otherwise no changes are applied. This process is applied sequentially for all the nodes in the network, and repeatedly with several passes, until no improvement for modularity can be found. In that case, the phase is complete and we move to the second phase. Several studies show that the ordering of the visiting of the nodes can influence the computation time [8].

Part of the algorithm efficiency is the result of the fact that the increase of modularity obtained by moving a node i from one community B to another community C can be easily computed using the following equation:

$$\Delta Q = [\frac{\sum_{in} + k_{i,in}}{2m} - (\frac{\sum_{tot} + k_i}{2m})^2] - [\frac{\sum_{in}}{2m} - (\frac{\sum_{tot}}{2m})^2 - (\frac{k_i}{2m})^2] \tag{3.14}$$

where $\sum_{in}$ is the sum of the weights of the links inside C, $\sum_{tot}$ is the sum of the weights of the links incident to nodes in C, k_i is the sum of the weights of the links incident to node i, $k_{i,in}$ is the sum of the weights of the links from i to nodes in C, and m is the sum of the weights of all the links in the network. A similar expression is used to calculate the increase of modularity by removing node i from community B.

The second phase of the algorithm consists in generating a new network where each node represents a community found in the original network by the first phase of the algorithm. The nodes of the new network are connected with links having weights that are the number of links connecting the communities that the nodes are representing in the original network. Within community links in the original network are represented as a self link in the new network with weight equal to the number of within community links. A pass is the execution of the first and second phases. The passes are iterated until no gain for modularity is possible. This methodology naturally incorporates a notion of hierarchy as communities of communities are built during the process.

Louvain method has been successfully used to efficiently detect and track stories

about real-world events on Twitter [70]. Based on the ideas from [70], we build a network with its nodes representing a tweet. The edges are weighted on the basis of the similarity score calculated between the two tweet embeddings. The universe of this network is restricted by the topic associated with tweets. We apply Louvain method to obtain communities within each of the topics. The final step is to rank the tweets within each community identified by the Louvain method. The next section gives a brief description about PageRank used to assert the importance of a tweet.

3.3.9 PageRank

Iterative graph-based ranking algorithms are essentially a way of deciding the importance of a vertex within a graph; in the context of search engines, it is a way of deciding how important a page is on the Web. Drawing parallels to our system, we employ such techniques to rank tweets in our semantic network. In this model, when one vertex links to another one, it is casting a vote for that other vertex. The higher the number of votes that are cast for a vertex, the higher the importance of the vertex. Moreover, the importance of the vertex casting the vote determines how important the vote itself is, and this information is also taken into account by the ranking model. Hence, the score associated with a vertex is determined based on the votes that are cast for it, and the score of the vertices casting these votes.

The original PageRank definition for graph-based ranking is assuming unweighted graphs. However, in our model, the graphs contain implicitly devised links, i.e., the edges carry similarity scores, which needs to be accounted for. In this direction we apply a modified version of the Pagerank algorithm introduced by [45].

Let $G = (V, E)$ be a directed graph with the set of vertices V and set of edges E, where E is a subset of $V \times V$. For a given vertex V_i, let $In(V_i)$ be the set of vertices that point to it, and let $Out(V_i)$ be the set of edges going out of vertex V_i. The modified PageRank is defined as follows

$$S(V) = (1 - d) + d * \sum_{j \in In(v_i)} \frac{S(V_j) * w_{ij}}{\sum_{V_k \in Out(V_j)} w_{jk}} \tag{3.15}$$

where d is a damping factor that can be set between 0 and 1.

Starting from arbitrary values assigned to each node in the graph, the computation iterates until convergence below a given threshold is achieved. After running the algorithm, a fast in-place sorting algorithm is applied to the ranked graph vertices to sort them in decreasing order. The modified PageRank can be also applied on undirected graphs, in which case the out-degree of a vertex is equal to the in-degree of the vertex, and convergence is usually achieved after a fewer number of iterations.

Therefore, the modified PageRank is used to find the most representative tweet node within each of the communities inside a topic. All the high ranked tweets from each of these communities symbolize diverse views and perspectives around the topic. Though we use this at the topic level, it would not be possible to apply these steps for unsupervised learning of topics as (a) there are overlaps between various topics, (b) it would be difficult to let the system learn to represent all the major election issues without imposing a top down structure. The next chapter evaluates the output from representative tweet extractor based on a debate and compares our results with the conclusions of the newsroom based on their partnership with Twitter.

Chapter 4

Experiments & Analysis

The ability of the Tweet Exemplifier to recognize tweets that best represent the nature of distinct conversations can be really useful for journalists, political analysts and news consumers. In this chapter, we evaluate the representative tweets based on the Twitter discourse during a democratic debate. Besides that, it is important to delineate the contribution of some components like Media knowledge miner and Election tweet aggregator in this entire system. The following section describes our experiments with a democratic debate.

4.1 Tweet Exemplifier Perspective of Debate

There was a democratic debate hosted by CBS news[1] in Iowa on November 14, 2015. Recall that our system comprises of Twitter data and analysis pipeline, as explained in Chapters 2 and 3. This enables us to look up and track any election event. Given this framing, we evaluate the performance of the Tweet Exemplifier by a series of steps listed below.

1. Look up all the election related tweets before and during the debate
2. Categorize these tweets across 22 different election topics
3. Generate representative tweets from each of these topics

[1] http://www.cbsnews.com/news/democratic-debate-transcript-clinton-sanders-omalley-in-iowa/

4. Compare these tweets with the CBS news transcript[2], which gives details of the questions selected from tweets and prominent spike moments during the debate based on their partnership with Twitter.

5. Analyze the results of our system to verify how well the significant issues are being represented as the conversations unfold prior to and during the debate.

As the Twitter Exemplifier processes the data through its various stages, there are a number of tweets which are either opinions about the personality or the election process. These tweets are called Non-Issue tweets, while the tweets that fall under one of the 22 categories are called Issue-based tweets. The details of the data used for the analysis are given in Table 4.1.

Details	*Values*
No. of hours data	8 hrs [6 hrs (before debate) + 2 hrs (during debate)]
No. of election tweets	$\sim 200,000$
% of Issue-based Tweets	35%
% of Non-Issue Tweets	65%

Table 4.1: Details about the data used for debate analysis

Table 4.2 shows the questions that were asked to the candidate with reference to the tweets, while Table 4.3 lists the biggest spike moments recorded based on the analysis made by a team at Twitter. The questions in Table 4.2 are associated with Immigration and Campaign Finance topics. Though representative tweet extractor, described in Section 3.3, selects the highly ranked tweets from every cluster within each topic, we focus on Immigration (see Table 4.4) and Campaign Finance (see Table 4.5) topics initially. This helps us draw parallels between our model outcomes and the questions in Table 4.2. The top ranked tweets from communities of a subset of topics are given in Appendix. For the purpose of visualization, we reduce the dimensionality of the tweet representation (encoder output of Tweet2Vec: $enc_N = 512$) to a three dimensional space using truncated-SVD [30].

[2]http://www.cbsnews.com/news/democratic-debate-transcript-clinton-sanders-omalley-in-iowa/

Questions
Secretary Clinton, let me ask you a question from twitter which has come in and this is a question on this issue of refugees. The question is, with the U.S. preparing to absorb Syrian refugees, how do you propose we screen those coming in to keep citizens safe?
And Secretary Clinton, one of the tweets we saw said this, "I've never seen a candidate invoke 9/11 to justify millions of Wall Street donations until now." The idea being, yes, you were a champion of the community after 9/11, but what does that have to do with taking big donations?

Table 4.2: Questions that were asked to candidates during the debate with reference to tweets

Biggest Spike Moments
But it's what drove the conversation most – in order, Hillary Clinton, when she defended her integrity on campaign contributions and mentioned 60 percent of her donors are women. That was her biggest spike moment.
Martin O'Malley's big spike moment was when he called Donald Trump an "immigrant-bashing carnival barker." Remember that, as a two-phased (inaudible) from Martin O'Malley - "immigrant bashing carnival barker" for Donald Trump. Those were the three spike moments for the three candidates as recorded by twitter.
For Bernie Sanders, it's when we called Dwight D. Eisenhower a noted socialist for referring to his income tax brackets being very high, and much higher than they are now.

Table 4.3: Biggest spike moments recorded based on the analysis by Twitter

The top ranked tweets in Table 4.4 focuses on three main immigration issues: Immigration reforms (*Cluster* 1), Syrian Refugees (*Cluster* 3) and Martin 0'Malley's comment on Trump (*Cluster* 2). Comparing our results with the first question in Table 4.2, we see that *Cluster* 3, the largest cluster among the three clusters in Immigration topic, is very much in alignment with the issue raised by the CBS moderator. The tweets in *Cluster* 1 about immigration reforms are also noteworthy. The biggest spike moment for Martin O'Malley, mentioned in Table 4.3, is quite evident from the top tweets in *Cluster* 2, which contains tweet about his remarks on Donald Trump. The system is able to identify the diverse subtexts within the immigration topic.

Similar comparisons between Table 4.5 and Table 4.2, 4.3 can draw parallels between *Cluster* 2 and the question raised by the moderator. Furthermore, Hillary

Clusters	*Top*-4 *Tweets*
Cluster 1	#DemDebate what does immigration reform mean to you?
	#DemDebate What is your plan for Immigration Law Reform? How will you deal with the Illegal Immigration Crisis?
	Finally, someone talks about immigration reform, not border reform, which IS THE ISSUE. #MartinOMalley #DemDebate
	#immigrationreform is being addressed. We need reform! @HillaryClinton @BernieSanders @MartinOMalley #DemDebate
Cluster 2	Martin O'Malley called Donald Trump an "immigrant-bashing carnival barker" #DemDebate I love him now.
	Immigrant bashing, carnival barker, Donald Trump. You go, O'Mally #DemDebate
	"That immigrant bashing carnival barker." Nice. #DemDebate
	Martin O'Malley just called Donald Trump an "immigrant-bashing carnival barker." The #DemDebate is heating up: https://t.co/2cmDVnHSzZ
Cluster 3	@charliekirk11 Democrats don't want OpenBordersInSouth. They'd like CIR and screen all immigrants. Have 2 deal w/immigrants AllowedInForYears.
	#DemDebate How would you screen 65k immigrants to ensure they are not terrorist affiliated or any for that matter ?
	It's possible to accept so much more Syrians and other refugees into the United States. #DemDebate we need a screening process though
	How do you "screen and vet" refugees? No ID, no luggage, no proof. #DemDebate

Table 4.4: Representative Tweets for Immigration

Clinton's biggest spike moment is conspicuous from the nature of tweets in *Cluster* 3. Taking note of the almost similar size of clusters (refer Figure4-2), questions about corrupt campaign finance system is also very much relevant in this context. Likewise, we can find Bernie Sanders' biggest spike moment in one of the clusters in Budget/Taxation topic (see Figure A-1 in Appendix). We see that our system is able to capture different types of discussions around a particular topic on Twitter. The issues brought up by the moderators in correspondence to tweets were mainly

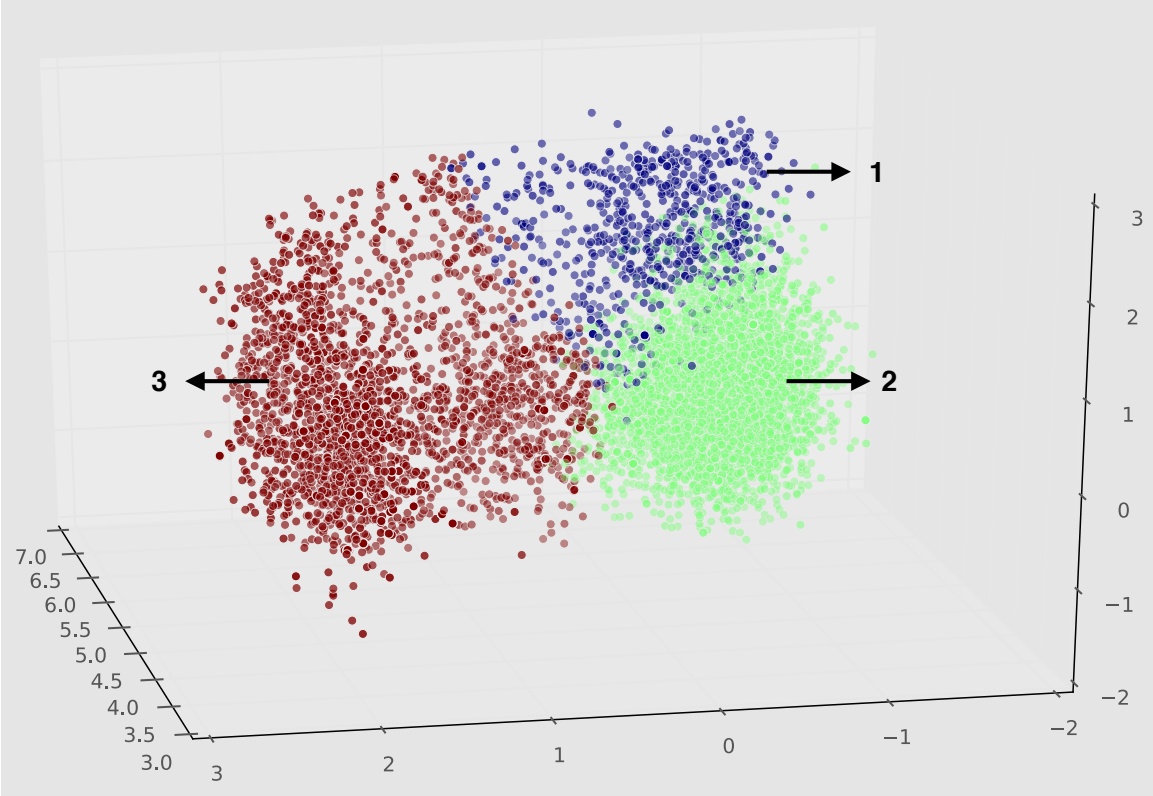


Figure 4-1: Clusters of Immigration tweets.

from Immigration and Campaign Finance topics. Relative to the share of Issue-based tweets, it is necessary to understand how representative these two topics are. The top seven topics and their share of conversations among those issue-based tweets are listed below.

1. Foreign Policy/National Security: 38.5%
2. Economy: 9.9%
3. Immigration: 9.7%
4. Health Care: 7.4%
5. Jobs/Employment: 5.4%
6. Guns: 4.8%
7. Campaign Finance 3.1%

It is quite evident from the above list that tweets about Foreign Policy/Nation Security and Economy roughly contribute to 50% of the issue-based conversation on Twitter. The prominent clusters in the Foreign Policy topic include ISIS, Paris attacks and Benghazi issue while the discussions in Economy topic varied from Wall Street reform/bailout to Socialism/Capitalism . The issues raised in those topics are also worthy of a question. Some of the prominent Twitter questions that the system would have selected based on the results as given in Tables A.3, A.1, 4.5 and 4.4 are as follows (some of which match the questions asked by the moderators. Note that questions 5 and 6 might not have been the top-pick of the system because Campaign Finance is the 7^{th} largest topic):

1. #DemDebate Looking at the #ParisAttacks what will you do as President to make sure countries in the middle east, stop funding ISIS?

2. Senator Sanders, does your socialist economic policy affect our right to economic freedoms? #DemDebate

3. #DemDebate How would you screen 65k immigrants to ensure they are not terrorist affiliated or any for that matter ?

4. #DemDebate What is your plan for Immigration Law Reform? How will you deal with the Illegal Immigration Crisis?

5. #DemDebate how do we address the problem of a corrupt campaign finance system?

6. #DemDebate How does @HillaryClinton expect us to believe she is going to reform Wall St. whn they r among her biggest donors?

Our system is capable of identifying topics that attract widespread Twitter audience and those that are underrepresented. Since the Tweet Exemplifier ranks tweets from each cluster in every topic, it is easier to keep track of the ensuing issues within any of those topics.

Clusters	*Top-4 Tweets*
Cluster 1	#DemDebate how do we address the problem of a corrupt campaign finance system?
	#DemDebate . How will you keep away the corrupt campaign finance?
	Not backing Sanders, but I appreciate his continuing emphasis on "the corrupt campaign finance system" #DemDebates
	Wonder if @BernieSanders would call campaign finance corrupt if the billionaires wrote checks to him. #DemDebate #SoreLoser
Cluster 2	The only thing #Hillary wants to go after wallstreet for is more campaign donations who is she kidding #DemDebate
	#DemDebate How does @HillaryClinton expect us to believe she is going to reform Wall St. whn they r among her biggest donors?
	Why are Hillary's biggest donors, big banks, despots and corrupt billionaires? https://t.co/z97kplwlcJ #DemDebate https://t.co/7oTIS8bjYu
	WOW, Hilary STILL trying to justify massive corporate campaign donations #DemDebate
Cluster 3	Why did people clap when Hilary said most of her donors were women. I don't get it. How is that an accomplishment. Gender card? #DemDebate
	Hillary said the majority of her donors are women. #DEMDebate
	Majority of @HillaryClinton's donors are women. #DemDebate
	A majority of your donors may be women, @HillaryClinton , but it's also possible that they could be wrong. Corporate women. #DemDebate

Table 4.5: Representative Tweets for Campaign Finance

4.2 Contribution of the System Components

The key components of Twitter Analysis Pipeline and Representative Tweet Extractor have been evaluated in previous sections. In this Section, we discuss the contribution of the sub-component of Twitter Data Pipeline: Election Tweet Aggregator. A list of election-related personalities discerned by the person categorizer model is added

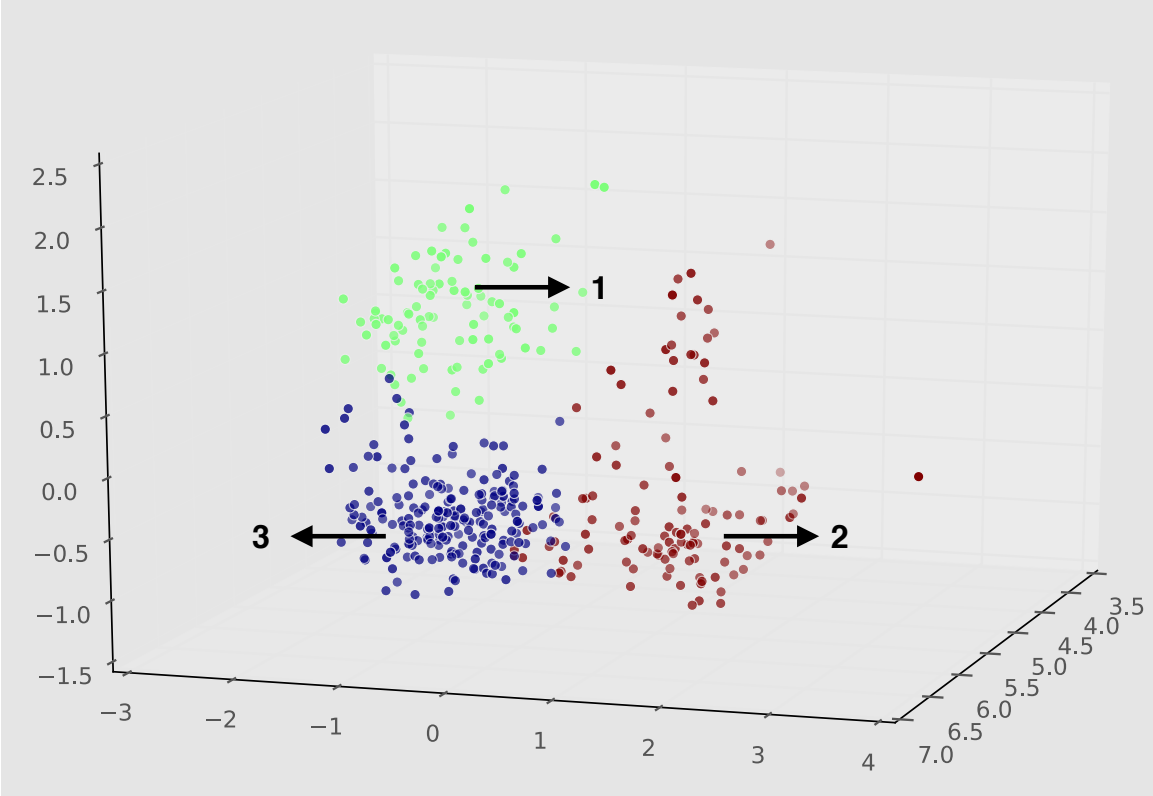


Figure 4-2: Clusters of Campaign Finance tweets.

to the boolean query on a weekly basis. The number of personalities added to the boolean query were initially 20 per week and it has currently hit a saturation and stands at 3-4 per week. We calculate the percentage change in the number of tweets before and after adding new terms, given by:

$$Change\% = \frac{N(D + NT) - N(D)}{N(D)} * 100 \tag{4.1}$$

where N(X) refers to the number of tweets obtained using the terms in X, D refers to the domain knowledge terms and NT refers to new terms (Media or Twitter). Table 4.6 compares the media and Twitter terms and their corresponding percentage changes.

The percentage change in number of tweets retrieved after query expansion (i.e., Twitter terms) is really high compared to the change when knowledge from media is induced. The gap is understandable as the query expansion adapts to the Twitter

Details	*Values*
Media Terms	Susana Martinez, Brynne Craig, Brooke Sammon, John McAfee, Guy Cecil
Twitter Terms	#MakeDonaldDrumpfAgain, #NeverHillary, #BernieOrBust
% change - Twitter Terms	66%
% change - Media Terms	11%

Table 4.6: Media Vs Twitter Terms

vocabulary while knowledge from media is restricted to personality names at this point in time.

Chapter 5

Related Work

Tweet Exemplifier is a natural language processing(NLP) framework applied in the context of elections. In this chapter, we will review the research from the NLP domain and areas of intersection between social media and election analysis. Related work include literature on election analysis using Twitter and media, work about capturing semantic structure from short texts and combining it with ideas from network science for community detection.

5.1 Social Media Analysis of Elections

In the last few decades, many research methods have been used to analyze the complex relationship between politicians and the media and how this shapes the development of the narratives across these groups. The public's interests have traditionally been captured and analyzed via polling, surveys, interviews, etc. and other representative samples of the local conversations. With the advent of the ubiquitous social media, it is much easier to get the digital imprints of the public opinion thereby enabling researchers to analyze voter-generated content. Analyzing the highly decentralized and fragmented public conversation at scale was all but impossible historically.

There has been significant number of recent studies on elections through the lens of Twitter. Conover, Michael et al. [15] used a combination of network clustering algorithms and manually-annotated data to investigate how social media facilitates

communication between communities with different political orientations. Using label propagation, they observed that the network of political retweets exhibits a highly segregated partisan structure, with extremely limited connectivity between left- and right-leaning users. Surprisingly this was not the case for the user-to-user mention network and it concluded that the politically motivated individuals provoke interaction by injecting partisan content into information streams whose primary audience consists of ideologically-opposed users. Livne et al. [41] analyzed the significant differences in the usage patterns of social media and how conservative candidates used this medium more effectively, conveying a coherent message and maintaining a dense graph of connections. They investigate the relation between network structure, content and election results by creating a proof-of-concept model that predicts candidate victory. Pennacchiotti et al. [54] automatically inferred the Twitter user's political orientation using information such as the user behavior, network structure and the linguistic content of the user's Twitter feed. While, Barbera, Pablo [5], used the structure of network as a source of information about their ideological positions and applied a Bayesian Spatial Following model that considered ideology as a latent variable to predict user's ideological stand. Olteanu, Alexandra et al. [51] presented a comparative analysis on online news and social media by covering news events associated with climate change using a method that combines automatic and manual annotations. Using a heuristic for activity peak detection based on attention patterns of Twitter keywords, each detected peak was annotated with the most likely event that triggered it. [64] applied statistical measurement models to the Polity indicators, used widely in studies of international relations to measure democracy as well. Most of the election analysis have focused mainly on predicting political ideologies of users or the outcome of elections.

5.2 Natural Language Processing

Tweet Exemplifier uses traditional approaches and deep learning techniques to solve plethora of subproblems within our framework. However, the overall goal of extracting

representative tweets given a context shares some common thread with short text summarization techniques. So we provide a brief review of research in this field of short text summarization. Besides this, we also provide a literature review of natural language processing (NLP) techniques developed for significant components of our system.

5.2.1 Short Text Summarization

There are many text summarization techniques that are adapted to short texts or developed exclusively for them. Because of the noise and redundancy in social media posts, the performance of off-the-shelf news-trained natural language process systems do not give promising results. SumBasic [65] uses simple word probabilities with an update function to compute the best k posts. Mead summarizer [55] is a well-known flexible and extensible multi-document summarization system and was adapted to tweets. All these summarization techniques extract tweets based on word frequencies and redundancy. The summaries are reduced to majority topic rather than representation from the various conversational realms. A number of recent works [6, 31, 79, 40, 39] rely on lexical clues with similarity scores calculated using different modifications of Tf-Idf. Some of them focus on redundancy and social network specific signals (e.g. user relationship) as a metric to summarize tweets.

Some of the interesting summarization algorithms like LexRank and TextRank [45, 20], use a graph based method to summarize tweets. They compute pairwise similarity between two tweets and make the similarity score the weight of the edge between them. The final score of a tweet is computed using metric based on these edge weights. Though our technique (explained in Section 3.3) is loosely based on TextRank, we provide a novel method to calculate similarity scores. Besides that, the summaries generated by these methods do not represent diverse discussions within the topics as they directly rank the tweets in the graph. Our technique detects communities within the semantic network and hence, achieves the goal of representation than mere summaries.

5.2.2 NLP Approaches for News Stories

Media Knowledge Miner (refer section 3.1.2) applies a number NLP techniques on news stories in the process of extracting influential election related personalities from news stream. There has been extensive amount of work in handling text documents. Standard text classification models have existed to classify news stories [32, 44, 50, 63]. They focus on novel features and techniques to handle unlabeled data. Similarly, a significant amount of work has focused on extracting entities and categorizing them from large document collections [1, 23, 4, 59, 16, 21]. Most of the work related to entity categorization involve a lot of feature engineering and can be restrictive to the domain that it is being applied to. He, et al. [26] proposed a novel entity disambiguation model, based on stacked denoising autoencoders. However, we used a combination of distributed word vector representation and stacked denosing autoencoder (explained in Section 3.1.2.4) to categorize entities extracted from the news stories.

5.2.3 Traditional NLP Approaches for Short Texts

Some of the crucial NLP sub-problems in our system are contained in Tweet Analysis pipeline. They require tweet classification models (Sections 3.2.1 & 3.2.2). There has been a wide spectrum of approaches involved in inferring semantics in texts. Several schemes such as Latent Semantic Analysis [17], Probabilistic Latent Semantic Analysis (pLSA) [28] and Latent Dirichlet Allocation (LDA) [7] have been used to good success in inferring the high level meaning of documents through a set of representative words. However, these techniques do not readily work with short texts like tweets. Topic modeling and classification on tweets is much more challenging and is still an open problem. Several schemes to train a standard topic model were proposed and their quality compared from both qualitative and quantitative perspectives were not satisfying. Many efforts [73, 29] have been made to address the application of topic models to short texts. They apply LDA for inducing topics using different aggregation methods (e.g., user level aggregation, etc.) aggregated message. Moreover, through the experiments, it is clear that these models do not yield better modeling

for tweets and indeed it is worse than training a standard LDA model on user aggregated tweets. Ramage et al. [56] applied Labeled-LDA, which extends LDA by incorporating supervision with implied tweet-level labels, enabling explicit models of text content associated with hashtags, replies or emoticons. Unfortunately the model relies heavily on hashtags, which may not map well to all topics.

5.2.4 Deep Learning Approaches for Short Text

With recent advances in natural language processing, a number of deep learning models have been effective and have achieved excellent results in semantic parsing [76], search query retrieval [60], sentence modeling [34], and other traditional NLP tasks [14]. Deep learning models require large amounts of data and it is possible to learn the semantic composition of tweets using models ranging from recurrent neural networks (sequential feature learning) [27] to convolutional neural networks (hierarchical feature learning) [34]. Tweet Exemplifier utilizes deep learning models for classifying tweets, explained in sections 3.2.1 and 3.2.2, adapted to robustly classify tweets in the context of elections.

Besides topic modeling and classification, developing learning algorithms for distributed compositional semantics of tweets can be a challenging problem at the intersection of language understanding and machine learning. Recently, a number of approaches have been developed for learning composition operators that map word vectors to sentence vectors including recursive networks [61], recurrent networks [27], convolutional networks [34] and recursive-convolutional methods [10, 78] among others. All of these methods produce sentence representations that are passed to a supervised task and depend on a class label in order to backpropagate through the composition weights. Consequently, these methods learn high quality sentence representations but are tuned only for their respective task. The paragraph vector [38] is an alternative to the above models in that it can learn unsupervised sentence representations by introducing a distributed sentence indicator as part of a neural language model.

There are number of encoder-decoder models which can be used for learning dis-

tributed representations. That is, an encoder take words as input words and maps it to a sentence vector and a decoder, in turn, is used to generate the surrounding sentences. Encoder-decoder models have gained a lot of attention for neural machine translation. Encoder maps the sentence from on the source language to a vector representation, while the decoder conditions on this encoded vector for translating it to the target language. A number of different choices of encoder-decoder pairs have been explored, including CNN-RNN [33], RNN-RNN [11] and LSTM-LSTM [62]. The source sentence representation can also dynamically change through the use of an attention mechanism [3] to take into account only the relevant words for translation at any given time. However, all these models work at word level and can be restrictive for tweets as they are noisy and idiosyncratic. Tweet2Vec (explained in Section 3.3.1) clearly describes a model that can learn representation from characters in detail. We also show the power of such tweet representations generated using this approach to solve other classification tasks.

Chapter 6

Conclusion

The thesis explained a framework that can identify representative tweets from discussions surrounding various election issues on Twitter. This is accomplished by robustly tracking the shifting conversations around elections on Twitter with the assistance of knowledge mined from traditional media. Below we summarize the contributions of this thesis and explore possible future directions for extending this work.

6.1 Future Directions

There are many possible ways to take this work forward. Some of the rewarding directions are explained below.

Near Real-time Representative Tweet Generation

An immediate extension to the current framework is to have a near real-time representative tweet generation system that can constantly listen to the Twitter stream and be reflective of the changing dynamics of conversations on Twitter around various election topics. This can be really useful tool for not just journalists but also for people to understand the diverse views that sometimes go unnoticed and analyze how the demographic bias on Twitter plays a role in this.

Demographic segmentation of Twitter users

Tweet2Vec is a general purpose tool that can generate a vector representation

for tweets. The power of those embeddings can be used to segment the users demographically. This in turn can open new avenues of analysis in identifying the preferences of those demographic groups. Since we have implemented a sophisticated technique to generate tweet embeddings, it would be interesting to explore ways to combine these vectors with image representations as features for such segmentation of users.

Cross-Domain Interoperability

Since significant portions of the framework are generic, there is a natural inclination to extend the features of the framework to other domains like food, health, etc.

Finally, a natural language processing framework that semantically analyzes and maps textual content can be really useful for understanding large collections of tweets across several domains.

6.2 Contributions

In this thesis, we described the Tweet Exemplifier - a comprehensive natural language processing framework for tracking and analyzing election related conversation on Twitter. The framework has capabilities to highlight tweets that best symbolizes the conversation under various topics. The framework utilizes recent advances in natural language processing and deep neural networks in order to dynamically learn new election terms for data ingest, categorize them into crucial election topics and generate diverse conversational spheres within each of these topics.

We achieved it by implementing a robust tweet aggregation mechanism combined with a character-level spam filtering election classifier. We used character-level and world-level convolutional models to deal with idiosyncrasies in tweets. These models have a precision greater than 90%. We introduced Tweet2Vec, a novel method to produce vector-based representation of tweets. Tweet2Vec was able to outperform the best performing system in the tasks like semantic relatedness and sentiment clas-

sification. The ingested election tweets, distributed among different topic buckets, were encoded into their vector representations and clustered into communities of semantically similar tweets using the popular Louvain method. The Louvain method finds high modularity partitions in such semantic network of tweets and the outcome is a group of diverse issue realms bound by the topic structure imposed on tweets. We compared the results of our system with that of the democratic debate. The system was able to find the biggest spike moments and come up with prominent topics and topic-centric discussions that were really promising.

6.3 Concluding Remarks

Our framework has a lot of potential to be applied in cross-domain research and analysis. Also, there is scope for improving our models like introducing attention mechanism in Tweet2Vec that enhances the quality of our embeddings. The results of our framework are promising and can already serve as a precursor to automatically generate gist of collection of tweets acting as plausible inputs to data-grounded journalism and debates. Though we had full access to Twitter's historical data, most of the work described in this thesis can be replicated using the Twitter public API.

Appendix A

Results: Representative Tweets

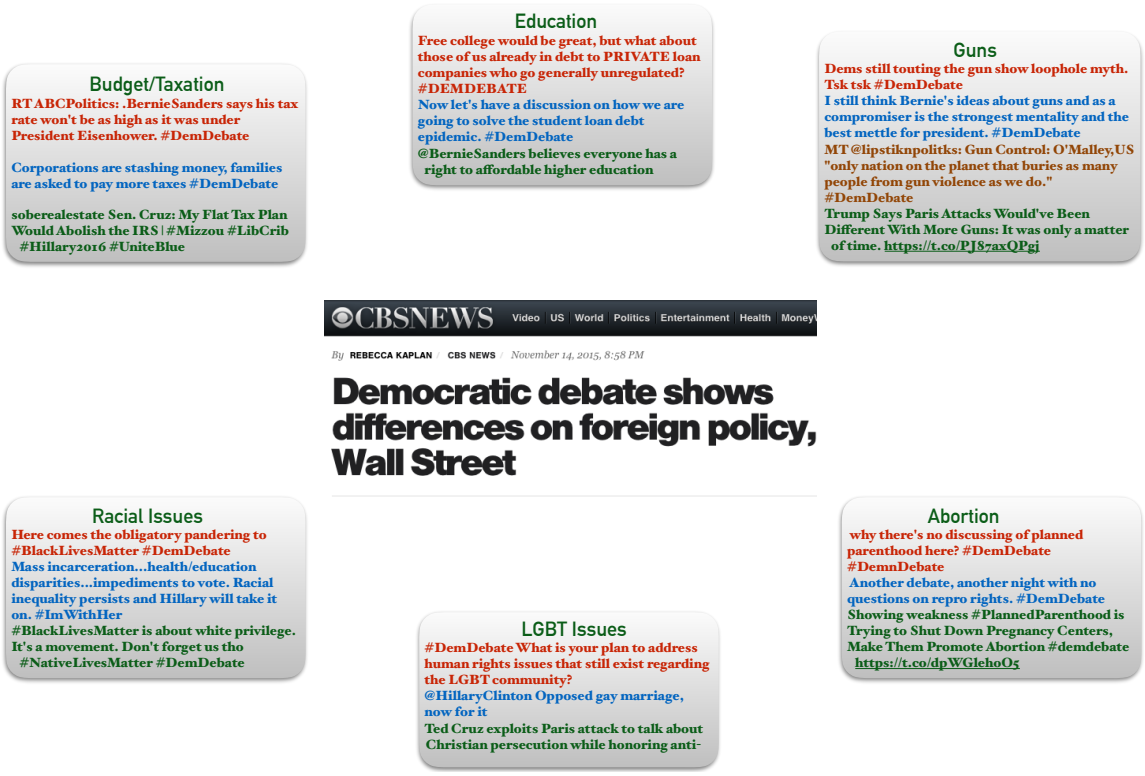


Figure A-1: Representative Tweets for topics during the Democratic debate on November 14, 2015: Budget/Taxation, Education, Guns, Abortion, LGBT Issues and Racial Issues.

Clusters	*Top*-4 *Tweets*
Cluster 1	Interesting to see what Mr. Berny will do about Mr. Isis... #DemDebate
	If a primary candidate "has a plan to defeat ISIS" right now, wouldn't believe them. #sadbuttrue #DemDebate https://t.co/qE4NEhfZ6x
	#DemDebate what should the US response be towards the aggression by ISIS in Europe?
	@MartinOMalley on ISIS: "No nation better than ours to face it"" #DemDebate
Cluster 2	#ISIS #ParisAttacks #IStandWithParis #Obama and Democrats want voters... that's why they are pushing to let them in. https://t.co/uVbvV9Zqz7
	#DemDebate Looking at the #ParisAttacks what will you do as President to make sure countries in the middle east, stop funding ISIS?
	The #DemDebate starting now. Interested to see how they'll address what has happened around the world in Paris, Baghdad, Syria recently.
	GOP presidential candidates point to #ParisAttacks to criticize President Barack Obama's foreign policy. https://t.co/6lu4xliuK4
Cluster 3	#DemDebate How can you insure America's security at home when you couldn't help them in Benghazi? #DemDebate
	@HillaryClinton Then what happened in Benghazi?? Where was that smart leadership.when we needed it??? 4 dead American.
	You are talking Libya now, maybe a Benghazi question? #DemDebate
	Speak about benghazi when you speak about libya ?#DemDebate

Table A.1: Representative Tweets for Foreign Policy/National Security during the Democratic debate on November 14, 2015

Clusters	*Top*-4 *Tweets*
Cluster 1	Wall Street's turn to bail out the middle class. Thank you. #DemDebate
	Its time for Wall Street to bail out the middle class. #DemDebate #DemDebate https://t.co/qE4NEhfZ6x
	fliganan:RT BernieSanders: We bailed out Wall Street, it's their turn to bail out the middle class and help our kids go to college tuition
	Poor O'Malley trying to get in this Wall Street discussion #DemDebate
Cluster 2	"We'll resume our conversation about the evils of capitalism, right after you watch these paid advertisements!" #DemDebate
	Can the candidates speak on "inclusive capitalism" vs "shareholder capitalism" #DemDebate
	Has capitalism worked? #DemDebate
	All three Idiots want to get rid of capitalism.#DemDebate
Cluster 3	Senator Sanders, does your socialist economic policy affect our right to economic freedoms? #DemDebate
	Sanders: I'm not as much of a socialist as Ike: Bernie Sanders says his tax policies aren't nearly as left-wing asâĂę https://t.co/RzeVi71eWV
	Why the hell is @BernieSanders on this stage anyway. Democratic Socialist? This is the Democratic Party. No Socialism.
	#DemDebate Bernie, can you explain to voters the difference between Socialist & Democratic Socialist? https://newrepublic.com/article/121680/bernie-sanders-democratic-socialist-not-just-socialist...

Table A.2: Representative Tweets for Economy during the Democratic debate on November 14, 2015

Clusters	*Top-4 Tweets*
Cluster 1	Free tuition and healthcare sounds good. How @BernieSanders would pay for it? handle congress? https://t.co/5S4JeZJEy2 #DemDebate
	Free tuition and healthcare sounds good. How BernieSanders would pay for it https://t.co/zsfqItxxhz «#DemDebate https://t.co/PGAOseuV5p
	It's been hard enough to keep Obamacare in place. @BernieSanders could never get a single-payer system through Congress. #PCChat #DemDebate
	Very curious how Sanders' healthcare plan will hold under constitutional challenges #DemDebate
Cluster 2	Health care is a right says @SenSanders. Hell yeah! #DemDebate #DemDebate
	I don't remember healthcare being in the Bill of Rights.
	Is healthcare a right for all people in the US or not? #BernieSanders says it is a HUMAN RIGHT #Hillary2016 does not.
	#demdebate Where in any of the founding documents is health care a right? Sanders was there when it was written, he should know it's not
Cluster 3	Clinton's Health Plans Do Nothing To Tackle Rising Costs https://t.co/EBrycfq48h #DemDebate
	Democrats raised the price of health care and reduced the ability to access. #ACA has failed, and the public rejects it. @cspanwj
	It was government subsidies that caused prices of college and health care prices to skyrocket. Yet Dems want more of the same.
	.@johndickerson please ask @HillaryClinton what if anything she will do to improve the #affordablecareact and bring costs down? @CBSNews #DemDebate

Table A.3: Representative Tweets for Health Care

Bibliography

[1] Eugene Agichtein. Scaling information extraction to large document collections. *IEEE Data Eng. Bull.*, 28(4):3–10, 2005.

[2] Monica Anderson. *More Americans are using social media to connect with politicians*, 2015.

[3] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[4] Michele Banko, Michael J Cafarella, Stephen Soderland, Matthew Broadhead, and Oren Etzioni. Open information extraction for the web. In *IJCAI*, volume 7, pages 2670–2676, 2007.

[5] Pablo Barberá. Birds of the same feather tweet together: Bayesian ideal point estimation using twitter data. *Political Analysis*, 23(1):76–91, 2015.

[6] Hila Becker, Mor Naaman, and Luis Gravano. Selecting quality twitter content for events. *ICWSM*, 11, 2011.

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

[8] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment*, 2008(10):P10008, 2008.

[9] Gregory Brazeal. How much does a belief cost?: Revisiting the marketplace of ideas. *Southern California Interdisciplinary Law Journal*, 21, 2011.

[10] Kyunghyun Cho, Bart van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*, 2014.

[11] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[12] Scott Clement. *TwitterâĂŹs political debate focuses on much different issues than Americans at large do*, 2016.

[13] Cathy J Cohen and Joseph Kahne. participatory politics. *New media and youth political action*, 2012.

[14] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. Natural language processing (almost) from scratch. *The Journal of Machine Learning Research*, 12:2493–2537, 2011.

[15] Michael Conover, Jacob Ratkiewicz, Matthew R Francisco, Bruno Gonçalves, Filippo Menczer, and Alessandro Flammini. Political polarization on twitter. *ICWSM*, 133:89–96, 2011.

[16] Silviu Cucerzan. Large-scale named entity disambiguation based on wikipedia data. In *EMNLP-CoNLL*, volume 7, pages 708–716, 2007.

[17] Scott Deerwester, Susan T Dumais, George W Furnas, Thomas K Landauer, and Richard Harshman. Indexing by latent semantic analysis. *Journal of the American society for information science*, 41(6):391, 1990.

[18] Lauren deLisa Coleman. *How Millennials Will Impact the 2016 Election Without Voting*, 2016.

[19] Maeve Duggan. *The Demographics of Social Media Users*, 2016.

[20] Günes Erkan and Dragomir R Radev. Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of Artificial Intelligence Research*, pages 457–479, 2004.

[21] Ronen Feldman, Benjamin Rosenfeld, and Moshe Fresko. TegâĂŤa hybrid approach to information extraction. *Knowledge and Information Systems*, 9(1):1–18, 2006.

[22] Christiane Fellbaum. *WordNet*. Wiley Online Library, 1998.

[23] Venkatesh Ganti, Arnd C König, and Rares Vernica. Entity categorization over large document collections. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 274–282. ACM, 2008.

[24] Kevin Gimpel, Nathan Schneider, Brendan O'Connor, Dipanjan Das, Daniel Mills, Jacob Eisenstein, Michael Heilman, Dani Yogatama, Jeffrey Flanigan, and Noah A Smith. Part-of-speech tagging for twitter: Annotation, features, and experiments. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2*, pages 42–47. Association for Computational Linguistics, 2011.

[25] R. Kay Green. *The Game Changer: Social Media and the 2016 Presidential Election*, 2016.

[26] Zhengyan He, Shujie Liu, Mu Li, Ming Zhou, Longkai Zhang, and Houfeng Wang. Learning entity representation for entity disambiguation. In *ACL (2)*, pages 30–34, 2013.

[27] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

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

[29] Liangjie Hong and Brian D Davison. Empirical study of topic modeling in twitter. In *Proceedings of the first workshop on social media analytics*, pages 80–88. ACM, 2010.

[30] Shiping Huang, Matthew O Ward, and Elke A Rundensteiner. Exploration of dimensionality reduction for text visualization. In *Coordinated and Multiple Views in Exploratory Visualization, 2005.(CMV 2005). Proceedings. Third International Conference on*, pages 63–74. IEEE, 2005.

[31] David Inouye and Jugal K Kalita. Comparing twitter summarization algorithms for multiple post summaries. In *Privacy, Security, Risk and Trust (PASSAT) and 2011 IEEE Third Inernational Conference on Social Computing (SocialCom), 2011 IEEE Third International Conference on*, pages 298–306. IEEE, 2011.

[32] Thorsten Joachims. Transductive inference for text classification using support vector machines. In *ICML*, volume 99, pages 200–209, 1999.

[33] Nal Kalchbrenner and Phil Blunsom. Recurrent continuous translation models. In *EMNLP*, volume 3, page 413, 2013.

[34] Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. A convolutional neural network for modelling sentences. *arXiv preprint arXiv:1404.2188*, 2014.

[35] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[36] Ryan Kiros, Yukun Zhu, Ruslan R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Skip-thought vectors. In *Advances in Neural Information Processing Systems*, pages 3276–3284, 2015.

[37] Marissa Lang. *2016 Presidential Election Circus: Is Social Media the Cause?*, 2016.

[38] Quoc V Le and Tomas Mikolov. Distributed representations of sentences and documents. *arXiv preprint arXiv:1405.4053*, 2014.

[39] Wenjie Li, Mingli Wu, Qin Lu, Wei Xu, and Chunfa Yuan. Extractive summarization using inter-and intra-event relevance. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 369–376. Association for Computational Linguistics, 2006.

[40] Fei Liu, Yang Liu, and Fuliang Weng. Why is sxsw trending?: exploring multiple text sources for twitter topic summarization. In *Proceedings of the Workshop on Languages in Social Media*, pages 66–75. Association for Computational Linguistics, 2011.

[41] Avishay Livne, Matthew P Simmons, Eytan Adar, and Lada A Adamic. The party is over here: Structure and content in the 2010 election. *ICWSM*, 11:17–21, 2011.

[42] Oded Maimon and Lior Rokach. *Data mining and knowledge discovery handbook*, volume 2. Springer, 2005.

[43] Christopher D Manning, Mihai Surdeanu, John Bauer, Jenny Rose Finkel, Steven Bethard, and David McClosky. The stanford corenlp natural language processing toolkit. In *ACL (System Demonstrations)*, pages 55–60, 2014.

[44] Andrew McCallum, Kamal Nigam, et al. A comparison of event models for naive bayes text classification. In *AAAI-98 workshop on learning for text categorization*, volume 752, pages 41–48. Citeseer, 1998.

[45] Rada Mihalcea and Paul Tarau. Textrank: Bringing order into texts. Association for Computational Linguistics, 2004.

[46] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[47] Frederic Morin and Yoshua Bengio. Hierarchical probabilistic neural network language model. In *Proceedings of the international workshop on artificial intelligence and statistics*, pages 246–252. Citeseer, 2005.

[48] Mark EJ Newman. Modularity and community structure in networks. *Proceedings of the national academy of sciences*, 103(23):8577–8582, 2006.

[49] Kamal Nigam, John Lafferty, and Andrew McCallum. Using maximum entropy for text classification. In *IJCAI-99 workshop on machine learning for information filtering*, volume 1, pages 61–67, 1999.

[50] Kamal Nigam, Andrew Kachites McCallum, Sebastian Thrun, and Tom Mitchell. Text classification from labeled and unlabeled documents using em. *Machine learning*, 39(2-3):103–134, 2000.

[51] Alexandra Olteanu, Carlos Castillo, Nicholas Diakopoulos, and Karl Aberer. Comparing events coverage in online news and social media: The case of climate change. In *Proceedings of the Ninth International AAAI Conference on Web and Social Media*, number EPFL-CONF-211214, 2015.

[52] Alexander Pak and Patrick Paroubek. Twitter as a corpus for sentiment analysis and opinion mining. In *LREc*, volume 10, pages 1320–1326, 2010.

[53] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *The Journal of Machine Learning Research*, 12:2825–2830, 2011.

[54] Marco Pennacchiotti and Ana-Maria Popescu. A machine learning approach to twitter user classification. *ICWSM*, 11(1):281–288, 2011.

[55] Dragomir Radev, Timothy Allison, Sasha Blair-Goldensohn, John Blitzer, Arda Celebi, Stanko Dimitrov, Elliott Drabek, Ali Hakim, Wai Lam, Danyu Liu, et al. Mead-a platform for multidocument multilingual text summarization. 2004.

[56] Daniel Ramage, Susan T Dumais, and Daniel J Liebling. Characterizing microblogs with topic models. *ICWSM*, 10:1–1, 2010.

[57] Alan Ritter, Oren Etzioni, Sam Clark, et al. Open domain event extraction from twitter. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1104–1112. ACM, 2012.

[58] Sara Rosenthal, Preslav Nakov, Svetlana Kiritchenko, Saif M Mohammad, Alan Ritter, and Veselin Stoyanov. Semeval-2015 task 10: Sentiment analysis in twitter. *Proceedings of SemEval-2015*, 2015.

[59] Benjamin Rozenfeld and Ronen Feldman. Self-supervised relation extraction from the web. *Knowledge and Information Systems*, 17(1):17–33, 2008.

[60] Yelong Shen, Xiaodong He, Jianfeng Gao, Li Deng, and Grégoire Mesnil. Learning semantic representations using convolutional neural networks for web search. In *Proceedings of the companion publication of the 23rd international conference on World wide web companion*, pages 373–374. International World Wide Web Conferences Steering Committee, 2014.

[61] Richard Socher, Alex Perelygin, Jean Y Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the conference on empirical methods in natural language processing (EMNLP)*, volume 1631, page 1642. Citeseer, 2013.

[62] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, 2014.

[63] Simon Tong and Daphne Koller. Support vector machine active learning with applications to text classification. *The Journal of Machine Learning Research*, 2:45–66, 2002.

[64] Shawn Treier and Simon Jackman. Democracy as a latent variable. *American Journal of Political Science*, 52(1):201–217, 2008.

[65] Lucy Vanderwende, Hisami Suzuki, Chris Brockett, and Ani Nenkova. Beyond sumbasic: Task-focused summarization with sentence simplification and lexical expansion. *Information Processing & Management*, 43(6):1606–1618, 2007.

[66] Prashanth Vijayaraghavan, Ivan Sysoev, Soroush Vosoughi, and Deb Roy. Deepstance at semeval-2016 task 6: Detecting stance in tweets using character and word-level cnns. *SemEval-2016*, 2016.

[67] Prashanth Vijayaraghavan, Soroush Vosoughi, and Deb Roy. Automatic detection and categorization of election-related tweets. In *10th International AAAI Conference on Web and Social Media*, 2016.

[68] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pages 1096–1103. ACM, 2008.

[69] Soroush Vosoughi. *Automatic Detection and Verification of Rumors on Twitter*. PhD thesis, Massachusetts Institute of Technology, 2015.

[70] Soroush Vosoughi and Deb Roy. A semi-automatic method for efficient detection of stories on social media. 2016.

[71] Soroush Vosoughi and Deb Roy. Tweet acts: A speech act classifier for twitter. In *proceedings of the 10th ICWSM*, 2016.

[72] Soroush Vosoughi, Prashanth Vijayaraghavan, and Deb Roy. Tweet2vec: Learning tweet embeddings using character-level cnn-lstm encoder-decoder. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Informaion Retrieval*. ACM, 2016.

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

[74] John Wihbey. *How does social media use influence political participation and civic engagement? A meta-analysis*, 2015.

[75] Wei Xu, Chris Callison-Burch, and William B Dolan. Semeval-2015 task 1: Paraphrase and semantic similarity in twitter (pit). *Proceedings of SemEval*, 2015.

[76] Wen-tau Yih, Xiaodong He, and Christopher Meek. Semantic parsing for single-relation question answering. In *ACL (2)*, pages 643–648. Citeseer, 2014.

[77] Xiang Zhang and Yann LeCun. Text understanding from scratch. *arXiv preprint arXiv:1502.01710*, 2015.

[78] Han Zhao, Zhengdong Lu, and Pascal Poupart. Self-adaptive hierarchical sentence model. *arXiv preprint arXiv:1504.05070*, 2015.

[79] Arkaitz Zubiaga, Damiano Spina, Enrique Amigó, and Julio Gonzalo. Towards real-time summarization of scheduled events from twitter streams. In *Proceedings of the 23rd ACM conference on Hypertext and social media*, pages 319–320. ACM, 2012.