Summer 2019

Determining Political Inclination in Tweets Using Transfer Learning

Mehtab Iqbal

DETERMINING POLITICAL INCLINATION IN TWEETS USING TRANSFER LEARNING

by

MEHTAB IQBAL

(Under the Direction of Jeffrey Kaleta)

ABSTRACT

The last few years have seen tremendous development in neural language modeling for transfer learning and downstream applications. In this research, I used Howard and Ruder's Universal Language Model Fine Tuning (ULMFiT) pipeline to develop a classifier that can determine whether a tweet is politically left leaning or right leaning by likening the content to tweets posted by @TheDemocrats or @GOP accounts on Twitter. We achieved 87.7% accuracy in predicting political ideological inclination.

INDEX WORDS: Transfer learning, Language models, Tweet classification, Stance classification, Computer mediated communication, Deep learning

DETERMINING POLITICAL INCLINATION IN TWEETS USING TRANSFER LEARNING

by

MEHTAB IQBAL

B.S, BRAC University, Bangladesh, 2012
M.S, Georgia Southern University, 2019

A Thesis Submitted to the Graduate Faculty of Georgia Southern University in Partial Fulfillment of the Requirements for the Degree

MASTER OF SCIENCE

STATESBORO, GEORGIA

DETERMINING POLITICAL INCLINATION IN TWEETS USING TRANSFER LEARNING

by

MEHTAB IQBAL

Major Professor:	Jeffrey Kaleta
Committee:	Cheryl Aasheim
	Christopher Kadlec

Electronic Version Approved:
July 2019

DEDICATION

This document is dedicated to all those people thinking on going back to school for a graduate degree.

If I can do it, so can anyone.

ACKNOWLEDGMENTS

I could not have done this without the myriad of people supporting me. This includes my professors from undergraduate who believed in me and encouraged me to get back to school, my graduate adviser(s), mom, dad, siblings, the single person who has been my true friend throughout this journey.

TABLE OF CONTENTS

LIST OF TABLES

Page

LIST OF FIGURES

Page

CHAPTER 1

INTRODUCTION

Technological development and adoption of the internet in our day to day lives made written language a central medium through which we communicate. This information disseminated in written forms has taken intricate and novel forms (Choi and Lee, 2015). Social media such as Twitter and Facebook are widely used to communicate at an interpersonal level and as a medium to broadcast information (Zhao et al., 2011). This climate of social media usage has changed the paradigm of information diffusion by removing the bottleneck of infrastructure (Stieglitz and Dang-Xuan, 2013).

This leads to the widespread adoption of social media in the dispersion of information and it has been shown, through the use of computer-mediated communication, affective information can easily be transferred (Stieglitz and Dang-Xuan, 2013). Researchers also found that the affective dimensions of communication can trigger cognitive involvement (Stieglitz and Dang-Xuan, 2013). Also, given the polarizing nature of politics (Stieglitz and Dang-Xuan, 2013) and the participatory nature of social media (Hermida, 2010), many political sources and news media outlets have taken to social media.

News media outlets provide provisions for users to interact with news articles through social media presence or directly on their website (Diakopoulos and Naaman, 2011; Hille and Bakker, 2014; Ruiz et al., 2011). These news outlets, keeping up with the features of Facebook and Twitter, provide mechanisms for users to post comments on their news articles. These comments are considered to be important online participation by most news media outlets due to the wide reach of the platform (Oeldorf-Hirsch and Sundar, 2015). On Facebook, these comments are structured at two levels. The original post can garner comments and, each comment can, in turn, be replied to by a user. This nested structure of comments provides a conducive environment for conversation, and debates (Iqbal and Khan, 2018). Several studies have found that up to 50% of comments have, at least, a single response (reply) made by another user (Ziegele et al., 2014). As cited by Iqbal and Khan (2018), these interactions are democratically valuable and contain rich interpersonal discourse on topics of public interest.

Discourse on Facebook pages of news media outlets tends to degenerate when two polarized communities interact with one another (Del Vicario et al., 2017). Users on Facebook typically select the information that reinforces their confirmation bias and ignores dissenting information (Quattrociocchi et al., 2016). This

behavior catalyzes the formation of polarized groups (echo chamber) which further strengthens the community (Zollo et al., 2015). I have seen in a previous study (Iqbal and Khan, 2018), interactions vary in different communities based on their partisan bias, and the partisan polarity of the community (news media followers).

Last few years have seen a similar structure of features introduced in other social websites, including news outlets sites, and Twitter, enabling conversation. This threaded conversation structure ultimately culminates to a similar discourse structure as exhibited in Facebook page posts. It is seen that behavioral patterns can be extracted from within the linguistic components (Chung and Pennebaker, 2007) of each comment, and reply. Linguistic components, both syntactic and semantic, can elicit information on a person's mental state, emotions (Chung and Pennebaker, 2007), and social positioning (Milroy and Milroy, 1992). How people infer information, and then, in turn, respond to it takes place at a meta-cognitive level (Menyuk, 1985), which can be extracted from the meta-linguistics characteristics of the language being used (Jean Emile Gombert, 1992).

This research aims to investigate the characteristics of conversations in the social media space and more specifically how the language used may show a linguistic predisposition as it relates to a person's political bias. For this study, I picked two notably polarized content sources on Twitter. First, I chose the official Twitter account of the Democratic Party — @TheDemocrats. Then, the official Twitter account of the Republican National Committee — @GOP. I hypothesize that the topic content and the language usage are distinctly different in the two accounts since they cater to unique group demographics of different political ideologies. It is with this hypothesis, I derive a second hypothesis that it should be possible to develop a classifier pipeline that can determine the political ideology of a tweet's author (e.g. politically left or right) with sufficient accuracy and confidence.

To test my hypotheses, I first performed term frequency analysis and visualized the resulting frequencies using the Scattertext (Kessler, 2017) library. I found that my first hypothesis holds, as a result, to further strengthen my point I extracted Latent Dirichlet Allocation (LDA) topics from the @TheDemocrats tweets and @GOP tweets separately. I compared the LDA topics to find that the topics do not overlap between the two tweet authors. Additionally, whenever the topics do intersect, the stance polarity is usually opposite relative to one another. Finally, I used a pre-trained weight dropped Long Short Term Memory (LSTM) language model to train a classifier to determine the tweet author and my Neural Network converged with a high enough accuracy towards the task, confirming my second hypothesis.

CHAPTER 2

BACKGROUND AND LITERATURE REVIEW

Studying the emotional content, or ideological biases in texts is not new. Through emotion extraction and analysis, it is possible to determine how people agree or disagree on online forum debates. Over the years the domain of this research has become pointedly specific towards understanding an author's stance on a given topic provided the topic being discussed is already known. Until recently, most natural language processing (NLP) related work has been domain-specific requiring both domain knowledge and model development within the respective domain.

Furthermore, recent years have seen research in language processing to be heavily dependent on the distributed representation of words as vectors popularly termed as word embedding models such as the skip-gram model (Mikolov et al., 2013a), or Stanford's GloVe (Pennington et al., 2014). However, robust language models have been introduced recently made it possible to perform target specific tasks without having to train an entire model from scratch (Howard and Ruder, 2018). This new concept of transfer learning from a general pre-trained language model started a whole slew of development in state-of-the-art classification models.

In the following literature review, I briefly discuss the progression of how emotional content leads to the foundation of argumentation and stance modeling of specific topics. Then I provide an overview of language models and how they apply to this research.

Social Aspects of Written Language

Empirical studies have shown how people use language can reveal information about their thoughts and emotions (Chung and Pennebaker, 2007). Linguistic Inquiry and Word Count (LIWC) has been successfully used to identify relationships between individuals in social interactions, including relative status (Sexton and Helmreich, 2000), deception (Newman et al., 2003), and the quality of close relationships (Slatcher and Pennebaker, 2006).

Social language processing is primarily concerned with inferring individual traits, such as sex, age, relative status, or mental health based on the use of language (Pennebaker et al., 2003). Sociolinguistics combine social network analysis and linguistic style to determine social position using variation in linguistic patterns

(Milroy and Milroy, 1992). For example, the use of pronouns provides information about how participants in a conversation address each other. The first person singular pronoun usage tends to increase while someone interacts with a person of higher status (Kacewicz et al., 2014). The degree of emotionality can be determined from the use of social and affective words. Punctuation, referred to as discourse markers, can show how formal or informal the language being used is (Scholand et al., 2010).

Argument Extraction

In a social media context, arguments are unstructured and often, repetitive (Swanson et al., 2015). Summarization of these arguments and the extraction of relevant argument statements is an important factor towards further analyses of online argument and debate (Swanson et al., 2015). To this extent, Swanson, et al., (2015) posed a hypothesis termed as the "Implicit Markup Hypothesis." The implicit markup hypothesis is a multifarious hypothesis that takes into consideration, several other hypotheses previously presented through various linguistic and metalinguistic understanding. The hypothesis states that a good argument can be inferred as an argument from the surface realization of its linguistic components.

According to the "Discourse Relation Hypothesis," arguments containing good argumentative segments tend to have an explicit connective between argument 1 and argument 2. Connectives are categorized into four components, specification ("first"), contrast ("but"), contingency ("if"), and concession ("so") (Prasad et al., 2007). Additionally, the "Syntactic Properties Hypothesis" states that the syntactic properties of a sentence may provide a good argument indicator or even a sentential complement of mental state (Marcu, 1999). The "Dialogue Structure Hypothesis" provides a strong relationship indicator between arguments by utilizing the dialogic structure of a conversation (Swanson et al., 2015). In other words, collecting information on whether an argument is a direct reply to a statement or not is a straightforward feature extraction from argument statements. Finally, the "Semantic Density Hypothesis" states that the richness of a sentence is often a requirement when dealing with the surface realization of an argument statement (Louis and Nenkova, 2011).

Stance Classification

While the agreement in a discourse model relies on a dialogic structure, stance classification tries to determine whether a text is in favor of, against, or neutral towards a proposed target (Mohammad et al.,

2016). Stance classification has similarities to sentiment analysis but poses a lot more complexity. Where sentiment analysis is concerned with whether a given text positive or negative in its tone, stance focuses on the favorability concerning a target entity (Mohammad et al., 2017). A good way to think about stance classification would be to consider a target A such that a sentence in favor of it could be either positive towards A or negative towards an opposing target B, thus favoring A by negation. An example from the SemEval-2016 (Mohammad et al., 2016) dataset is shown.

Stance classification has seen a lot of interesting and high accuracy classifiers as submissions to the SemEval-2016 contest. The state-of-the-art model utilized transfer learning techniques using a recurrent neural network (RNN) architecture (Zarrella and Marsh, 2016). The model consisted of a pre-trained embedding layer, followed by a deep pre-trained RNN layer made up of 128 Long Short Term Memory (LSTM) units. The embedding layer was trained on a stream of 2.1 million tweets using the skip-gram (Mikolov et al., 2013a) model, followed by extraction of phrased up to four words long using the word2phrase mechanism (Mikolov et al., 2013b).

The recurrent layer was trained on in-domain tweets determined by the hashtag content (Zarrella and Marsh, 2016). These in-domain tweets consisted of the five target topics introduced in the SemEval-2016 dataset - "Atheism," "Climate change is a concern," "Feminist Movement," "Hilary Clinton," and "Legalization of abortion" (Mohammad et al., 2016). Zarella & Marsh (2016) used hashtags to determine target specific hashtags for example in case of "Climate change is a concern", they picked tweets containing "#climatechange" to be in favor of the topic and "#climatescam" representing tweets against the topic of climate change. Finally, the layer was pre-trained using about 300,000 tweets containing their curated list of 197 hashtags that convey favoring information towards each of the five topics (Zarrella and Marsh, 2016).

Language Models

Language modeling is a fundamental component of natural language processing. The task usually involves the determination of a word in a sentence given a fragment of that sentence (Merity et al., 2018). Tokens are the building blocks of language models and can be abstracted in varying levels of granularity. For example, a token can be comprised of words in a sentence, sub-words in a word, or even characters (Merity et al., 2018). While each level of granularity can provide their specific benefits, it is seen that word-level language models are more reliable for classifiers relying on generalized models (Merity et al., 2018). However, word-based language models aren't without their limitations either, on the one hand, if the vocabulary

is too small, the model needs to account for out of vocabulary tokens. On the other hand, as the vocabulary size increases, so does the dimension of the vector space (Bengio et al., 2003).

This representation of words in high dimensional space can easily cause the overall volume of the space to increase, which in turn renders the available data to become sparse. This phenomenon is known as the curse of dimensionality. "The curse of dimensionality" is one obvious problem when dealing with words in a sentence to make language models (Bengio et al., 2003). Consider a typical tweet of 250 characters, that is an average of about 40 words. If I were to model the joint distribution of 40 consecutive words in a collection of tweets with a vocabulary size of 50,000, the potentiality of free parameters could be as high as $50,000^{40}$–1. While continuous models can generalize more easily with smoothing techniques such as the Gaussian mixture models (Li and Sporleder, 2010). For language models which is a discrete space, small changes can alter function output drastically (Bengio et al., 2003).

In 2013, Mikolov et al. in their paper Distributed Representation of Words in Vector Spaces, introduced a syntactical distribution model based on its semantic weight in a corpus. This word embedding (Li and Sporleder, 2010) model introduced two key algorithms to model language in continuous vector space. One is the continuous bag of words (CBOW) and the other is the skip-gram model. Continuous bag of words tries to predict the next word in a sequence given the preceding window of words. This allows each word to be placed within an n-dimensional vector space where n is typically the size of the vocabulary. Similarly, the skip-gram strategy is to predict a window of words given a specific word. For example, consider the sentence "a cat sat on a mat." Using CBOW, one would try to determine from a fragment "a cat sat on a," that the next word is "mat." In case of skip-gram given a word "sat," one would predict "a cat," and "on a mat." This provides context to the word "cat" and would eventually determine its position in the corpus vector space (Goldberg and Levy, 2014).

A similar embedding model based on word co-occurrence called Global Vectors (GloVe) (Pennington et al., 2014) successfully managed to represent words in a vector space too. Even though their approaches to how they built were different the two embedding models were both great leaps in unsupervised text categorization techniques and have been relied on for most NLP tasks until recently (Howard and Ruder, 2018). In 2018, Howard & Ruder proposed, what they termed as the Universal Language Model Fine Tuning (ULMFiT) which uses a large corpus on a given language to train a general model. This model is then fine-tuned towards a target corpus with techniques to achieve good accuracy at a much less computational cost than training a model from scratch (Howard and Ruder, 2018). ULMFit leverages the Weight Dropped LSTM

(Merity et al., 2018) as its base language embedding layer.

The weight dropped LSTM is based on LSTM layer without any significant internal modification of the recurrent neural network (RNN). The weight dropped LSTM instead relies on a novel Averaged Stochastic Gradient Descent (ASGD), application of Drop Connect (Wan et al., 2013), and modified regularization method for sequential language modeling.

CHAPTER 3

METHODS

As a goal of this research to examine the language content of conversations in the social media space and discover if there is a linguistic predisposition as it relates to a person's political bias, I looked for textual data representative of political demography. With this aim, I collected user tweets from @TheDemocrats and @GOP accounts on Twitter using the Twitter timeline API. For the first hypothesis, both text visualization and, term frequency analysis, were conducted to explore the characteristics of the data. Then I set up my ULMFiT pipeline and loaded weights from a pre-trained language model. I then fine-tuned my language model on a large corpus of Tweets. Finally, I used my data from @TheDemocrats and @GOP to train my classifier using the feature generated by the fine-tuned ULMFit model.

Data

Due to the structure of tweets, the language model inferred from them tend to be very different and specific (Go et al., 2009). Twitter messages, while originally introduced at 140 character limits is now 280 characters long. However, it has been seen that an average tweet consists of around 14 words (Go et al., 2009). This small length poses a challenge when thinking in the context of language models since typical language models are trained on much larger bodies of text.

Besides, since Tweets are composed using various devices, the frequency of misspelling is a lot higher (Go et al., 2009). Tweets also have their own set of nuances such as using a hashtag (a word preceded by the pound sign #) to display a concise point of the message or perhaps to show agreement with a class of tweets using the same hashtag. Similarly, the "@" symbol is used to mention other users. Finally, a retweet is a feature that allows users to quote and tweet another tweet. These retweets are preceded by a capitalized "RT." Most analysis and research up until now have cleaned tweet data by removing these nuanced elements of a tweet. Some recent researches included hashtags as a part of their analysis to determine topic or stance (Mohammad et al., 2017).

Recent work in language modeling has seen the importance of character level alteration. This is especially true when dealing with tweets because of their already small length. For example, the "RT" in a tweet signifies a level of agreement, and as a result, the content of the tweet might be weighted differently

	Source	Full Text
0	dem	We marched. We protested. Now, we #VoteThemOut. Confirm your polling place at and let's get ready to show up at the polls like never before...
1	dem	Democrats put hope on the ballot, and we never backed down from our values of inclusion and opportunity, because we know that those are not ...
2	gop	RT @JohnCornyn: They are not serious: Today, the President offered both Democrats and Republicans the chance to meet for lunch at the White
3	gop	Warren should not be able to escape liability for her deceptions. The Texas Bar should open an ethics investigation of Warrens false represe...
4	gop	My family and my name have been totally and permanently destroyed by vicious and false additional allegations. -Judge Kavanaugh

Figure 3.1. Dataframe preview of the tweets from @TheDemocrats and @GOP.

depending on the topic of the corpus.

In this study, I collected a random sample of tweets from two Twitter accounts - @TheDemocrats, and the @GOP. Since each of those accounts is targeting specific groups, I wanted to start my analysis with these. First, I used the Twitter API and a Python wrapper for the API called Tweepy to collect timeline data from each account. Since I was using the timeline endpoint, there was a limitation to the amount of data I could extract. I collected several snapshots of Tweets from March 1st, 2019 till March 28th, 2019.

Once the data was collected, I loaded the dataset containing tweets from @TheDemocrats into a Pandas Dataframe object. I then dropped all the columns and retained only the "full_text" column which contains the tweet text. I then iterated through the data and removed all URL contents in the tweets, also removed some selected special characters that do not exist in the ASCII table (ordinals above 128). Note that I did not remove mentions or hashtags because they contain information that is both important in the Twitter community, and thus can be leveraged for my analysis. I then added a column specifying the source of the tweet in each row, in the case of @TheDemocrats I used the term "dem."

I repeated the process for the tweets collected from the @GOP account and used the term "gop" in my source column. Once the two Dataframes were ready, I concatenated them along the vertical axis (row-wise) to form my final Dataframe with only the source and the full-text columns. Figure 3.1 shows a snippet of my Dataframe.

The next step was done with three simple manipulations of the text – raw text, text without stop words, lemmatized bigrams and trigrams. Trigrams captured all the information we needed and n-grams of higher n values did not yield anything interesting. To be more specific, n-grams higher than a trigram were usually sentence fragments, and in the off chance that they did result in meaningful phrases were rare enough to be insignificant. Length of tweets added to the insignificance of higher valued ngrams. Another technique used in the process was to first convert the tweets into a bigram model and then using the trigram phraser to

extract trigrams from the bigram models. This helped us in reducing redundancy or fragmented phrases.

Since the language model I used was pre-trained on selected curation of text in Wikipedia, and my target classification model was data from Twitter, I needed to fine-tune my model using a corpus of tweets. For this task, I used the Twitter dataset compiled by Go et al. (2009) containing 1.6 million tweets.

Model 1: Term Frequency Analysis and Text Visualization

To test my first hypothesis, I performed several text analytic techniques, which include but are not limited to topic analysis using LDA, term frequency analysis, n-gram phrase extraction, etc. However, in the end, a variant of term frequency analysis using a visualization library called Scattertext (Kessler, 2017) provided deterministic results that I could rely on to proceed forward with this project.

Visualization techniques allow depiction of nuances in data that help to either tell a story or provide valuable insight into the nature of data. Text visualization can be either complex or straightforward depending on the level of features depicted. Scattertext (Kessler, 2017), aid in visualizing words based on four-pointed features – precision, recall, non-redundancy, and characteristicness.

According to Kessler (2017), precision shows a word's discriminating power without any context to frequency. In the case of Scattertext, words close to the x and y-axis have high precision. A perfect precision would mean a word appears only once in a categorized corpus. Recall, on the other hand, determines the word frequency relative to a specific class. If a word has high recall it tends to have low precision, for example, stop words, and appears close to the top right side of a Scattertext plot. Additionally, the two other measures Kessler (2017) introduced in Scattertext is non-redundancy, which is a non-trivial measurement based on the co-occurrence of two words where one of the word pair has high precision and recall, as a result, the other word is less important. Finally, the characteristicness of a word is calculated by comparing the frequency of a word in a given category versus its frequency in the entire corpus. The characteristic coefficient of a word that tends to appear only in a single category is considered high. Scattertext weighs each word and n-gram phrase based on the four measures, and plots it as a scatter plot. In my usage, since I am comparing two distinct sources of information, I plotted the data as categories against one another. That is to say, I had one category (Tweets from @TheDemocrats) along the vertical axis and the other (Tweets from @GOP) along the horizontal axis.

Scattertext has a few convenience-methods to build a corpus from specified Dataframe columns. It uses the Spacy NLP library to tokenize and build n-gram phrases (bigram and trigram) to construct a bag of words

and dictionary. I generated two Scattertext plots, one, the text I used to build my corpus contained all stop words, and two, I removed the stop words before generating my visuals. The idea was to see the difference in how words interact with one another with or without the stop words.

Result Summary

Figure 3.2 shows the output of Scattertext on the input text where the vertical axis represents words and phrases from @TheDemocrats and the horizontal axis shows words and phrases from the @GOP. The higher up in the vertical axis, the higher the frequency for the words and phrases used by the @TheDemocrats Twitter account, and the horizontal axis shows the same for @GOP.

It is easy to note that the top right corner depicts words and phrases used by both accounts, while the words in the bottom left corner are infrequent word and thus is mostly blank spaces. The shape of the spread is interesting because it shows distinct sets of words that are dominant in by tweets derived from one or the other account but not both. Also, since I did not remove any stop words at this stage, one can note that the top right corner (shared frequent words) are grammatical constructs of language – conjunctions, prepositions, etc., further reinforcing my hypothesis that the two accounts are a good candidate for source classification.

Figure 3.3 shows the Scattertext output where the stop words have been removed. This was done as an exercise to see what kinds of words fill out the top right corner of the plot once the common language words have been removed. It isn't surprising that the top three words are "vote," "america," and "people." All of these are relevant political words that are expected to be within the vernacular of any American political party.

Model 2: Classification through Transfer Learning

Following my findings from my term frequency analysis and Scattertext plots, I decided to test my second hypothesis by using the Universal Language Model fine tuning (ULMFiT) pipeline (Howard and Ruder, 2018). The universal language model uses the Weight Dropped LSTM (Merity et al., 2017). I used the FastAI (Howard, 2017/2018) library API to load the pre-trained weights. The pre-trained weights used were trained on the Wikitext-103 dataset (Merity et al., 2016). Once I had the weights loaded, I prepared my data according to the FastAI documentation, which can load directly from a CSV file or use data structured in a Pandas Dataframe.

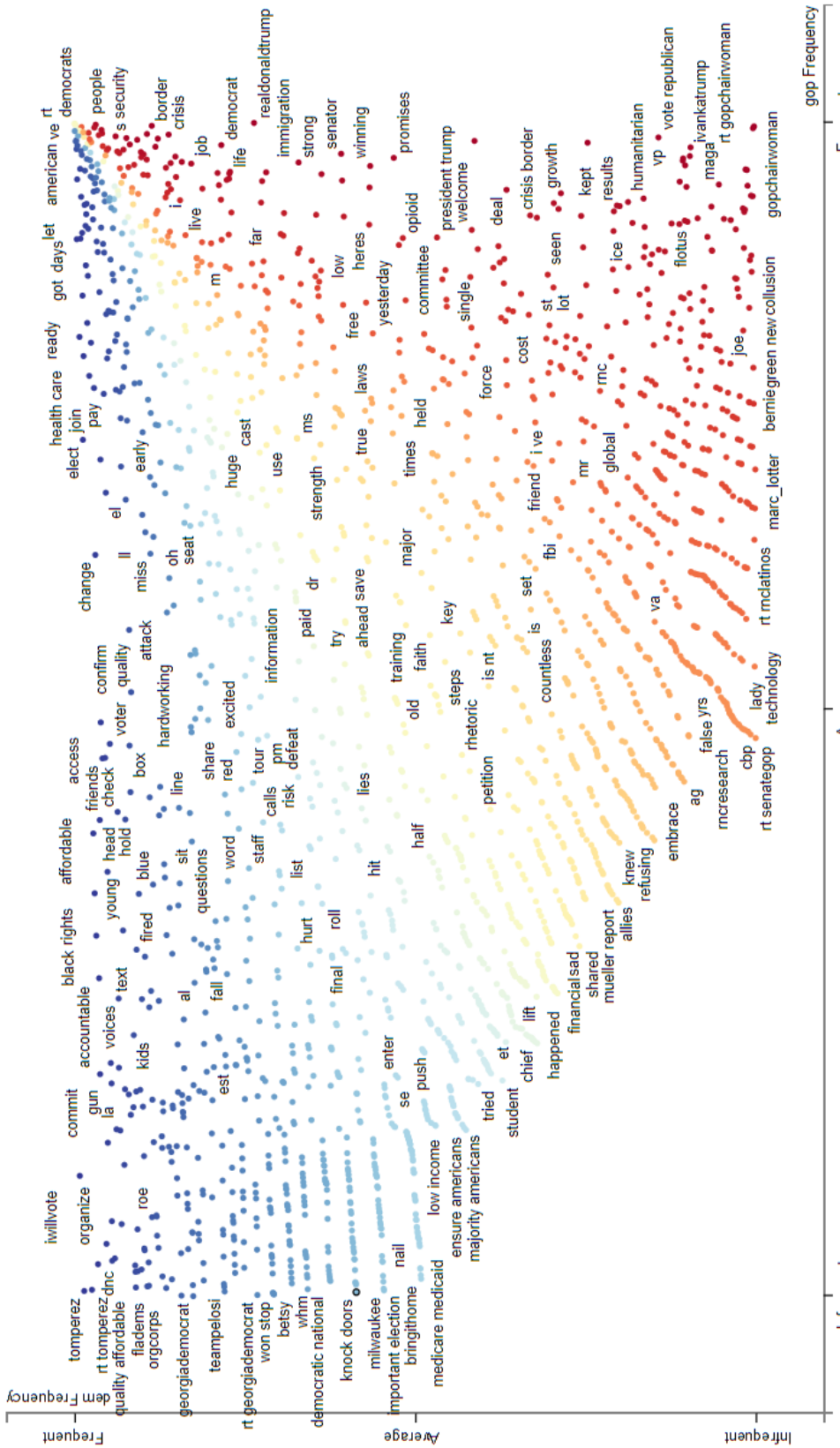

Figure 3.2. Scattertext output showing the terms usage relationship weighted term frequency

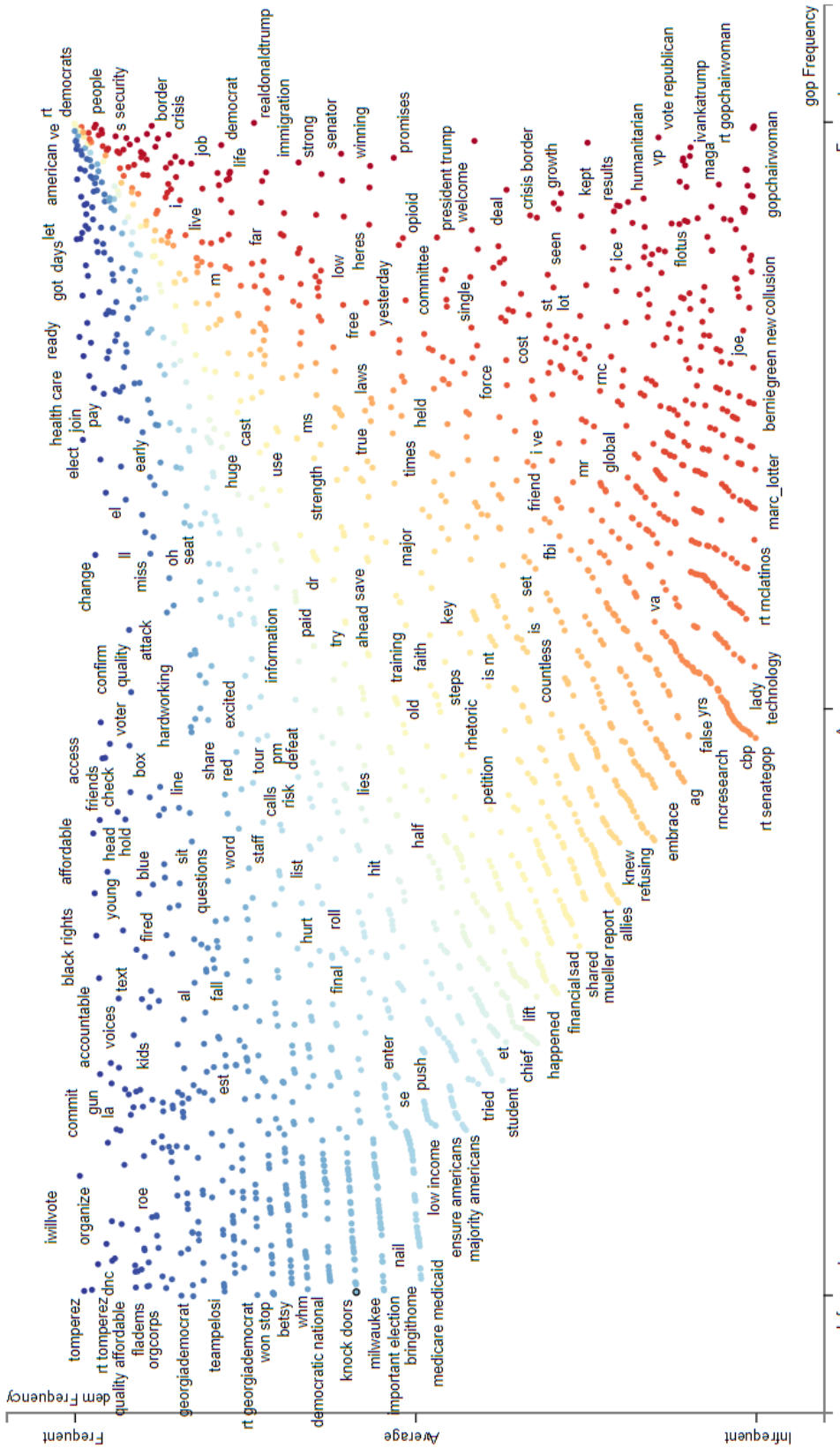

Figure 3.3. Scattertext output showing term frequency relationship without stop words

The Dataframe is expected to have the dependent variable in the first column and the input data in the second column. My Dataframe was already prepared for such a task, where my dependent variable was the source column containing a binary category - “dem,” or “gop” and my input data was the full text from the tweets. The source column was coded as either 1 for “dem,” or 0 for “gop.” The full text was prepared transformed using complex language rules to retain a high level of contextual information.

Preparing the Data

I split the dataset into training and test set. I started by randomly sampled 80% of the dataset, amounting to 5,162 tweets for training and the remaining 1,290 tweets were used for testing the classifier.

The FastAI library provides implementations of convenience methods to preprocess the text in a fast and efficient manner using graphics processing units and multi-threading. On top of the computational code optimization, the library also provides tokenization, and creation of language indexes to retain as much language information as possible. As a result, I use the TextLMDataBunch class that returns the text as an object called DataBunch containing the tokens of the input text weighted with the pre-trained language model among the other earlier mentioned optimized setup.

The tokenization is performed using Spacy tokenizer but FastAI adds some advanced character and word level indexes using several rules which are shown in Table 3.1. The returned text is all lowercase annotated by special language rules as shown in Figure 3.4.

idx	text
0	threat to a woman 's right to choose . xxmaj help us # stopkavanaugh : xxbos xxup rt xxup @nmdems : xxmaj every bit of evidence suggests that the # goptaxscam is not working for anyone other than xxmaj republican donors . @ladysunshinenm ref xxbos xxup rt @reppressley : xxmaj an attack on the worker is an attack on all of us as far as i m concerned .
1	if you want to help defeat xxmaj trump i xxbos xxmaj donald xxmaj trump set a dangerous precedent when he decided to attack families seeking asylum . xxmaj this is not normal , and it 's not who we are a nation . xxbos xxmaj we ca n't pick a new sticker design , so we want you to choose . xxmaj vote on your favorite below xxbos xxmaj
2	inaccurate census data , which would harm the governments ability to provide necessary services to communities across the country . xxbos xxmaj heather xxmaj nauert , @realdonaldtrumps nominee for xxup u.n. xxmaj ambassador : \n -served in a senior role at the xxmaj state xxmaj dept . \n -designated as xxmaj acting xxmaj under xxmaj secretary for xxmaj public xxmaj diplomacy & & xxmaj public xxmaj affairs \n
3	the xxmaj american people are raising their voices to a deafening roar today . xxmaj we will not stop marching , we will not stop fighting , xxbos xxmaj we are not going to fold , we are on the right side of this argument . -@johncornyn xxbos xxmaj they re all so out of touch with xxmaj american workers . xxbos xxup rt @senatedems : .@senatordurbin : xxmaj
4	: xxmaj if you live in xxmaj clark or xxmaj washoe xxmaj county you can vote at xxup any voting center this year ! xxmaj no assigned polls which ma xxbos xxmaj we could n't be happier to celebrate @justicecbeasley on becoming the first xxmaj african xxmaj american woman to serve as xxmaj north xxmaj carolina 's chief justice . # xxup bhm xxbos .@realdonaldtrump wants xxup you to

Figure 3.4. Snippet of the text after applying language model rules

Table 3.1

Rules and Examples for Markup

Rule	**Example**
Add a new token xxmaj in front of a capital letter.	Twitter is a social media → xxmaj twitter is a social media
Add a new token xxup in from of words written in all capital letters.	THIS IS GREAT → xxup this xxup is xxup great
Add a new token xxrep followed by an integer n for characters repeated more than three times in a word.	this is awesome!!!! → this is awesome xxrep 4 !
Add a new token xxwrep followed by an integer n for words repeated more than four times in a row.	this is so so so so so amazing! → this is xxwrep 5 so amazing !
Remove two or more occurrence of continuous spaces with a single space.	inconsistent spacing is not fun → inconsistent spacing is not fun
Add spaces forward slash and pound (#)	twitter has #hashtags and /url/to/somewhere → twitter has # hashtags and / url/ to/ somewhere

Fine Tuning

To test my classification setup, I fit my data as-is into the model and did not witness any learning for 20 cycles. The resultant accuracy was 46% where the last few iterations so no improvement in the learning. Table 3.2 shows the twenty cycles of learning. Details of the classification model are outlined later in the classification section.

This made sense because the Weight Dropped LSTM used in the pipeline was pre-trained on the Wikipedia dataset (Merity et al., 2016) which is vastly different from Tweets and required fine-tuning on a large Twitter corpus. I used the 1.6 million tweet dataset (Go et al., 2009) as a large enough Twitter corpus to fine-tune the model.

Since 1.6 million is a very large number and training Neural Networks is a time-consuming affair, I decided to start with a random sample of the dataset. I took a random sample of 200, 000 tweets and used FastAI's language model learning to update the weights of the pre-trained LSTM.

I first unfroze the entire sequential model and then updated the model a low 10^{-3} learning-rate to ensure there is no catastrophic forgetting (Howard and Ruder, 2018). I fit the model for a total of 10 cycles before classifying my data. As an added experiment I fine-tuned the model on a larger sample of 400,000 tweets to see if it provides any improvement in my classification result, which is further discussed in the resulting

Table 3.2

Learning per epoch based on the general model

epoch	train_loss	valid_loss	accuracy	time
0	4.176107	3.627949	0.372210	00:18
1	4.023833	3.461562	0.385201	00:18
2	3.839609	3.306321	0.409174	00:18
3	3.650410	3.195148	0.416562	00:18
4	3.462623	3.093271	0.431540	00:18
5	3.270577	3.022630	0.437857	00:18
6	3.080475	2.985003	0.447009	00:18
7	2.910681	2.969759	0.449978	00:18
8	2.741373	2.959977	0.452054	00:18
9	2.595567	2.964245	0.457277	00:18
10	2.463413	2.958068	0.460045	00:18
11	2.349846	2.971541	0.460179	00:18
12	2.250453	2.981655	0.461674	00:18
13	2.169481	3.001044	0.463125	00:18
14	2.094010	3.014725	0.461116	00:18
15	2.043698	3.028789	0.461585	00:18
16	1.994101	3.031569	0.462277	00:18
17	1.969386	3.041887	0.461183	00:18
18	1.943084	3.043045	0.461897	00:18
19	1.934984	3.043704	0.461920	00:18

summary section.

Classification

The classifier takes the output of the RNN and concatenates the output with blocks of batch normalization layer, dropout layer with ReLU activation that is then connected to a Softmax activation layer for the classification (Howard and Ruder, 2018). Following the findings of the paper presented by Howard & Ruder (2018), I updated the weights of the network by gradually unfreezing each layer to fit my classifier. This process of unfreezing a network layer by layer to make minor updates to the weights is the main component of transfer learning.

I first trained the classifier layer for one cycle (Smith, 2018). To be able to use the one cycle policy learning method, I first determined the learning rate by performing mock training on batches starting with a small learning rate and slowly increasing the learning rate, which is then plotted against the corresponding loss (Smith, 2018). I picked a learning rate, from the graph, that still has decreasing loss and about an order of magnitude before the minimum point as shown in Figure 3.5.

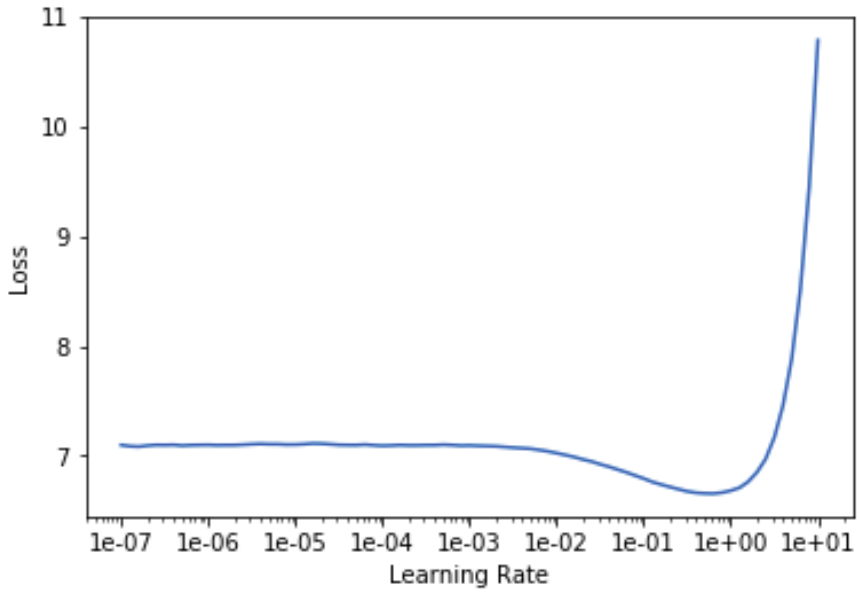

Figure 3.5. Learning rate curve for the classifier model

Then I used the one cycle policy (Smith, 2018) where the idea is to set a maximum learning rate and a momentum. The learning is slowly increased from a fraction of the maximum learning rate to the maximum learning rate while at the same time decreasing the momentum. Then I repeated the process but this time starting at the maximum learning rate and decrease the learning rate while increasing the momentum. During the decreasing phase once the minimum rate has been reached I kept the momentum fixed while continuing further towards a learning rate which is 100 times smaller. Figure 3.6 shows the one-cycle learning that was performed on the data.

I started by training the RNN and Softmax layers which are the final classification layer of the neural network while the rest of the network weights were kept frozen. Then I sequentially unfroze each layer and trained the weights until all the layers were unfrozen. The final network was trained at a very slow pace to avoid catastrophic forgetfulness (Howard and Ruder, 2018) until the learning plateaus. Details of the training process are presented in appendix B.

Results Summary

Once my training was through, I used Panda's "crosstab" method to generate a confusion matrix. The "crosstab" method computes a simple cross-tabulation of the frequencies of two or more factors by default.

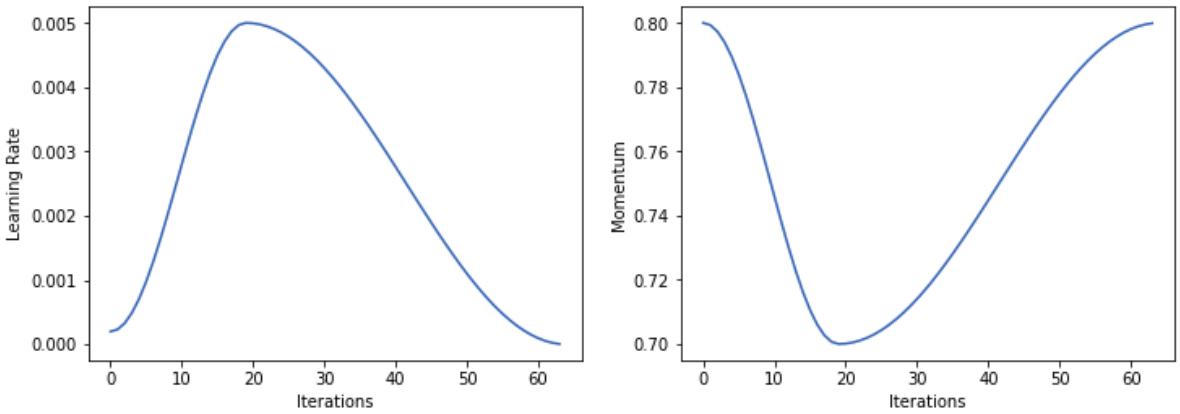

Figure 3.6. One Cycle Learning based on Smith (2018)

In my case, I used a sample of 1040 rows of training data to produce the table. I then used the classifier to predict on my original test set which is completely unseen by the model to generate another confusion matrix to compare variations between classification results between seen and unseen data.

My fine-tuned model extended on 400,000 tweets produced much better results than the one tuned on 200,000 tweets. Both Table 3.3 and Table 3.4 shows the result from the fine-tuned classifier based on the 400,000 tweets. Table 3.3 shows the confusion matrix for the classification result on a sample of the training data the classifier has come across some time during the training phase. From the matrix, I can easily calculate the accuracy to be 89.4%.

Table 3.3

Confusion Matrix for seen data

$n = 1040$	**@TheDemocrats**	**@GOP**
@TheDemocrats	471	53
@GOP	57	459

Table 3.4 shows the confusion matrix is generated from the test data that is completely unseen by the classifier. The accuracy calculated is 87.7%.

Table 3.4

Confusion Matrix for unseen test data

$n = 1255$	**@TheDemocrats**	**@GOP**
@TheDemocrats	520	75
@GOP	79	581

For the sake of completeness and to keep true to my words, the confusion matrix in Table 3.5 shows the results on the fine-tuned model from 200,000 tweets.

Table 3.5

Confusion Matrix on seen data for model tuned to 200,000 tweets

$n = 1040$	**@TheDemocrats**	**@GOP**
@TheDemocrats	487	98
@GOP	39	416
Accuracy	86.6%	

Fine-tuning my model on 200,000 tweets took about 1.5 hours using an nVidia Tesla K80 card provisioned from the Google Cloud Platform (GCP). The same task on 400,000 tweets took a total of 4 hours. Fitting on entire 1.6 million tweets was projected to 13 hours, which was abandoned due to the expectation that the increase in model accuracy will be insignificant.

CHAPTER 4

DISCUSSION

NLP research relied on a domain-specific contextual analysis of syntax for semantic evaluation. Text processing such as removal of stop words, or even special characters, especially in informal communication such as the ones used in social media loses a lot of information to build such syntactic models. Language models allow us to retain more information while providing strong semantic relationships.

Leveraging current advancement in language models and neural networks, I have managed to train a classifier with 87.7% accuracy. This confirms our hypotheses regarding the distinct usage of vernacular between the two accounts @GOP and @TheDemocrats. We have seen both from our term frequency analysis, and classifier model that both the semantic and syntactic content of a tweet is strongly coupled with the ideological source or target of said tweet.

I further analyzed the tweets that were mis-classified as shown in Appendix C. My sampling of misclassification elicits the overlaps between the two ideological polarities (political left and political right). As is the nature of the content in individual tweets, some tweets just do not contain enough textual information for a language model to properly classify the content. In other cases, domain knowledge of the entities was expected. For example, it is easy for us to denote a tweet as being politically right if the content was in favor of or contained a quote by someone who was famously ideologically right inclined. However, the classifier is blind to such existing climatic knowledge and predicts solely based on the language used.

This elucidates another limitation in the model which is imposed due to being a binary classifier. Language model-based classifiers which can compute complex semantic relationships would seem to perform better if there was an option for another class that specified the ”neither/nor” relationship. Specifically speaking, a ”neutral” class so that the model can learn from content which is neither politically left or right-leaning. This will reduce the number of misclassifications and thus possibly increase accuracy. However, creating a neutral class is not trivial because it will require building a training set that is carefully examined to contain politically neutral content.

It can be argued that selecting tweets from specific domains such as scientific research, entertainment, sports, etc can provide neutral content. In this day and age of political ubiquity, it cannot be said for certain that those would be bereft of political slants. Utilizing curated tweets with annotated stance could be future

research and an experimental avenue for a more robust classifier in this domain.

Based on the findings of this research I expect to be able to extend this classification model to classify between unseen authors by setting up a premise of how similar is one tweet to either the @TheDemocrats or @GOP and by extension conclude whether a tweet is politically left or right inclined. Such classifiers can be of high value to both businesses and society at large. The myriad of controversies regarding external manipulation (Boatwright et al., 2018) in the election of 2016, it is becoming crucial to help the public make an informed decision. Additionally, the requirement to determine troll accounts or misinformed manipulative content sources is more than ever before. Not only can research in these techniques help the political arena but organization can also gain from better understanding the subtle slants their information present. Finally, it can provide much needed quantitative background to the debate of political bias in news media outlets.

CHAPTER 5

CONCLUSION

Text communication lack interlocutor information such as facial expressions and voice intonation. However, the written text can be very expressive and contain a lot of emotional, cognitive, and ideological information. One could argue that social media content such as tweets, due to their character limitations could be bereft of deep emotional content. However, due to the lack of formalization on how a tweet can be written, they tend to pack a lot of emotional and ideological content. These can be determined using precise NLP techniques.

In this project, I set out with two Twitter accounts which are notably idealistic in their partisan position - @TheDemocrats, and the @GOP. I explored a sample of the tweets from these two accounts to look for either difference in topics, information presentation style. While it is expected that they would have different topics to talk about, however, how they would use their language was of greater interest. To test my expectations, I used text visualization techniques to determine the distribution of terms, and phrases among each source and how they relate to one another.

Once I confirmed through my visualization that there is a significant difference in the terminologies used by either source, I leveraged recent advancement in NLP and language models to build a classifier using transfer learning. I transferred learning from a pre-trained language model, altering as little as possible in the process to see if a general language model can converge to my dataset to see if there are indeed language nuances that separate the two sources. I found that with careful fine-tuning of hyperparameters in the universal language model, the classifier does indeed converge with considerably high accuracy. My model achieved an accuracy of 87.7%.

REFERENCES

Bengio, Y., Ducharme, R., Vincent, P., and Jauvin, C. (2003). A neural probabilistic language model. *Journal of machine learning research*, 3(Feb):1137–1155.

Boatwright, B. C., Linvill, D. L., and Warren, P. L. (2018). Troll factories: The internet research agency and state-sponsored agenda building. *Resource Centre on Media Freedom in Europe*.

Choi, J. and Lee, J. K. (2015). Investigating the effects of news sharing and political interest on social media network heterogeneity. *Computers in Human Behavior*, 44:258–266.

Chung, C. and Pennebaker, J. W. (2007). The psychological functions of function words. *Social communication*, 1:343–359.

Del Vicario, M., Zollo, F., Caldarelli, G., Scala, A., and Quattrociocchi, W. (2017). Mapping social dynamics on Facebook: The Brexit debate. *Social Networks*, 50:6–16.

Diakopoulos, N. and Naaman, M. (2011). Towards Quality Discourse in Online News Comments. In *Proceedings of the ACM 2011 Conference on Computer Supported Cooperative Work*, CSCW '11, pages 133–142, New York, NY, USA. ACM.

Go, A., Bhayani, R., and Huang, L. (2009). Twitter Sentiment Classification using Distant Supervision. page 6.

Goldberg, Y. and Levy, O. (2014). Word2vec Explained: Deriving Mikolov et al.'s negative-sampling word-embedding method. *arXiv:1402.3722 [cs, stat]*.

Hermida, A. (2010). From TV to Twitter: How ambient news became ambient journalism.

Hille, S. and Bakker, P. (2014). Engaging the Social News User: Comments on news sites and Facebook. *Journalism Practice*, 8(5):563–572.

Howard, J. and Ruder, S. (2018). Universal Language Model Fine-tuning for Text Classification. *arXiv:1801.06146 [cs, stat]*.

Iqbal, M. and Khan, S. (2018). Mining Facebook Page for Bi-Partisan Analysis. In *SAIS Proceedings 2018*. Southern AIS.

Jean Emile Gombert (1992). *Metalinguistic Development*. University of Chicago Press.

Kacewicz, E., Pennebaker, J. W., Davis, M., Jeon, M., and Graesser, A. C. (2014). Pronoun Use Reflects Standings in Social Hierarchies. *Journal of Language and Social Psychology*, 33(2):125–143.

Kessler, J. S. (2017). Scattertext: A Browser-Based Tool for Visualizing how Corpora Differ. *arXiv:1703.00565 [cs]*.

Li, L. and Sporleder, C. (2010). Using Gaussian Mixture Models to Detect Figurative Language in Context. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 297–300, Los Angeles, California. Association for Computational Linguistics.

Louis, A. and Nenkova, A. (2011). Automatic identification of general and specific sentences by leveraging discourse annotations. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 605–613, Chiang Mai, Thailand. Asian Federation of Natural Language Processing.

Marcu, D. (1999). Discourse trees are good indicators of importance in text. *Advances in automatic text summarization*, 293:123–136.

Menyuk, P. (1985). Wherefore Metalinguistic Skills? A Commentary on Bialystok and Ryan. *Merrill-Palmer Quarterly*, 31(3):253–259.

Merity, S., Keskar, N. S., and Socher, R. (2018). An Analysis of Neural Language Modeling at Multiple Scales. *arXiv:1803.08240 [cs]*.

Merity, S., Xiong, C., Bradbury, J., and Socher, R. (2016). Pointer Sentinel Mixture Models. *arXiv:1609.07843 [cs]*.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013a). Efficient Estimation of Word Representations in Vector Space. *arXiv:1301.3781 [cs]*.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013b). Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, pages 3111–3119.

Milroy, L. and Milroy, J. (1992). Social network and social class: Toward an integrated sociolinguistic model. *Language in society*, 21(1):1–26. cites: milroySocialNetworkSocial1992.

Mohammad, S., Kiritchenko, S., Sobhani, P., Zhu, X., and Cherry, C. (2016). SemEval-2016 Task 6: Detecting Stance in Tweets. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 31–41, San Diego, California. Association for Computational Linguistics.

Mohammad, S. M., Sobhani, P., and Kiritchenko, S. (2017). Stance and Sentiment in Tweets. *ACM Trans. Internet Technol.*, 17(3):26:1–26:23.

Newman, M. L., Pennebaker, J. W., Berry, D. S., and Richards, J. M. (2003). Lying words: Predicting deception from linguistic styles. *Personality and social psychology bulletin*, 29(5):665–675.

Oeldorf-Hirsch, A. and Sundar, S. S. (2015). Posting, commenting, and tagging: Effects of sharing news stories on Facebook. *Computers in Human Behavior*, 44:240–249.

Pennebaker, J. W., Mehl, M. R., and Niederhoffer, K. G. (2003). Psychological aspects of natural language use: Our words, our selves. *Annual review of psychology*, 54(1):547–577.

Pennington, J., Socher, R., and Manning, C. (2014). Glove: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Prasad, R., Miltsakaki, E., Dinesh, N., Lee, A., Joshi, A., Robaldo, L., and Webber, B. (2007). The Penn Discourse Treebank 2.0 Annotation Manual. *IRCS Technical Reports Series*.

Quattrociocchi, W., Scala, A., and Sunstein, C. R. (2016). Echo chambers on facebook.

Ruiz, C., Domingo, D., Micó, J. L., Díaz-Noci, J., Meso, K., and Masip, P. (2011). Public Sphere 2.0? The Democratic Qualities of Citizen Debates in Online Newspapers. *The International Journal of Press/Politics*, 16(4):463–487.

Scholand, A. J., Tausczik, Y. R., and Pennebaker, J. W. (2010). Social language network analysis. In *Proceedings of the 2010 ACM Conference on Computer Supported Cooperative Work*, pages 23–26. ACM.

Sexton, J. B. and Helmreich, R. L. (2000). Analyzing cockpit communications: The links between language, performance, error, and workload. *Human Performance in Extreme Environments*, 5(1):63–68.

Slatcher, R. B. and Pennebaker, J. W. (2006). How do I love thee? Let me count the words: The social effects of expressive writing. *Psychological Science*, 17(8):660–664.

Smith, L. N. (2018). A disciplined approach to neural network hyper-parameters: Part 1 – learning rate, batch size, momentum, and weight decay. *arXiv:1803.09820 [cs, stat]*.

Stieglitz, S. and Dang-Xuan, L. (2013). Emotions and Information Diffusion in Social Media—Sentiment of Microblogs and Sharing Behavior. *Journal of Management Information Systems*, 29(4):217–248.

Swanson, R., Ecker, B., and Walker, M. (2015). Argument mining: Extracting arguments from online dialogue. In *Proceedings of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 217–226.

Wan, L., Zeiler, M., Zhang, S., Le Cun, Y., and Fergus, R. (2013). Regularization of neural networks using dropconnect. In *International Conference on Machine Learning*, pages 1058–1066.

Zarrella, G. and Marsh, A. (2016). MITRE at SemEval-2016 Task 6: Transfer Learning for Stance Detection. *arXiv:1606.03784 [cs]*.

Zhao, W. X., Jiang, J., Weng, J., He, J., Lim, E.-P., Yan, H., and Li, X. (2011). Comparing Twitter and Traditional Media Using Topic Models. In *Advances in Information Retrieval*, Lecture Notes in Computer Science, pages 338–349. Springer, Berlin, Heidelberg.

Ziegele, M., Breiner, T., and Quiring, O. (2014). What Creates Interactivity in Online News Discussions? An Exploratory Analysis of Discussion Factors in User Comments on News Items. *Journal of Communication*, 64(6):1111–1138.

Zollo, F., Novak, P. K., Del Vicario, M., Bessi, A., Mozetič, I., Scala, A., Caldarelli, G., and Quattrociocchi, W. (2015). Emotional Dynamics in the Age of Misinformation. *PLOS ONE*, 10(9):e0138740.

APPENDIX A

TRAINING AND FINE TUNING

Table A.1

Learning parameters for first cycle

Variable	**Value**
Number of Training Cycles	1
Maximum Learning Rate	10^{-3}
Minimum Learning Rate	$10^{-3}/25$
Momentum Range	$(0.8, 0.7)$
Final Accuracy	76.5%

Table A.2

One cycle training of the classifier layer

epoch	**train_loss**	**valid_loss**	**accuracy**	**time**
0	0.598945	0.515739	0.765385	00:08

Table A.3

Learning parameters for the last 2 layers

Variable	**Value**
Number of Training Epochs	3
Maximum Learning Rate	10^{-4}
Minimum Learning Rate	$10^{-2}/25$
Momentum Range	$(0.8, 0.7)$
Final Accuracy	86.9%

Table A.4

Training the last two layers

epoch	train_loss	valid_loss	accuracy	time
0	0.515463	0.411548	0.810577	00:09
1	0.408737	0.336093	0.850962	00:09
2	0.321439	0.283817	0.869231	00:09

Table A.5

Learning parameters for last 3 layers

Variable	Value
Number of Training Epochs	3
Maximum Learning Rate	10^{-5}
Minimum Learning Rate	$10^{-5}/25$
Momentum Range	$(0.8, 0.7)$
Final Accuracy	76.5%

Table A.6

Training the last 3 layers

epoch	train_loss	valid_loss	accuracy	time
0	0.268129	0.285971	0.887500	00:15
1	0.260361	0.270025	0.882692	00:14
2	0.227808	0.269915	0.889423	00:14

Table A.7

Training the whole model at a slow learning rate

epoch	train_loss	valid_loss	accuracy	time
0	0.214624	0.273522	0.884615	00:20
1	0.199118	0.272517	0.893269	00:19
2	0.199700	0.270088	0.894231	00:19

APPENDIX B

MISCLASSIFIED TWEETS

Figure B.1. Tweet misclassified as @GOP

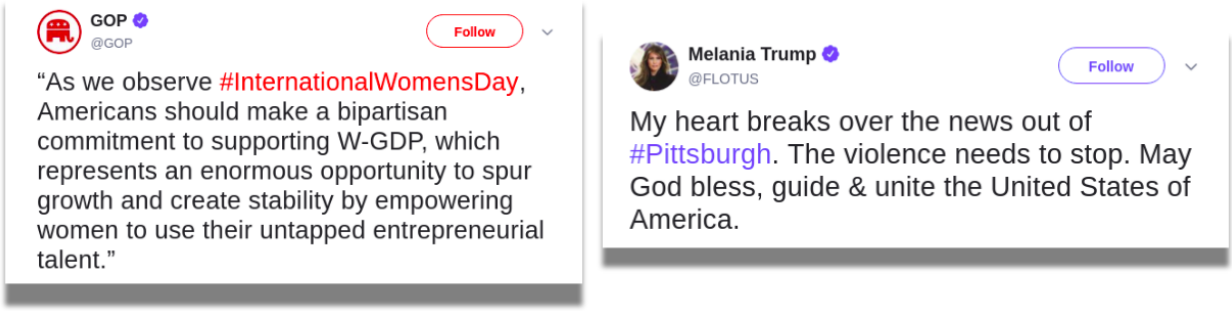

Figure B.2. Tweet misclassified as @TheDemocrats

Table B.1

Sample of misclassified tweets

Stance	Tweet	Predicted
gop	RT @ScottforFlorida: .@SenBillNelson regarding your slanderous attacks on me & my wife- do you think my wife cant manage her money without	dem
gop	RT @GOPChairwoman: All the political theatrics in the world wont change that Democrats sat on information for months & leaked it to the me	dem
dem	RT @SenateDems: This week, the Senate will vote on Chad Readler to be a Circuit Court Judge. Readler filed the Trump admin brief calling f	gop
gop	RT @GOPChairwoman: All 5 of these Democrats went to law school, but theyre willing to totally disregard the bedrock of our justice system	dem
gop	Today, we remember the 35th anniversary of the Beirut Barracks Bombing. May we always honor the immortal sacrifice of 241 heroes who gave their lives for our freedom. https://t.co/h7VYalCu8C	dem
gop	RT @SenKevinCramer: If I sponsored a resolution that was brought to the floor, I would not hesitate to vote for it. Democrats refusing to	dem
gop	RT @GOPChairwoman: The mainstream media always finds time for guests who attack @realDonaldTrump for wanting to secure our borders. How a	dem
gop	It is completely inconceivable to me that he did the things she is alleging. https://t.co/5lPQC9E3WW	dem
dem	RT @sabrinasingh24: HEADLINE: New data: Democrats crushing Republicans in 2018 elections https://t.co/3mAXo0KhJQ	gop
gop	We all know that many blue collar workers actually support @realDonaldTrump, but when labor leaders are coming out and slamming the Green New Deal as bad for jobsbad for their membersthat's a bad sign for democrats.@marc_lotter `https://t.co/OjFmLqkyJR`	dem

Table B.2

Sample of misclassified tweets

Stance	Tweet	Predicted
dem	So while he's patting himself on the back, Americans are dying every day from a crisis that the government has the power to address. Democrats will stay committed to providing access to care so we don't lose any more lives.	gop
gop	What they're doing is just trying to find any excuse to go and appease their base who wants the President impeached. They want to reverse the outcome of the election where the American people said Donald Trump will be our president. -@SteveScalise https://t.co/xPZeW7WyIC	dem
gop	RT @GOPChairwoman: The southern border is a dangerous, horrible disaster. We've done a great job, but you can't really do the kind of job	dem
gop	Clarence Henderson was one of the first students to take part in the historic Greensboro sit-in. Tonight, he was honored at the White House for his courage. https://t.co/5AHPTi6Rt2	dem
dem	RT @dncpress: FACT CHECK: ISIS was already on the decline when Trump took office. https://t.co/zr5YrORRqF	gop
dem	RT @BrandonBG_: Yes. I needed help. The house I grew up in was under 12 feet of water for two weeks. Now Im here to help remove Steve Kin	gop
dem	RT @HRC: In todays narrow ruling against the Colorado Civil Rights Commission in #MasterpieceCakeshop case, the Supreme Court acknowledged	gop
dem	`https://t.co/21AQpAgV7L`	gop
dem	We don't think taking money from health care programs for Americans who need it most is considered "savings." `https://t.co/NOnqCTOUnn`	gop
gop	RT @IvankaTrump: When Washington works, America wins. `https://t.co/Bu7622m6Ai`	dem