TONGS: TLDR; OPINION NETWORK GUIDE SYSTEM

A Thesis

presented to

the Faculty of California Polytechnic State University,

San Luis Obispo

In Partial Fulfillment

of the Requirements for the Degree

Master of Science in Computer Science

by

Andrew Wang

December 2017

COMMITTEE MEMBERSHIP

TITLE:	TONGS: TLDR; Opinion Network Guide System
AUTHOR:	Andrew Wang
DATE SUBMITTED:	December 2017
COMMITTEE CHAIR:	Alexander Dekhtyar, Ph.D. Professor of Computer Science
COMMITTEE MEMBER:	Foaad Khosmood, Ph.D. Professor of Computer Science
COMMITTEE MEMBER:	Lubomir Stanchev, Ph.D. Professor of Computer Science
COMMITTEE MEMBER:	Chris Lupo, Ph.D. Professor of Computer Science

ABSTRACT

TONGS: TLDR; Opinion Network Guide System

Andrew Wang

In the modern world, huge amounts of text are being generated every minute. For example, Twitter users post their current emotions in tweets, while Facebook users vent about their experience in posts. In just one minute, Twitter users upload 350,000 tweets, and Facebook users post anywhere from 2.5 million to 3 million posts [8, 30]. To keep up with this growth in data, almost all of this information goes through automated text processing. To extract features such as the opinion and subjectivity in text, sentiment analysis is applied to the corpus. In this thesis, we present the TONGS library for conducting sentiment analysis. TONGS uses Word2Vec within the TensorFlow library to convert words into vector space representations. The TONGS library contains four different methods built upon previous research in sentiment analysis and Word2Vec. We further experiment and analyze these methods using the IMDB dataset. Finally, we introduce and test a new sentiment dataset from government hearings obtained through Digital Democracy, challenging the accuracy of the TONGS library in an unknown topic.

ACKNOWLEDGMENTS

Thanks to:

- Michelle Lam for being the best person ever.
- Sam Wu for being the best friend and lab partner to work with.
- My brother, Chris Wang, and sister-in-law, Jessi Chow, for letting me stay with them as I worked on this.
- Andrew Guenther, for uploading this template and adding testing.
- Corey Ford, Carson Caroll, Drew Schultz, and more for keeping the thesis template up to date.
- Leanne Fiorentino for looking out for all the students.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

INTRODUCTION

In today's society, it is important to understand the feelings and emotions of others. If someone is expressing the feeling of sadness, it means they could use some comforting. If someone is showing the emotion of joy, then they are experiencing the feeling of happiness. Emotion is detected with our five senses. Sight is a primary sense of obtaining information in the world. Hearing helps us communicate with others. Touch helps define our bodily sensations, such as tingling or tension in the skin or muscles. Taste and smell contribute a small portion to detecting emotion, such as keeping a distance from a terrible odor or bitter-tasting food to warn of poison. But, if all senses are removed, what happens to emotion?

Authors deal with this when writing books. They have to convey feelings and emotions through their writing. In this case, authors have to use the sense of sight combined with their usage of words in a book to stimulate all the senses of the reader. When people read books, they seem to understand the emotions and feelings the author was trying to convey. An author can write a sentence that can cover a range of expressions. So how do humans know that certain sentences are positive, negative, or neutral? How are visual symbols understood to be meaningful feelings and emotions instead of confusing and meaningless?

Humans have perception that helps them notice patterns. From a young age, they are taught that certain visual symbols make up the alphabet. They are then taught that each symbol has a sound associated with it. This leads them to group these symbols, letters, in different patterns that describe the world around them [13]. An example, the pattern of symbols, word, "apple" is used to identify the description of a red spherical edible object. When a child sees or hears the word "apple", they know

it represents a specific thing in this world. As a human's vocabulary grows, they can relate words to more things in this world, such as, "fresh" apple is "good" to eat but a "rotten" apple is "bad." Through learning sentence structure and growing their vocabulary, words describe more kinds of situations, and then that word will relate to that description, again. This is how the image of a "rotten apple" or using the word "rotten" in a different context such as "rotting corpse" will still convey the negative connotation of the emotion behind this phrase. Another relatable area is in their reaction for certain situations. For example, a funeral is not a positive event, so the feeling of sadness and the emotion of crying is associated with it.

Writers use these associations between word and experiences to help convey their feelings. Consider the example, "I know not all that may be coming, but be it what it will, I'll go to it laughing" by Herman Melville in Moby-Dick. The experiences of not knowing anything might be associated with fear, and when people are happy, they laugh. Therefore, there are two feelings here, one is the fear of the unknown which is, "I know not all that may be coming" and the other is joy with "I'll go to it laughing".

So, if only textual information is given, it is possible to retrieve sentiment from it. Normally, we use all five senses to determine the emotion, but words themselves can be assigned an emotional value. "Joy" is a word with positive emotion and "depression" is a word with negative emotion. The normative and emotional ranges for a large number of English words were obtained by Bradley and Lang as a part of their work on the ANEW (`Affective Norms for English Words`) corpus [4].

In the modern world there is access to huge collections of text. Every day there are news articles being written, reviews for movies, court hearing transcripts, and more. Almost all of it goes through automated text processing. For example, given a new article, what is the main topic or summary of it? Or given a text of a book,

what is the translation of it in another language? Given a product review, what is the customer's sentiment of the product? These are some of the questions that the natural language processing field of computer science tries to answer.

Natural language processing addresses tasks such as automatic summarization, machine translation, question answering, speech recognition, and sentiment analysis [15].

Sentiment analysis is the task of determining the opinion and subjectivity of text. This task can be seen with a large e-commerce website like `Amazon.com`. Customers can buy a product and have the chance to review it. Afterwards, the merchant can read the review and may modify their business based on the review. However, due to `Amazon.com` having millions of customers, even though only a small percentage can give a review, there is still a lot of sentiment to find for the merchant. Therefore, sentiment analysis can automatically summarize what all the reviewers are talking about and make it easier for the merchant to use it to improve their business.

Every year there are advancements on the techniques of sentiment analysis. Current methods use word-embeddings and machine learning algorithms to determine sentiment. Word-embeddings are numerical vectors per word that can be created in many different ways. The most popular way is using term-frequency - inverse document frequency. Machine learning algorithms such as Random Forest, Support Vector Machines, and others are popular in helping with categorization problems [5, 23, 24]. However, the best word-embedding and algorithm combination has an accuracy of 83% [26]. Therefore, there is room to improve the accuracy rate.

Recently, Google has released their open source software library for machine intelligence, TensorFlow. Within TensorFlow, there is a program called Word2Vec [1]. Word2Vec takes in sentences and creates numerical vectors for each word that it sees. These numerical vectors preserve semantic and syntactic relationships between words.

An example for using Word2Vec in the paper is:

$$vector(\text{“}King\text{”}) - vector(\text{“}Man\text{”}) + vector(\text{“}Woman\text{”}) = vector(\text{“}Queen\text{”}) \quad (1.1)$$

In other words, the vector of the word ‘king’ plus the vector of the word ‘woman’ results in a vector close to the vector of the word ‘queen’. These vectors are word-embeddings that are currently used for collaborative filtering, recommendation systems and bioinformatics. Google has also released their Word2Vec that has been trained on Google News’ data. It contains about three million words and is available for public use.

TLDR; Opinion Network Guide System (TONGS) is a library that uses Word2Vec for sentiment analysis. First, it checks prior successful sentiment analysis to use as a comparison. Second, it attempts sentiment analysis using Word2Vec. Last, TONGS studies which sentiment analysis methods Word2Vec works with and does not work with.

To test TONGS, we used the IMDB datasets from Pang et al. [26]. It is a popular data set across sentiment analysis projects to compare their accuracy. TONGS evaluations presented in this thesis also use a dataset from Digital Democracy. The Digital Democracy project by the Institute for Advanced Technology and Public Policy provides transcripts of legislative bills for public use. These hearings will be useful for conducting sentiment analysis on political data. Because there are many legislators and hearings, using sentiment analysis can help discover the viewpoint of a legislator on a specific bill.

The contributions of this thesis are as follows:

- We tested if sentiment analysis can be conducted using Word2Vec.
- We discovered clusters of words that have a similar sentiment using Word2Vec.

- We compared different machine learning algorithms for sentiment prediction trained with Word2Vec.
- We investigated the user of preprocessing techniques with Word2Vec-based sentiment analysis.
- We introduced a political dataset for sentiment analysis and tested the accuracy of sentiment analysis of our Word2Vec methods on political data

The rest of this work is organized as follows. Chapter 2 of the thesis discusses some of the background and related work in the field of sentiment analysis and Google's Word2Vec. Chapter 3 goes over the different methods in the TONGS library. Chapter 4 covers the experiments and analysis of the different methods. Finally, Chapter 5 reviews the contribution of this work.

Chapter 2

BACKGROUND AND RELATED WORK

A sentiment is a specific emotion, attitude or opinion prompted by feeling. Emotion is something that psychologists have tried to tackle for the past few decades. In 1972, Paul Ekman defined six basic emotions that are expressed universally across different cultures. With each basic emotion, there are secondary emotions as seen in the figure below. The six basic emotions are: anger, disgust, fear, happiness (joy), sadness and surprise [6].

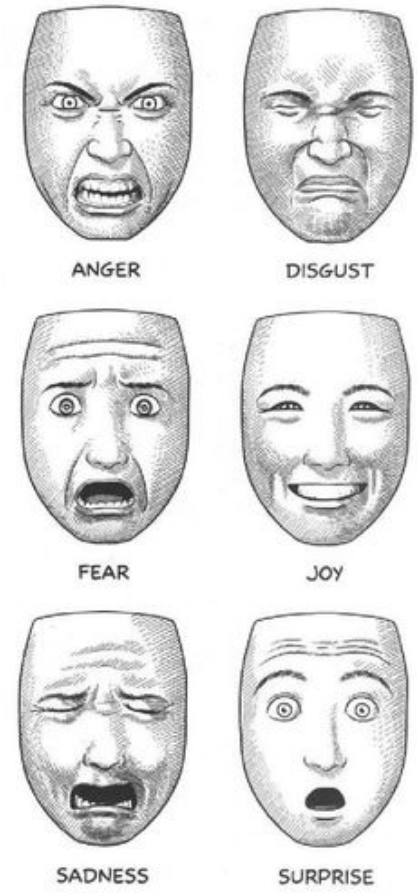

Figure 2.1: Ekman's emotions from Scott McCloud's Understanding Comics section about Emotions [17]

In 1980, Robert Plutchik created the "wheel of emotions." The wheel of emotions shows that basic emotions can blend and create new emotions. Plutchik suggests eight primary emotions, where each has a polar opposite: sadness and joy, anger and fear, trust and disgust, and surprise and anticipation.

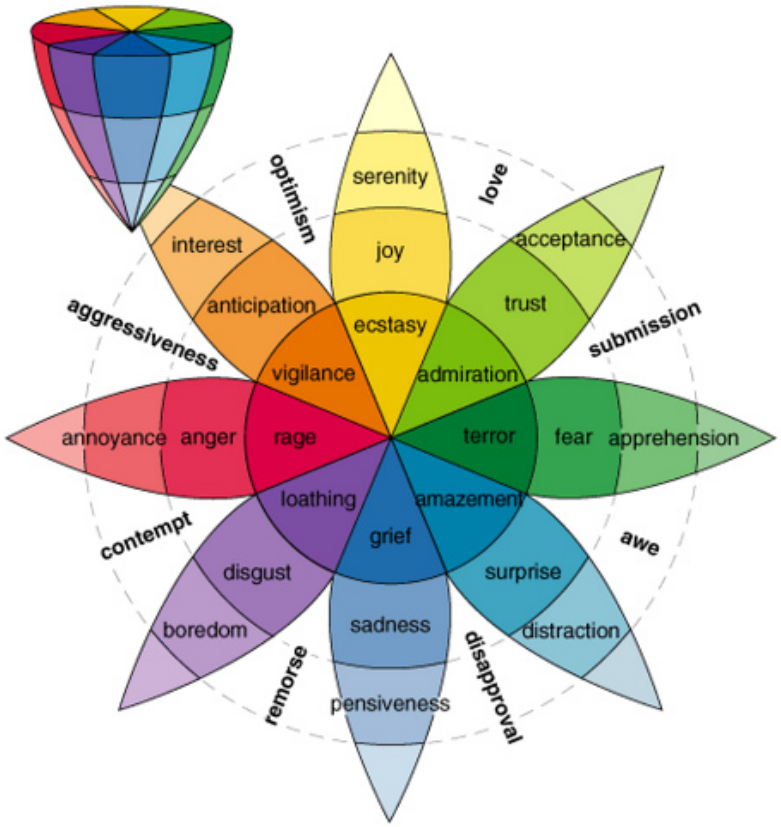

Figure 2.2: Plutchik Emotions [28]

Psychologists still do not have a clear cut answer to the question, "how many emotions do we have?" But using the idea of the base emotions that both Ekman and Plutchik have stated helps define a baseline for both positive and negative emotions. So, which emotions are positive or negative?

When someone wins the lottery, they express joy. When someone is walking alone and crying in the rain, there is a feeling of sadness. How are these emotions expressed? Emotional expression comes from facial movements such as smiling, scowling, and

other behaviors. People experience and express emotions by using their body to make sounds and movements. Certain emotions are expressed for specific situations. Someone being injured brings out surprise and fear causing the one experiencing these emotions to jump, close their eyes, or shake. Someone experiencing a sewage leak would cause them to scrunch up their face and walk away in disgust.

These experiences were originally shared via story telling. Legends of heroes and what they did were told to inspire children from generation to generation. Storytellers would describe the hero so children could have an image in their head. They would use descriptive words that their listeners can relate to, such as "the hero stood high above their enemies in victory" or "during the tiring battle, our hero took one deep breath before they swung their sword towards victory." Children do not personally have to experience the situation in the story to understand the emotions portrayed in it. As they listen, their experiences grow via the story they just heard. These stories were not only shared via voice, but also from images or scriptures on walls. These stories were then recorded into books, such as the bible. The same stories that were heard from storytellers or from walls were now saved into books that could be mass produced for many to read. As people read the text in books, they also feel the emotions as if they listened to a storyteller.

Yet, emotions are also felt with experiences that are hard to imagine. This can be seen with fiction writers. They are the professionals in sharing the emotions of the characters that they develop. They begin with introducing the theme, world, and time of their story, then slowly introduce their characters. Sometimes authors start in the middle of their story, but show how their character develops physically and emotionally. The readers follow along and feel the same happiness and sadness as the characters experience. For example. if readers learn about that the main character grew up with abusive parents, that is emotionally hurtful for both the main character and reader. Luckily, not everyone has to experience that event, but everyone knows

for a fact that it does not represent a positive emotion.

This is also seen with today's news. Headlines show that there are mistreated children in the world and people react to the headlines by being outraged or sad. On the opposite end, there are also headlines about world peace or people helping other people. Both of these are not experienced by everyone, yet the appropriate emotions are felt. News was originally spread by word of mouth, then by text on paper, and now by text on the Internet.

The Internet has large amounts of text other than news: there are also reviews, suggestions, and more, that are easily accessible for people to read. The amount of time required to read all of content on the Internet is difficult to achieve. When people read product reviews, they wish to discover the feeling of the previous buyer towards the product, whether if it is positive or negative. Or for news articles, was the author expressing his positive or negative emotions towards a specific event? These emotions can help us define sentiment.

Sentiment can either be positive, negative, or neutral. Positive sentiment can be the expression of love, sympathy, kindness, and joy. For instance, "Adrian Pasdar is excellent is this film." Negative sentiment can be angry, aggressiveness, sadness, and fearfulness. For example, "The menu cards are very ABSOLUTELY disgusting, covered in dirty oil with the foil peeling away from the paper." Neutral sentiment is when there is no emotion at all. For example, "There is a book on the desk."

How does one determine sentiment given only text? How was the sentiment determined for those two quotes above? A simple method is to look at the text and to pinpoint specific words that have sentiment related to them. For example, given a list of words, "lightbulb", "excellent", "cat", and "above", only the word "excellent" by itself seems to have sentiment related to it. So the first quote above contains the word "excellent", and none of the words in the rest of the sentence contribute to a

different sentiment. On the other hand, the other sentence seems to have multiple negative sentiment words, for example, "disgusting" and "dirty." Therefore, the second quote has to be a sentence with negative sentiment.

There are also ambiguous sentences that make it hard to determine what is the sentiment. A classic example from Wall Street Journal, "Republicans Grill IRS Chief Over Lost Emails." Without the context, there are two possible meanings of this quote. Either the Republicans are questioning the IRS chief about emails, or the Republicans are cooking the IRS chef with email as fuel. In either case, the sentiment is negative. But there are cases where sentences are difficult to understand and determine if its positive or negative. One instance would be sarcasm, "I'm happy for my browser to crash right in the middle of my coursework." The sentence contains the word *happy*, but unless if the context is known, people would not know that the browser crashing creates a negative emotion. Another would be "Can you recommend a good tool I could use?", which is a sentence that uses words of sentiment but does not necessarily express any sentiment.

Determining sentiment based on individual words was studied by Bradley and Lang in their research, `Affective norms for English words (ANEW)`. They conducted a study to develop a set of normative emotional ranges for a large number of words in the English language [4]. Each word was rated in terms of pleasure, arousal, and dominance. Therefore, by looking up each word in a sentence, the sentence's sentiment can be determined by grabbing its sentiment in the ANEW dataset. This technique works great on sentences that contain positive words, "I loved everything about her, so I introduced her to all my friends, and they loved her too." Or ones with negative words, "We feel angry or frustrated with others or ourselves." If positive words outweigh negative words, it's a positive sentence. Or if the negatives outweigh the positive words, then it is a negative sentence.

However, this technique is not perfect. First, ANEW covers only about 10,000 words, while the English language has over 3 million words in active and passive usage. Second, it fails to capture sentences such as, "Nice perfume. Must you marinate in it?" It has the word "nice", which may seem like a compliment to the perfume wearer, but the word "marinate" for putting on perfume puts it in a negative connotation. Sentences with multiple meanings also are hard to understand, such as, "the person walked into a mine." Was this person exploring coal mines or was this a military person whose life just ended? One could be an exciting experience, while the other is deadly. Humans are great at judging sentiment in a given context, however, for a computer it is a challenging task. Information on the Internet grows exponentially every day. Terabytes of data is uploaded and information is spread worldwide. A human would spend years reading the amount of text that is uploaded in only a couple of minutes.

An example text could be from an online e-commerce review such as this one from Amazon.com about chocolates, "I buy this candy a lot and it is always so good but not this time the inside was oily. Not smooth and creamy like it should be." The reviewer could be deciding whether to purchase this product or not. Amazon has star ranking system to evaluate the overall ratings of the reviews. However, these ratings become damaged by reviews such as a 1-star rating with a product review saying, "good!" or a 5-star rating that says, "not too bad." The text of the review is better indicator of what the reviewer means. So they read another review about the same product, "Ordered this a couple of weeks back and worried that they may melt while en route to my house due to what other customers have said in different posts. Once I received them, everyone loved them! They seem to have been stored in the right temperatures, and they taste great. They are definitely safe to buy from this seller - they don't seem to be last year's chocolates." There are about eight hundred other reviews to read, and it would be hours to decide if other buyers liked or disliked the

product.

Computers have the power to process gigabytes of data in minutes. Within seconds, those eight hundred reviews could be processed and return the total number of likes and dislikes. The field of text processing and converting it into meaningful data is called natural language processing. Natural language processing can summarize the text of all of the reviews, translate them into different languages, and determine sentiment. Sentiment analysis can solve the problem of what is the overall sentiment of the reviews of a product for a product.

Sentiment analysis is the process of computationally identifying and categorizing opinions expressed in a piece of text, especially to decide whether the author's attitude is either positive, negative or neutral [22]. Therefore, the problem here is, given only a body of text, is the sentiment positive or negative?

The first approach to sentiment analysis takes the same basic idea of determining the sentence's sentiment based on which words the sentence contains. The algorithm would have a preselected list of positive and negative words, go through a sentence and count the number of positive and negative words. If there are more positive words than negative words, then the sentiment would be positive. But if the sentence contains more negative words, then sentiment would be negative.

As simple as it seems, the downside of this approach is that it requires the preselected list of positive and negative words to be context aware. To illustrate, "Stop! We Beat Everybody!", would be a negative sentiment if this was a case of warning of trespassing. But in the case of commercial sales, this would be positive because it means that their deals are a bargain. So researchers began to tailor these lists of positive and negative words to specific themes manually. In 1963, Stone et al, studied polarity of words and manually constructed these lists, called lexicons, for the English language [31].

However, sentiment analysis is usually conducted with a large amount of text. Luckily, there are words that are positive all the time, such as "happy", "excited", and more, and words that are negative all the time, "depressed", "death", and more. Using these base words, the original lexicons could be expanded automatically using lexical relations or parts of speech patterns [32].

However, this basic algorithm and the limited dictionary did not give a great accuracy. Annet et al. tried preprocessing the input data to see if stemming words helped improve the accuracy. Stemming is the process of reducing inflected (or sometimes derived) words to their word stem, base or root form. Annet et al. also expanded their lexicon dictionary with WordNet, which increased their accuracy by 10%. Their highest accuracy rate was 60.4% only using the lexicon and WordNet addition. They recommend that the lexicon that used should maintain a 50–50 relationship of positive words to negative words or else classification will be skewed in one direction [3].

The most prominent work with the lexicon-based approach comes from Turney using Pointwise Mutual Information (PMI) for sentiment analysis. The PMI of two words, is:

$$PMI(word_1, word_2) = \log_2(\frac{p(word_1 \& word_2)}{p(word_1)p(word_2)}) \tag{2.1}$$

where $p(word_1 \& word_2)$ is the probability that $word_1$ and $word_2$ appear in the same sentence and where $p(word)$ is the probability that $word$ would occur in a sentence.

Turney first extracted phrases containing adjectives or adverbs using this equation to calculate the semantic orientation of a phrase:

$$SO(phrase) = PMI(phrase, \text{“}excellent\text{”}) - PMI(phrase, \text{“}poor\text{”}) \tag{2.2}$$

The reference words *"excellent"* and *"poor"* were chosen because it is common in five star ratings systems that five stars is *"excellent"* and one star is *"poor"*. Then he classified the text based on the average semantic orientation of the phrases. His accuracy on the movie dataset was 66% [32]. The lexicon-based approach was only

slightly better than guessing compared to the other techniques that used supervised machine learning to get a higher accuracy rate.

The machine learning approach uses specialized algorithms that are trained on prelabeled data and then determine sentiment by classifying the unlabeled data. Machine learning algorithms require the data to be in vectors of real numbers. These numerical vectors can be created in many ways, but popular methods used in previous research are Bag of Words and TF-IDF.

The Bag of Words approach gathers important words across the set of text and converts them into a vector. The vectors are created by initializing an empty vector with the length set to the number of unique words in the text. Then, for each document d, it will be assigned a vector v_d such that v_{d_i} is i'th word in the set of all words and the number of times it has appeared in that document d. Due to the set of all words being too large, it is very common to limit the vector to be the top n most frequent words.

TF-IDF, also known as term frequency – inverse document frequency, is a popular way to convert words to a vector space model and measure important words from a collection of text. The value of TF-IDF weight increases with the number of times a word appears in a document, but decreases with the number of documents in the corpus the word occurs. TF-IDF works by collecting all the words in a given collection of documents. The set of all terms becomes its vocabulary. Term frequency (tf_{d_w}) is the number of times a word appears in a document. The inverse document frequency (idf_w) measures how common a word w is across the corpus. N is the number of documents across the corpus.

$$idf_w = \log \frac{N}{df_w} \tag{2.3}$$

Then TF-IDF can be calculated for a word w as follows:

$$tfidf_w = tf_{d_w} * idf_w \tag{2.4}$$

Both of these ways to convert words to numerical vectors allow the use of algorithms such as Naive Bayes, Support Vector Machines, Maximum Entropy, Random Forest, and Logistic Regression [5, 24, 23, 25] to categorize the vectors into different classes. These algorithms take in numerical vectors as input in order to help them predict an output.

These algorithms are used in Pang et al's research, which analyzes the IMDB movie review dataset for sentiment. They use the bag of words approach along with extra information, such as unigrams, bigrams, parts of speech, and position of word as part of their word vector. They have shown that frequency is not as important as presence of word. Instead of using the number of times a word has shown up in a document, v_{d_i} will be 1 if the word appears at all in the document and 0 if it does not. They obtained an accuracy of 82.9% using Support Vector Machines with unigrams and word presence [26].

Other works take similar approaches of adding more information to their vector. Maas et al. builds on top of Pang et al's research and created a word vector that learns word representations that capture semantic and sentiment information. This achieved an accuracy rate of 88.9% [14]. Annett builds their vector using number of positive words, negative words, and words that negate the sentence. The algorithms that they used are Support Vector Machines, Naive Bayes, and Alternating Decision Trees. Their results show that the machine learning approach is better than the lexicon-based one. Their highest accuracy result for machine learning is using Support Vector Machines at 77.4% [3].

Even though Annett claims that the machine learning approach is better than the lexicon-based method, the requirement of prelabeled training set is usually a limitation. Generating the training set requires experts to annotate the data and if there is not enough data, the algorithm may fail.

There have also been approaches for combining machine learning and lexicon-based approaches. Hybrid approaches have been designed with a well-built lexicon and the accuracy of a strong supervised learning algorithm.

Mudinas et al.'s *pSenti* iterates both lexicon-based and learning-based sentiment analysis. It obtains higher accuracy than only lexicon-based systems, and similarities in line with learning-based systems. pSenti achieved an accuracy of 82.3% [21].

2.1 Word2Vec

Recently, Google released Word2Vec by Mikolov et al. [20]. Word2Vec represents the distribution of words in vector, unlike the two previous methods that relied on word presence or frequency. Mokolov et al. wanted to figure out if they could predict the words $w_{i-2}, w_{i-1}, w_{i+1}, w_{i+2}$ around a given word w_i. They created a neural network in order to attempt to solve this problem, but the result of the neural network were not promising. However, they noticed that the layer before the final output was encoded in such a way that it could be used to detect words with similar semantics. Further investigating this vector, they realized it captured linguistic properties such as gender, tense, plurality, and semantic concepts such as "is the capital city of."

This model that Mikolov et al. created is called Skip-Gram. It takes in a word W_i and predicts the words around it W_{i-2}, W_{i-1}, W_{i+1}, W_{i+2}.

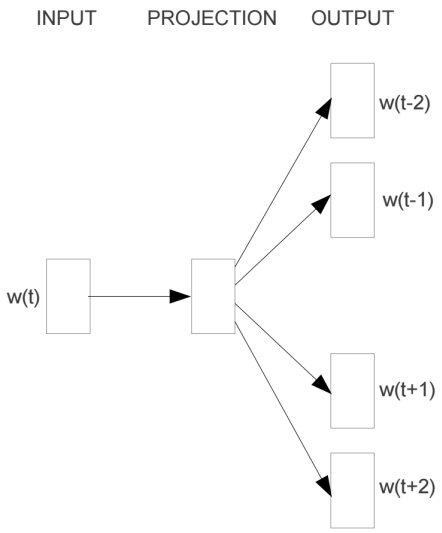

Figure 2.3: Diagram of Skip-gram

2.1.1 Uses for Word2Vec

Mikolov et al. claim that Word2Vec's word embeddings not only have good word representations as vectors but can be shown that words can have multiple degrees of similarity. The authors show an example of this equation

$$vector(\text{“}King\text{”}) - vector(\text{“}Man\text{”}) + vector(\text{“}Woman\text{”}) \tag{2.5}$$

which results in a vector that is closest to *vector(“Queen”)*. This shows that Word2Vec can be used to show lexical relations between words and can be used in a way that is similar to Turney's PMI equation.

The current methods of utilizing Word2Vec is to either use Google's TensorFlow open source library for Machine Intelligence or use Gensim. TensorFlow works by constructing a large neural net. The other option is with Gensim, which is originally ported the original Word2Vec implementation in C to python.

One of the more recent papers utilizing Word2Vec for sentiment analysis is Xue, Fu and Shaobin's research [34]. They have created a model to build a sentiment dictio-

nary using Word2Vec with their Semantic Orientation Pointwise Similarity Distance (SO-SD) model. SO-SD is defined as:

$$SO - SD(word) = \sum_{pword \in Pwords} SD(word, pword) - \sum_{nword \in NWords} SD(word, nword) \tag{2.6}$$

Where SD is the similarity distance between words, $Pwords$ is a list of positive words, and $Nwords$ is a list of negative words. Using this equation, they generated a lexicon to help them classify sentences. They have obtained a 83% accuracy rate with Weibo contents.

2.2 Classification Techniques

In our work, we use the Random Forest, Support Vector Machine, Naive Bayes, and Logistic Regression classification algorithms.

2.2.1 Random Forest

The general Random Forest method was proposed by Tin Kam Ho in 1995 [11]. and then Leo Breiman used it for classification and regression trees [5].

The Random Forest classifier is built upon the idea of Decision Tree classifiers. The decision tree classifier uses a model that takes in multiple input variables and tries to determine the class. The example below shows a decision tree, where the left path is *yes* and the right path is *no*. The final classifications are either *died* or *survived*.

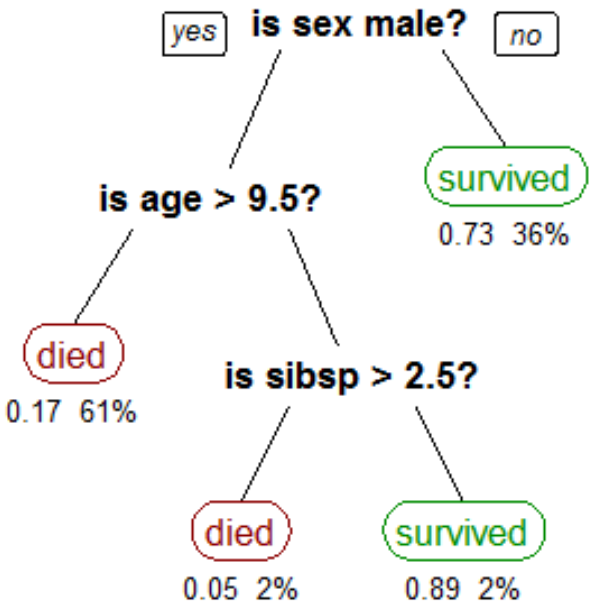

Figure 2.4: Decision Tree Example: Survival of passengers on the Titanic

The random forest classifier builds multiple decision trees and each tree is built based on a subset of the training data and a subset of features. This idea of combining multiple noisy and unbiased models to create a model with low variance is called bagging. Each decision tree randomly selects 63.2% of the original training data to build an individual tree, and a small subset of features. Across all the decision trees, all of the training data will be used. Then when there is data to fit, it goes through multiple decision trees and selects the most common class selected among all the

trees [5].

Data: Training Data

Result: Random Forest Classifier

Select n number of trees.;

i = 0;

while $i \leq n$ **do**

Select 63.2% of training data.;

Select m predictors out of all predictors;

Build decision tree with m predictors;

Calculate out of bag error rate using the 36.8% leftover training data;

$i = i + 1$;

end

Algorithm 1: Random Forest Decision Tree Algorithm

Random forest runs great on large datasets, has thousands of input variables without deletion, finds key variables for classification, unbiased estimate of generalization error, and does not overfit.

2.2.2 Support Vector Machines

Support Vector Machines take in labeled training data and output an optimal hyperplane. A hyperplane is a n-1 dimensional subset of n dimensional space of labeled training data that divides the space into two disconnected parts.

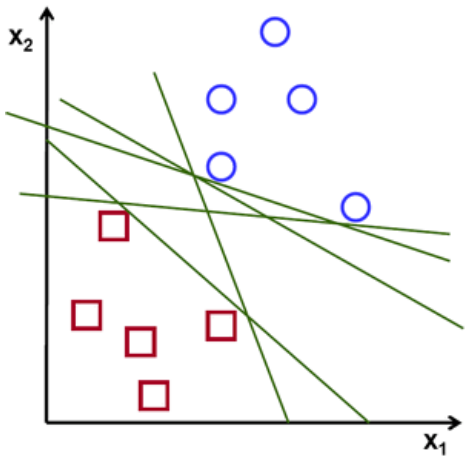

Figure 2.5: Hyperplane Example: 2D graph with potential hyperplanes separating the two spaces [24]

.

The hyperplane is defined as $w\dot{x}+b$. Since there are so many potential hyperplanes, the optimal hyperplane is one that has the largest minimum distance to the closest hyper plane. Support vectors are the elements of the training set that would change the position of the dividing hyperplane if removed. Support Vector Machines try to classify the data by selecting the best support vectors. Support Vector Machines try to balance finding the support vectors far enough from the detection planes with potentially misclassified points from the wide margins. Therefore, it wants to be as far as a possible from the plane, but penalizes for points that appear within the vectors.

The problem can be defined as:

$$\min_{w,b,\{\xi_i\}} (\frac{\|w\|^2}{2} + C\sum_{i=1}^{n}(\xi_i)) \tag{2.7}$$

subject to $y_i(w\dot{x}_i) \geq 1-\xi_i, \forall \bar{x_i} \in X, w+x_i \geq 0$. where C is a constant that determines how important linear separability is to the decision making. ξ_i are slack variables and are expected to be non-zero. $\xi_i > 0$ is when point x_i is within the margin.

This problem is actually solved by introducing a new set of variables α_i and setting $w = 2$.

This problem can be represented as the following equation:

$$\max L = \sum \alpha_i - \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j x_i \cdot x_j \tag{2.8}$$

This equation is in the form of the Lagrangian. It maximizes the distance between two support vectors, where x_i and x_j are the input vectors and α_i and α_j are constraints that must be greater than or equal to 0.

$$\alpha_i^{t+1} = \alpha_i^t + \eta(1 - y_k \sum_{j=1}^{m} (\alpha_j^t y_j x_i^T x_j)) \tag{2.9}$$

However, not all datasets are linearly separable. Therefore, a kernel function that maps data to a higher dimensional space to gain linear separation can be applied to transform the dataset.

$$K(x_i, x_j) = \phi(x_i) \cdot \phi(x_j) \tag{2.10}$$

Common kernel functions are:

- Polynomial: $(\bar{u} \cdot \bar{v} + 1)^n$
- Radial basis function: $e^{-\frac{\|x_i - x_j\|}{\sigma}}$
- Sigmoid: $\tanh(\kappa x \cdot y - \delta)$

2.2.3 Naive Bayes

The Naive Bayes classification technique is based on Bayes' Theorem. It assumes all features independently contribute to the probability. The Bayes' Theorem is defined as followed:

$$P(c|x) = \frac{P(x|c)P(c)}{P(x)} \tag{2.11}$$

Here, c is the class, x is the predictor. $P(c|x)$ is the posterior probability of class given predictor. $P(c)$ is the prior probability of class. $P(x)$ is the prior probability of observing the datapoint x. Given $x = (x_1, ..., x_n)$, the Naive Bayes assumption is the belief that the probability of observing each of x is independeant of observing other features. Therefore,

$$P(c|X) = P(x_1|c) \times P(x_2|c) \times \cdots \times P(x_n|c) \times P(c), \tag{2.12}$$

where $P(x|c)$ is the likelihood of the probability of predictor given class. Then the classifier is defined as this function, where K is the total number of classes and n is the total number of predictors.

$$y = \underset{k \in \{1,...,K\}}{argmax}\, p(C_k) \prod_{i=1}^{n} p(x_i|C_k), \tag{2.13}$$

where `argsmax` is the maximum a posteriori probability estimate. In other words, the mode of the posterior distribution. The class with the highest probability will be the chosen class for the input vector.

Naive Bayes is simple, fast and not sensitive to irrelevant features. However, it does assume every feature is independent.

2.2.4 Logistic Regression

In the 1930's, Fisher and Yates invented logistic regression [18]. It is named for the function that is used, the logistic function or sigmoid function. This function can take any real-valued number and map it to a value between 0 and 1.

$$\frac{1}{1 + e^{-value}} \tag{2.14}$$

This equation helps calculate the probability of different classes of an input vector

with n-dimensional features. The equation is rewritten as

$$h_\theta(x) = \frac{1}{1 + e^{-\theta x}} \tag{2.15}$$

Where θ is used to minimize the logistic cost function and is created from the input training set.

Consider a point x in feature space can be projected and converted into a real number. Then map the real number to the range of 0 to 1 with the logistic function. This gives us a probability value using any vector.

One way to use it for binary classification, is if the probability is greater than or equal to 0.5, classify it as one category and if it is less than 0.5, categorize it as the other.

2.2.5 Neural Networks

Neural networks are great at deriving meaning from complicated or imprecise data [2]. Similarly to the previous classifiers, the neural network takes in training data as input and then develops a system that optimizes the loss function of the predictor. Neural networks are built with neurons. These neurons take in many inputs and produce a single output with the system rules [23].

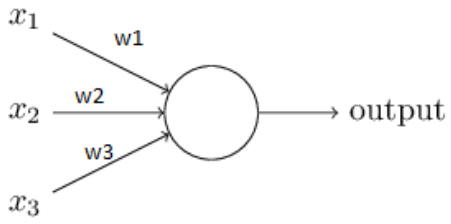

Figure 2.6: Diagram of a neuron

The output of a neuron is computed as $g(\sum_{w_i x_i})$ where g is a differentiable function called "activation function". A Neural network has at least two layers, an input layer

and an output layer. It can have more layers by having neurons taking the output of a neuron from the previous layer, applying a rule, and produce another output. These in between layers are called hidden layers.

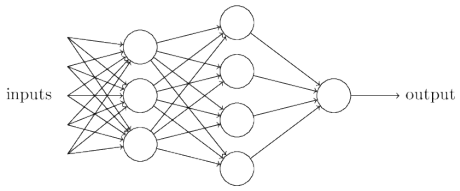

Figure 2.7: Diagram of a neural network with a single hidden layer.

Determining handwritten digits is a common example of how a neural network can be used for classification. In the figure below, it shows how using each pixel in the image, the neural network outputs the digit. It is possible for the neural network to be unsure and predict it to be multiple digits at once.

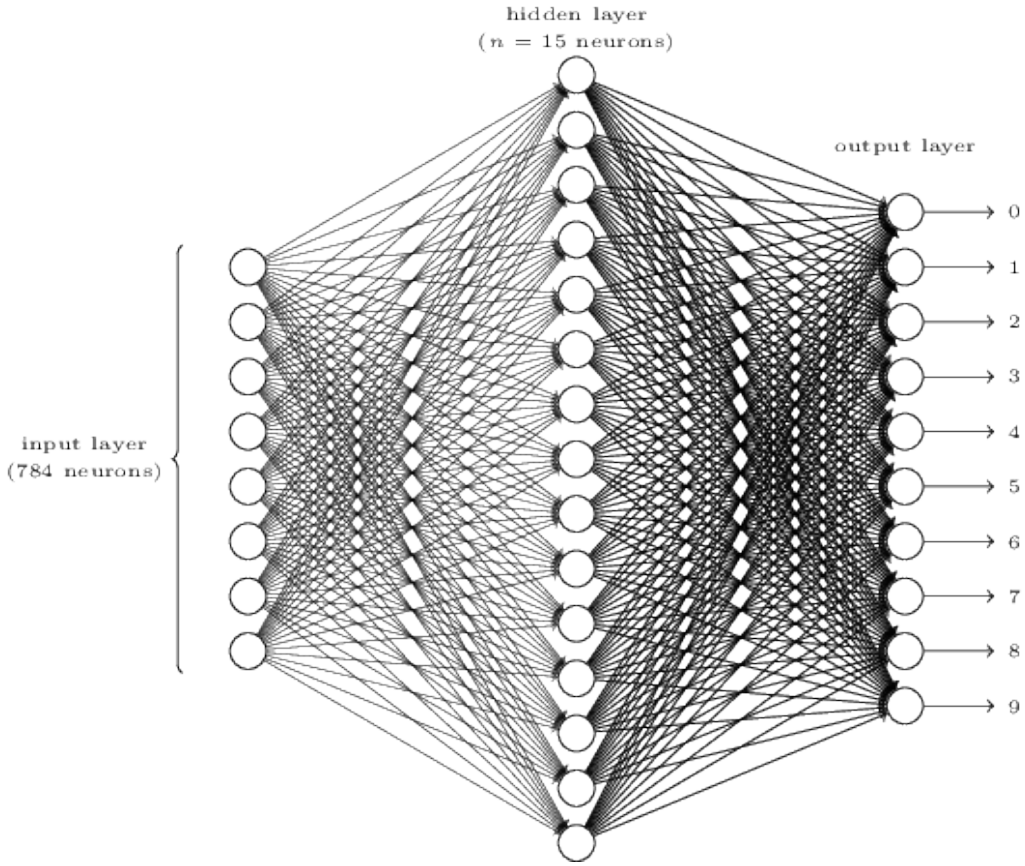

Figure 2.8: Diagram of a neural network with a single hidden layer.

Chapter 3

DESIGN AND IMPLEMENTATION

TLDR; Opinion Network Guide System (TONGS) researches a variety of ways to use Word2Vec for sentiment analysis. Word2Vec converts any word into a vector space representation. With the word's vector space representation, it may be possible to determine its sentiment based on its proximity to other sentiment-carrying words. Word2Vec can also be used to create a vector space representation of a sentence. The sentiment of a sentence is determined by using the sentence vector space representation with a machine learning algorithm.

TONGS consists of four different methods using Word2Vec for sentiment analysis.

- Discovering Vector Space Sentiment (DVSS)
- Similarity Pairs with Word2Vec Sentiment Analysis (SPWSA)
- Supervised Word2Vec Sentiment Analysis (SWSA)
- Supervised Word2Vec Sentiment Analysis with preprocessing techniques (SWSAPT)

3.1 Discovering Vector Space Sentiment

Word2Vec converts words into a vector space representation. The vector space represents the word's location in a multidimensional space. Then, calculating the distance between two words simply becomes the calculation of the distance between two vectors. Discovering Vector Space Sentiment (DVSS) analyzes the possibility of determining sentiment based on the word's location in the Word2Vec vector space.

The DVSS method makes an assuption that words with similar sentiment cluster together in Word2Vec space. For instance, the words *"happy"* and *"sad"* are positive and negative, respectively. If the word *"good"* has no sentiment assigned to it, then we can compute its distance in Word2Vec space to *"happy"* and *"sad"*. If it is closer to *"happy"*, then *"good"* has a positive sentiment. But if it is closer to *"sad"*, then it has negative sentiment. However, only using a single positive word and a single negative word may not be an accurate way to classify sentiment based on proximity. The DVSS method is improved by comparing the distance of a word to a collection of preselected known positive and negative words.

TONGS builds this method using GenSim's Word2Vec [29]. TONGS utilizes the pretrained Word2Vec vector from Google News' corpus and concentrates on the Word2Vec representations of Bing Liu's list of positive and negative words [12].

Thus, we expect positive words to be closer in Word2Vec space to other words with positive sentiment, and negative words closer to other words with negative sentiment.

Let pos be a set of known positive words and neg be a set of known negative words. Let $w \in pos \cup neg$ be a word with unknown sentiment.

Then $dist(w, pos) = \frac{1}{|pos|} \sum_{p \in pos} dist(w, p)$ and and $dist(w, neg) = \frac{1}{|neg|} \sum_{n \in neg} dist(w, n)$.

TONGS implements the DVSS method by following the algorithm below:

input: input-word

output: sentiment

Data: model = Word2Vec()

Data: $positives = GetPositiveListOfWords()$

Data: $negatives = GetNegativeListOfWords()$

foreach *word in positives* **do**

    Retrieve word from model and into positiveVectors;

end

foreach *word in negatives* **do**

    Retrieve word from model and into negativeVectors;

end

Retrieve the Word2Vec vector representation of input-word from model;

NegativeDistance = 0;

PositiveDistance = 0;

foreach *vec in positiveVectors* **do**

    positiveDistance += distance(input-word, vec);

end

foreach *vec in negativeVectors* **do**

    negativeDistance += distance(input-word, vec);

end

if $positiveDistance \leq negativeDistance$ **then**

    mark as positive;

end

else

    mark as negative;

end

Algorithm 2: Discovering Vector Space Sentiment Algorithm

This algorithm takes each word in the sentence and compares it to every positive word and negative word. The algorithmic complexity of this algorithm is:

$$\mathcal{O}(N + M) \tag{3.1}$$

where N is the number of positive words and M is the number of negative words.

In order to use this algorithm, each positive and negative word from Bing Liu's list [12] is converted into its Word2Vec vector space representation. The DVSS method takes in as input, `input-word`, which is the word of interest and outputs its discovered sentiment. Then, the `input-word` is converted into its vector space representation from the Word2Vec model. Then we calculate the distances from the `input-word` to the positive words, and do the same for the negative words. The sentiment of the `input-word` is most similar to the sentiment of whichever group had the smallest calculated distance.

For our evaluation we ask the following question: does DVSS accurately capture the sentiment of English words?

3.2 Similarity Pairs with Word2Vec Sentiment Analysis

Where the DVSS method predicts the sentiment of individual words, our second method, Similarity Pairs with Word2Vec Sentiment Analysis (SPWSA) method predicts the sentiment of full sentences. Informally, the method works as follows: for each word, it calculates the difference between the words similarity to the ideal positive word and, similarly, its similarity to the ideal negative word. The ideal positive and negative words are chosen from the lists of positive and negative words to optimize the classification function. They are selected through a training process which takes

in a set of positive and negative words and a set of sentiment labeled sentences and outputs the positive word that is the closest (in aggregation) in Word2Vec space to the words from positive sentences, and the negative word that is closest in aggregation to the words from negative sentences.

Once the ideal positive and negative words are selected, the sentence sentiment procedure works as follows: it computes the cumulative similarity of each word in the sentence to the ideal positive word, and the cumulative similarity of each word in the sentence to the ideal negative word. The sentiment of the sentence is set as the sign of the difference between these two cumulative similarity scores:

$$sentiment(sentence) = sign(\sum_{word in sentence} similarity(word, ideal\_positive\_word) - \sum_{word in sentence} similarity(word, ideal\_negative\_word)$$

In our implementation (see Algorithm 3 below), we use the cosine similarity as the similarity function for this method.

input: sentence

output: sentiment

sentiment = 0;

foreach *word in sentence* **do**

    sentiment += similarity(vector(word), vector(positive_word)) -

     similarity(vector(word), vector(negative_word));

end

return sentiment;

Algorithm 3: Word2Vec Sentence Sentiment Calculation

The similarity function here is the cosine similarity function. Therefore, the value of similarity will be closer to 1 if a word is more similar to another word; otherwise it will be closer to 0. The idea of the algorithm is that if the word is similar to a

positive word, it will be closer to `positive_word`. But if the word is similar to a negative word, it will be adjacent to `negative_word`. More positive similar words than negative words will result in a positive sentiment score, and if there are more negative similar words than positive, the result will be negative. Therefore, if the result of the computation is positive, the word has positive sentiment; otherwise if the word is negative, it has negative sentiment.

TONGS creates this function using Gensim and the similarity function exists under the `Word2VecModel.similarity(word1, word2)` [29]. This method still uses the Word2Vec model trained by the Google News' corpus.

The method is implemented as defined by the algorithm below:

input: sentence

output: sentiment

Data: model = Word2Vec()

SentenceSentiment = 0;

foreach *word in sentence* **do**

    SentenceSentiment += model.similarity(word, positive_word) -
     model.similarity(word, negative_word);

end

if *SentenceSentiment* ≥ 0 **then**

    return positive;

end

else if *SentenceSentiment*<0 **then**

    return negative;

end

Algorithm 4: Similarity Pair Word2Vec Sentiment Calculation

The algorithmic complexity of this algorithm is:

$$\mathcal{O}(N) \tag{3.2}$$

where N is the number of words in the sentence.

This algorithm takes in a string, `sentence`, and outputs the discovered sentiment. TONGS researches which `positive_word` and `negative_word` would produce the highest accuracy. It generates the enumeration of positive and negative word pairs from Bing Liu's list of positive and negative words [12]. TONGS has concluded that the words "nice" and "mean" produces the highest accuracy. The *finalize* method calculates the distance between each word in the sentence to "nice" and "mean". The result of this method returns the sentiment as positive, if the calculated sentiment is greater than or equal to zero, and negative, if it's less than zero.

The question, "does the SPWSC method obtain an accuracy feasible for sentiment analysis with Word2Vec?" will be answered in Section 5.

3.3 Supervised Word2Vec Sentiment Analysis

Word2Vec extends the conversion of words-to-vector space representations into sentence-to-vector space representations. This is a challenge since Word2Vec only converts words to vectors. A sentence is a word, clause, or phrase or a group of clauses or phrases forming a syntactic unit which expresses an assertion, a question, a command, a wish, an exclamation, or the performance of an action [19]. As a result, we can take the vector representation of each word and combine them to create a new vector to represent as the sentence vector. At the same time, just having the vectorized form of a sentence and its associated sentiment is not enough to determine the sentiment of other sentences. A supervised machine learning algorithm is trained with a training

set, allowing this classifier to classify new sentences' sentiment.

TONGS implements this method using multiple supervised machine learning algorithms. The user has a choice between Naive Bayes, Random Forest, Support Vector Machines, or Logistic Regression (See Section 2). The only data desired from these classifiers is whether or not the sentence is positive or negative. The chosen classifiers excel at binary classification. TONGS uses GenSim's Word2Vec and ScikitLearn packages for the classifiers.

The model fitting processes are based on the algorithm below:

input: classifier
output: sentiment
Data: model = Word2Vec()
Data: trainingdata = GetTrainingData()
foreach *sentence in trainingdata* **do**
    Train Word2Vec with words in sentence;
end
foreach *sentence, sentiment in trainingdata* **do**
    sentenceVector = [];
    foreach *word in sentence* **do**
        sentenceVector += model[word];
    end
    Train classifier with sentenceVector and sentiment;
end
return classifier;

Algorithm 5: Supervised Word2Vec Sentiment Model-Fitting Algorithm

Then the classification method is:

input: input-sentence, classifier

output: sentiment

sentenceVector = [];

foreach *word in input-sentence* **do**

    sentenceVector += model[word];

end

Classify sentenceVector and return sentiment;

Algorithm 6: Supervised Word2Vec Sentiment Analysis Classification Algorithm

The model-fitting algorithm takes in a classifier type to determine which classifying algorithm to create. Then the training dataset is retrieved to train the Word2Vec model and then the Word2Vec model converts sentences into their vector representation. Sentences are converted to sentence vectors by retrieving each vector representation of the words in the sentence and adding them together. Each vector representation of the word have values based on the linguistic context of words that Word2Vec determines. Adding each word vector in the sentence produces a vector representation of the sentence, since the linguistic context of the overall sentence is based on all the values of words in the sentence. We use these sentence vector representations and the associated sentiments to train the classifiers.

The method takes in an input-sentence and the classifier created by the pre-trained process. It converts the input sentence into a vector and uses the classifier to obtain the resulting sentiment.

Our evaluation question is: In the Supervised Word2Vec Sentiment Analysis (SWSA) method, which supervised machine learning algorithm produces the highest accuracy for sentiment analysis with Word2Vec?

3.4 Supervised Word2Vec Sentiment Analysis with Preprocessing Techniques

Prior research has shown that preprocessing techniques such as stopword filtering, parts of speech tagging, negation tagging, stemming, or expanding contractions helps the accuracy rating of sentiment analysis [26].

TONGS implements the following preprocessing steps:

3.4.1 Stopword Filtering

Stopword filtering removes extremely common words such as 'a', 'you', and more. The removal of stopwords reduces the impact of common words in Word2Vec.

3.4.2 Parts of Speech Tagging

Parts of speech tagging include a unique identifier after the word to indicate its part of speech. With this intention, words which have multiple parts of speech can be disambiguated. The word *fast* can be a noun, verb, adjective, or an adverb. Consider these two sentences: "That person is fast" and "They fast during Lent." In the first sentence, the word *fast* is an adjective and in the second sentence, the word *fast* is a verb. So to clarify usage of the word, the preprocessor would differentiate them by adding "_JJ" to the adjective *fast* in the first sentence, producing "fast_JJ". The second sentence will be tagged as "fast_VB" for being an verb. These tag labels come from Penn Treebank Parts of Speech [16]. This will increase the number of words in Word2Vec but will potentially clarify which word is used during the classification of new sentences.

3.4.3 Negation Tagging

Negation tagging detects words that have been negated, and this is done by adding "_NEG" to the word. To illustrate, the sentence "I am not happy" will produce the tokens "I", "am", "not", and "happy_NEG". This was used to further increase the difference between positive and negative sentences. If this sentence was looked at word by word, the only sentiment carrying word is *happy*, which is a positive word. However, this sentiment is negative. With negation tagging, the sentiment carrying word would be "happy_NEG", which is the negative form of the originally positive word. Therefore, we have the original from of the sentiment word and the reverse form to differentiate positive and negative

3.4.4 Stemming

Stemming combines different forms of a word into its base form. For instance, the words "argue", "argues", "argued", and "arguing" will be reduced to the stem form "argu". Stemming reduces the number of total words in Word2Vec but increases uses of the stem word. The more occurrences of a word in the training corpus helps Word2Vec find the semantic similarities between other words.

3.4.5 Expanding Contractions

The expanding contractions preprocessing technique expands contractions in order to keep sentences such as "I can't do that" and "I can not do that" the same.

3.4.6 Normalization

Another preprocessing technique does not apply to the words themselves, but rather on the sentence vector created. The sentence vector can be normalized using Eu-

clidean norm. Depending on the size of the sentence, the values of the vector can grow to be large values if not normalized. Normalizing the sentence vector can help with the accuracy rating.

Data preprocessing in TONGS is implemented as flags that can be toggled when running the application.

The preprocessing steps modify the previous model-fitting algorithm and are rewritten as follows:

input: classifier, preprocessFlags

output: classifier

Data: model = Word2Vec()

Data: trainingdata = GetTrainingData()

foreach *sentence in trainingdata* **do**

    preprocessedSentence Preprocess(preprocessFlags, sentence);

    Train Word2Vec with words in preprocessedSentence;

end

foreach *sentence, sentiment in trainingdata* **do**

    preprocessedSentence = Preprocess(preprocessFlags, sentence);

    sentenceVector = [] **foreach** *word in preprocessedSentence* **do**
     | sentenceVector += model[word]

    end

    Normalize(preprocessFlag, sentenceVector);

    Train classifier with sentenceVector and sentiment;

end

return classifier;

Algorithm 7: Supervised Word2Vec Sentiment Calculation with Preprocessing Model-Fitting Algorithm

Our final method is a modification of the SWSA method with the addition of the specific preprocessing techniques.

input: input-sentence, classifier, preprocessFlags

output: sentiment

sentenceVector = [] preprocessSentence = preprocess(preprocessFlags,

input-sentence) **foreach** *word in preprocessSentence* **do**
| sentenceVector += model[word]

end

Normalize(preprocessFlags, sentenceVector);

Classify sentenceVector and return sentiment;

Algorithm 8: Supervised Word2Vec Sentiment Calculation with Preprocessing Classification Algorithm

This method takes the sentence to classify as an input, and the classifier from the model-fitting algorithm. Similar to SWSA, it uses the training data to train Word2Vec and then to train the classifier. However, before training the classifier, it preprocesses the input data. The preprocessing depends on which flags are enabled, but it will be a combination of these techniques: stopword filtering, parts of speech tagging, negation tagging, stemming, or expanding contractions. Before the sentence vector is consumed by the classifier, it may be normalized if enabled by the preprocessing flags. The result of the method is the same as the previous experiment, being the sentiment determined by the classifier.

Our evaluation question is: In the Supervised Word2Vec Sentiment Analysis with Preprocessing Techniques (SWSAPT) method, which preprocessing technique produces the highest accuracy for sentiment analysis with Word2Vec?

Chapter 4

RESULTS AND ANALYSIS

The four methods discussed in the last chapter can compute the sentiment of words or sentences. However, the validity of these methods has not been discussed. In order to be useful functional methods of sentiment analysis, they need to have reasonable accuracy. An accuracy similar or better than previous experiments would validate the methods. The very least, the accuracy rating should be better than 50%.

For each of our experiments we used Bing Liu's list of positive and negative of words [12], the Google News' corpus, and the pre-labeled IMDB dataset. In order to ensure that these experiments are not tailored to calculating the sentiment of movie reviews, we include an additional experiment that tests with a political dataset from Digital Democracy [9].

The five experiments TONGS conducts in order to validate the methods are:

- Discovering Vector Space Sentiment
- Similarity Pairs with Word2Vec Sentiment Analysis
- Supervised Word2Vec Sentiment Analysis
- Supervised Word2Vec Sentiment Analysis with preprocessing techniques
- Supervised Word2Vec Sentiment Analysis with preprocessing on Digital Democracy

4.1 Discovering Vector Space Sentiment

This method determines the sentiment of a word based on its vector space representation. An experiment to validate the method needs to test the accuracy of the computed sentiment. This can be done by excluding a set of known words with sentiment and running them through the method. The accuracy rate can be calculated as the number of correctly labeled sentiment divided by the total number of words tested.

We randomly selected 100 positive and 100 negative words and TONGS classified the words using the algorithm below:

output: accuracy

Data: $testSentimentWords = GetTestPositiveAndNegativeListOfWords()$

foreach *word, sentiment in testSentimentWords* **do**

guessedSentiment = DiscoverVectorSpaceSentiment(word) **if**

gussedSentiment == 1 and guessedSentiment == sentiment **then**

$acutal_positive+ = 1;$

end

if *gussedSentiment == 0 and guessedSentiment == sentiment* **then**

$acutal_negative+ = 1;$

end

if *gussedSentiment == 1 and guessedSentiment != sentiment* **then**

$false_positive+ = 1;$

end

if *gussedSentiment == 0 and guessedSentiment != sentiment* **then**

$false_negative+ = 1;$

end

end

The confusion matrix for this experiment is:

Table 4.1: Confusion Matrix for DVSS

	Classified Positive	Classified Negative
Actual Positive	1	99
Actual Negative	53	47

This experiment failed to do better than 50%. To further investigate why this

method failed, we attempted to visually see the groups created. However, multidimensional vectors are not easy to visualize on a two-dimensional plane. However, the t-distributed stochastic neighbor embedding (t-SNE) algorithm helps with reducing the dimension of the vectors to at least two dimensions while preserving the higher dimensional representation [33]. Using t-SNE, each word is reduced from multiple dimensions to two dimensions in order to plot on a 2D plane. On this 2D plane, positive words are marked as a blue plus sign and negative words are marked as a red minus sign. The axis do not have any logical meaning. The algorithm preserves distances in high-dimensional space, where the actual point position does not have much meaning. The image below shows the result of this investigation.

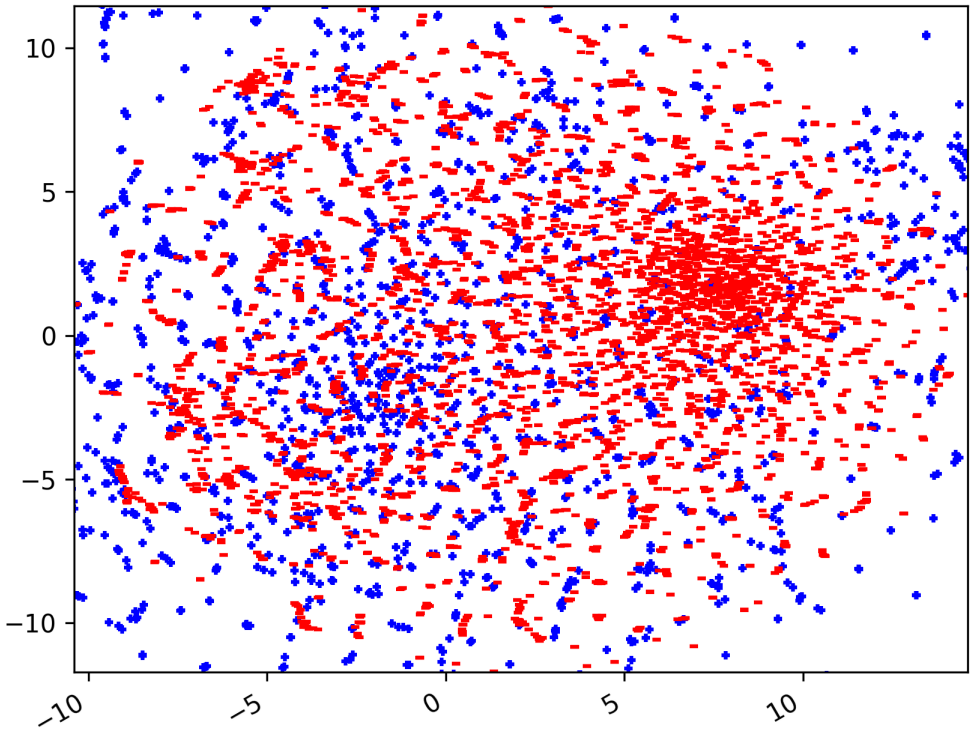

Figure 4.1: Graph of all positive and negative word vectors in 2D via TSNE. Axis are irrelevalent due to dimention reducing algorithm.

As seen in the image, positive words and negative words are located randomly next

to each other and not in an organized way. Therefore, it is difficult to assert sentiment based on location relative to other words. Because there is not a distinct separation between positive and negative words, this method fails to compute sentiment better than guessing. Therefore, determining sentiment based on the vector space is not possible with Word2Vec.

4.2 Similarity Pairs with Word2Vec Sentiment Analysis

In Similarity Pairs with Word2Vec Sentiment Analysis, we had to reduce the number of words to investigate to ten popular opposite word pairs. Each pair of words were tested with the IMDB training dataset to see which one achieved the highest accuracy.

The training phase is as follows:

output: accuracy

Data: $trainingSentences = GetTrainingSet()$

$ideal\_pos = None$;

$ideal\_neg = None$;

$bestAccuracy = 0$;

foreach *positiveWord, negativeWord in SentimentPairs* **do**

    $total = 0$;

    $correct = 0$;

    foreach *sentence, sentiment in testSentences* **do**

        guessed-sentiment = SPWSASentiment(sentence, positiveWord,

         negativeWord) **if** *guessed-sentiment == sentiment* **then**

            $correct+ = 1$;

        end

        $total+ = 1$;

    end

    $accuracy = correct/total$ **if** $bestAccuracy \leq accuracy$ **then**

        $ideal\_pos = positiveWord$;

        $ideal\_neg = negativeWord$;

        $bestAccuracy = accuracy$;

    end

end

Below is the table of words and their associated training accuracy.

Table 4.2: Positive and Negative word pairs

Positive Word	Negative Word	Training Accuracy
nice	mean	62.956%
joyful	depressed	59.95%
love	hate	57.488%
good	bad	51.352%
excellent	poor	51.292%
happy	sad	51.156%
alive	dead	50.868%
pretty	ugly	50.028%
good	evil	49.992%
yes	no	49.32%

Of the pairs tested, the ideal positive word is *nice* and the ideal negative word is *mean*.

This pair is used for the validation process. The validation process requires a set of sentences with labeled sentiment to test against. The process will need to run each computed sentence through this method and then compare the sentiment with the actual sentiment. The accuracy of this method can be calculated as the number of correctly labeled sentences divided by the number of sentences tested.

The validation phase is as follows:

output: accuracy

Data: $testSentences = GetTestSet()$

$total\_pos = 0$;

$total\_neg = 0$;

$correct\_pos = 0$;

$correct\_neg = 0$;

foreach *sentence, sentiment in testSentences* **do**

guessed-sentiment = SPWSASentiment(sentence, ('nice', 'mean') **if**

guessed-sentiment == 1 **then**

if *guessed-sentiment == sentiment* **then**

$correct\_pos+ = 1$;

end

$total\_pos+ = 1$;

end

else

if *guessed-sentiment == sentiment* **then**

$correct\_neg+ = 1$;

end

$total\_neg+ = 1$;

end

end

Here are the results from the experiment:

Table 4.3: Accuracy Ratings using *nice* and *mean* for SPWSA

Positive Accuracy	61.648%
Negative Accuracy	79.52%
Overall Accuracy	70.584%

This experiment shows that sentence sentiment analysis with Word2Vec is feasible. Word2Vec claims to preserve the relationship between words, so replacing the PMI equation from Turney's research with the Word2Vec similarity equation is reasonable.

Turney's accuracy on the movie dataset was 66% [32]. The same method of word similarity via Word2Vec achieved an overall score of 70%. However, breaking down the results into specific positive or negative accuracy ratings shows that this method is better at detecting negative sentences rather than of positive ones. The dataset itself has an equal amount of overall positive and negative sentences, but the sentences in each review may vary.

4.3 Supervised Word2Vec Sentiment Analysis

This method uses a supervised machine learning algorithm for sentiment analysis. Supervised machine learning requires a training dataset, and in order to calculate its accuracy rate, it also has an paired test dataset. This experiment runs through multiple classifiers and each one is tested to investigate which classifier would obtain the highest accuracy rate.

output: accuracy

Data: $testSentences = GetTestSet()$

Data: $classifiers = GetClassifiers()$

$true\_pos = 0$;

$true\_neg = 0$;

$false\_pos = 0$;

$false\_neg = 0$;

foreach *classifier in classifiers* **do**

  foreach *sentence, sentiment in testSentences* **do**

    guessed-sentiment = SupervisedWord2VecSentimentAnalysis(sentence, classifier) **if** *guessed-sentiment == 1* **then**

      if *guessed-sentiment == sentiment* **then**

        $true\_pos+ = 1$;

      end

      else

        $false\_pos+ = 1$;

      end

    end

    else

      if *guessed-sentiment == sentiment* **then**

        $true\_neg+ = 1$;

      end

      else

        $false\_neg+ = 1$;

      end

    end

  end

end

return correct/total;

In the algorithm, each classifier will be tested their accuracy based on the test dataset. The algorithm will calculate the number of true positives, true negatives, false positive, and false negatives.

Here are the confusion matrices for each classifier and its accuracy ratings.

Table 4.4: Confusion Matrix with Entropy and Purity for Word2Vec with Naive Bayes

	Classified Positive	Classified Negative	Entropy	Purity
Actual Positive	6549	5951	0.998	0.523
Actual Negative	2766	9734	0.762	0.779
Total	9315	15685	0.880	0.651

Table 4.5: Confusion Matrix with Entropy and Purity for Word2Vec with Random Forest

	Classified Positive	Classified Negative	Entropy	Purity
Actual Positive	5440	7060	0.988	0.565
Actual Negative	1851	10649	0.605	0.852
Total	7291	17709	0.797	0.709

Table 4.6: Confusion Matrix with Entropy and Purity for Word2Vec with SVM

	Classified Positive	Classified Negative	Entropy	Purity
Actual Positive	10391	2109	0.655	0.831
Actual Negative	2149	10351	0.662	0.828
Total	12540	12460	0.658	0.830

Table 4.7: Confusion Matrix with Entropy and Purity for Word2Vec with Logistic Regression

	Classified Positive	Classified Negative	Entropy	Purity
Actual Positive	10715	1785	0.592	0.8572
Actual Negative	1540	10960	0.538	0.877
Total	12255	12745	0.565	0.867

The best classifier is Logistic Regression with Word2Vec with an accuracy of 87.6%, which is better than Pang and Lee's word-embeddings [26]. Naive Bayes and Random Forest do a better job at classifying negative sentences than positive sentences. SVM and Logistic Regression classify both positive and negative sentences at the same accuracy.

4.4 Supervised Word2Vec Sentiment Analysis with preprocessing techniques

This experiment goes through all the different preprocessing techniques and see which one obtains an accuracy better than not having any preprocessing techniques. For this validation method, only the classifier with the highest accuracy rating is used. This experiment runs through a similar process as the last experiment, but it will also be switching the preprocessing technique.

The table below shows how the accuracy rating changed per technique for each classifier.

Table 4.8: Accuracy measures for different preprocessing techniques using Word2Vec with Naive Bayes

Stopword Filtering	82.3%
Stemming	65.3%
Expanding Contractions	65.1%
POS Tagging	63.8%
Negation Tagging	54.1%

Table 4.9: Accuracy measures for different preprocessing techniques using Word2Vec with Random Forest

Stopword Filtering	66.8%
Stemming	66.7%
Expanding Contractions	66.2%
POS Tagging	67.2%
Negation Tagging	66.2%

Table 4.10: Accuracy measures for different preprocessing techniques using Word2Vec with SVM

Stopword Filtering	84%
Stemming	82.8%
Expanding Contractions	83.6%
POS Tagging	83%
Negation Tagging	81.1%

Table 4.11: Accuracy measures for different preprocessing techniques using Word2Vec with Logistic Regression

Stopword Filtering	87.96%
Stemming	86.3%
Expanding Contractions	87.6 %
POS Tagging	86.4%
Negation Tagging	83.1%

The next step was to combine different preprocessing techniques to see if that increased accuracy ratings. TONGS carefully choses the order of the preprocessing techniques in order to prevent loss of information. For example, adding parts of speech tags after stopword filtering will lose the original sentence structure for the tagger. Therefore, TONGS chooses preprocessing in this order: Expand Contractions, Stemming, POS Tagging, Stopword Filtering, and then Negation Tagging.

One issue here would be the collision of Parts of Speech tagging and Negation Tagging. TONGS handles this by detecting if Parts of Speech tagging was enabled and will extract the original word and add `_NEG` after the Parts of Speech tagging. An example of what a negated word would look like is: `sad_JJ_NEG`.

Below are tables of enabled preprocessing techniques and its accuracy rating on the IMDB dataset for SVM and Logistic Regression classifier. Tables for Random Forest and Naive Bayes can be found in Appendix B.

Table 4.12: Accuracy measures for different combinations of preprocessing techniques using Word2Vec with SVM

POS Tagging	Stopword Filtering	Stemming	Expanding Contractions	Negation Tagging	Accuracy
X	X				84.1%
	X				84%
	X		X		83.9%
X	X		X		83.7%
			X		83.6%
					83.5%
	X	X	X		83.3%
X			X		83.1%
X	X	X			83.1%
X	X	X	X		83.1%
X					83%
	X	X			83%
	X			X	83%
		X	X		83%
	X		X	X	83%
		X			82.8%
X	X			X	82.8%
	X	X		X	82.6%
X	X		X	X	82.6%
X		X			82.5%
X		X	X		82.3%
	X	X	X	X	82.3%
X	X	X	X	X	82%
X	X	X		X	81.9%
			X	X	81.3%
		X		X	81.2%
				X	81.1%
X				X	80.4%
		X	X	X	80.4%
X			X	X	80.1%
X		X		X	79.7%
X		X	X	X	79.6%

Table 4.13: Accuracy measures for different combinations of preprocessing techniques using Word2Vec with Logistic Regression

POS Tagging	Stopword Filtering	Stemming	Expanding Contractions	Negation Tagging	Accuracy
X	X				88.2%
	X		X		88.2%
X	X		X		88.1%
	X				88%
			X		87.7%
					87.5%
X			X		87.5%
	X	X			87.5%
	X	X	X		87.5%
X	X	X			87.4%
X					87.3%
		X	X		87.3%
X	X	X	X		87.2%
		X			87.1%
	X			X	86.8%
X		X	X		86.7%
X		X			86.6%
	X	X		X	86.6%
	X		X	X	86.6%
X	X		X	X	86.6%
	X	X	X	X	86.6%
X	X			X	86.4%
X	X	X		X	86.3%
X	X	X	X	X	86.1%
			X	X	85%
		X	X	X	84.9%
				X	84.7%
		X		X	84.7%
X				X	84.6%
X			X	X	84.4%
X		X	X	X	83.8%
X		X		X	83.7%

From the table, we can see that adding multiple preprocessing techniques reduce the accuracy rating. This would be due to reducing the number of words seen in the

Word2Vec model to train.

We can see that including stopword increases the accuracy, but is it worth the improvement from the baseline logistic regression at 87.6%? What is the cost of the approximately 0.4% gain?

The graph below shows how long it took each phase of the SWSAPT algorithm for each SVM and Logistic Regression. Graphs for Random Forest and Naive Bayes can be found in Appendix B.

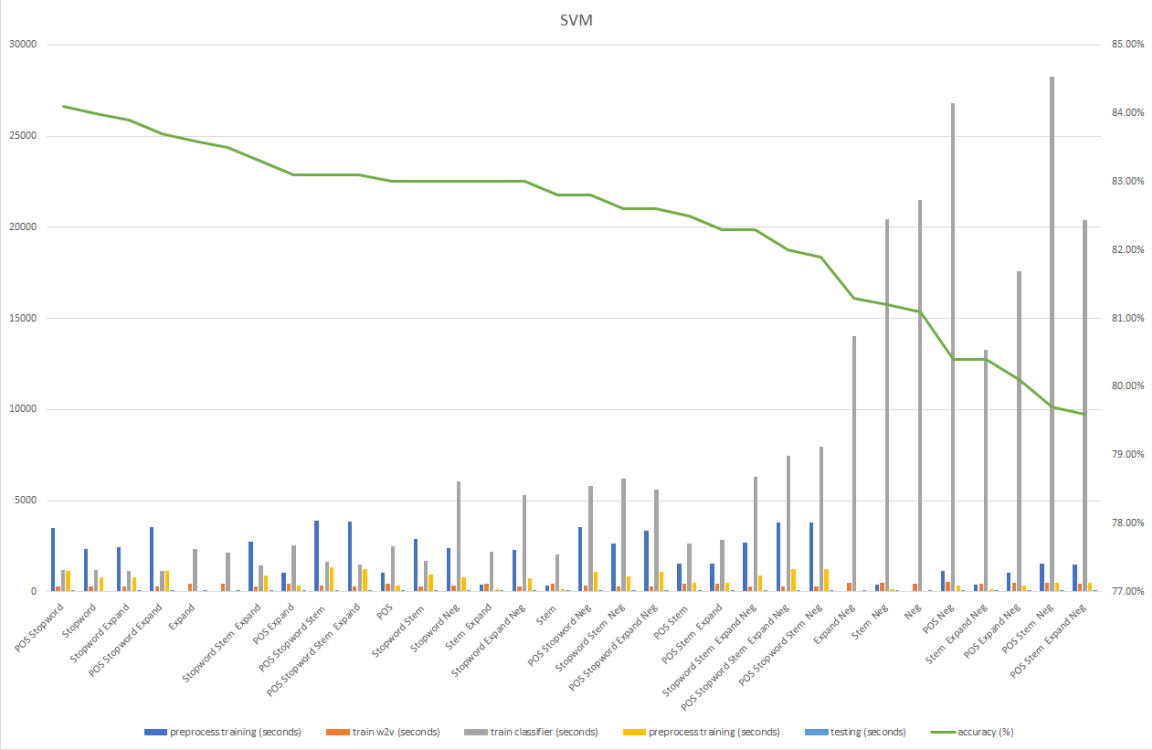

Figure 4.2: Graph of time and accuracy of SWSAPT with SVM

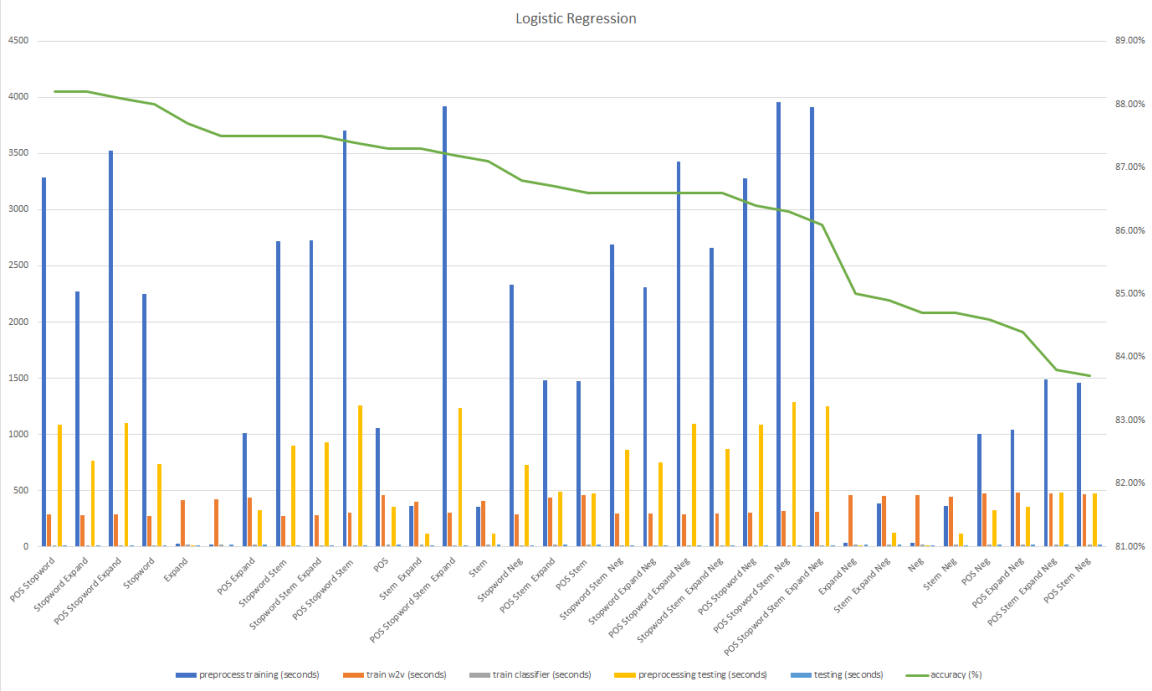

Figure 4.3: Graph of time and accuracy of SWSAPT with Logistic Regression

From the timings, we see that enabling stopword filtering increases the time from a couple minutes to nearly an hour. With the SVM chart, we see that enabling more preprocessing techniques affects the classifier and increase the amount of time it takes to train the classifier.

We know that that including stopword filtering increases the accuracy. Furthermore, any additional word level preprocessing techniques would reduce the accuracy rating. We have yet to investigate the sentence level preprocessing technique of normalization of the sentence vector. Therefore, we tested all the classifiers with and without stopword filtering and with or without sentence normalization.

The results of that experiment are shown in the table below.

Table 4.14: Accuracy measurements for different experiments using Word2Vec and classifiers

Method	Accuracy
Logistic Regression with stopword filtering and normalization	88.02%
Logistic Regression with stopword filtering	88.0%
Logistic Regression	87.536%
Logistic Regression with normalization without stopword filtering	86.74%
SVM	83.092%
SVM with stopword filtering	82.924%
SVM with stopword filtering and normalization	72.048%
Naive Bayes with stopword filtering and normalization	71.288%
Random Forest with stopword filtering and normalization	68.412%
Naive Bayes with normalization without stopword filtering	65.036%
Random Forest with normalization without stopword filtering	64.312%
Naive Bayes	64.28%
Random Forest	62.92%
Random Forest with stopword filtering	62.228%
SVM with normalization without stopword filtering	61.736%
Naive Bayes with stopword filtering	57.192%

From the table, we can see that Logistic Regression with stopword removal and normalization gives us the best accuracy with 88.02%. We also found that stopword removal and normalization increases the accuracy rating. Stopword removal removes excess vectors from being added into the result vector, and normalization ensures that no matter the sentence length, it results in all the vectors trained in the classifier to be the same length.

Another check before moving onto the Digital Democracy experiment is to ensure that the classifier was not overfitting on the IMDB dataset. The previous experiments were conducted with the Word2Vec model and classifier training on the same dataset. So we conducted another set of tests using the Google News' vector for Word2Vec.

Table 4.15: Accuracy measurements for different experiments using Word2Vec and classifiers

Method	Accuracy
Logistic Regression	86.012%
SVM	82.924%
Random Forest	63.204%
Naive Bayes	57.19%

4.5 Word2Vec Sentiment Analysis with Digital Democracy

This experiment clarifies that the validation on the previous sentiment methods were not simply tailored towards movie reviews. The experiment runs similarly to the supervised Word2Vec sentiment analysis with preprocessing technique validation experiment. However, the training dataset is a political dataset from Digital Democracy [9]. This political dataset includes quotes from politicians agreeing or disagreeing with California legislative bills. These quotes can be found in Appendix A, and each quote has their sentiment determined. Once TONGS has determined sentiments for all the utterances, we randomly selected 100 quotes to be reviewed by 32 professionals. These professionals were given 10 sentences and their associated sentiment and judged if the associated sentiment was either "correct", "incorrect", or "no idea". The majority vote designated the sentiment of the quote.

The overall accuracy compared to the TONGS' output was 71% accurate. Out

of one hundred quotes, 71 were correctly labeled by TONGS, 26 were wrong and 2 were tied for positive and negative sentiment. Y confusion matrix for the experiment is located in the table below:

Table 4.16: Digital Democracy Experiment Majority Voting Confusion Matrix

	Classified Positive	Classified Negative
Actual Positive	41	16
Actual Negative	11	30

Another type of voting that was investigated was consensus voting. Consensus voting is the situation in which a quote had a majority vote in a certain sentiment, and the opposite sentiment did not have more than one. Below is the confusion matrix for that:

Table 4.17: Digital Democracy Experiment Consensus Voting Confusion Matrix

	Classified Positive	Classified Negative
Actual Positive	22	13
Actual Negative	10	32

One thing to note is that 63 of the quotes were biased towards a certain sentiment. Another notable observation is that 17 quotes had a high count of "I don't know." This sentiment engine exceeded expectations of detecting sentiment where the quote had either none or a low count of "I don't know". The engine also did a better job of detecting sentiment of quotes with higher positive to negative votes. The lower the ratio, the more errors TONGS makes. Also, when there is a higher count of "I don't know," TONGS has a lower accuracy rate.

Chapter 5

FUTURE WORK AND CONCLUSION

As the field of natural language processing and sentiment analysis grows, new techniques and methods will be discovered. What should and should not be used is recorded and used for future work. This thesis presented TONGS as a test to see if Word2Vec is feasible to use for sentiment analysis.

Although Word2Vec failed to compute sentiment based on a word's vector space representation, TONGS has shown that sentiment analysis is feasible with Word2Vec. The accuracy of TONGS scores similarly to its counterpart from Pang and Lee's experiments. The word embeddings returned from the Word2Vec model do not produce usable vector space models to traverse and find specific positive and negative words in a given direction. Nor were there clusters of words with sentiment that could be easily grouped. Unsupervised classification with Word2Vec can be used for sentiment analysis and will be on the same accuracy rating as other unsupervised classification models.

Supervised learning was experimented on using multiple algorithms. The algorithms are Naive Bayes, Random Forest, Support Vector Machine, and Logistic Regression. It was shown that logistic regression with stopword removal and normalization produced the best results. In other research, preprocessing has been shown to improve accuracy, but preprocessing the input data does not improve the accuracy rating with Word2Vec.

The final experiment tested sentiment analysis in previously not studied data. The Word2Vec model used was trained with Google News' vector, the classifier was trained on IMDB movie reviews, and the data to be tested was political data. This political

data was provided in collaboration with the Digital Democracy project. TONGS classified the given utterance and a subset was presented on a survey for people to vote on the sentiment. The sentiment was then later compared to see the accuracy of TONGS, which was 72%.

Overall, Word2Vec can be used for sentiment analysis and is capable of working with themes from different datasets.

5.1 Future Work

TONGS only utilized the Gensim implementation of Word2Vec. Google's TensorFlow may have more toggles to increase the accuracy rate of sentiment analysis with Word2Vec [1].

There are other models for representing words as vectors. TONGS used Word2Vec, but Stanford also has an unsupervised learning algorithm for obtaining vector representation of words called GloVe: Global Vectors for Word Representation [27].

TONGS conducted its experiments based on binary classification of sentiment, in which sentences were classified. Sentences were classified as either positive or negative. There is more to sentiment than positive and negative. SentiWordNet is similar to WordNet but creates three sentiment scores, positive, negative, and objectivity [7].

TONGS also only focused on the sentence level of sentiment. Sentiment analysis is also applicable on documents. There is another form of Word2Vec for documents called Doc2Vec. Further research can investigate feasibility of using Doc2Vec for classifying documents [29].

BIBLIOGRAPHY

[1] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[2] H. Adeli and S.-L. Hung. *Machine learning: neural networks, genetic algorithms, and fuzzy systems*. John Wiley & Sons, Inc., 1994.

[3] M. Annett and G. Kondrak. A comparison of sentiment analysis techniques: Polarizing movie blogs. In *Proceedings of the Twenty-First Canadian Conference on Artificial Intelligence*, pages 25–35, 2008.

[4] M. M. Bradley, P. J. Lang, M. M. Bradley, and P. J. Lang. Affective norms for english words (anew): Instruction manual and affective ratings, 1999.

[5] L. Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[6] P. Ekman and W. V. Friesen. Constants across cultures in the face and emotion. *Journal of Personality and Social Psychology*, 17(2):124–129, 1971.

[7] A. Esuli and F. Sebastiani. Sentiwordnet: A publicly available lexical resource for opinion mining. In *In Proceedings of the 5th Conference on Language Resources and Evaluation (LREC06*, pages 417–422, 2006.

[8] Facebook. Facebook second quarter 2017 results, 2017.

[9] T. I. for Advanced Technology and P. Policy. Digital democracy, 2012.

[10] S. G. Hadi Puransari. Deep learning for sentiment analysis of movie reviews. Stanford University, 2014.

[11] T. K. Ho. The random subspace method for constructing decision forests. *IEEE transactions on pattern analysis and machine intelligence*, 20(8):832–844, 1998.

[12] M. Hu and B. Liu. Mining and summarizing customer reviews. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '04, pages 168–177, New York, NY, USA, 2004. ACM.

[13] P. H. Lindsay and D. A. Normal. *Human Information Processing: An Introduction to Psychology.* Academic Press, 1977.

[14] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, pages 142–150, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics.

[15] C. D. Manning and H. Schütze. *Foundations of Statistical Natural Language Processing.* MIT Press, Cambridge, MA, USA, 1999.

[16] M. Marcus, G. Kim, M. A. Marcinkiewicz, R. MacIntyre, A. Bies, M. Ferguson, K. Katz, and B. Schasberger. The penn treebank: Annotating predicate argument structure. In *Proceedings of the Workshop on Human Language Technology*, HLT '94, pages 114–119, Stroudsburg, PA, USA, 1994. Association for Computational Linguistics.

[17] S. McCloud. *Understanding Comics.* A Kitchen Sink book. HarperCollins, 1994.

[18] P. McCullagh. Generalized linear models. *European Journal of Operational Research*, 16(3):285–292, 1984.

[19] Merriam-Webster. Sentence.

[20] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781, 2013.

[21] A. Mudinas, D. Zhang, and M. Levene. Combining lexicon and learning based approaches for concept-level sentiment analysis. In *Proceedings of the First International Workshop on Issues of Sentiment Discovery and Opinion Mining*, WISDOM '12, pages 5:1–5:8, New York, NY, USA, 2012. ACM.

[22] T. Nasukawa and J. Yi. Sentiment analysis: Capturing favorability using natural language processing. In *Proceedings of the 2nd international conference on Knowledge capture*, pages 70–77. ACM, 2003.

[23] M. Nielsen. Using neural nets to recognize handwritten digits.

[24] OpenCV. Introduction to support vector machines.

[25] B. Pang and L. Lee. Opinion mining and sentiment analysis. *Found. Trends Inf. Retr.*, 2(1-2):1–135, Jan. 2008.

[26] B. Pang, L. Lee, and S. Vaithyanathan. Thumbs up?: Sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing - Volume 10*, EMNLP '02, pages 79–86, Stroudsburg, PA, USA, 2002. Association for Computational Linguistics.

[27] J. Pennington, R. Socher, and C. D. Manning. Glove: Global vectors for word representation. In *EMNLP*, volume 14, pages 1532–1543, 2014.

[28] R. Plutchik. *Emotion: A Psychoevolutionary Synthesis*. Harper and Row, 1980.

[29] R. Rehurek. Gensim, 2009.

[30] I. L. Stats. Twitter usage statistics, 2017.

[31] P. J. Stone and E. B. Hunt. A computer approach to content analysis: Studies using the general inquirer system. In *Proceedings of the May 21-23, 1963, Spring Joint Computer Conference*, AFIPS '63 (Spring), pages 241–256, New York, NY, USA, 1963. ACM.

[32] P. D. Turney. Thumbs up or thumbs down?: Semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, pages 417–424, Stroudsburg, PA, USA, 2002. Association for Computational Linguistics.

[33] L. van der Maaten and G. E. Hinton. Visualizing high-dimensional data using t-sne. *Journal of Machine Learning Research*, 9:2579–2605, 2008.

[34] B. Xue, C. Fu, and Z. Shaobin. A study on sentiment computing and classification of sina weibo with word2vec. In *Proceedings of the 2014 IEEE International Congress on Big Data*, BIGDATACONGRESS '14, pages 358–363, Washington, DC, USA, 2014. IEEE Computer Society.

APPENDICES

Appendix A

DIGITAL DEMOCRACY

Table A.1: Quotes used in Digital Democracy survey

Id	Quote
1	I request an audit, there's gonna be an audit done by the State Auditor. If the Audit Committee, which I'm a member of approves it, then it will go subsequent to develop with my colleagues in human services here, some legislation to put the coordination back where it should be.
2	check in about how we are doing and, it, you know, how our students are doing, and I know no parents, etc. And I know for me in my district, the LCAP process was a little bumpy. Not exactly what you would wish for.
3	So, we're listening to your answers to the questions that were proposed. It doesn't seem as if we really do have a handle on seismic safety for our state hospital structure. We have some anecdotal information, but we really don't know what the picture.
4	I think, as we look forward to the next session here, and challenges and opportunities, I think this will become apparent. I do have, actually, one question for Mr. Brown. You said that you currently connected 300 facilities. In a perfect world, how far do we have to go?

5	And reality is actions that community members, community leaders are being forced to take. The leader of STEP issued a letter to her local regional center, June 19th. Stating that she was no longer able to accept referrals. This is an organization that has successfully placed residents from developmental
6	In terms of not only the state, but also localities also suffered, made cuts to public health infrastructure during the recession. And you've listed the litany of infectious disease threats that we're facing right now.
7	What concerns me about the way this is structured however is that we agreed a year ago in initially establishing the cap and trade program that 60% of the funds would be continually appropriated. And 40% would be available for the legislature at our discretion to annually allocate in reviewing the governor's proposal and our own desires.
8	The bottom line is this, recounts only very rarely change election results, and when there is no financial incentive to end a recount when tax payers are footing the bill, we're going to get a lot more recounts with elections that are unlikely to change.
9	So I'd ask the author to take a look at that and see if there was some, perhaps the prohibition should not apply until the hosting platform has been notified that there is a problem. I didn't see that in the bill. Thank you, Mr. Chair.
10	and addictive nicotine is to begin to regulate the product as a tobacco product. And they are tobacco products, not just by definition of this bill, but by definition of federal law. And, proposed regulations that the food and drug administration have out right now and will be formalized later this calendar year. Centers for Disease Control along with our

11	Although what you clarified right now I think helps alleviate some of the concerns on the issue, but why don't we hold this item open. If you don't mind staying here and Mr Cooper you made a motion.
12	Well there's some, there's some states that don't allow the Tesla to be purchased by their, their members. So Texas is the big, is the big culprit here. So if you're a Texan and you want to buy a made in California Tesla, you can't do it.
13	I will be relatively brief. Key word is relatively. SB 559 adds regular full-time lifeguards employed by Imperial Beach, the City of Imperial Beach to the list of public safety employees entitled to a benefit known is 4850 leave. Under Labor Code section 4850, current law provides a leave of absence of up to one year with full salary and benefits for specified employees who have been injured on-the-job and become disabled or unable to perform their job responsibilities as described in their job description.
14	Well, the Chair's recommended do pass with the current amendment. So the Chair's not gonna be requiring the bill to come back to the committee. If there are additional amendments, those amendments will be considered as it moves forward to the next committee, and then to the floor. I hope I gave you the short answer.
15	point is that I hope that we are not constrained in our discussions about what we can do in the future, because we really first of all need to look at the needs. And we're not gonna say because we can only negotiate on a new deal on revising MCO or whatever and that brings in, who knows, let's say $500 million less money than we got in past years.

16	So, I just want to, again, the distinction that I'm suggesting that we have a conversation about to see if you might be willing to take is, rather than looking at the increase from the previous year, the language would then read such as, "starting from 2017".
17	Okay just wanted that on. On the record. Just so the committee knows, the physicians did ask for more amendments. One regarded the business practices, the models that were referred to as the AOC and the IPA.
18	SB 573 will allow the state to open its data in a secure in efficient manner which has the potential to vastly improve our government's efficiency in accountability, and foster economic development.
19	One good point that is made about the Career Tech Ed is that Career Tech Ed has embraced excellence in education and standards and those are important and should be built-in to this.
20	Thank you, Mr. Vice Chairman. I would say in response to some of the comments, and I'm happy to answer any questions or take comments from your members. I think were all sort in an agreement that this is an important way to approach addressing recidivism and providing significant resources that are necessary to make someone successful.
21	So, I think that this bill is appropriate for the time. I think the greenhouse gas reduction fund is appropriate to use. We can manage our forest effectively and reduce carbon and help the wildlife and increase the water yields. I have with me, Julee Malinowski-Ball from the Biomass industry.
22	From Assembly Member Mike Gipson's district. Please welcome Doctor Reginald Pope. Doctor Pope has been the Pastor of Bethel Missionary Baptist Church in the Watts community of Los Angeles since 1972. He was a trail blazer in helping to reduce crime during the Watts riots.

23	We're grateful that Dr. Karen Baylor is here, she's Deputy Director of Mental Health and Substance Abuse Services with the Department of Health Care Services. We're grateful that Dina Kokkos-Gonzales is here. She's Chief in the Mental Health Services Division, Department of Health Care Services.
24	Chair, before we get to opposition, could I ask a question to the author? So, clearly AB-32 got our ball rolling, and from that it made sense that we would establish a renewable portfolio standard. It would make sense that we deal with each of the sectors that contribute greenhouse gas emissions.
25	Prior to serving, joining the Senate, I served as the County Supervisor in that capacity. As some of you might know, I served on CalOptima, which is Orange County's organized health system.
26	So I would say is, first of all, I appreciate looking at outcomes. As someone whose been involved in quality improvement in my own field I know how important outcomes are, but as we look at quality improvement we recognize, you know you have outcomes, and then you also need to look at inputs that go into outcomes.
27	And this year SB 612 builds on efforts of the previous past two years where we want to bring clarity and consistency on some of the more problematic issues that have necessitated additional conversations with stake holders, including issues that have arisen with some of the other programs CUPAs are charged with overseeing.
28	So it's clearly going forward given California's long term water challenges, the State will need to access deeper aquifers and adopt and develop the technology necessary to utilize water, that we might not consider feasible today.

29	Selma was many miles down a very long highway stretch. There's a federal park along that road that commemorates the respite tent city that marchers made on the journey for freedom and access to the ballot box.
30	And I think it's because of this word called affirmative action. JFK and Executive Order 10, 109, 25, and 61, he ordered the government contractors to take affirmative action to ensure applicants are employed and employees are treated fairly during employment without regard to their race, creed, color or natural origin.
31	Great. Thank you. We will go to now the remaining parts of the agenda. The consent calendar includes eight items, today. Are any questions on the consent calendar items? Questions or comments? If not, a motion is an order on the eight items on the consent calendar. There's a motion and second. Roll call please.
32	I encourage you to ask a member of your staff just to kinda learn what's there and take note of it as a general matter of public policy. I think the opportunity for our group of legislators is to kinda understand what is the state of the law, which is bequeathed to us by our predecessors.
33	Sounds like the votes may be here and may not who knows but I'm saying are you willing to work, willing to work with Mr. Gordon, he's the one, I rectify this.
34	Many of us have not had to risk losing our job to care for a loved one with a serious illness, but life happens. And we wanna ensure that Californians can keep their jobs and keep contributing to our economy, while taking on the added burden and responsibility.

35	This esteemed delegation is visiting Sacramento as participants of the US state department's International Visitor Leadership Program to learn about US politics and elections law and economic development in innovation in California.
36	In the past two years, I've worked with Cindy on several issues. She brings a wealth of knowledge and enthusiasm to the district. Most importantly, Cindy cares about the kids, and she knows what makes kids successful in the classroom.
37	Thank you very much, Mr. Pro Tem. Thank you for the time we've spent together. As I listened to the Pro Tem's last question, I think it really is important and I think we spend a lot of time in this committee digging deeply into performance metrics of various department heads, but it really is a matter of executive management.
38	I would like to see the HSA be deductible in California, which it is not currently as I understand it. To give people a tool to be able to do that. That would take away the argument that this somehow benefits employers, but I believe it also would stabilize things and give employees an opportunity to participate in these plans.
39	Thank you, and of course you would make recommendations for change and allocation in the following fiscal year. Can you at some point give us some of those methods that the state uses to go after grant recipients that don't perform?
40	And so, we'll be talking about this alarming trend, today as well as, ways how we can make sure we increase funding for the system to serve more students as we look to increase enrollment at all systems of higher education to serve the workforce needs of the, of the 21st century.

41	And of course anybody on the panel, this can be a bill that you have. You can talk about when the bills you have or perhaps a bill you'd like this committee tackle or just go over some of the jurisdiction.
42	So first, I wanna thank you for the comprehensive overview and certainly the department is the umbrella for multiple programs and campaigns. You did focus on the current efforts around measles and protasis. Do you have your fingertips today just a quick update on where we are with either of those and what the department's doing to promote appropriate vaccination protocols?
43	this year is a century of black life, history, and culture which highlights the great accomplishments of African-Americans. African-Americans who were able to overcome segregation and racism to achieve what to some may seem impossible and who were more times than not denied their basic civil and human rights, but still played an integral role in shaping history of this nation. Whether it's Marshall Major Taylor, a world champion cyclist in 1899 who had to
44	Lots of things that we all know in our hearts use our energy and are very restorative. Thought we might wanna look at something like that. There are methods of doing this kinda cost benefit analysis that could show that it is very cost effective.
45	He grew up work as a lifeguard, and in fact set a record. He saved 79 lives as a lifeguard. Many those were young women so I think, you know, they may not have been all that serious, but it's official tally of 79. From there, obviously, we all know he was a broadcaster.
46	Having a capitated system, they can be more efficient, bless you, they can be more efficient. They come in, not only see them that day, they can see that same patient for a different problem that same day as well, and make sure that that patient is taken care of.

47	I don't know if my statistics are right, but I think the general thought is there, so. We have Jessica Peters, thank you for coming Jessica, from the Legislative Analyst's Office, and you're presenting us with your report, so why don't you proceed. Thank you.
48	I don't have an analysis of how this will be all impacted, and how we find that sweet spot, but I think they're points are very valid, and my goal is to create something that's better for everyone involved.
49	And the device states that, it sounds like it tells you a lot. When you're using it, so it reminds you to call 911. So it's part of the process.
50	The Senate Bill 117 extends sunset of the existing law that allows alarm companies to form limited liability companies. I'm accepting the amendment set forth in your committee report that will shorten the period from five years to three years.
51	I think so. Let's do all three cuz I'm sure the folks who are lined up want to probably touch on all three issues, so we'll do public comment all at once. So keep rolling.
52	I have to say that I would come down in favor of the ABA. Because in my view, a three-year limited term funding needs to allow all districts. Large, small, medium, whatever, to plan with some idea of stability. And I've experienced when I worked for the US Department of Education.
53	A petition was presented to State Water Resources Control Board, where we also had over 100 plus entities, individuals from the local community, coming before the water board to ensure that the state's attention to this issue becomes a priority.

54	Thank you very much. We want to ask my colleagues to please just introduce themselves very briefly, and then we can get started and commence with the business before us. We're gonna start on my right.
55	On sunset review and please, we'll invite the California Architects Board and Landscape Architects Technical Committee to please come forward and make yourselves as comfortable as possible, as you can in this room. Thank you for coming this afternoon and being willing to participate.
56	Mr. Chairman and members, thank you for the opportunity to be here today. AB 41 requires the Department of Community Services and Development CSD to recommend a plan for state-wide low-income water rate assistance program.
57	Makes it simpler so that we save that money and put it back into services to allow these kids to be all day in the daycare program run by Kidango.
58	He was loyal to his family. He was loyal to this country. He was loyal to his party and he as a man was well loved by Doris, by his children, Marsha Ward, Peggy Macini, and had five grandchildren and one great-grandchild and also his sister Jene Baker of Camario. I ask that we adjourn in the honor of a wonderful man, a wonderful husband and father and grandfather, Buzz Folder.
59	And so, this bill simply expands the state preschool program to insure that every low income four-year-old has the opportunity to attend preschool following up on, on last year's budget act.
60	Yeah, well, thank you Miss Weber and Miss Huffman for your, for your testimony and Ms. Weber for the bill. Totally support it. It's really important that we work harder on this.

61	Well, really, really appreciate your very, very thoughtful approach and the issues brought up. I've learned a lot from this. So thank you so much, to the two of you. Thank you. Alright, let's turn on now to kind of looking, expanding out a bit to really deep dive into this voter turnout in Los Angeles County issue.
62	Thanks very much Mr. Chair. Thank you very much for allowing me to present Assembly Bill 883. This bill simply prohibits employers from publishing or posting a job advertisement that disqualifies an applicant who is currently or was at one point a public employee.
63	Thank you Mr. Chair and Members. I really appreciate your support I look for this measure moving forward and being able to work with every single one of you I respectfully ask for your aye vote.
64	Under SB 353, counties will have additional funds to address these issues. The funds are flexible, allowing each county to address the public safety needs most critical to its residents.
65	Mr. Speaker and members. I know this bill really took a lot of time to finally get here, a little bit of turmoil, but finally made it here to this floor. So I'm very appreciative to get here, but like a fine distilled beverage it takes a little time it takes a little age and now it's finally here for us. It's peanut butter jelly time. I urge your support for this important bill. Thank you.
66	Thank you Mr. Chair, this bill is important, necessary. It changes federal law and we appreciate your vote. This is Andy Foster, the executive director of the commission who's here and is doing such an outstanding job. That's why this is such a good-looking bill today.

67	And many of these students face multiple barriers and early education is critical and my district intends to demonstrate the effectiveness of early learning and community engagement. This bill would effectively create an all day program for the preschool program there at no cost to the State of California.
68	I appreciate you for bringing this comment forward. I think it's something that the Education Committee ought to be looking at carefully as we talk about how we're going to basically create a school district that's going to be responsive to the needs of our children and put us in a position where we can be proud of every school that we have in, in the state of California.
69	And you have my commitment to definitely take this into serious consideration, and we respect that and we make it easier for that process to work. I do appreciate these concerns, and I think that they merit our attention.
70	Yeah, Richard Mullen at Cal Iso was the potential exportation of thermal coal in event that the merger happens. Obviously the IOUs have extremely limited coal exportation to California to generate electricity we have some MOUs obviously, Los Angeles, primarily LADWP.
71	Well-intentioned people worked on this bill, and now we've come to a point of where things that people have spoken about tonight on the floor, they're not in there anymore, it's not there anymore.
72	The law provides that breach notifications must be written in plain language, but is otherwise silent about how the information should be presented or organized. Unfortunately, data breach notices are often written in a way that obscures the information they contain, making them ineffective in communicating critical information to affected California residents.

73	So again if people want to do bills that's like the Sacramento Kings bill what they would want to do is identify a specific project, go to the supporters in opposition, have them meet together come up with the mitigation measures, this is not been done in either case. That is the reason for our noe recommendation.
74	And, so that's the problem I'm still frustrated with. I'm not sure there's an answer, but at least let's try to be fair to those people and those boards that they're not bearing even additional costs for failures in the system.
75	So they're kind of protected as they go along with regards to who can and cannot sign for their loans and all those kinds of things that's really unfortunate that happens. But unfortunately, it happens to those who can least afford to have this in their lives, to have a debt of 30, $40,000 and you're unemployed.
76	More incomes, right? But then, what you actually look at for care claims. You have more women taking time off to actually care for a sick child or a relative.
77	If that law is thrown out, that's gonna happen on the local level. But right now, there are laws on the books, because city councils have decided for whatever reason, and I don't, I'm not sure why they would decide that they decided to not have vacation rentals.
78	Can we remind you, we're at name, organization, and position only. You've had two substantive witnesses that have gone beyond the six minutes of allotted time. So, please just state your name, organization, and position.
79	In San Francisco like in Sonoma County, we shouldn't be forced to negotiate or in the case of Malibu, use a subpoena in a two year court fight to have big corporations simply follow local laws. Furthermore, all this bill does is ensure that those who wish to rent their properties can do so legally.

80	But I'd also like to know a little bit of maybe about the rationale for how the fee structures were set if there can be some clarity on that because certainly when we're going about this new structure for an MCO tax.
81	You don't know when a drone is over your head. And of course, they're operated remotely as well, so they can get into all sorts of areas that helicopter's not going to. So I know this is going to get to judiciary as well, and I'll be seeing– oh excuse me.
82	I can remember, I was a faculty at San Diego State for 40 years, and during that time near the end there was an effort to increase student fees and every time it came up to talk about student success fees, the students said no, because they couldn't afford additional fees.
83	Currently, they're using the public right-of-way for this task and using temporary facilities. This project has been in the envisioning process, and it's as you know, development projects often require lots of preparation.
84	Senator Mendoza, you have 3 ayes, 5 nos. The majors on call. You have some members absent. So we'll leave it on call until the end of the hearing. If at the end of the hearing it does fail, would you appreciate reconsideration?
85	A recent survey of 46 100% affordable housing projects in the Bay Area, found that about 2,000 parking spaces out of 5,600 went unused. Those unused parking spaces cost well over $100 million to build. The funding for affordable housing comes from public sources, including voter bonds and low income housing credits.

86	That said, I share many of the concerns raised today of how words do matter in the way formulate resolutions that may seem well intentioned or innocuous can be misinterpreted or used for other purposes and we've heard some, I think, important examples that are not theoretical. That's what's playing out unfortunately in higher education.
87	How can we give a secure feeling to those that have traumatic brain injury that want to get back into the workforce, want to be independent once again without the fear that they're going to have to go through mazes of federal and state bureaucracies to get back on assistance if they need to.
88	48% of foster youth are given antidepressants that have an FDA black box label warning for use by children. We, the state of California, assume the role of the parent for foster youth. It's the job of the Legislature to address this problem for a voice that doesn't often get heard.
89	That didn't happen. That didn't happen because people became myopic in a kind of me, me, where's my money kinda attitude, or just don't want it. And there's nothing, absolutely nothing, no matter how you bend over backwards, to satisfy them.
90	So I think to draw the comparison is not totally correct because you can be elected with no education and fill that role. In fact, I think there's somebody running for president that doesn't even have a college degree. But you cannot probably be a superintendent or a city manager without a number of degrees and special skills that do that. So, I think the comparison isn't appropriate. So, thank you, sir.
91	High-level question for you, I know it's probably a litle more complicated than my question but, what level of resources do you think we need to be investing as a state to start to reverse the affordable housing crisis?

92	I happen to support high quality, low risk securities in my life. And certainly when it comes to the tax payer dollar, or students, or agencies, I happen to support that. I think we've gone awry with risk and risk-taking behavior, and finance at the local government level occasionally. We know about that, and at the state as well.
93	I guess have a question, because most parents put their kids in kindergarten because they're ready to have a little break there for either that half or full day.
94	That remains solid, if that's something that we want to eliminate, then that would be appropriate to me in a natural environment. I guess that's, I don't think we should look for the future in a pathway.
95	It does. Yeah, I think that's helpful. I could drill down on that a little bit. Just the region I'm in and I guess the final question, is the department with these resources, responsive to either local health departments? Or even, what if there were a neighborhood or a citizen's group concerned about clusters of birth defects? Is there a responsive mechanism in this? Or is it just based on your risk strategy vectors around the state?
96	Or in Orange County its only 620,000. We as senators represent a million should we then split the senators and add 40 more senators? To me, I get the LA County part. But putting San Bernardino, Orange County, Riverside County on the same boat, that we only represent half a million people is, to me it is so unnecessary.
97	But we've never had a consistent program that really addresses this population, and as a result, we often wonder why is it that students who are born in this country, who speak English, actually score less on test scores than kids who are coming into the country not speaking English.

98	Yeah, I would just like to applaud the fact that we're attempting to make it more difficult for people who have this problem and this compulsion. However, my biggest concern happens to be with the fact that generics are not able to be controlled and take advantage of this particular.
99	The financial burden of children, of child, of childcare becomes more of a reality for family I'm describing, especially living in an urban area, such as Orange County, Los Angeles, San Diego, or even San Francisco.
100	We came into Birmingham where the fabric of a community was torn suddenly with the killing of those four little girls whose only crimes were the color of skin and showing up to the 16th Street Baptist Church for Sunday school.

Table A.2: Results of Digital Democracy survey

Id	Positive	Negative	I don't know	TONGS' sentiment
1	0	4	0	1
2	0	4	0	1
3	0	5	2	1
4	0	5	2	1
5	0	5	1	1
6	0	7	0	1
7	0	8	1	1
8	0	8	2	1
9	1	4	3	1
10	1	5	4	1
11	1	5	3	1
12	1	7	0	1
13	2	3	6	1

14	2	4	1	1
15	3	4	2	1
16	2	1	5	0
17	2	1	5	0
18	4	0	2	0
19	4	0	0	0
20	5	0	2	0
21	5	0	1	0
22	5	0	2	0
23	6	0	1	0
24	6	1	4	0
25	7	0	2	0
26	7	3	1	0
27	2	2	4	1
28	2	2	2	0
29	1	0	4	1
30	2	0	1	1
31	3	0	4	1
32	3	1	2	1
33	3	2	3	1
34	3	2	1	1
35	4	0	0	1
36	4	0	0	1
37	4	1	1	1
38	4	1	3	1
39	4	2	3	1

40	4	2	0	1
41	4	2	3	1
42	4	3	1	1
43	5	0	0	1
44	5	0	2	1
45	5	1	0	1
46	5	1	0	1
47	5	2	3	1
48	5	3	0	1
49	5	3	7	1
50	6	0	3	1
51	6	0	0	1
52	6	1	1	1
53	6	1	3	1
54	7	0	1	1
55	7	0	1	1
56	7	0	3	1
57	7	0	2	1
58	7	0	0	1
59	7	1	1	1
60	8	0	1	1
61	8	0	1	1
62	8	0	2	1
63	8	0	0	1
64	9	0	1	1
65	9	1	0	1

66	11	0	0	1
67	11	0	1	1
68	11	1	0	1
69	11	1	1	1
70	0	2	6	0
71	0	2	1	0
72	0	4	1	0
73	0	4	3	0
74	0	5	0	0
75	0	6	1	0
76	0	7	4	0
77	0	7	3	0
78	0	7	1	0
79	0	9	0	0
80	1	2	4	0
81	1	3	0	0
82	1	3	1	0
83	1	3	2	0
84	1	4	0	0
85	1	5	1	0
86	1	6	2	0
87	1	7	3	0
88	1	7	0	0
89	1	8	0	0
90	1	8	1	0
91	1	8	4	0

92	2	3	0	0
93	2	3	4	0
94	2	5	3	0
95	2	5	0	0
96	2	10	1	0
97	2	10	0	0
98	3	4	0	0
99	3	6	2	0
100	0	11	0	1

Appendix B

ADDITIONAL SUPERVISED WORD2VEC SENTIMENT ANALYSIS WITH PREPROCESSING TECHNIQUES EXPERIMENTS

Table B.1: Accuracy measures for different combinations of preprocessing techniques using Word2Vec with Naive Bayes

POS Tagging	Stopword Filtering	Stemming	Expanding Contractions	Negation Tagging	Accuracy
	X		X		72.3%
	X				72.2%
X	X		X		72.1%
	X	X	X		72%
X	X				71.6%
	X	X			71.3%
X	X	X	X		70.3%
X	X	X			69.5%
					65.5%
		X			65.3%
		X	X		65.2%
			X		65.1%
X			X		64.2%
X					63.8%
X		X	X		61.9%
X		X			61.7%
X	X	X		X	60.6%
	X	X		X	59.7%
	X	X	X	X	58.9%
X	X		X	X	57.4%
X	X			X	56.9%
X	X	X	X	X	56.8%
	X		X	X	56.7%
	X			X	56.6%
			X	X	55.4%
X			X	X	55%
		X	X	X	54.5%
X		X	X	X	54.2%
				X	54.1%
		X		X	54%
X				X	53.7%
X		X		X	53.4%

Table B.2: Accuracy measures for different combinations of preprocessing techniques using Word2Vec with Random Forest

POS Tagging	Stopword Filtering	Stemming	Expanding Contractions	Negation Tagging	Accuracy
					67.6%
X					67.2%
	X				66.8%
		X			66.7%
			X		66.2%
				X	66.2%
X	X				65.3%
X		X			64.9%
X			X		64.3%
X				X	64.3%
	X	X			64.2%
	X		X		64%
	X			X	63.9%
		X	X		63.8%
		X		X	63.6%
			X	X	63.5%
X	X	X			63.2%
X	X		X		63.2%
X	X			X	63%
X		X	X		62.9%
X		X		X	62.7%
X			X	X	62.6%
	X	X	X		62.4%
	X	X		X	62.2%
	X		X	X	59.7%
		X	X	X	59.5%
X	X	X	X		59.4%
X	X	X		X	59.3%
X	X		X	X	59.2%
X		X	X	X	59%
	X	X	X	X	59%
X	X	X	X	X	58.9%

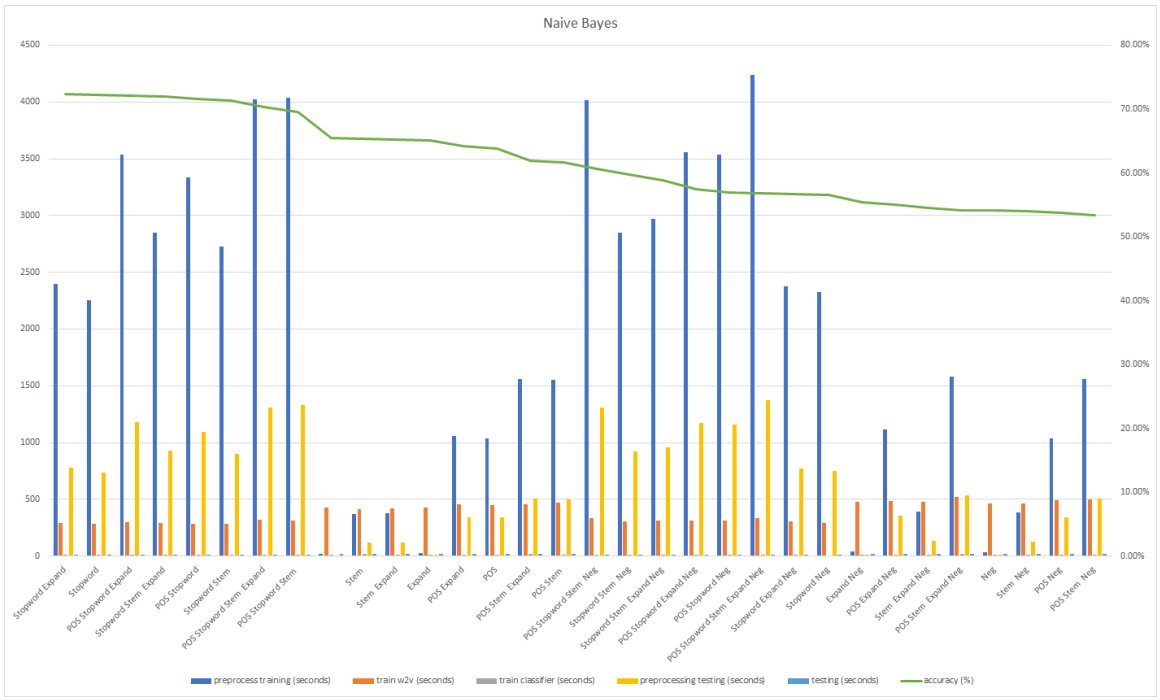

Figure B.1: Graph of time and accuracy of SWSAPT with Naive Bayes

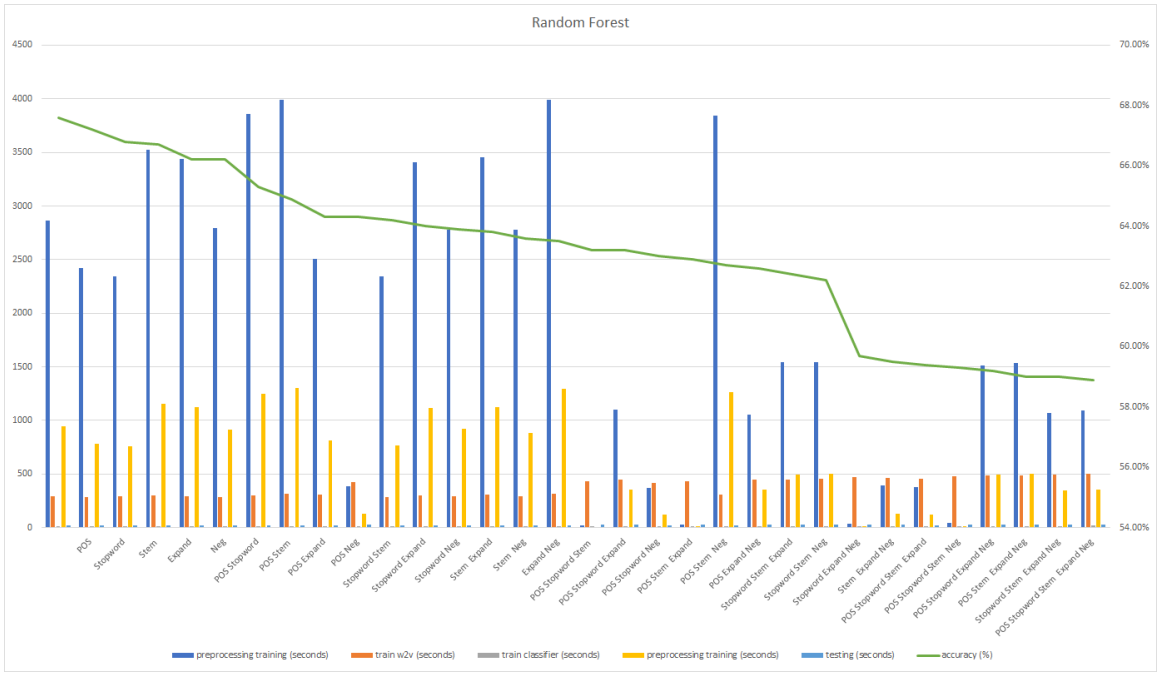

Figure B.2: Graph of time and accuracy of SWSAPT with Random Forest