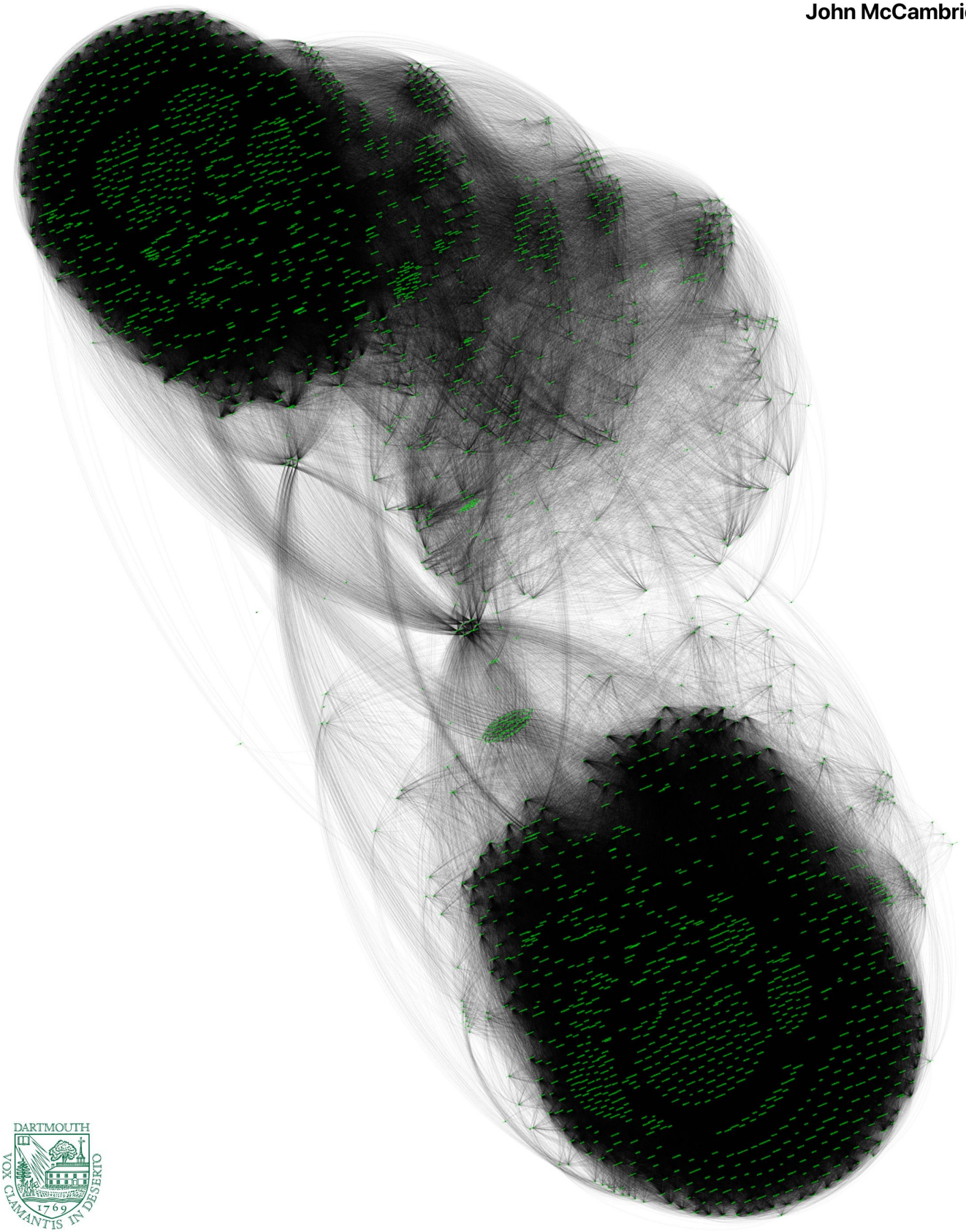
SentiGraph: a graph-based approach to determine overall passage sentiment.

John McCambridge



DARTMOUTH
VOX CLAMANTIS IN DESERTO
1769

SentiGraph: a graph-based approach to determine passage sentiment.

John McCambridge

Dartmouth College - *john.l.macdonald.22@dartmouth.edu*

Abstract

Sentiment Analysis is the fundamental understanding of the underlying meaning of text. Many pre-existing sentiment analysis engines work by classifying words into distinct categories, however, an ongoing struggle is to determine quantitatively the sentiment of a passage as a spectrum of values. Here, a novel approach of determining sentiment is introduced by building a bidirectional graph that holds words as nodes with edge weights as sentiment relationships between other words which we call a **SentiGraph**. This gives the ability to classify the sentiment of passages into certain emotional groups by finding the shortest path from words in the graph, to the respective emotional terminals of the graph. From the results, the model performs well classifying sentiment three groups of emotions: joy, anger, and sadness and has the potential to add exponentially more categories in the future.

1 Introduction

The two main approaches toward sentiment analysis are either lexicon based or supervised classification. Many existing systems such as Google's Word2Vec to accomplish this [1] with context based word-embeddings; however the nature of these models and associated vectors don't relate well when comparing sentiment. This is because these models are trained on context, and not the emotional content [2], making the comparison of such vectors (with reference to sentiment) almost impossible. The second approach utilizes Lexicons, such as the SO-CAL model [3] designed to extract the subjectivity and polarity from a passage of speech; yet, this model is constrained to only assigning a positive or negative label to a piece of text. Supervised-Learning applications involving sentiment analysis often involve the creation of a SVM (Support Vector Machine) to categorize content into specific emotional groups; these SVM models are usually trained using a large n-gram corpus along with phrases that are tagged with labels associating emotion to content. This allows the model to detect patterns where emotional language will be used and classify accordingly. A lexicon approach offers more flexibility of domain choice over that of supervised learning like Word2Vec; although models which are trained with supervised learning perform well on the information which they were trained on, the accuracy of the model drops significantly when tested on an unfamiliar domain [3]. In the case of the lexicon approach, which the SentiGraph model uses, the associated word dictionaries are collected manually. The 'EmoLex' lexicon contains over 14,000 words with 10 categories of emotions assigned to each word [4]. Work has previously been conducted predict emotional reactions within social networks [5] by extracting vectors from the 'EmoLex' lexicon, however, this technique has not been expanded for the use of sentiment analysis.

This paper plans to utilize this technique to present a novel approach in performing sentiment analysis using a word-relational graph based on the contents of the 'EmoLex' lexicon; this will allow any given word in a passage to have an sentimental value quantitatively determined by finding the closest emotional terminal. An emotional terminal is a node in the graph with no associated word, but strong polarity of emotional content - it will be used as a benchmark reading as nodes which are further away will be less sentimentally related. By utilizing this 'SentiGraph', we have created a sentiment analysis engine to analyze professor performance based on reviews left by students; this is to allow students to make better informed decisions during the college professor by better selecting their professors.

2 Methodology

2.1 Professor Review Data

The data which will be analysed in this paper originates from 'LayupList.com' [6] - a Dartmouth student run professor review site with over 25,000 reviews left by students. We obtained a collection of 5000 reviews left by students in a text format of professor, course, and review information. From this data-set, we gathered subsets of this and categorised them into 10 specific examples each relating to angry, sad, and joyful reviews left by students. In each of these categories, we truncated some of the reviews to only include the most important parts of speech removing redundant parts as to reduce computation time.

2.2 Constructing SentiGraph

The construction of the bidirectional sentiment graph begins by converting the words included in the 'EmoLex' lexicon into their respective vectors. For example, the word **hate** translates to:

$$v_{hate} = [1, 0, 1, 1, 0, 1, 0, 1, 0, 0]^T$$

This vector represents 1 for a classification in that specific category, and 0 if else. For reference, the categories inside EmoLex are cited in the references [7]. To allow for accurate comparison between vectors, every 0 element is normalized to become 0.01. Next, the emotional terminals are defined by setting each vector to their respective EmoLex classification; this step required some tuning with the methodology that they could form a basis and consequently all be linearly independent:

$$\mathbf{A}_{anger} = [1, 0, 0, 0, 0, 0, 0, 0, 0, 0]^T$$

$$\mathbf{J}_{joy} = [0, 0, 0, 0, 1, 0, 0, 0, 0, 0]^T$$

$$\mathbf{S}_{sad} = [0, 0, 0, 0, 0, 0, 0, 1, 0, 0]^T$$

For the terminals, the 0 elements are not normalized as we want words to form relationships with them only when they are extremely sentimentally similar. For each word w_i inside the emotional lexicon, we only consider non-zero vectors and classify zero vectors as neutral; the word is added as a node to the graph and the similarity between the word w_i and each of the terminals is computed using the cosine similarity.

To determine the relationship between two vectors, ϵ is defined as the cosine similarity between the vectors w_i and w_j:

$$\epsilon = \arccos\left(\frac{w_i \cdot w_j}{\|\mathbf{w_i}\|\|\mathbf{w_j}\|}\right)$$

Which can be computed as:

$$\epsilon = \arccos\left(\frac{\sum_{i=1}^{n} A_i B_j}{\sqrt{\sum_{i=1}^{n} A_i^2} * \sqrt{\sum_{i=1}^{n} B_i^2}}\right)$$

Where $A = w_i$ and $B = w_j$ for the last equation.

We define an terminal radius **R** as the radius a word similarity must be in order to create a relationship between the word and a terminal; in this case $\mathbf{R} = 0.25$ was found to be suitable.

Then, for each other node w_j in the graph we compute the Euclidean distance between w_i and w_j to determine if they are close enough to make a relationship; if the distance between the two words is smaller than the threshold **D** then an edge is formed. During testing, $\mathbf{D} = 2.0$ was found to be suitable. An edge weigh ϕ is formed between two words w_i and w_j defined as:

$$\phi = max(1 - \epsilon, 0.01)$$

The intuition behind this that as the similarity tends towards 1 the cost of traversal should tend toward 0, but traversal should never be free. If the edge weight is less than a cost threshold **T**, that is the vectors are not completely dissimilar, a relationship in the graph between the two words in formed. In testing, **T** = 0.95 was found to be suitable.

By utilizing the vector similarities of words inside EmoLex, a SentiGraph has been success-

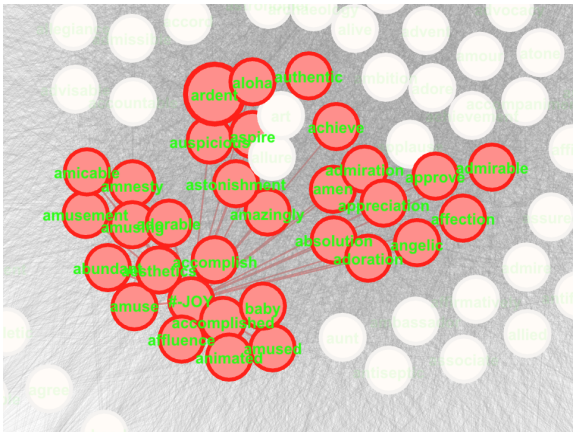


Figure 1: The terminal of joy (with node #-JOY) with surrounding words of related sentiment.

fully created with **over 6,000 words** and **over 8 million edges** between them. Each word inside the SentiGraph has a relationship to a neighbour based on sentiment and a path towards each of the three emotional terminals; the further it is away from one emotional terminal, the less emotionally related it is to that emotion.

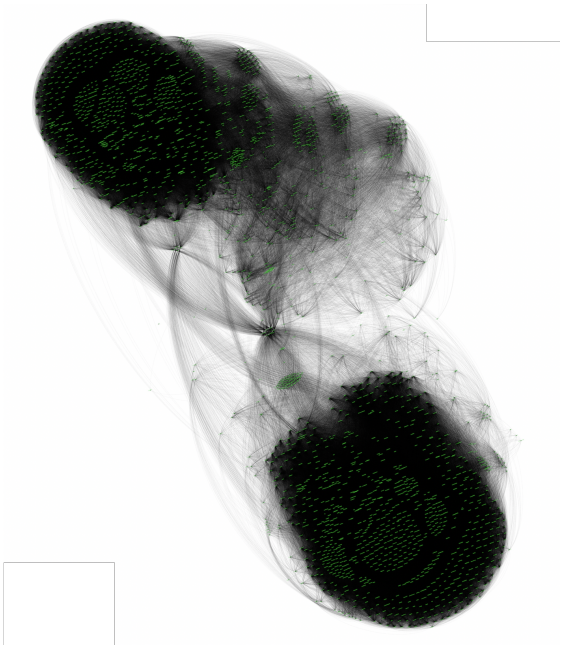


Figure 2: SentiGraph with 2,000 nodes and around 1 million edges.

2.3 Classification

We now take a review left by a student $\{w_1, w_2, ..., w_n\}$ and break it into sentences through tokenization. If the word does not exist within the graph, it is discarded. For each keyword in the tokenized sentence, the distance is computed between each of the terminals and the keyword using Djikstra's algorithm for finding the shortest distance between two nodes in a weighted graph [8]. This is computed for each of the terminals and returns a quantitative measurement of the sentiment of a specific word. For example, the word $heartless$ has a sentiment of sadness:

$$\textbf{Classify}(\text{"}heartless\text{"}) = \\ Anger : \textbf{Null}, Sad : \textbf{0.706965}, Joy : \textbf{Null}$$

We also divide the score by a γ value ($\gamma = 1.5$ if related to professor else $\gamma = 1.0$) if the sentence is related to a professor which is done by checking for title and pronoun usage; this will increase the weight of a reading specific to a professor.

After a collection of scores have been gathered for a specific passage, we extract the minimum score from each keyword (the shortest path between the keyword and terminal) compute the total sum for all terminals across the passage, and then normalize each score by dividing by the total number of occurrences for each of the labels - finally the inverse of each score is taken to compute a measure mapping larger values to more related sentiment and thus a quantitative measure of sentiment in a passage has been created.

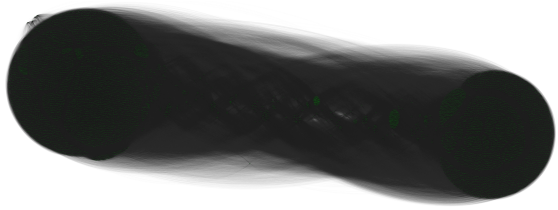


Figure 3: SentiGraph with 6387 nodes and around 8 million edges.

3 Experiments & Results

3.1 Tuning Hyper-parameters

In order to experiment with the model, the hyper-parameters associated with the model such as the terminal radius **R** or the vectors associated with the terminals were tuned; this process was tedious as the generation of a full SentiGraph takes around 20-30 minutes to create. To determine the vectors for the emotional terminals, they need to be as distinct as possible to avoid interference; to increase the hit-rate for determining the sentiment of a path, **R** was slowly increased until a viable hit-rate was reached.

3.2 Results & Performance

During the testing phase, ten excerpts were taken from the professor review data-set for each of the emotions; the model achieved a performance of accurately classifying anger **82.0**%, sadness **71.0**%, and joy **76.0**% of the time.

4 Discussion

From these results, it can be seen that the model is able to classify with sufficient accuracy categories of joy and anger, but struggles to determine the classification of items which are sad. This may be due to the polarity of the emotion of sadness being

	$\textbf{Joy}_a$	$\textbf{Ang}_a$	$\textbf{Sad}_a$
$\textbf{Joy}_p$	**8**	1	1
$\textbf{Ang}_p$	1	**9**	0
$\textbf{Sad}_p$	2	2	**6**

Figure 4: The confusion matrix generated from the testing of the classifier showing strong ability to categorise anger and joy, however less accuracy for sadness.

P_{joy} = **73**%, R_{joy} = **80**%
P_{anger} = **75**%, R_{anger} = **90**%
P_{sad} = **86**%, R_{sad} = **60**%
Overall Accuracy: 78%.

Figure 5: The precision **P** and recall **R** of the model reveals good performance for detecting joy and anger, but sub-par performance detecting sadness.

confused with other categories which can be seen above in the confusion matrix.

The results of this model have produced usable classifications for three basic categories; we have also determined a quantitative measure of sentiment for keywords in a sentence to accomplish sentiment analysis of a passage.

A moral question however must be addressed when utilizing this method in the assessment of professor performance; research has shown a strong bias existing when students leave evaluations of their female professors [9]; another important point which must be addressed is the bias in distribution of comments left by students as the reviews left by students are from those only choosing to file them, causing a bi-modal distribution rather than a more representative distribution of the student body [10].

5 Conclusion

In conclusion, the creation of a graph relating words with each other in the context of sentiment can be applicable to determining the sentiment of a passage. As an application of this, we were able to determine the sentiment of a series of reviews left by students for professors to gauge the effectiveness of a professor; however work can still be done in order to improve this model's effectiveness.

Firstly, the emotional terminals can be improved as they are quite sparse vectors and as a consequence many words cannot be capture by just three groups of emotions as certain words will have more nuanced sentiment.

Secondly, the addition of more word categories would reduce false positive as the number of possible paths from a word to a terminal would increase and give it more possibilities for classification.

Thirdly, hosting the algorithm on a cloud-based system with large computational access would be necessary as the time taken to traverse a graph will millions (and with potential improvements in lexicon size potentially billions) of edges requires sufficient access to CPU processing power.

With these changes in place, it seems probable that a graph-based sentiment analysis engine would perform best with nuanced passages to detect underlying sentiment.

5.1 In The Future...

In the SentiGraph v1.0, words are separated by their sentimental relationship, yet these words are not often contextually relevant; for the SentiGraph v2.0, the edge between two words would be a factor of not only sentiment but the probability of both words occurring within a sentence. This would establish a graph where words cluster not just by sentiment, but contextual relevance.

This would allow for the overall sentiment of a passage of text to be 'tuned' as a word like 'hate' could be replaced with 'dislike' by walking to a neighbour of hate, a neighbour of which with lower sentiment, but still being contextually relevant, revealing the ability to sentimentally adjust text.

Acknowledgments

I would like to give thanks to **Prof. Rolando A. Coto Solano** of Dartmouth College for his strong

guidance and informative criticism during the pursuit of this project.

References

[1] District Data Labs. Modern methods for sentiment analysis. *Medium*, 2017.

[2] Chris McCormick. Google's trained word2vec model in python. *https://en.wikipedia.org/wiki/Dijkstra*

[3] Maite Taboada et Al. Lexicon-Based Methods for Sentiment Analysis*, volume 1. MIT, 2010.*

[4] Saif Mohammad. Nrc word-emotion association lexicon. https://saifmohammad.com/WebPages/NRC-Emotion-Lexicon.htm*, 2016.*

[5] Jeremie Clos et Al. Predicting emotional reaction in social networks. 39th European Colloquium on Information Retrieval*, 2018.*

[6] Dartmouth student run webpage. Dartmouth's course review site. https://www.layuplist.com/*, 2020.*

[7] Emolex vector classifications (in order) are: anger, anticipation, disgust, fear, joy, negative, positive, sadness, surprise, and trust. https://saifmohammad.com/WebPages/NRC-Emotion-Lexicon.htm.

[8] Edsger W. Dijkstra. Dijkstra's shortest path algorithm. McCormickML.com*, 1956.*

[9] Kristina Mitchell et Al. Gender bias in student evaluations. Cambridge University Press*, 2018.*

[10] Rolando Coto Solano. Comment made during project discussion. Dartmouth College*, 2020.*