BIGGERS, FREDERICK BROWN, M.S. Deep Semantic Matching Approach Towards Dynamic Text Filtering in Micro-Blog Messages. (2020)
Directed by Dr. Somya D. Mohanty. 53 pp.

There is a growing interest in using social media content for Natural Language Processing applications. This paper seeks to demonstrate a way to present the changing semantics of Twitter within the context of a crisis event, specifically tweets during Hurricane Irma. Using an implementation of the Word2Vec method of Neural Network training mechanisms developed by Mikolov, et al to create Word Embeddings, this paper will: discuss how the relative meaning of words changes as events unfold; present a mechanism for scoring tweets based upon dynamic, relative context relatedness; and show that similarity between words is not necessarily static.

DEEP SEMANTIC MATCHING APPROACH TOWARDS DYNAMIC TEXT FILTERING IN MICRO-BLOG MESSAGES

by

Frederick Brown Biggers

A Thesis Submitted to
the Faculty of The Graduate School at
The University of North Carolina At Greensboro
in Partial Fulfillment
of the Requirements for the Degree
Master of Science

Greensboro
2020

Approved by

Committee Chair

for carmen, jt, and jamey

APPROVAL PAGE

This thesis written by Frederick Brown Biggers has been approved by the following committee of the Faculty of The Graduate School at The University of North Carolina at Greensboro.

Committee Chair _____
Somya D. Mohanty

Committee Members _____
Minjeong Kim

_____
Prashanti Manda

_____
Date of Acceptance By Committee

_____
Date of Final Oral Examination

ACKNOWLEDGEMENTS

This project could not have been completed without the contributions of some marvelous people. It was a long journey for this project, and any successes found in this document are due to the stable footing upon the shoulders of very tall giants.

To begin, I would like to acknowledge the enthusiasm, knowledge, and sheer academic prowess of my advisor, Dr. Somya D. Mohanty. He has been an extremely inspiring mentor on this journey, and I am fortunate to have been able to work with him on as many projects as I have. The goals of this paper were borne out of his vision, and I greatly appreciate all that he has done.

I would also like to acknowledge Dr. Mohanty's son, Kavi, whose birth last summer protected me from roughly twelve weeks of sentences that begin with "see if you can...".

I would not have been able to complete any of this work without the flexibility and genuine consideration of my supervisor and department head at the UNC Greensboro University Libraries, Franklin Graves and Tim Bucknall. Both were willing to allow me to adjust my schedule and workload to ensure a proper balance between work and study.

I would also like to thank my thesis committee members, Dr. Minjeong Kim and Dr. Prashanti Manda. They too have been very enthusiastic about this project, and I greatly appreciate their feedback and involvement during the defense.

This thesis was part of a larger study and research project, *Leveraging Twitter and Big Data Analytics for Natural Disaster Management and Recov-*

ery: A Case Study for Hurricanes Irma and Harvey, at UNC Greensboro as part of the Giant Steps seed grant project. As such, I would like to thank the contributing members of these projects: Dr. Fred Sadri, Dr. Rick Bunch, Dr. Evan Goldstein, Dr. Thomas McCoy, Jo Klein, Nastaran Pourebrahim, and Saed Sayedahmed.

I would like to acknowledge the contributions and suggestions from my readers: Mary-Katherine Amos and Maggie Murphy. Their recommendations helped me adhere to proper formatting and citations, but more importantly, helped my work maintain the appearance of being written by a dutiful library employee.

But last, and most certainly not least, I would like to thank my wife, K. Brooke Spangler, who endured the emotional rollercoaster of a husband struggling with a writing project that far exceeded any expectations of where we thought this might go.

TABLE OF CONTENTS

Page

LIST OF TABLES

Page

LIST OF FIGURES

Page

CHAPTER I

INTRODUCTION

Twitter is one of the preeminent microblogging platforms worldwide. With a reach of nearly 27 million Monetizeable Daily Active Users (mDAU)[1] in the US and 126 million mDAU worldwide, Twitter users generate nearly 500 million tweets per day [2]. Twitter's ubiquity, combined with its functionality, ease of use, and API configuration make it an frequent tool for harvesting data. Examples of this type of implementation include: pairing the metadata associated with each tweet to datasets [3] or applying spatio-temporal metadata to isolate tweets for the purpose of analyzing regionally relevant events as they occur [4]. With the prevalence of cellphone use during emergency situations, and the above mentioned features, Twitter might be an effective asset for first responders in times of crisis.

Determining what tweets would be considered relevant to the needs of emergency personnel presents a different problem. Tweets can contain any manner of content, be it observations of weather related phenomena, commentary on sports events, or social discussion. Isolating relevant tweets requires analysis of a multitude of characteristics such as from location and time based metadata, but also the content of the tweet itself. With events occurring in varying locations,

---

1 The Twitter Q4 2018 shareholder letter "...defines monetizable daily active usage or users (mDAU) as Twitter users who logged in or were otherwise authenticated and accessed Twitter on any given day through twitter.com or Twitter applications that are able to show ads [1]."

each with their own regional parlance, metalinguistics, and iconography, while addressing the meaning(s) of text changing relative to the circumstances at hand, a dynamic interpretation of linguistics is necessary. This study tested methods of optimizing context analysis for event related semiotics within tweets generated during Hurricane Irma.

Hurricane Irma made landfall on the Florida coast on September 10, 2017. At this point in its progression, it was a Category 4 storm with "...maximum winds of 115 [knots]..." and "...sustained winds of 62 [knots] and a gust of 81 [knots] were measured." [5]. Rain and wind resulted in a storm maximum of "...21.66 inches of rain... measured between 9 and 12 September..."[5] and "... produc[ing] 25 confirmed tornadoes: 21 in Florida and 4 in South Carolina."[5]. Hurricane Irma, as of 2017, was the fifth most costly Tropical Cyclone to hit the United States, with an estimated cost of damage nearly $50 billion [6].

The purpose of this project was to analyze and compare methods for Word Embeddings using vectorization in tweets generated during Hurricane Irma. A series of Neural Networks were trained via Word2Vec to convert words in tweets into numerical representations of meaningful context relationships. These contexts were then applied to find tweets which were connected to designated search terms. The resulting processes were used to identify a more comprehensive set of related tweets beyond those indicated by the presence of the initial search term(s). Findings from this project may be applicable for emergency response personnel who seek to retrieve geolocated tweets associated with disasters, without using a predetermined set of search criteria.

I.1. Basic Terms

I.1.1. Word Embeddings

Word embedding (or word embeddings) is the generic term for assigning numeric values to words, with the mathematical operations between those numeric values implying some semantic or syntactic relevance [7]. These numeric values are assigned based on a computer generated algebraic representation of observed contextual relationships. Such representations are critical in designating syntactic intent in a manner such that it is capable of being interpreted by a computer. To provide this function within such a model, word embeddings must be created based upon an algorithmic approximation of natural language. Without such a framework, words would lack the necessary connections to each other. To clarify, it is possible to take every word in the English language, alphabetize them, and assign them a numerical value. While this would provide organization and structure, there is no inherent meaning in how word no. 45 is related to word no. 50. A computer in this example, when queried, could return *trapezoid* from *trapeze*, an unlikely semantic connection.

Numerical values must therefore be established based upon a uniformly consistent translation encapsulating context and meaning between words. This process is defined as isolating commonalities between words, determining a dimensional model capable of representing relationships between these words, and assigning numeric values to words based upon their individual spatial locations. Each word then has a corresponding *vector* within this dimensionality. This vectorization of words thus *embeds* meaning into these numerical representations.

I.1.2. Corpus

Training a computer to determine word meanings requires a sufficient and relevant body of text. This body is known as a corpus. It is important that a corpus be similar in purpose to the text that is intended to be analyzed. To clarify, an algorithm trained on text retrieved from business emails may not be adequately trained to determine ingredients in cookbooks. As illustrated in Yang, et al, analysis of Twitter content by a neural network trained on "aligned" content performs better than a neural network trained on a Wikipedia dump [8]. Likewise, the meaning of an individual word is governed by its context; inconsistency across contexts can introduce an element of ambiguity, thus reducing the effectiveness of machine learning.

With the recent study conducted by Tshitoyan, et al, it was demonstrated that a sufficiently large corpus could be used to make predictions in scientific discovery [9]. This study "...collected and processed approximately 3.3 million scientific abstracts published between 1922 and 2018 in more than 1,000 journals...". These texts were then processed via the Word2Vec library, with word embeddings generated based upon the context gleaned from these abstracts. The information contained in the corpus was comprised of journal articles specifically pertaining to research on thermoelectric compounds and their properties. The articles were separated into historic timeframes, and the word embeddings created for each period. By comparing these word embeddings with publications that occurred after each period, this system was able to predict the discovery and development of thermoelectric materials well. For example, by "...analyzing abstracts published before the year 2009...", this system was able to "predict" the existence of "...a top five compound four years before its publication in 2012."[9]. While scientific dis-

covery can certainly change over nearly a century, and the language used evolves with each subsequent advance in technology, this study shows there is significant benefit to allowing a system to train on texts with consistent linguistic norms.

In the case of Twitter, the process of training via a corpus must be done with allowances to compensate for linguistic variations in grammar and syntax, as well as restrictions due to character limits. In addition to these variables, topics within Twitter can trend and the meaning of words can change based upon dominant topics. Tweets generated during a natural disaster, such as a hurricane, can change the context of concepts and words (e.g.: the difference between literal: *there is a flood on my street* and metaphorical: *a flood of tears*). As word relationships can often be derived from the relative placement of words, the context in which these words appear will add another potential avenue of complexity to the vectorization process.

Searching for tweets associated with a named occurrence, such as a natural disaster, can yield artificially limited results even when the name is used as part of the search criteria. For Twitter to provide data to emergency responders during a natural disaster, a system must be employed to help isolate tweets that are relevant to that event. Training such a system for natural disaster context recognition requires a body of temporally relevant data. Once this training is complete, a metric must be implemented by which the relatedness of terms or text can be evaluated.

If contextual information contained in tweets is to be relevant to emergency responders, two primary factors must be addressed. The first factor is that the semantic accuracy of any given system of analysis is relative to the topics trending at that point in time. The overall meaning of a given tweet is dependent

on how the words it contains are used under immediate circumstances. Changes in topics or contexts influences the interpretation of individual words [10]. Static training of machine learning systems on enormous corpora is effective for probabilistic interpretation of consistent meaning across a uniform body, but lacks the nuance necessary for interpreting polysemy as it changes from moment to moment.

The second factor is matter of available resources. It is important that the analysis functionality of this system be efficient at a level of computational infrastructure investment attainable in situations where funds and capability are limited on short notice [11]. Again, while corpora of millions or billions of lines of text are necessary to train more universal text recognition machine learning models, their efficiency can often be measured in hours or days. The typical response in cases of emergency must be significantly shorter.

I.1.3. Cosine Similarity

$$\cos\theta = \frac{\overrightarrow{i} \cdot \overrightarrow{j}}{\| \overrightarrow{i} \| \times \| \overrightarrow{j} \|} \tag{I.1}$$

Once the vectors have been constructed in a manner where spatial relationships imply syntactic relevance or similarity, mathematical comparisons of these vectors can be used to interpolate meaning. In the vector dimensional space of word embeddings, vectors of words with similar context or meaning will tend to congregate. One way to quantify vectors' spatial proximity can be done by comparing their internal angles.

The cosine trigonometric function has the property where two coincident vectors will have a cosine of 1, as their internal angle has a measure of zero. As two vectors diverge, their internal angle increases. An internal angle measure of 90 degrees has a cosine of zero. Between zero and 90 degrees, the cosine of the angle has a real, positive value between one and zero, respectively.

As the angle continues to increase above 90, and up to 180, degrees, there is a commensurate relationship with the cosine of this angle as well. The cosine of 180 degrees has a value of negative one, and the cosine of the angles between 90 and 180 degrees have a range of real, negative values between zero and negative one.

Envisioning each term within the context of a corpus as having a vector, and that vector's spatial position related to the term's context or meaning allows the relatedness of two vectors to be interpreted as inversely proportional to the degree of the internal angle formed by the two vectors.

Thus, the phrase *cosine similarity* is used as a real number representing how close two terms are within the context vector space. Two similar or related terms will have a cosine similarity as a real value close to one, where two lesser-related terms will have a lower cosine value, to a minimum at negative one.

I.1.4. Word Terminology and Designations

Throughout this paper, there will be mentions of input and output words; target and center words; as well as words and their contexts. It may seem, through repeated use, that some of these terms are interchangeable; in some particular uses, they may be.

The *target word* and the *center word* are often used as equivalent. In both cases, this is the *word* when associating training with a *context*. Often, target will

be associated with the relationship indicated by the CBOW training mechanism (See: II.2.2.1) where context is used to predict a word. Regardless, training is done on a word that exists at the 'center' of its context(s), and the terminology varies between sources.

The input and output *words* refer to the word pair currently training the neural network on a particular iteration. Whether the input or output is the center (or target) word depends on the method used. (See: II.2.2.1 and II.2.2.2)

I.2. Related Work

I.2.1. Social Media As Crisis Resource

Social media has been shown to be an effective means of addressing crisis events [13,14]. The study and responsive analyses of social media and its applicability to crisis events has been termed crisis informatics [15,16]. Crisis informatics can encompass natural disasters, such as floods [4], hurricanes, and wildfires [13], or can be applied to social and medical crises such as opioid addiction [17] and the spread of disease [14,18].

In the study of crisis informatics, social media can function as part of the toolset used in crisis preparation and emergency preparedness [19]; and for response and communication during the event [20-22]. Poblet et al. describe the roles of social media separated across distinct data types as a crowdsourced, multi-tiered tool [20]. Social media can be used as a source of data, because it can function as the product of the "crowd as a sensor" [20] by providing location data or other metadata that can be correlated with known datasets "...especially in the mitigation and preparedness phases [of disaster management]" [20]. Of particular interest is the "crowd as a reporter" [20] , wherein social media users

report "first-hand information on events as they are unfolding" to a specific social media platform [20].

Reporting data to a social media platform is the first component of the crowd as a sensor. Reuter, et al. categorizes interaction aspects of communication within crisis informatics into four categories: Authorities-to-Citizens (A2C), Authorities-to-Authorities (A2A), Citizens-to-Citizens (C2C), and Citizens-to-Authorities (C2A) [16]. In the C2C quadrant, communications are categorized as "Self-Help Communities" where private citizens are sharing crisis-related information relevant to their locality; this data is intended for other regionally coincident private citizens and is not specifically broadcast to, or for, emergency responders [16]. Such crisis-related information can have some overlap with the C2A category, which is "use of citizen-generated content." [16] Finding and assessing user-generated social media content intended either for other citizens or authorities in times of crisis, without necessarily distinguishing between the two, is essential to this study.

I.2.2. Natural Language Processing and Text Mining

Data Processing is most effective when the data input is formatted in a manner that acknowledges the idiosyncrasies of the processor. Very often, there is an precursory set of operations before import where the data is *cleaned*.

Data, when taken as recorded in its default state, usually carries with it additional information beyond what is necessary. Data in this state is often described as *messy*. This is an especially common occurrence with data intended for natural language processing. Extraneous characters or words that contribute no additional semantic value can impair processing. Therefore, removing irrelevant material is an essential step prior to analysis. While there are operations which

can handle such material, there is another component of language that presents issues. Schöch argues that natural language is comprised of "analog, non-discrete data, which cannot be analyzed or transformed computationally," and languages are "semiotic systems that have dimensions beyond the physically measurable, dimensions which depend on semantics and pragmatics, that is on meaning in context." [23] Social media content, like that contained in Twitter, exhibits many of the pitfalls of processing natural language and presents unique challenges depending on objective.

One way to mine data largely comprised of natural language is to correlate the unstructured content with more structured datasets via unique identifiers and metadata. Longley and Adnan have leveraged both the structured and unstructured data in Twitter to produce effective demographic analyses in London [3]. In their study "...represent[ing] a small and self-selecting sample of all Twitter users in London...", their methods were used to correlate geo-temporal metadata with other datasets, and employ natural language processing techniques to determine ethnicity, age, residence, and commuting routes, among other demographic data. This study further extrapolated using the "UK government Generalised Land Use Database for 2005" to pinpoint "...the probable residence of all 75,522 London-based Twitter users." [3] By combining structured data with known "clean" data sources, the study was able to use unstructured data and derive new findings.

With Twitter as a conversational vehicle, there are concerns with attempting to parse meaning out of text. Aslan and Vásquez delve into the idea of citizen sociolinguistics in internet-based discourse [24]. In their research, they observe how users co-opted the dialectic idiosyncracies of a viral video, and were able

to convey meaning via social media using a community derived metalinguistic understanding [24]. This is especially relevant for Twitter, where tweets are often analyzed in a regionally coincident context. As such it is important to acknowledge that sociolinguistic norms can affect semantics. An emphasis on semantic consistency within NLP is important in many contexts. Training NLP to recognize, or compensate for, these sociolinguistic patterns can be costly, but failing to acknowledge their impact may adversely affect effectiveness of the analysis functions.

In cases where consistent semantic interpretation over a large number of documents is important, methods have been employed to increase the immutability of the vocabulary. In Pedersen, et al. one such mechanism is to reduce the vocabulary, while minimizing the reduction's impact on meaning [25]. This has been accomplished by swapping words within an acceptable range based upon semantic similarity [25]. Priority is placed upon enforcing semantics in an absolute sense, where the meaning (or meanings) of a word remain relatively static within the context of the document, e.g. where bi-grams like *heart attack* should be correlated with *myocardial infarction* or *coronary thrombosis* [25]. Analysis on semantics, therefore, can be compared across the entire corpus despite similar concepts being represented by analogous phrases.

I.2.3. Twitter and Word2Vec

Many studies have approached analyzing the symantec content of Twitter data by using Word2Vec as a mechanism for creating word embeddings. In Yang, et al. Word2Vec was employed with various tests of hyperparameter values for analysis of tweets related to an election [8]. This study compared the effectiveness of training Word2Vec neural networks on Spanish Wikipedia with those trained

on Twitter data sets. Their training data was labeled as "election related" or "non election related" and focused on tweets that occurred during a parliamentary election in Venezuela in 2015. Their objective was to attempt to predict whether a tweet could be identified as election related based upon the vector representations of words contained in the tweet. The study found that training on an aligned data set (using Twitter data instead of a more generalized corpus, such as content from Wikipedia) and proper configuration of Word2Vec parameters (specifically increased word/context window and dimensionality sizes) proved effective at creating representations of the tweets themselves [8].

In Benton, et al, Word2Vec was one of the components used to create vector representations based upon the text of Twitter users. In their study, the intention was to create embeddings to illustrate relationships for users, rather than words, and then use these embeddings for predictive tasks. To do this, each user "representation" is a set of embeddings aggregated from "...several different types of data (views)...the text of messages they post, neighbors in their local network, articles they link to, images they upload, etc." [26]. The *views* in this context are collated and grouped based upon the testing criteria. For example, to predict user created content, a view of tweets created by a particular user would be isolated, and the neural network trained on the user's tweets as a single document. If, instead, the intended goal is predicting friends of a particular user, then the view would focus on tweets that are either liked by the user or reference other users in the text. In either case, this study uses Word2Vec to create word embeddings, and "[represents] each view as the simple average of the word embeddings for all tokens within that view." [26]

I.3. Encompassing Study

The text analysis presented here is a component of a grant funded research project taking place at UNC Greensboro: *Leveraging Twitter and Big Data Analytics for Natural Disaster Management and Recovery: A Case Study for Hurricanes Irma and Harvey*. This study analyzes selected tweets based upon predetermined criteria, and then provides emergency responders with access to the tweets that meet these criteria.

Tweets are first isolated for a specified time period and location. This working set of tweets is then processed via a group of machine learning models optimized for four different categories of geotagged and related data: weather, image processing, user reliability, and text relatedness. These processing models generate a 4-tuple of scores for each tweet, min-max scaled to 0-100. Personnel would then be presented a web interface with a tunable control mechanism associated with each score. Once a value is selected for each category, the user is provided a list of all tweets whose scores exceed the selected value.

The weather scores are derived from observed rainfall, wind speed, and distance. In the case of the Hurricane Irma study, distance component is calculated from the tweet's location relative to the NOAA observation of the hurricane's eye within the time delta. The user reliability score is based upon a supervised machine learning model trained on prediction of Twitter's own verified users. The image score is also the product of a supervised machine learning model using human-coded images categorized for the depiction of various weather effects. In this case, pictures are tagged for flood, wind, and destruction.

CHAPTER II

METHODOLOGY AND MECHANISMS

II.1. Tokenization and Cleaning

II.1.1. Tokenization

The text was first processed using regular expressions and tweet tokenization functions. One of the libraries leveraged for this process is NLTK, the Natural Language Toolkit. The NLTK `reduce_lengthening` under `nltk.tokenize.casual` will reduce concurrent repeated characters to three incidents. For example, 'OOOOOMMMMGGGGGGG' would be reduced to 'OOOMMMGGG'. It is assumed that homographs separated only by character quantity could be reduced to the same word. This operation decreases the overall vocabulary size, with minimal impact on individual token meaning.

Further token removal for stopwords was performed by removing entries in the NLTK English stopwords library. This process was in addition to the Frequent Word Subsampling formula contained in the Word2Vec specification (see: II.2.1) shown here.

$$P(w_i) = 1 - \sqrt{\frac{t}{f(w_i)}} \tag{II.1}$$

This function removes frequent terms from corpora based upon frequency, as opposed to a static list of words observed to add no additional syntactic

import. As stated by Mikolov, et al, this formula evaluates "...each word w_i in the training set..." and discards it based upon the "...probability computed by [Formula II.1] where $f(w_i)$ is the frequency of word w_i and t is a chosen threshold, typically around 10^{-5}." [27]

II.1.2. Cleaning

The terms were cleaned using regular expressions, and a custom cleaning function was defined to remove the following from all tweets:

1. Uppercase letters
2. URLs beginning with http:// or https://
3. @mentions, including those with a leading '-' or '.'
4. Punctuation, but not hashtags (#)
5. Non-hashtag # (e.g. bounded on left by word character, single-character instance, etc.)
6. Word-bounded numbers
7. encoded HTML

While there are incidents where character case might denote semantic difference, such as march (to travel in regular pattern) or March (the third month), patterns of case vary widely through tweets. In this study, as there might be the presence of inconsistent capitalization, all words were converted to lower case first, before further processing.

As strings containing URLs impart no semantic value to text, any appended URLs were stripped from text. While studies may be able to parse out hyperlinks as a possible feature for machine learning, this study prioritized the non-hyperlink content of the text of the tweet.

Once cleaned as above, remaining word tokens were processed through a stemmer function. The purpose of the stemmer is to further eliminate redundancy in the vocabulary, by treating words with the same stems as semantically equivalent. The words *heavy*, *heavier*, and *heaviest* would be reduced to *heavi*.

II.2. Word2Vec and Parameters

II.2.1. Word2Vec

Word2Vec is the result of research performed by Mikolov, et al, seeking a method for representing meaning as vectors while maintaining "multiple degrees of similarity" [12]. In their research, they were able to analyze text and observe relationships that could be illustrated by vector operations.

The Word2Vec vectorization method has been shown to be an effective way to derive meaning from a large corpus, and then use that meaning to show relationships between words. In the example: King - man + woman = Queen, when the vector representation for *man* is subtracted from the vector representation for *king* this new vector difference implies some sort of *monarch* meaning. When this *monarch* vector is added to the vector representation for *woman*, the new vector is roughly equivalent to the vector for *queen*. When converted to vectors and using vector operations, even word meaning in this framework obeys some of the rules of linear algebra [12,28]. Even concepts as esoteric as physical properties of molecules can be represented in such a manner (e.g. the word embeddings for "ferromagnetic − NiFe + IrMn ≈ antiferromagnetic" [9])

To begin this process, the vocabulary of the corpus is defined and its size determined, W [29]. This first vector, I, is a $W \times 1$ one-hot vector, where the

single 1 in this matrix represents the input word's position in the vocabulary list.[2] The product of the first matrix transformed, $I^\top$, and a second $W \times D$ matrix v, where D is the arbitrary dimensionality of the word embeddings, yields a $1 \times D$ vector which is the vector representation of the input word embedding, v_{w_i} for the input word w_i [29].

$$I^\top \cdot v = v_{w_i}$$

$$\begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \cdot \begin{bmatrix} v_{11} & v_{12} & v_{13} & \dots & v_{1D} \\ v_{21} & v_{22} & v_{23} & \dots & v_{2D} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ v_{W1} & v_{W2} & v_{W3} & \dots & v_{WD} \end{bmatrix} = \begin{bmatrix} I_1 \\ I_2 \\ \vdots \\ I_d \end{bmatrix}$$

The product of this vector, v_{w_i} and the $D \times W$ output word matrix, v' , gives a $W \times 1$ vector, v'_{w_0}. The input and output word vectors correspond to either center word and context words depending on mode (see II.2.2.1 Continuous Bag of Words and II.2.2.2 Skip-Gram under Word2Vec Parameters).

2 When possible, the equations and variables indicated throughout this paper were refactored and relabeled to coincide with the definitions in the works by Mikolov, et al [12,27]. Any apparent disparity between other cited works and the formulas illustrated in this paper are from the preference to unify representations based on consistency with the seminal works on Word2Vec.

$$\begin{bmatrix} I_1 \\ I_2 \\ \vdots \\ I_d \end{bmatrix} \cdot \begin{bmatrix} v'_{11} & v'_{12} & v'_{13} & \dots & v'_{1D} \\ v'_{21} & v'_{22} & v'_{23} & \dots & v'_{2D} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ v'_{W1} & v'_{W2} & v'_{W3} & \dots & v'_{WD} \end{bmatrix} = \begin{bmatrix} O_1 \\ O_2 \\ \vdots \\ O_n \end{bmatrix}$$

This v'_{w_0} vector is then softmax scaled. By performing this transformation, the resulting $1 \times W$ vector behaves similarly to a probability distribution. Its values, now all positive, are compared to the $1 \times W$ one-hot vector representing the output word, w_O, in the vocabulary. This relationship is illustrated in the following formula: [27]

$$p(w_O|w_I) = \frac{\exp\left({v'_{w_0}}^{\top} v_{w_i}\right)}{\sum_{w=1}^{W} \exp\left({v'_w}^{\top} v_{w_i}\right)} \tag{II.2}$$

Backpropagation occurs via stochastic gradient descent, and the process begins again with the next word within the context window. Once all context terms are processed within the word window for the center word, the process begins again with the next center word and its context words. The update functions are further discussed in II.2.2.6: Negative Sampling.

II.2.2. Word2Vec Parameters

II.2.2.1. Continuous Bag of Words

In the Word2Vec module, there are two different methods of training the vector model, and they are nearly opposites of each other. The first, Continuous Bag-of-Words, or CBOW, trains the neural network by using the context words as the input and the expected target word as the output. The intended use here is to predict a single word based upon an input of one or more context words.

II.2.2.2. Skip-Gram

The other method for training the Neural Network is the Skip-Gram model. In this model, the center word is the single input; the context words are the output. This model aims to predict context words based on a single word.

The neighboring words are also scored by their relative location to the center word, and weighted with a proportional function to emphasize a context word when it is closer to the center word. In this way, a context word that is directly adjacent to the center word carries more weight for context than a word that is a few positions away. "For example, a size-5 window will weigh its contexts by $\frac{5}{5}, \frac{4}{5}, \frac{3}{5}, \frac{2}{5}, \frac{1}{5}$." [28]

Both methods are built upon maximizing the probabilistic pairing of the correct word, w, with the correct context c. The difference comes from the conditional event notation: $P(w|c)$ indicates the CBOW relationship, while $P(c|w)$ indicates Skip-Gram, for any given word-context pair, (w, c).

II.2.2.3. Minimum Word Frequency

Word frequency can play an important role in analysis of large bodies of text. Setting a floor on the occurrences of a word below which it is ignored can prevent a word from being included in the vocabulary entirely. This can be important if a corpus contains jargon or slang that is not necessarily endemic to the work(s) in question. It is possible, however, that too aggressive of a floor on occurrence frequency could diminish some of the nuanced meaning desired by this study. Furthermore, wholly unique tweets could be eliminated from consideration entirely. This presents a problem both from a comprehensive standpoint, as a unique tweet may convey information essential to analysis, but also a programmatic problem, as tweets with zero tokens require special handling.

II.2.2.4. Word Window

The word window argument sets the maximum distance on either side of a center word where neighboring words are considered for context. For example, a word window of 3 would look both three words ahead and behind the center word to include any words found in the context part of the neural network construction. Though words outside of this window are considered to be part of the same document, words within the same document will share context words where the word windows overlap. For CBOW, these words are the input values for the neural network, and for Skip-Gram, these words are the output values.

II.2.2.5. Word Vector/Hidden Layer Dimensionality

As mentioned above in section II.2.1, the construction of the neural network is based upon inputs and outputs, but the internal weights are used as a representation for each of the word embeddings [29,30]. For the purpose of this project, the dimensionality of the word embedding vectors and the hidden layer of the neural network are equivalent, and the terminology will be used interchangeably. To correspond with the Gensim documentation, 'Hidden Layer Dimensionality' is represented by the argument `size` within the Gensim implementation of the Word2Vec function [30].

Negative Sampling

If all words in a vocabulary V are combined such that $\binom{V}{2}$ represents all possible word-context pairs, far more pairs exist than true word-context relationships within the training corpus. Assume for all valid word-context pairs (w, c) there exists an N such that $(w, c) \in N$ and an N' such that $(w, c) \notin N'$, and $N \cup N' = V$.

If the neural network is only trained on (w, c) pairs in N, then any single $(w, c) \in N$ has tremendous significance. And, as each (w, c) pair represents only a portion of the total number of contexts for any given word, updating each row of the input and output matrices can be costly in terms of processing. Instead, for each w, the parameter for the negative sampling function, k, indicates a choice of k negative values for c_{ns} such that $(w, c_{ns}) \notin N$. This limits the impact of any single (w, c) pair, and further discourages any recognition of a $(w, c) \in N'$, while minimizing the processing overhead.

Mikolov, et al, define the Negative Sampling objective function as follows, and this is used to "...replace every $\log P(w_O|w_I)$ term in the Skip-Gram

$$\log \sigma \left({v'_{w_O}}^{\top} v_{w_i} \right) + \sum_{i=1}^{k} \mathbb{E}_{w_i \sim P_n(w)} \left[\log \sigma \left(-{v'_{w_i}}^{\top} v_{w_i} \right) \right] \tag{II.3}$$

objective" (See: II.2). [27] This function seeks to "...distinguish the target word w_O from draws from the noise distribution $P_n(w)$ ...where there are k negative samples for each data sample." [27]

By default, the Gensim implementation of Word2Vec for Python uses a negative sampling value of 5, where the recommended range is 5-20 [27,20,32]. For each iteration of positive training on a word-context pair, the algorithm also selects a set of "noise words" [30] where the neural network associates these words negatively with the input layer.

As the objective for training involves numerous rows on both the input and output layers, the update equations must be similarly adjusted. Rong describes the update equation for the input to hidden layer for Skip-Gram below [29].

$$v_{w_i}^{(\text{new})} = v_{w_i}^{(\text{old})} - \eta \cdot \text{EH}^{\top} \tag{II.4}$$

Here EH is a vector representing the summation of the prediction errors, and η represents the learning rate.

For the Negative Sampling aspect of Skip-Gram, the update equations for the hidden to output layers is described here [29].

$$v'_{w_j}{}^{(\text{new})} = v'_{w_j}{}^{(\text{old})} - \eta \left(\sigma \left(v'_{w_j}{}^{\top} h\right) - t_j\right) h \tag{II.5}$$

The equation above "...only needs to be applied to $w_j \in \{w_O\} \cup \mathcal{W}_{\text{neg}}$ [.]" [29] To address the additional functionality of the Negative Sampling, Rong expands the variables indicated earlier with the following description: "$\mathcal{W}_{\text{neg}} = \{w_j | j = 1, \cdots, K\}$ is the set of words that are based on $P_n(w)$, i.e., negative samples... t_j is the 'label' of word w_j. $t = 1$ when w_j is a positive sample; $t = 0$ otherwise." [29] Furthermore, "...j [is] the subscript for...output layer units" [29] and "...h is simply copying (and transposing) a row of the input $\rightarrow$ hidden weight matrix...associated with the input word w_i." [29]

II.3. Scalar Comparison Formulas

After training, the Word2Vec neural network produces vectors for terms but not tweets. For the results of this analysis to be compatible with the other scoring mechanisms within the encompassing study (see: I.3), a single scalar value would need to be determined for each tweet. The following formulas were used to derive a scalar score for the tweet from an amalgamation of the component term vectors. In the initial testing, each formula was executed in tandem, and the equations would be used to compare the effect of variation in the parameters.

For purposes of consistency, and to distinguish from previous terminology, new symbols will be used for the components necessary for these comparisons. The symbol a designates the initial search or *seed* term, the basis of all comparisons for these formulas. The symbol τ will refer to a token contained within a processed tweet, where τ_i indicates one of many such tokens in any given tweet.

II.3.1. Cosine Similarity From Cosine Distance of One Dimensional Arrays (CSTVS)

$$1 - \frac{\alpha \cdot \sum_{i=1}^{k} \tau_i}{\|\alpha\| \|\sum_{i=1}^{k} \tau_i\|} \tag{II.6}$$

The SciPy `spatial.distance` library has a built-in function for cosine distance between two 1D arrays, interpreted as vectors. In this function, a separate $1 \times D$ zero matrix is initialized, with D as the dimensionality of the word vectors. In this formula (as with II.7) this new matrix is calculated as the summation of the word vectors for each tweet. Using this $1 \times D$ matrix as a vector itself, the cosine distance between the matrix-as-vector and the word vector for the seed term *irma* is calculated. Cosine distance can be further converted to cosine similarity by subtracting from one.

This formula was selected to leverage the efficiency of optimized pre-generated code over other possible functions. If the performance of this scoring mechanism proved to be nearly equivalent to others of the formulas, then it could be evaluated on the basis of resource and time consumption.

II.3.2. Dot Product of Search Term Vector and Tweet Vector Sum (DP)

$$\|\alpha\| \times \|\sum_{i=1}^{n} \tau_i\| \times \cos\theta \tag{II.7}$$

Cosine similarity (see: Equation I.1) is proportional to the dot product of two vectors. It has been observed within the vector constructs for Word2Vec that vector operations, such as addition and subtraction, yield meaning [12,28]. This was used as the predicate for interpreting the meaning of a tweet as the sum of its component word vectors. Summation of all of the token vectors, τ_i, within a tweet returned a vector itself in the same dimensionality as, and therefore could be compared to, the vector for the seed term *irma*, a, via the cosine similarity of the two. Using Equation: II.7 gives a scalar value for the tweet comprised of related word vectors.

II.3.3 Mean Cosine Similarity of Tweet Terms in Vector Vocabulary (MCS)

$$\frac{1}{n}\sum_{i=1}^{n} \tau_i \tag{II.8}$$

For this process, after tokenization and cleaning, each remaining token, τ_i, in each tweet was scored based upon its cosine similarity (see: Figure II.8) to the seed term *irma*. If a term was not present in the vocabulary, due to minimum word count or other restricting criteria, the term was given a zero, which evaluates to a neutral context relation due to cosine similarity. The mean of all cosine similarity values for tokens τ within the tweet, including zeroes, was calculated,

and this value was designated as the score for the tweet. As stated in section: I.2.3, Benton, et al used a similar mechanism for representing an aggregation of tweets and their metadata within each "view". [26]

II.3.4 Sum of Cosine Similarity of Tokens Over Square Root of Token Count (SCSSC)

$$\frac{1}{\sqrt{n}} \sum_{i=1}^{n} \tau_i \tag{II.9}$$

Like the equation in II.8, this formula scores a tweet based upon a summation of the tweet's component token vectors. However, the scalar value calculated in II.8 could disproportionately favor shorter tweets, as each token would contribute a greater proportion of the score. In an attempt to minimize the impact of word count in any given tweet, the mean operation was replaced by dividing by the *square root* of the word count.

CHAPTER III

EXPERIMENTATION AND EVALUATION

III.1. Methods

III.1.1. Human-Coded Tweets

There were 19088 tweets in this dataset for the time period 2017-09-10 00:00 GMT through 2017-09-11 00:00 GMT, inclusive. These tweets were human-coded for relatedness to Hurricane Irma. For purposes of identifying relatedness, a tweet whose context was interpreted by a human reader as being associated with Hurricane Irma was given a boolean *True* value. The presumption being: tweets whose content implied the composer's present awareness of the hurricane would contain context sensitive terms as well as location based metadata. Both are considered essential to the studies associated with this research.

These tweets were further isolated to exclude non-English content. While the functionality of the training mechanism is sufficiently language agnostic, words that are *interlingual homographs* could potentially alter context for a particular spelling (i.e. *done*, an adjective indicating a completed state in English, is also the first-person singular present subjunctive form of *donar*, to donate, in Spanish.)

To ensure independence between human-coded data and the training mechanism, the value for human-coding was not introduced into the neural network as a feature during training. This attribute was only used when evaluating

the effectiveness of a particular scoring formula, and to assess the impact of variation on a parameter. See III.1.4 for details on the AU-ROC score.

III.1.2. Cleaning and Tokenization

To account for changes in the vocabulary size before and after the various transformations, an initial operation to split solely on whitespace characters was performed. The remaining unprocessed tokens were grouped and counted, leaving 43387 tokens in the vocabulary. The following table (III.1) shows the twenty most frequent tokens and their counts prior to any transformations.

Table III.1. Pre-Transformation Count of Tokens in Tweets

token	**count**
the	4900
I	4133
to	3853
@	3337
a	3020
in	2998
and	2843
of	2796
is	2619
for	1977
my	1943
s	1772
you	1647
Florida	1592
this	1572
on	1491
t	1357
from	1236
it	1202
at	1129

The first transformation performed was the `reduce_lengthening` functionality mentioned in II.1.1. This function reduced the total number of

tokens to 43254. While this represents a reduction of only .31% such a reduction is essential. Any superfluous tokens decrease the effectiveness of training; two (or more) words whose existence otherwise would be treated as identical, but whose spelling is only separated by the quantity of a character, and therefore completely different, dilutes the likelihood of the neural network recognizing their syntactic equivalence.

The second transformation operation is detailed in Section: II.1.2. Once these operations were performed, the quantity of tokens left in the vocabulary was 14439, a reduction of 66.9%. The list in Table: III.2 shows the top twenty words ordered by count after the combined transformations. When compared with the initial list in Table: III.1 it is immediately apparent that case-sensitivity is significant in minimizing vocabulary. In the first table, *Florida* occurs 1592 times. After cleaning, *florida* appears 1809 times and is the most frequently used word. Note: the incident of the word *hurrican* could likely be attributed to misspelling, but also to the effect of the stemmer function (i.e. truncating both *hurricane* and *hurricanes* to their root.)

Table III.2. Post-Transformation Count of Tokens in Tweets

token	**token count**
florida	1809
#hurricaneirma	1623
fl	1587
irma	1374
hurrican	1360
#irma	1193
wind	946
get	936
report	886
go	830
storm	775
power	715
miami	705
rain	682
mph	661
like	657
beach	656
gust	655
safe	633
aso	544

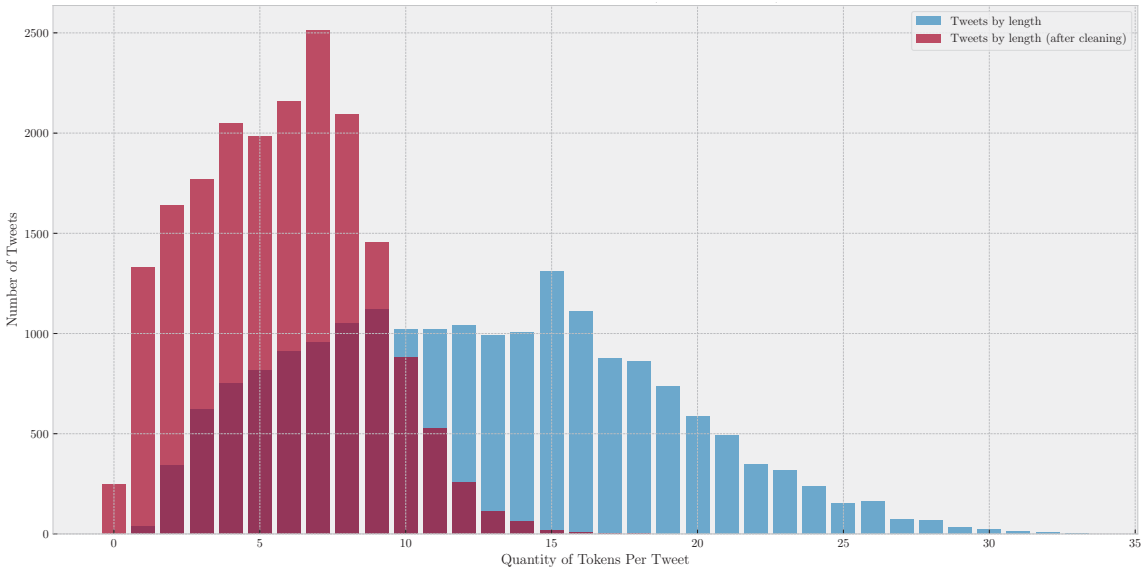

Figure III.1. Histogram of Tweets by Length (Token Quantity)

The graph in III.1 shows the quantity of tweets by number of tokens before and after processing.

The tweets had a maximum length of 33 tokens, separated by whitespace characters, prior to cleaning and tokenization. The performance of these operations reduced the maximum number of tokens in a tweet to 20. 15971 of the 19088 tweets, or 83.7%, contained 10 or fewer tokens.

III.1.3. Universal Constraints

With the stated purpose of interpreting content related to a single search term, the Skip-Gram mode was selected for training. By prioritizing a mode where word-predicts-context, it is hoped that the similarity of two single words could be compared based upon the syntactic equivalence of their contexts. The potential groups of terms that would predict individual target words would be comprised of temporally relevant, yet likely unknown contexts, and therefore

CBOW might not be as effective, i.e. the desired result of this operation is generative, not reductive. Therefore, the mechanism that derives many output contexts from a single word fits the preferred mode of training.

Additionally, for purposes of consistency, one other element was held constant through all the first sets of tests: the number of epochs for training. Since the first rounds of testing were to suggest ranges for a later grid search, and not for optimal settings, it was decided to prioritize minimal training time over accuracy. As such, all experimental neural networks were trained through 10 epochs for each iteration of the parameter being tested. The Python `timeit` was used to calculate the time efficiency on training the neural network once optimal parameters are determined.

III.1.4. Comparison Metrics

To compare the effectiveness of changing a particular parameter, the vector model was created using at least ten different values for that parameter (with all other parameters held constant), and the results of the scalar comparison function compared via the Area Under Receiver Operating Characteristics curve (AU-ROC) value. The Receiver Operating Characteristics curve is a function comparing the relative rates of increase of true-positives versus false-positives as the values corresponding to an independent threshold are increased. The area under this curve is calculated as a real number value between 0 and 1, with 1 implying a perfect recognition of true-positives, and a value of .5 indicating performance roughly equivalent to random selection [11]. If the initial trial range(s) for the parameter would indicate a monotonic relationship between the parameter and AU-ROC, a new range of values were considered for the parameter, and the tests were performed again.

While each test of a range for a parameter shows its impact on a set of defaults, it does not properly show how the combined variability of two (or more) parameters affects performance. In order to compensate for these compositions, an iterative grid search test of all variables was performed. This comprised of testing the cross-product of all ranges for the parameters, along with testing a range of training epochs, each as a discrete set of parameters for training the neural network, with the AU-ROC used as the scoring parameter. See IV.1.2 for results of the Grid Search.

CHAPTER IV

RESULTS AND DISCUSSION

IV.1. Selection of Scalar Formulas

The initial sets of tests compared the AU-ROC of each scalar formula as applied to tweets relative to the search term: *irma*. Each iteration of the testing involved training the neural network with default values for each parameter, isolating one parameter and determining a window which contained a local maximum for AU-ROC.

IV.1.1. Tuning Parameters

IV.1.1.1. Word Window

The initial test for the Word Window Size parameter variability set a ceiling at 10 tokens on either side of the center word. The other parameters were set at constants: minimum word count 1, word vector dimensionality 100, negative sampling 5, and using the Skip-Gram model. As stated in section III.1.2, the maximum token count for a tweet within this data set was 20. A word window value of 10 as the upper bound for the testing range ensured that all center words were provided at least half of the encompassing tweet as context. This also ensured that any given word potentially had the entire tweet as context for 83.7% of tweets. See III.1 for the distribution of tweets by length.

Table IV.1. AU-ROC of Word Window Size Values 1 to 10

Formula	1	2	3	4	5	6	7	8	9	10
CSTVS	0.706	0.708	0.724	0.739	0.734	0.737	0.748	0.754	0.750	0.755
DP	0.814	0.811	0.816	0.817	0.818	0.819	0.821	**0.822**	0.821	0.821
MCS	0.512	0.548	0.588	0.619	0.617	0.630	0.649	0.657	0.654	0.659
SCSSC	0.717	0.724	0.742	0.758	0.760	0.770	0.781	0.784	0.783	0.790

For word window values 1 through 10 in Table: IV.1, the four scalar comparison formulas have a maximum observed AU-ROC at window size 8 for the Dot Product formula II.7. While the difference in scores was negligible, it did indicate a trend towards a local maximum, therefore further tests were not performed.

IV.1.1.2. Minimum Word Frequency

Testing Minimum Word Frequency presented a different problem than most of the other parameter tests. By setting a threshold on frequency, it would be possible for a tweet to be comprised entirely of words that would not exist in the vocabulary of the vector sets. With the scalar comparison formulas dependent on the cosine similarity of a term and the search term, if a vector did not exist, it is possible for some of the tweets to end up with component elements in the denominator equal to zero. This required additional error handling in the code representing the scoring formulas.

Variation in Minimum Word Frequency also affected the maximums for each scalar comparison formula differently. With each of the other parameters, the maximum AU-ROC score consistently correlated with the same value for all scalar comparison formulas (e.g. the optimal value for Word Window Size,

8, corresponded to a maximum AU-ROC for all four formulas, See: IV.1). With Minimum Word Frequency, the optimal value for three of the four formulas was 8. However, for the Dot Product formula, the optimum value for Minimum Word Frequency was 3.

Table IV.2. AU-ROC of Minimum Word Frequency Values 0 to 9

Formula	0	1	2	3	4	5	6	7	8	9
CSTVS	0.736	0.733	0.740	0.744	0.732	0.735	0.743	0.745	0.751	0.728
DP	0.818	0.816	0.827	**0.829**	0.824	0.823	0.827	0.826	0.828	0.818
MCS	0.629	0.625	0.626	0.634	0.623	0.634	0.642	0.641	0.656	0.632
SCSSC	0.767	0.759	0.781	0.794	0.789	0.793	0.800	0.799	0.806	0.790

IV.1.1.3. Word Vector/Hidden Layer Dimensionality

As with the previous tests, the Dot Product formula (see: II.7) indicated the best performance for scoring a tweet. Changes in vector dimensionality yielded minimal performance changes, as indicated in Table IV.3. All formulas performed best with a dimensionality of 150, though the change from the default 100, showed little appreciable difference in the results.

Table IV.3. AU-ROC of Hidden Layer Dimensionality Values 50 to 500

Formula	50	100	150	200	250	300	350	400	450	500
CSTVS	0.744	0.754	0.755	0.751	0.749	0.749	0.748	0.747	0.746	0.746
DP	0.816	0.822	**0.823**	0.822	0.822	0.822	0.821	0.821	0.821	0.821
MCS	0.654	0.657	0.657	0.652	0.650	0.649	0.648	0.647	0.646	0.645
SCSSC	0.779	0.784	0.784	0.781	0.779	0.779	0.777	0.777	0.776	0.775

IV.1.1.4. Negative Sampling

The initial test of the negative sampling set out to compare the effectiveness of increased numbers of negatively sampled terms. The default value of 5 seemed to have minimal impact on the AU-ROC score. However, this test showed one of the more dramatic outliers for AU-ROC score over all tests of parameters. Changes from one value to the next for all parameter tests were measurable, but the variation rarely exceeded .02 in the subsequent calculation of AU-ROC. The difference between 0 and 1 for the negative sampling value showed a substantial increase from 0.560 to 0.854 for the Dot Product Formula: II.7. Similar increases were noted for the other scalar comparison formulas. The 0.854 for the Dot Product formula below also represents the highest AU-ROC score for all parameter tests. The remaining AU-ROC values for 2 through 9 negatively sampled words were also greater than the corresponding value for 0. This indicated that including a minimal number of negative context words in the training has an overall positive effect on the accuracy of the neural network.

Table IV.4. AU-ROC of Negative Sampling Values 0 to 9

Formula	0	1	2	3	4	5	6	7	8	9
CSTVS	0.564	0.771	0.745	0.748	0.749	0.754	0.753	0.760	0.752	0.756
DP	0.560	**0.854**	0.829	0.826	0.822	0.822	0.819	0.820	0.817	0.816
MCS	0.561	0.717	0.678	0.665	0.653	0.657	0.649	0.658	0.643	0.645
SCSSC	0.560	0.811	0.786	0.782	0.783	0.784	0.783	0.788	0.783	0.784

IV.1.2. Optimized Parameters and Grid Search

Once ranges containing a local maximum on the AU-ROC score were determined, these ranges were used as the testing values of a Grid Search, with one alteration. With minimal initial impact seen by variability in Hidden Layer Dimensionality, only vectors of 100D and 150D were tested. Below is the table of the top performing permutations of parameters.

Table IV.5. Grid Search Parameter Results

AU-ROC	HLD	MWF	WWS	NS	EP	SF
0.887560	150	5	1	1	25	DP
0.886191	100	5	1	1	25	DP
0.881556	150	3	1	1	25	DP
0.879418	150	7	1	1	25	DP
0.879235	150	6	1	1	25	DP
0.878688	150	8	1	1	25	DP
0.878547	100	6	1	1	25	DP
0.878196	100	3	1	1	25	DP
0.877670	100	7	1	1	25	DP
0.877067	150	9	1	1	25	DP

As expected, the Dot Product (DP) II.7 scalar formula performed the best overall. The Negative Sampling (NS) parameter value also reflected the observations in initial testing; a value of 1 was clearly optimal for this training. Another expected outcome was the apparent negligible impact in using 100D versus 150D for Hidden Layer Dimensionality (HLD).

The remainder of the parameters appeared to deviate somewhat from the values seen as local maximums in the initial testing. Minimum Word Frequency

(MWF) and Word Window Size (WWS) were apparently affected by the simultaneous adjustment of other parameters, as well as being somewhat more influenced by the number of training epochs (EP).

The violin plot below (IV.1) shows the distributions of AU-ROC scores for each of the four scalar formulas. The two halves of each distribution correspond to the two values tested for Hidden Layer Dimensionality.

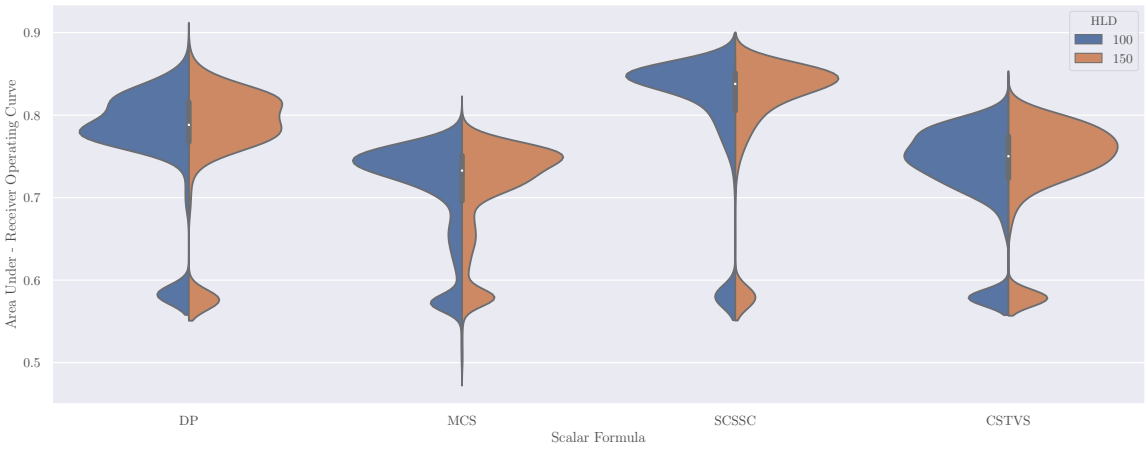

Figure IV.1. Effect of Scalar Formula on AU-ROC

The Dot Product (DP) II.7 scalar formula shows a higher overall maximum, although with slightly greater variance, when compared to the Sum of Cosine Similarity of Tokens over Square Root of Token Count (SCSSC) II.9.

IV.1.2.1. AU-ROC of Scalar Comparison Formulas

Using the neural network trained with optimal parameters, the tweets were again scored and their AU-ROC curves created. Figure IV.2 shows the scalar comparison formulas both with optimal parameters (indicated by **(O)** and solid lines)

and default parameters (indicated by **(D)** and dotted lines), color-matched, with a reference line for the .5 AU-ROC threshold. As was indicated in previous tests, the Dot Product (DP) formula proved to be the most effective and consistent method for scoring a tweet. The Mean Cosine Similarity score seemed the least effective, but somewhat more consistent than the Cosine Similarity of Tweet Vector Sum (CSTVS). It is worth noting that dividing by the square root of the tweet length (SCSSC) proved to be a significant improvement over the simple mean.

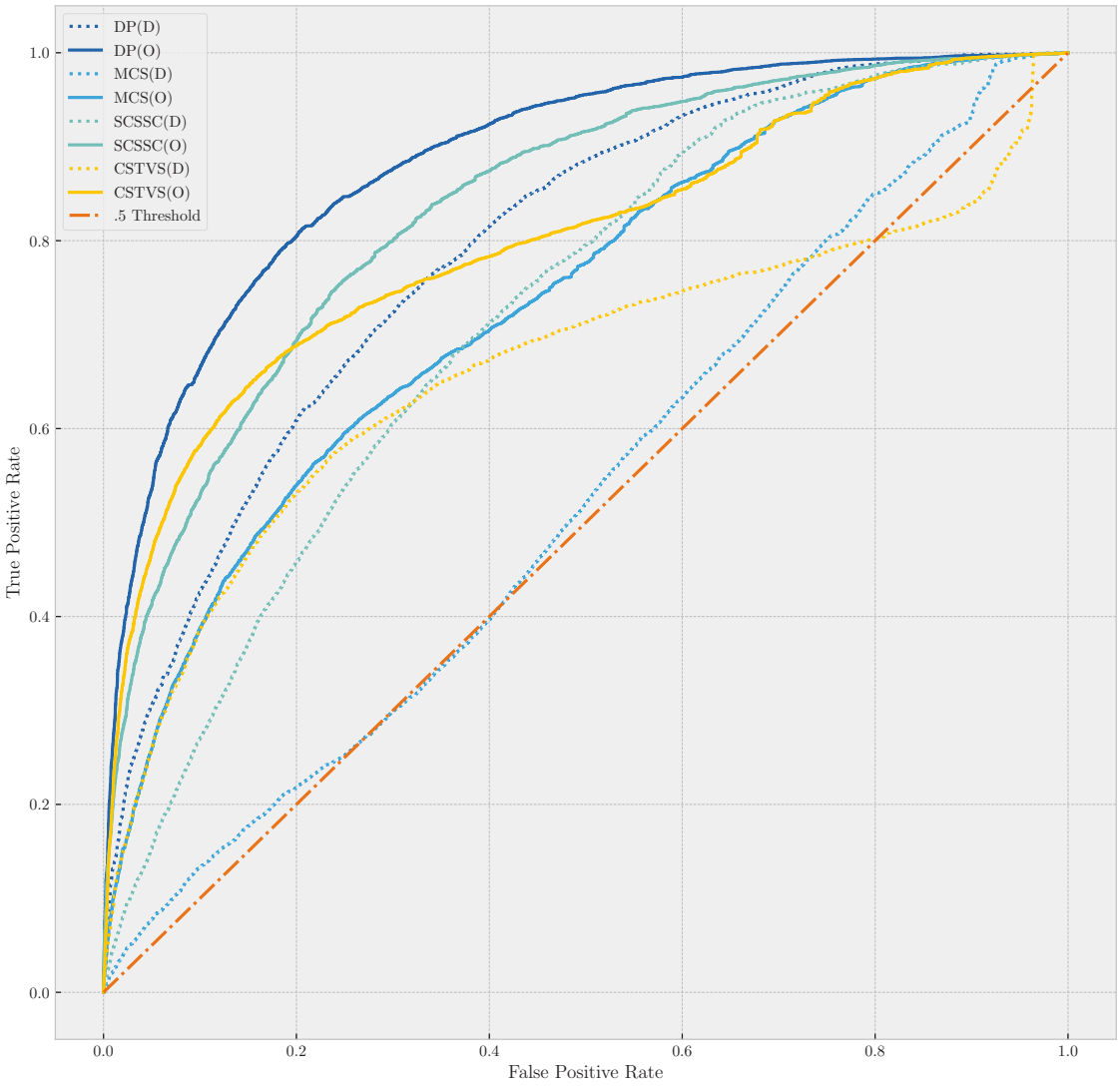

Figure IV.2. AU-ROC of Scalar Comparison Formulas

IV.1.3. Dynamic Relatedness

IV.1.3.1. Word Lists Per Hour

For the tables in Appendix: Related Word Lists A, each column represents one hour within in the 24-hour period starting at 00:00 GMT on September 10, 2017. For each hour, the Word2Vec neural network is trained on only the tweets that occur during that period, using the optimal parameter configuration determined by the grid

search IV.1.2 above. The list of words represent the top twenty most similar by cosine similarity in descending order as compared to the search term: “irma”.

Some of the interesting observations come from interpreting the possible context and reasoning for why certain terms are positioned in lists at particular times. For example, the word *shelter* appears in various locations throughout the lists. Perhaps more interestingly, it is the top word of the hourly list at the time of Hurricane Irma’s landfall, and the top word for the subsequent three hours. And *landfal*, the stem of landfall and landfalls, only appears once: during the landfall hour. The word *tomorrow* appears four times in the five hours, 00:00 – 04:00. Since local time is UTC-4, these hours correspond with 8:00PM – midnight on the day previous to landfall. *Tomorrow* does not appear on the lists for related words on the day of landfall.

The word *ese* presents another interesting linguistic observation. While this word has colloquial meaning in Spanish, its appearance in these lists is indicative of another meaning. Searching the graph of word communities (See: IV.1.3.2.), *ese* is found in a group of weather terms. By isolating the training to English only tweets, the meaning appears to have tended toward ESE, an abbreviation for *East by Southeast*. In this context, the probability of this particular interlingual homograph was higher when considering the direction from which the hurricane approached. Furthermore, when looking at the hourly lists of words, it appeared in the top four words in each of the four hour lists prior to landfall; only once in the lists prior to that; and never in the hours afterward.

Another word that has a fascinating set of positions on this list is the word *safe*. It appears only once in the twelve hourly lists prior to landfall, at the bottom of the 08:00AM UTC list. However, it appears seven times in the eleven hourly lists after landfall.

IV.1.3.2. Graphs of Word Communities

For the graph depicted in Appendix IV.3, each word is connected to terms based on cosine similarity. The edges in this graph represent values for cosine similarity greater than $\cos(45)$ or $\approx .7071$. This value was chosen as a lower bound on vector representation of similarity, as included values would be closer to coincident than orthogonal. The nodes are subjected to a gravity algorithm to encourage similar terms to cluster, and dissimilar terms to repel each other. The edges in this graph represent the cosine similarity between the vectors that represent the word embeddings of the words in the nodes. Each node's relative size is proportional to the related token's PageRank score.

In the graph, sections separated by color are designated based upon Louvain Modularity. The communities that formed depict topics, with some highlights in the figures below. For example, in Figure IV.4, there is the topic of famous Florida attractions as represented by the words: Magic Kingdom, Walt Disney World, Harry Potter's Wizarding World, and Hollywood Studios. Similarly, in Figure IV.5, there appear to be weather related words associated with windspeed (41mph, 80mph), pressure (994mb, 1002mb, baromet[ric], pressur[e]), weather phenomena (thunderstorm, gust, funnel, squall, drizzl[e], rain, mist, humid[ity]), measurements of compass direction (e, ese, sw, ene, nne), and terms of scale (light, heavi[est], moder[ate], intens[e]).

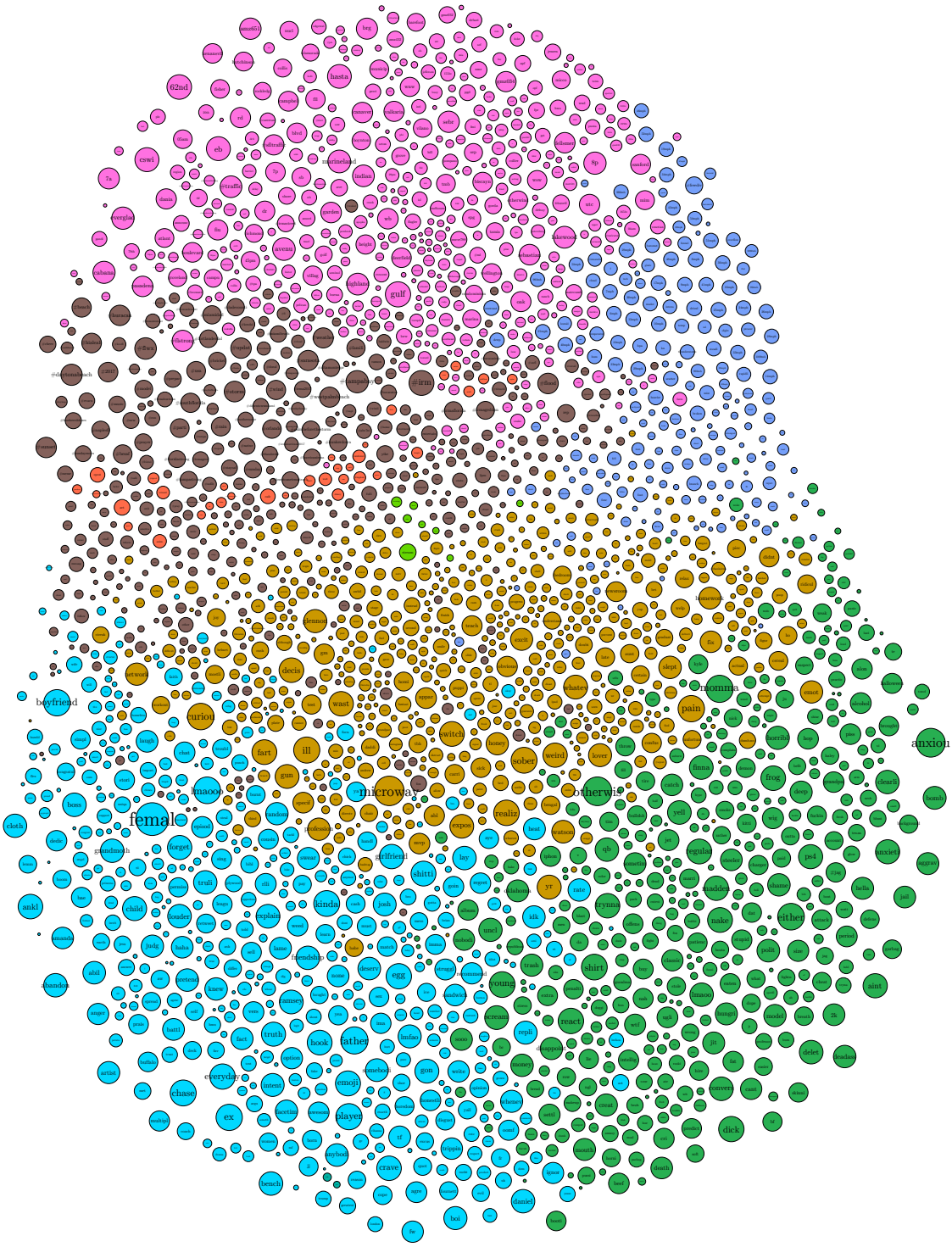

Figure IV.3. Graph of Topic Communities

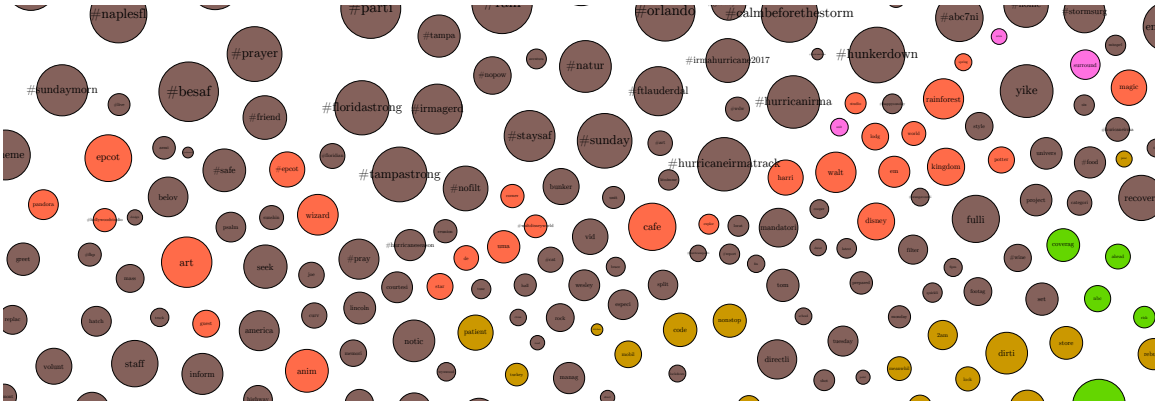

Figure IV.4. Graph of Topic Communities: Attractions

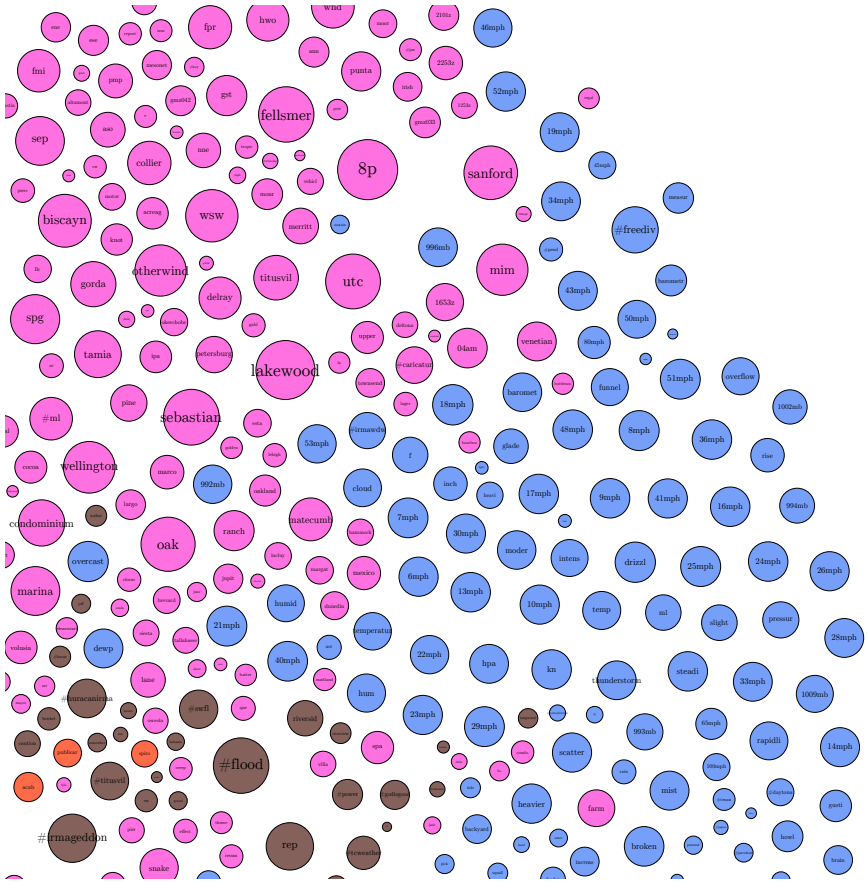

Figure IV.5. Graph of Topic Communities: Weather

CHAPTER V

CONCLUSION AND FUTURE WORK

For this paper, it is proposed that a regionally and temporally coincident corpora comprised of the text of tweets surrounding an emergency event provide a good basis for a dynamic syntactic construct. This construct can then be used to widen and improve results based upon a single search term, where many of these results may be omitted. This construct can also be used to infer meaning and significance by reviewing lists of related words based upon the corpora at hand.

Over the course of this research, a number of opportunities for future work presented themselves based upon the results of this study. While this methodology shows promise, it has not been performed at scale. Some of the next steps may be to pursue these tests on larger sets of tweets, with similar constraints on time and region. It also may be employed to study the transformation of a conversation topic as designated by a search term, and comparing this to other well known topic modeling methods, such as LSI and LDA. It may also be compared to other dimensionality reduction techniques such as PCA and SVD.

BIBLIOGRAPHY

[1] Twitter, "Q4 and Fiscal Year 2018 Letter to Shareholders," Tech. Rep., Feb. 2019. [Online]. Available: https://s22.q4cdn.com/826641620/files/doc_financials/2018/ q4/Q4-2018-Shareholder-Letter.pdf

[2] Internet Live Stats - Internet Usage & Social Media Statistics. [Online]. Available: https://www.internetlivestats.com/

[3] P. A. Longley and M. Adnan, "Geo-temporal Twitter demographics," *International Journal of Geographical Information Science*, vol. 30, no. 2, pp. 369–389, Feb. 2016. [Online]. Available: http://www.tandfonline.com/doi/full/10.1080/ 13658816.2015.1089441

[4] X. Liu, B. Kar, C. Zhang, and D. M. Cochran, "Assessing relevance of tweets for risk communication," *International Journal of Digital Earth*, pp. 1–21, Jun. 2018. [Online]. Available: https://www.tandfonline.com/doi/full/10.1080/17538947. 2018.1480670

[5] J. P. Cangialosi, A. S. Latto, and R. Berg, "Hurricane Irma," National Oceanic and Atmospheric Administration U.S. Department of Commerce, Tech. Rep. AL112017, Jun. 2018. [Online]. Available: https://www.nhc.noaa.gov/data/tcr/ AL112017_Irma.pdf

[6] U. S. N. H. Center, "Costliest U.S. Tropical Cyclones Tables Update," National Oceanic and Atmospheric Administration, Tech. Rep., Jan. 2018. [Online].Available: https://www.nhc.noaa.gov/news/UpdatedCostliest.pdf

[7] S. Arora, Y. Li, Y. Liang, T. Ma, and A. Risteski, "RAND-WALK: A Latent Variable Model Approach to Word Embeddings," *arXiv:1502.03520 [cs, stat]*, Feb. 2015, arXiv: 1502.03520. [Online]. Available: http://arxiv.org/abs/1502.03520

[8] X. Yang, C. Macdonald, and I. Ounis, "Using Word Embeddings in Twitter Election Classification," *arXiv:1606.07006 [cs]*, Jun. 2016, arXiv: 1606.07006. [Online]. Available: http://arxiv.org/abs/1606.07006

[9] V. Tshitoyan, J. Dagdelen, L. Weston, A. Dunn, Z. Rong, O. Kononova, K. A. Persson, G. Ceder, and A. Jain, "Unsupervised word embeddings capture latent knowledge from materials science literature," *Nature*, vol. 571, no. 7763, pp. 95–98, Jul. 2019. [Online]. Available: http://www.nature.com/articles/s41586-019-1335-8"

[10] Y. Xu, B. C. Malt, and M. Srinivasan, "Evolution of word meanings through metaphorical mapping: Systematicity over the past millennium," *Cognitive Psychology*, vol. 96, pp. 41–53, Aug. 2017. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0010028517300324

[11] M. Imran, P. Mitra, and C. Castillo, "Twitter as a Lifeline: Human-annotated Twitter Corpora for NLP of Crisis-related Messages," *arXiv:1605.05894 [cs]*, May 2016, arXiv: 1605.05894. [Online]. Available: http://arxiv.org/abs/1605.05894

[12] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient Estimation of Word Representations in Vector Space," *arXiv:1301.3781 [cs]*, Jan. 2013, arXiv: 1301.3781. [Online]. Available: http://arxiv.org/abs/1301.3781

[13] F. Alam, F. Ofli, and M. Imran, "CrisisMMD: Multimodal Twitter Datasets from Natural Disasters," p. 9. [Online]. Available: https://arxiv.org/pdf/1805.00713.pdf

[14] N. Savage, "Twitter as medium and message," *Communications of the ACM*, vol. 54, no. 3, p. 18, Mar. 2011. [Online]. Available: http://portal.acm.org/citation.cfm?doid=1897852.1897860

[15] R. Soden and L. Palen, "Informating Crisis: Expanding Critical Perspectives in Crisis Informatics," *Proceedings of the ACM on Human-Computer Interaction*, vol. 2, no. CSCW, pp. 1–22, Nov. 2018. [Online]. Available: http://dl.acm.org/citation.cfm?doid=3290265.3274431

[16] C. Reuter, A. L. Hughes, and M.-A. Kaufhold, "Social Media in Crisis Management: An Evaluation and Analysis of Crisis Informatics Research," *International Journal of Human-Computer Interaction*, vol. 34, no. 4, pp. 280–294, Apr. 2018. [Online]. Available: https://www.tandfonline.com/doi/full/10.1080/10447318.2018.1427832

[17] B. Chan, A. Lopez, and U. Sarkar, "The Canary in the Coal Mine Tweets: Social Media Reveals Public Perceptions of Non-Medical Use of Opioids," *PLOS ONE*, vol. 10, no. 8, p. e0135072, Aug. 2015. [Online]. Available: https://dx.plos.org/10.1371/journal.pone.0135072

[18] A. Culotta, "Lightweight methods to estimate influenza rates and alcohol sales volume from Twitter messages," *Language Resources and Evaluation*, vol. 47, no. 1, pp. 217–238, 2013. [Online]. Available: https://www.jstor.org/stable/42637261

[19] R. M. Merchant, S. Elmer, and N. Lurie, "Integrating Social Media into Emergency-Preparedness Efforts," *New England Journal of Medicine*, vol. 365, no. 4, pp. 289–291, Jul. 2011. [Online]. Available: http: //www.nejm.org/doi/abs/10.1056/NEJMp1103591

[20] M. Poblet, E. García-Cuesta, and P. Casanovas, "Crowdsourcing roles, methods and tools for data-intensive disaster management," *Information Systems Frontiers*, vol. 20, no. 6, pp. 1363–1379, Dec. 2018. [Online]. Available: https://doi.org/10.1007/s10796-017-9734-6

[21] S. Gunessee, N. Subramanian, S. Roscoe, and J. Ramanathan, "The social preferences of local citizens and spontaneous volunteerism during disaster relief operations," *International Journal of Production Research*, vol. 56, no. 21, pp. 6793–6808, Nov. 2018. [Online]. Available: https: //doi.org/10.1080/00207543.2017.1414330

[22] J. B. Houston, J. Hawthorne, M. F. Perreault, E. H. Park, M. Goldstein Hode, M.R. Halliwell, S. E. Turner McGowen, R. Davis, S. Vaid, J. A. McElderry, and S. A. Griffith, "Social media and disasters: a functional framework for social media use in disaster planning, response, and research," *Disasters*, vol. 39, no. 1, pp. 1–22, Jan. 2015. [Online]. Available: http://doi.wiley.com/10.1111/disa.12092

[23] C. Schöch, "Big? Smart? Clean? Messy? Data in the Humanities" *Journal of Digital Humanities*, vol. 2, no. 3, 2013. [Online]. Available: http://journalofdigitalhumanities.org/2-3/big-smart-clean-messy-data-in-the-humanities/

[24] E. Aslan and C. Vásquez, "'Cash me ousside': A citizen sociolinguistic analysis of online metalinguistic commentary," *Journal of Sociolinguistics*, vol. 22, no. 4, pp. 406–431, Sep. 2018. [Online]. Available: http://doi.wiley.com/10.1111/josl.12303

[25] T. Pedersen, S. V. Pakhomov, S. Patwardhan, and C. G. Chute, "Measures of semantic similarity and relatedness in the biomedical domain," *Journal of Biomedical Informatic*s, vol. 40, no. 3, pp. 288–299, Jun. 2007. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S1532046406000645

[26] A. Benton, R. Arora, and M. Dredze, "Learning Multiview Embeddings of Twitter Users," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Berlin, Germany: Association for Computational Linguistics, 2016, pp. 14–19. [Online]. Available: http://aclweb.org/anthology/P16-2003

[27] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed Representations of Words and Phrases and their Compositionality," p. 9. [Online]. Available: https://papers.nips.cc/paper/ 5021-distributed-representations-of-words-and-phrases-and-their-compositionality. pdf

[28] O. Levy, Y. Goldberg, and I. Dagan, "Improving Distributional Similarity with Lessons Learned from Word Embeddings," *Transactions of the Association for Computational Linguistics*, vol. 3, pp. 211–225, Jun. 2015.

[29] X. Rong, "word2vec Parameter Learning Explained," *arXiv:1411.2738 [cs]*, Nov. 2014, arXiv: 1411.2738. [Online]. Available: http://arxiv.org/abs/1411.2738

[30] R. Řehůřek, "gensim: topic modelling for humans," Apr. 2019. [Online]. Available: https://radimrehurek.com/gensim/models/word2vec.html

[31] Y. Goldberg and O. Levy, "word2vec Explained: deriving Mikolov et al.'s negative-sampling word-embedding method," arXiv:1402.3722 [cs, stat], Feb. 2014, arXiv: 1402.3722. [Online]. Available: http://arxiv.org/abs/1402.3722

[32] R. Řehůřek and P. Sojka, "Software Framework for Topic Modelling with Large Corpora," in *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*. Valletta, Malta: ELRA, May 2010, pp. 45–50, http://is.muni. cz/publication/884893/en.

APPENDIX A

RELATED WORD LISTS

Table A.1. Related Words 00:00 UTC – 05:00 UTC

00:00	01:00	02:00	03:00	04:00	05:00
tampa	whole	time	shelter	tampa	time
last	tampa	beauti	shift	tri	night
shelter	last	let	guess	time	make
make	read	storm	tri	made	friend
night	outsid	night	last	yet	alway
beauti	check	tomorrow	pet	whole	sleep
yet	ese	see	sleep	night	need
close	made	tri	outsid	tomorrow	want
help	tri	shelter	time	mom	world
outsid	tomorrow	#irma	check	friend	fuck
hit	time	like	strong	outsid	wind
whole	sleep	hurrican	made	make	watch
tri	night	last	cuba	alway	boy
#irmahurrican	make	make	help	sleep	nigga
time	yet	still	saturday	open	see
move	food	watch	night	could	pleas
ago	footbal	place	dawg	hit	wait
tomorrow	might	need	yet	great	beach
wait	lake	tampa	eye	want	tonight
#hurricanirma	spend	person	school	need	#irma

Table A.2. Related Words 06:00 UTC – 12:00 UTC

06:00	07:00	08:00	09:00	10:00	11:00	12:00
tri	sleep	last	ese	ese	tampa	tampa
time	offici	outsid	outsid	tri	yet	time
night	need	sleep	moder	help	time	check
close	e	heavi	valkaria	close	ese	ese
outsid	want	e	sleep	outsid	eye	tri
sleep	hit	#key	nation	eye	friend	might
alway	#key	wind	e	moder	first	night
need	wind	tropic	need	sleep	night	first
want	tropic	good	wind	heavi	last	close
well	see	beach	fuck	wellington	close	coffe
e	much	#irma	#sfltraffic	wind	#traffic	friend
wind	#irma	florida	pleas	fuck	strong	help
fuck	beach	storm	storm	tropic	outsid	last
watch	florida	#mfl	beach	good	make	follow
wait	storm	aso	peopl	see	well	outsid
good	know	lauderdal	#irma	pleas	want	make
beach	#mfl	power	florida	storm	phone	sleep
storm	power	mesonet	f	flood	sleep	strong
live	call	rain	rain	beach	hit	#irmageddon
florida	aso	safe	mesonet	rain	florida	open

Table A.3. Related Words 13:00 UTC – 18:00 UTC

13:00	14:00	15:00	16:00	17:00	18:00
shelter	shelter	shelter	shelter	#hurricaneirma	hit
first	whole	tampa	want	outsid	safe
wait	tampa	beauti	time	food	outsid
beauti	yet	see	good	get	open
tri	check	#hurricaneirma	tampa	safe	updat
make	open	prep	get	watch	hurrican
could	hit	food	guess	time	make
made	get	come	hurrican	peopl	prayer
see	friend	watch	last	know	first
eye	read	yet	check	see	get
#hurricaneirma	world	time	hit	love	wait
world	safe	sleep	peopl	power	everyon
night	good	ride	come	gonna	check
peopl	time	first	eye	#irma	see
close	come	get	friend	still	home
outsid	make	check	food	hurrican	power
help	first	go	day	#nfl	#hurricaneirma
landfal	beauti	know	see	make	okay
come	wait	open	make	home	watch
pleas	home	tri	way	want	yet

Table A.4. Related Words 19:00 UTC – 00:00 UTC

19:00	20:00	21:00	22:00	23:00	00:00
tampa	#hurricaneirma	shelter	#hurricaneirma	#hurricaneirma	power
eye	go	hurrican	day	get	back
bay	#irma	#irma	watch	still	get
first	come	still	live	go	come
time	watch	come	time	updat	even
wait	#napl	updat	updat	wait	got
hit	wait	time	get	power	updat
outsid	live	whole	make	last	#irma
#hurricaneirma2017	pass	#hurricaneirma	shelter	#irma	time
make	hit	make	go	first	outsid
food	right	watch	wait	yet	still
us	friend	wait	see	right	go
shelter	day	stay	still	time	storm
get	shelter	see	#irma	tampa	much
last	get	safe	hit	us	hurrican
point	safe	rain	need	everyon	light
safe	beauti	home	rain	home	let
open	look	tampa	safe	made	friend
alway	everyon	get	open	hour	watch
video	make	pleas	pleas	back	start