University of Ottawa

uOttawa

Faculty of Science

Dialog Systems in Mental Health

Thesis BSc Computer Science

Authors:
Abha Sharma
Rupsi Kaushik

Supervisor:
Caroline Barrière

December 16, 2019

Contents

1 Introduction

1.1 Background

Mental disorder is the reduced ability of a person to function effectively over a prolonged period of time. It can be caused by a significant level of distress, changes in thinking, mood, behaviour, and feelings of isolation and sadness [1]. Although there is still a stigma around mental disorders, the statistics show that they are highly prevalent. In any given year, 1 in 5 Canadians are affected by a mental disorder or an addiction problem. Moreover, by the time Canadians reach 40 years of age, 1 in 2 people will have or will have had a mental disorder [2]. Globally, 450 million people suffer from a mental disorder at any given time, which places mental disorder as one of the leading causes of ill-health and disability worldwide. Most mental illnesses can be treated through medications or psychotherapy. However, nearly two-thirds of people with mental disorders never seek treatment. In fact in Ontario itself, wait times of six months to one year are common [3].

1.2 Motivation

As therapy is being proven to be more and more effective in the treatment of mental disorders, this issue of inconvenient access has been a pressing yet neglected problem. Many people are unable to receive treatment because of the cost of treatment or simply because of physical limitations such as their place of residence or other mobility issues. In order to help facilitate the transition towards convenient mental health access and to motivate individuals to reach out in today's age, novel solutions must be introduced by leveraging a powerful tool: technology. In an attempt to provide a meaningful, intermediate phase while a person waits for professional help, we decided to create a machine learning enabled mental health chat bot, that is able to communicate with users who currently may not have anyone to confide in.

1.3 Objectives

The main objective of this project is to:

- Create a fully functioning application that is able to capture a sense of a therapy session, track moods, and provide an overview of these moods
- Further investigate the Natural Language Processing problem of capturing semantic equivalence in text within the context of Dialog Systems
- Propose a meaningful solution to this problem, evaluate the results, and dissect areas for improvement

2 Psychology Research

2.1 Cognitive Behavioural Therapy

Mood disorders such as depression and anxiety are the most common type of mental disorders [1]. Due to this reason, we decided to base our project solely on these two disorders. Among the many psychotherapy methods implemented, cognitive behavioural therapy (CBT) has proven to be the most effective while treating depression and anxiety [4]. The primary goal of CBT is to transform an individual's thought process through a structural, goal-oriented approach. In brief, the first step in CBT is identifying the problems in an individual's life and noticing the things that are bothering them the

most. Next, this approach encourages individuals to become aware of their thoughts, emotions, and behaviour patterns that are associated with those identified problems. Individuals are encouraged to assess these thoughts in order to examine how they contribute to the individual's feelings and, in turn, their behaviour and state. For example, one way anxiety can be triggered is by having an over exaggerated idea of reality. In this case, CBT would push the individual to further explore if the thoughts they are having are incorrect or exaggerated ideas of reality. The final step in CBT is to replace these incorrect thinking patterns with new patterns of thinking and shifting the way an individual looks at and interprets events [5].

In practice, CBT is usually administered by a psychotherapist or a therapist. However, many studies have shown that self-directed CBT can also be very beneficial in reducing anxiety and depression [6]. Taking this into consideration, we wanted to build a dialog system that is able to portray this method of CBT.

As there is an extensive list of mood disorders, we minimized our research focus to happy, sad, okay, stress and anxious. We found that these would best capture the symptoms of anxiety and depression. For each of the mood, we decided to research on ways to improve their mood if negative or to enhance their mood if positive based on the practices of CBT.

2.2 Moods

2.2.1 Happy

According to some experts, "happiness is a choice." The happiest people tend to show gratitude for small things in their daily lives. They recognize that the small, good things and events in their lives could have gone differently. As stated by Jacqueline Whitmore, one of the ways to make every day a happy day is to reflect on the blessings, to find joy in small things, and celebrate small successes [7].

2.2.2 Sad

The loss of a loved one, the feeling of hopelessness, and loneliness are among the most common reasons for feeling sad. Grieving is a natural response to the loss of someone and should not be suppressed. Focusing on the good things about one's relationship with their loved one and the time they had together may be a good coping strategy [9]. Feelings of hopelessness are often created due to negative thoughts that are unrealistic or illogical. Polarized thinking is a cognitive distortion where individuals tend to see things in either black or white - good or bad. These distortions can be overcome by taking different perspectives on the past and the present [10]. Finally, the feeling of loneliness can occasionally be a result of particular situations and environmental triggers. It is important to become aware that these feelings are common among multiple individuals and seeking contact from close friends and family is beneficial [11].

2.2.3 Okay

We have chosen to define the key term 'okay' as the state of not being happy or being sad, but being somewhere in between. It describes a state of lethargy and emptiness. This state of mind can be caused due to burnout, being constantly overwhelmed either by work or other responsibilities, or by believing that one is a failure in life. To combat these feelings, experts have recommended taking a break from one's usual schedule and making time for themselves, and to break away from judging themselves too harshly [12].

2.2.4 Stress

Stress can be caused by a number of reasons. The most common reasons are due to feeling overburdened from school, work, financial responsibilities, and family stress. If stress is caused by overburden, it is best to take a break in order to list and prioritize the things that need to be done. For financial and family stress, it is best to reach out to friends and family that could better assist the individual with their obstacles.

2.2.5 Anxious

The main difference between stress and anxiety is that stress is usually caused by external factors, while anxiety is caused by internal thoughts [13]. Feeling of anxiousness can be divided into the state of worrying and the state of panic or panic attacks. Anxiousness can be caused by negative thoughts or by being overwhelmed by daily events and blowing these events out of proportion. Believing that if you fail a quiz then the teacher will completely lose respect for you, that you will not graduate from college, that you will therefore never get a well-paying job, and will ultimately end up unhappy and dissatisfied with life is an example of blowing events out of proportion[14]. The key here is to change these thoughts and focus on doing one thing at a time. Panic attacks are sudden, intense surges of fear, panic, and anxiety. First and foremost step in helping an individual experiencing panic attack is to reduce the symptoms of panic and help bring them to a calmer state. This can be done by deep breathing: taking deep breaths in and out through the mouth, feeling the air slowly fill the chest and belly and then slowly leave them again [15]. Once they are calmer, trigger patterns and incorrect thinking can be recognized using the same techniques as used in the state of worrying.

Taking the above information from our research, we came up with sample dialogues that we could use in our project. Using these dialogues we were able to create dialog trees for each of our moods, which can be found in the appendix section.

3 Dialogflow

Another key tool that we researched is Dialogflow. Dialogflow is a development suite by Google, used to create conversational interfaces. It is commonly used in facilitating customer oriented services like with Dominos and Ticketmaster for their online ordering bots. Dialogflow uses Google's machine learning and NLP expertise. The main component to understand in Dialogflow is the concept of intents. Simply stating, intents categorize the user's intentions, as in what the user desires to accomplish. Each intent has training phrases, contexts, and responses. Training phrases are example phrases for what the user might say. When a user expression resembles these phrases, Dialogflow matches the intent and the corresponding response is returned back to the user. Contexts control the conversation flow, from one matched intent to another. [16]

To get started, we created a project in Dialogflow, where our first level of intents are the different moods, whose contexts then influences the flow of the conversation. We added dialogues from our sample trees of the moods as training phrases and responses.

4 Application Development

4.1 Introduction to Boon Bot

Boon Bot is the agent in the web application that we created as a first step to get closer to help tackle this issue of accessibility of therapy. The web application, as a whole, serves to act as a platform to help individuals manage their thoughts in one place in

a meaningful way. In other words, it is a platform that attempts to mimic a smaller version of therapy. The main purpose of our agent itself is to be able to continue short conversations with the user in a way to get them to talk more about their feelings. Boon Bot is able to detect the user's mood and respond accordingly, providing relevant tips based on this mood. This is achieved through leveraging Dialogflow's Natural Language Processing (NLP) capabilities, along with our own additional models. When the users express their moods or feelings, Dialogflow tries to match with one of the predetermined mood intents. If it is incapable of matching the expressed mood, we guide the system towards our own machine learning models in order to increase the chances of matching it to an intent.

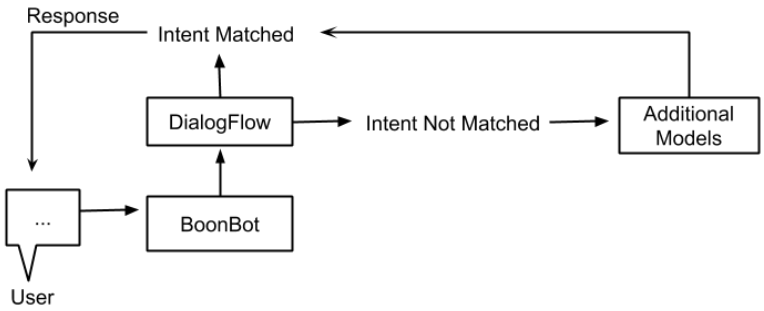

Figure 1: General Flow of Application's Dialog System

After a mood has been detected, it is tracked through the application's mood tracker, which is a calendar feature that displays the various moods of the user that have been detected throughout different check-in dates. Through this tool, the users have the chance to recognize their triggers and patterns in their daily lives. These tracked moods combine to provide a bi-weekly and monthly overview feature, using doughnut and line graphs as a visual aid to better discern the user's progress. Additionally, this application looks out for certain mood disorders and gently alerts the user if they are exhibiting common signs of these mood disorders and prompts them to consider seeking professional help.

4.2 List of Technologies

In order to implement Boon Bot, we got the opportunity to experiment with numerous technologies. In addition to psychology research, we did research on the best ways to implement our front-end and back-end components.

4.2.1 Front-end Technologies

Figma
Figma is a collaborative interface design tool. It allows users to design and prototype digital experiences. It is built for web browsers and therefore can be used across any OS platform without any downloads or updates. [17]

GIMP

GIMP is a free and open-source graphics editor used for image retouching and editing, free-form drawing, converting between different image formats, and more specialized tasks. [18]

ReactJS
React is a JavaScript library for building user interfaces and is optimal for fetching rapidly changing data. [19]

Figure 2: List of Technologies

4.2.2 Back-end Technologies

Firebase
Firebase is Google's development platform that provides functionality such as analytics, authentication, database management and more. [20]

Node
Node.js is a JavaScript runtime environment which enables users to compile React code.

Zerorpc
Zerorpc is a communication layer for distributed systems, which allows servers written in Python to easily communicate with those written in node.js. This is how we were able to efficiently separate machine learning tasks in python while still talking to our client. [21]

Python
For our machine learning tasks, we used Python. Within Python, we used libraries such as pandas, NumPy, NLTK, Keras and scikit-learn.

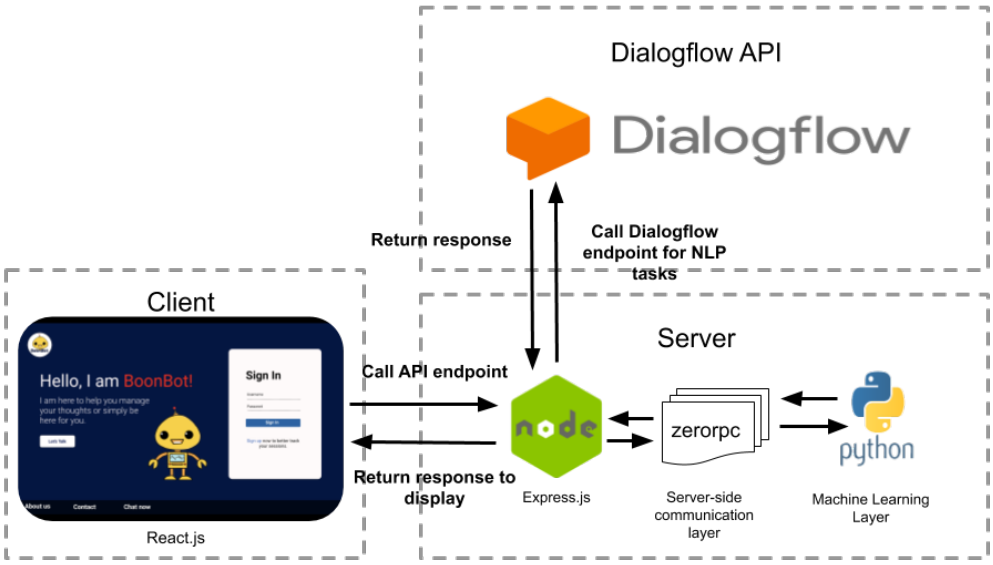

Figure 3: Boon Bot's Architectural Diagram

5 Paraphrase Detection

When Dialogflow fails to match a user query with one of the predefined mood intents, a Default Fallback intent is detected and the response "I didn't get that. Can you rephrase it?" is returned. This typically occurs when an intent does not have enough training phrases or Dialogflow's machine learning expertise is unable to predict an intent. With the help of paraphrase detection, an NLP classification problem, we were able to take significant steps in easing the issue. The problem of paraphrase detection involves determining whether a pair of sentences convey the same meaning or not. We decided to look at solving the problem of paraphrase detection separately from our application. Once a proper solution model was found, we integrated it back into our application.

6 Machine Learning

Machine Learning is a powerful tool due to its recent progressive developments in computing complex problems through models, with the goal of analyzing larger amounts of data and accurately predicting outcomes. It is able to extend these benefits within the context of solving NLP problems. From Google AI's Transformer-based models that consider a word's double-sided context to IBM's training data generator, today we have cutting edge resources to solving these problems, all thanks to the help of Machine Learning. Therefore, we leverage this ability in order to solve the paraphrase detection problem. We looked at two data sets, the Microsoft Paraphrase Research Corpus and the Quora Question Pairs corpus. We implemented a baseline model, initially to detect if two pairs of sentences were paraphrases. We then implemented a neural network with feature engineering and compared the results of this model with that of the baseline model to see if there were improvements.

7 Methodology

We have developed a system to identify whether a pair of sentences is identical. This was done through four main steps. First, preprocessing techniques such as tokenization, stopword removal, and lemmatization, were applied to the sentences. Secondly, a baseline approach was implemented. Afterwards, feature engineering was done in order to capture syntactic and semantic characteristics of these sentences. Finally, these features were passed on as input vectors to our multi-layer perceptron in order to determine whether a pair of sentences is semantically equivalent. The following section will provide more details on each of the steps.

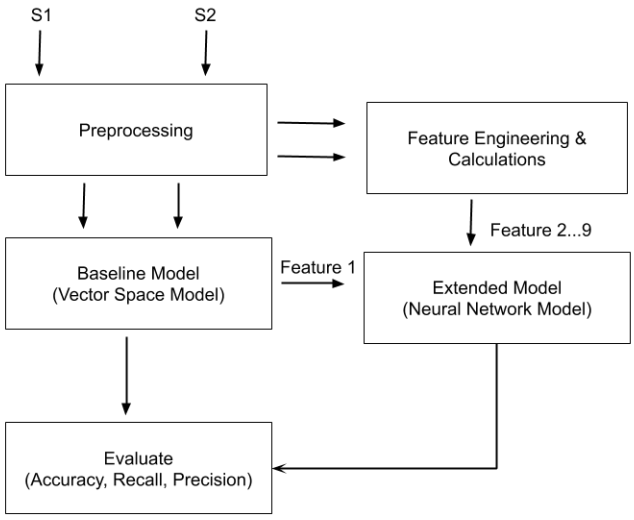

Figure 4: Illustration of the Paraphrase Identification Methodology

7.0.1 Preprocessing

Some standard preprocessing techniques that were applied to the sentences:

- Word Segmentation
  - *Tokenization*: each sentence is broken up into individual words or tokens.
- Data Cleaning
  - *Stopword removal*: common words that provide little to no value to the overall sentence are removed.
  - *Punctuation removal*: get rid of any punctuation (question marks, periods, etc.) that are attached to the tokens.
- Morphological Segmentation
  - *Lemmatization*: the root form of each word is captured such that it is still a valid word in the language.

Some extended preprocessing techniques that were applied to the sentences:

- Capturing Negation
  - *Tokenizing with Antonyms*: tokens that have negation tokens preceding them are taken into account and their antonym is added to the dictionary in the process of tokenization.
- Capturing relations
  - *WordNet Tokenization*: include hypernyms and synonyms of words in the dictionary in the process of tokenization

7.1 Baseline Model

For our baseline model, we implemented the Vector Space Model (VSM), an algebraic model that is commonly used in information retrieval. In this model, documents are rendered as n-dimensional vectors, where each dimension represents the terms. The similarity between the query vector and documents are calculated, whether it is through Euclidean Distance or other similarity metrics, ranking the relevant results based on the higher of the similarity scores. In our implementation, we decided to use Cosine Similarity as our similarity measure, which better suits our goal of capturing relevance. Unlike Euclidean, this measure is independent of vector size and is able to look at angular orientation between vectors in order to measure closeness [22]. We then explored our problem of identifying paraphrases with VSM, where each sentence in our datasets corresponds to a document. For example, the sentences from the Quora Question Pairs corpus are referred as documents:

D1: "What is the step by step guide to invest in share market in India?"
D2: "What is the step by step guide to invest in share market?"

7.1.1 Computing Pairwise Similarity

The main goal of our baseline approach is to calculate pairwise similarity of sentences and, depending on our chosen threshold, classify these sentences as a paraphrase or not a paraphrase. We first build a term-document-matrix in order to represent the counted frequency of word occurrences in each document, also known as term frequency (tf). We take into account the sentence lengths and normalize our term frequencies to build our final term-document-matrix.

	Document #	step	guide	invest	share	market	india
0	1	0.29	0.14	0.14	0.14	0.14	0.14
1	2	0.33	0.16	0.16	0.16	0.16	0.00

Figure 5: Normalized Term Document Matrix for the two sentences

After this, we use inverse document frequency (idf) in order to put a higher emphasis on rare terms and lower emphasis on frequently occurring terms throughout the sentences. This is done with the thought that a term like 'india' is more distinguishing than a word like 'step' in the given example documents. The calculation of idf(w) can be represented as:

$$log \frac{N}{df(w)},$$

where w = word, N = total number of documents, and df = document frequency

After we calculate our idf, we have everything we need to calculate our tf-idf vectors by multiplying the tf values with the idf values. By taking into account the number of times a word appears in a document and rarity of this word across documents, this tf-idf score reflects the importance of this word to a document in relation to a set of documents. Using these, we are now able to compute the similarity of these vectors using the cosine similarity measure:

$$\cos\theta = \frac{D2 \cdot D1}{||D2||||D1||},$$

where D2 and D1 represent the tf-idf vectors we previously calculated for D2 and D1, respectively.

	Sentence_1	Sentence_2	is_Paraphrase	Cosine_Similarity
0	What is the step by step guide to invest in sh...	What is the step by step guide to invest in sh...	0	0.895532
1	What is the story of Kohinoor (Koh-i-Noor) Dia...	What would happen if the Indian government sto...	0	0.410995

Figure 6: An example of a cosine calculation done by the baseline model

A more in depth explanation of this can be found at [23].

7.2 Feature Engineering

We hypothesized that syntactic and semantic similarity were both important while predicting if a pair of sentences were paraphrases. We built our features based on this hypothesis, creating both syntactic and semantic features that measures syntactic and semantic similarity for a pair of sentences.

Syntactic
Edit Distance
This measures similarity between a source and target string. It is the minimum number of operations (insertion, deletion, or substitution) required to transform the source string to target string. The edit distance between 'monkey' and 'money' is 1. Deletion of 'k' in 'monkey' will give us 'money.'

Change in Implementation

We have implemented edit distance but on the word level instead of character level. This was done by transforming the sentences to list and comparing each element of the source and the target.

Jaccard Coefficient
It is calculated by dividing the intersection (common words) of the two sentences over the union of the two sentences(length of sentence one + length of sentence two - intersection)

Sequence Matcher
SequenceMatcher is a class in the python module difflib. It finds the length of the longest contiguous matching subsequence. The ratio then divides it by the total length of characters of both sentences and multiplies it by 2. This

returns the similarity score (float in [0,1]). for example, [THANK]S[FOR][RESPONSE] and [THANK]ING[FOR] KIND[RESPONSE] has 18 characters in the longest subsequence, including spaces. Therefore, the ratio will output 0.8 (2*18/45).

N-gram Overlap
N-gram is a sequence of N words. We create N-grams of both sentences. We then look at the common grams and divide it by the union of grams (in other words, perform a jaccard coefficient with the n-grams).

Change in Implementation

N = 3 or 4 are common and are optimal at capturing the probability of a word given the previous words. We chose N = 3 because we felt that this would be able to capture context even for shorter sentences.

Semantic
Word Movers Distance
Word Mover's Distance uses normalized bag of words and word embeddings to calculate the distance between sentences. It retrieves vectors from pre-trained word embeddings models for the words of the sentences. The key assumption with this similarity measure is that similar words should have similar vectors. For example: 'Obama speaks to the media in Illinois' and 'The president greets the press in Chicago' have the same meaning, however they do not have any words in common. Word Mover's Distance helps with this.

Extended Cosine Similarity
This is the classic Cosine Similarity, extended using Wordnet's relations. We use WordNet to extend our sentences and, therefore, extend our dictionary.

Change in Implementation

We extend our semantic reach by including synonyms, hypernyms, and antonyms in our dictionary. We then call our existing calculate cosine similarity method to get the similarity of the extended documents.

Word Sense Disambiguation
Word Sense Disambiguation finds the best sense of a word from all the given senses of the word. The Lesk algorithm uses WordNet and gets the gloss of all the senses of the word in the sentence and then calculates the maximum overlap with the senses, returning whichever gives the maximum overlap. Let's take the phrase 'pine cone'. 'Pine' has two senses. Sense 1: kind of evergreen tree with needle-shaped leaves and Sense 2: waste away through sorrow or illness. 'Cone' has three senses. Sense 1: solid body which narrows to a point. Sense 2: something of this shape whether solid or hollow. Sense 3: fruit of a certain evergreen tree. Comparing the senses of the two words, we can see that 'evergreen tree' is common in one sense of each word. Therefore, Sense 1 of Pine and Sense 3 of Cone are the most appropriate when 'pine' and 'cone' are used together.

Change in Implementation

After Lesk was applied to each sentence, where the most appropriate senses of each word was detected, we looked for the common senses in the two sentences. To normalize our result, we again used Jaccard.

Named Entity Recognition Similarity
In this feature, we first collected NER words for each sentences along with their label.

Change in Implementation

For example, 'Washington' will have a label of 'GPE' for geo-political entities. We computed Jaccard Coefficient by dividing the common NER (with label) over the union of NER of both sentences. In the case where no NER was detected, in neither of the sentences, we simply returned 0, else we returned the Jaccard Coefficient.

7.3 Neural Network

For our extended model, a multi-layer perceptron consisting of one input layer, one output layer and two hidden layers was implemented. The input layer includes nine nodes, representing each of the input features. The hidden layers have seven nodes each and the output layer consists of a single node. A sigmoid function was used as the nonlinear function for our output layer. This ensures that all our output values are adjusted between 0 and 1.

7.4 Integration with BoonBot

After our models were trained, we were able to save these models through a Python library called 'pickle.' We were able to load this model into our web app in order to make a prediction through calling the 'predict classes' method. In the case of our baseline model, the highest cosine similarity was chosen as the result for the intent. In the case that they are equal, the user gets re-prompted from Dialogflow to help us figure out the intent. If no similarities are detected, then the Default Fallback Intent is returned. Unfortunately, due to time constraints, we were not able to deal with properly handling the situation where the MLP outputs multiple 1's, indicating match to several intents.

8 Results

8.1 Test Data Collection Methodology

For Microsoft and Quora datasets, we split the data into 70% training and 30% testing sets. In order to test for Boon Bot, we had to collect the test data ourselves. Given time constraints, we were only able to collect a little bit more than 40 pairs of sentences. We shared a google document with a group of students and asked them to write down sentences they believed to be paraphrases and sentences they believed to not be paraphrases, collecting a balanced class of 20 paraphrases and 20 non-paraphrases.

```
Sentence_1,Sentence_2,is_Paraphrase
I am feeling happy because I had a good day today, I am feeling glad due to the fact I had an amazing day today,1
I have been happy for a while, I have been satisfied for some time,1
I never thought I would say this but I am pretty happy, I in no way thought I would say this but I am pretty content,1
I am so sad because my dog died, I am so unhappy due to the fact that my dog passed away,1
I have been feeling so down lately, I have been feeling so low currently,1
```

Figure 7: Snippet of CSV file from collected test data

8.2 Threshold Determination for Baseline Model

Recall that our baseline cosine model calculates float values, indicating how similar pairs of sentences in our datasets are. In order to further classify these sentences as a para-

phrase, we pick a threshold we believe is adequate to represent a semantic equivalence between sentences. To achieve this, we accept different thresholds between the range of 0 and 1 to classify our sentences, incrementing the threshold by 0.5 each time. We assess the performance of these classifications through our evaluation metrics: accuracy, recall, and precision. Based on the results of this analysis and additional exploration of what these metrics tell us relative to our problem, we pick the threshold for the baseline model. For example, a high accuracy is a good indication that a model is performing great. It tells us the ratio of the correctly classified examples out of the total classifications that was done by our model. However, this accuracy becomes problematic when we have a really big imbalance of classes. If the majority of our samples belong to one class then the accuracy is biased towards that class since there is an equal cost assigned to false negatives and positives. As we'll see in a later section, the fact that there is a presence of imbalance in our dataset means that the accuracy measure should not be the only metric to take into account for the decision.

8.3 Paraphrase Detection Results

Test Data	Accuracy	Recall	Precision
Microsoft	70%	79%	75%
Quora	69%	57%	67%

Baseline Model Test Results at Threshold of 0.45

Test Data	Accuracy	Recall	Precision
Microsoft	71%	78%	78%
Quora	70%	87%	58%

Neural Network Model Test Results

8.4 Boon Bot Test Results

Test Data	Accuracy	Recall	Precision
BoonBot	53%	100%	6%

Boon Bot Test Results on Baseline at Threshold 0.45

Test Data	Accuracy	Recall	Precision
BoonBot(Microsoft)	64%	100%	28%
BoonBot(Quora)	69%	100%	39%

Boon Bot Test Results on Neural Network Model

8.5 Analysis

Overall, there is an improvement to be noted between the baseline model and the neural network model, even if it is a small one. In the following section, we further analyse different characteristics that might have attributed to this overall result.

Effects of Textual Content Characteristics

Language Level
In comparison to the Microsoft corpus, the Quora dataset is filled with heterogeneous

data in the sense that it is composed of varying questions submitted by a mass of people, meaning that the presence of informal language is more common. The question pairs also ranges in topics, from domain-specific questions filled with acronyms to cultural specific questions. Additionally, the dataset contains anomalies such as non-ASCII characters, as well as a few ASCII emoticons that are still difficult to decipher.

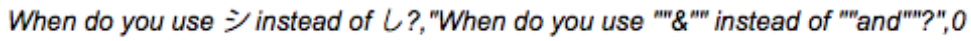

Figure 8: Sentence Pair containing indistinguishable characters

For example, through preprocessing and not being able to semantically represent those characters, these sentences would be incorrectly characterized as a paraphrase by our system due to the fact that now they have many words in common.

Length of Text
The length of texts can influence the input features. If sentences are long, there's more possibility for the input features to be able to calculate a similarity and output a value. If sentences are too short, input features may not be able to find similarities between two sentences. For example, "I am blessed" and "I am happy" should be classified as a happy mood intent. However, input features may not be able to provide any values to the neural network. This is because after tokenization and stop word removal is applied, we will only be left with the word 'blessed' and the word 'happy'.

Level of Typographical Errors
Typographic errors are mistakes made in the typing of printed material. The Quora dataset contains more typographic errors, as people tend to speak more casually in online forums that lack any formal editing process. This results in data that is syntactically incorrect due to spelling errors. This is in contrast with Microsoft dataset where it was observed that the sentences were well-formed and syntactically correct with little-to-no spelling errors.

Effect of Feature Selection

Number of features
We selected 9 features to serve as input vectors for our multi-layer perceptron, majority of them only encompassing classical text mining features. It would help to build a model to extract even more features rather than manual engineering and implementation, like we are doing now.

Quality of features
Through the inclusion of hypernyms, synonyms, and antonyms, our features succeeded in capturing many important semantic relations. Nevertheless, additional relations such as polysemy, homonymy, and more can further be included during the feature engineering process. We can move further than just classical approaches and seek to find features that help with the range of topics and also help with our domain. For example, possibly including features that look for words like "how" and "can" might have helped in the Quora set to distinguish for sentences that might be more mathematical, in the sense that it is looking for a step-by-step solution.

Balance of features

Experiment more with the balance of features. This means analyzing the effect of having more semantic features than syntactic features or vice versa.

Feature Ablation
We decided to remove each feature and see how the Neural Network performs without it. For Quora, we found that the worst feature was Word Mover's Distance. For Microsoft, the worst one was the word n-gram. However, we did not find a particular feature that outperformed another or did noticeably worse than the other, since all the metrics were similar to each other at each removal. In the future, we plan on looking for more efficient ways to do feature evaluation. One possible way would be to perform permutation feature importance, where the value of the features are shuffled rather than dropped entirely. This process would also tell us what our best features were in a more logical manner, which would be more helpful to us.

Effect of Pre-processing techniques

Part of Speech Tagging
Part of Speech tagging provides numerous benefits when it comes to text classification. It helps create a baseline structure that points us towards a direction to start understanding the meaning of a sentence through the grammatical classification of each word. While this still deals with ambiguity and may not have completely solved all of our problems, it would have initially simplified a lot of our issues and improved our overall results.

Expanding Abbreviation (disambiguation)
Another approach to a disambiguation problem and to help us better our results would be through the inclusion of abbreviation expansion.

Effects of Training Set Content

Upsampling
Class imbalance in the Quora dataset was indicated to be a big problem in the discussion, revealing that Quora has way more samples of the positive class than negative. This suggests that the minority class is neglected and, since our models are more likely to classify samples as the positive class, the results of our metrics are skewed. In order to handle this imbalance and move towards accurate results, resampling techniques could be used. For example, randomly sampling the minority class with replacement would serve as a simple fix, as it is generally accurate, does not require additional knowledge, and minimizes bias through randomization [36].

Effects of Boon Bot Implementation

Model choice
In our case, we are picking the best performing model. However, it is possible that experimenting more with combining data from the two sets and developing different methods towards domain-specific data collection in order to develop a model could possibly improve our overall results.

Other Effects

Human Annotation Errors
In the Microsoft and Quora datasets, there are many instances where the annotations

are debatable and a consensus is not achievable. In other cases, many of the samples are labelled incorrectly.

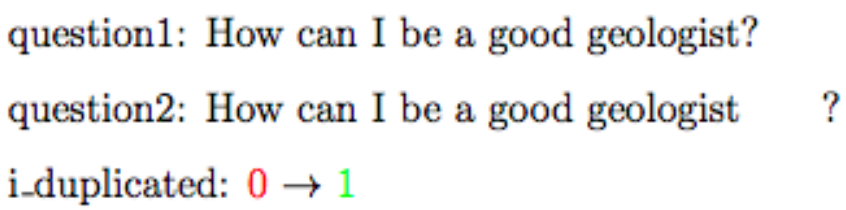

Figure 9: Clear Human Annotation Error

Test Data Collection Biases
To collect our test data for Boon Bot, we asked individuals to write down two sentences they thought were paraphrases for each of our six moods. These data instances were collected from acquaintances who have taken a machine learning course previously and have an understanding of how paraphrase detection and neural networks work. This may have influenced their answer in some overlooked way, adding an observer bias to our test data. This data set also has limited generalizability. This is due to the fact that our sample data set was only taken from students at a university. Individuals from different age groups could have different ways of constructing sentences and paraphrases.

9 Conclusion

In conclusion, through this project we had the opportunity to achieve our goal of building a full stack web application that is able to reproduce a smaller notion of therapy. Additionally, we had the chance to touch on the essentials of machine learning and further understand its impact on paraphrase detection. Through feature engineering, the different characteristics of text retrieval were investigated thoroughly in order to understand what an optimal feature may consist of. We were able to implement and analyze our proposed solution to this NLP problem, through the implementation of a multi-layer perceptron. Through this project, it is clear that machine learning can extend dialog systems from just routine customer service tasks to medical and health care fields. In a time and place, where mental disorders are highly prevalent, technologies like chatbots can provide short and, possibly, long term relief and treatment for individuals.

10 References

1. Public Health Agency of Canada, "Government of Canada," Canada.ca, 15-Sep-2017. [Online]. Available: https://www.canada.ca/en/public-health/services/about-mental-illness.html. [Accessed: 01-Dec-2019].

2. "Mental Illness and Addiction: Facts and Statistics," CAMH. [Online]. Available: https://www.camh.ca/en/driving-change/the-crisis-is-real/mental-health-statistics. [Accessed: 02-Dec-2019].

3. "Mental disorders affect one in four people," World Health Organization, 29-Jul-2013. [Online]. Available: https://www.who.int/whr/2001/media_centre/press_release/en/. [Accessed: 02-Dec-2019].

4. H. A. Flynn, R. Warren, S. P. Chand, and P. J. Maerov, "Using CBT effectively for treating depression and anxiety," MDedge Psychiatry, 28-Mar-2019. [Online]. Available: https://www.mdedge.com/psychiatry/article/82695/anxiety-disorders/ using-cbt-effectively-treating-depression-and-anxiety. [Accessed: 07-Dec-2019].

5. B. Wiest, "If You Want To Master Your Life, Learn To Organize Your Feelings," Forbes, 16-May-2018. [Online]. Available: https://www.forbes.com/sites/briannawiest/2018/05/14/ if-you-want-to-master-your-life-learn-to-organize-your-feelings /#12b85678cb0f. [Accessed: 07-Dec-2019].

6. "NAMI," Home. [Online]. Available: https://www.nami.org/Blogs/NAMI-Blog/November-2016/Discovering-New-Options-Self-Help-Cognitive-Behav. [Accessed: 07-Dec-2019].

7. J. Whitmore, "5 Ways to Make Every Day a Happy Day," HuffPost, 29-Jun-2015. [Online]. Available: https://www.huffpost.com/entry/five-ways-to-make-every-day-a-happy-day_b_7162858?guccounter=1. [Accessed: 07-Dec-2019].

8. M. Bruneau, "12 Things Being a Therapist Taught Me About Happiness," Thrillist, 14-Jan-2016. [Online]. Available: https://www.thrillist.com/health/nation/a-therapists-lessons-about-happiness. [Accessed: 07-Dec-2019].

9. "Grief: Coping with reminders after a loss," Mayo Clinic, 17-Nov-2018. [Online]. Available: https://www.mayoclinic.org/healthy-lifestyle/end-of-life/in-depth/grief/art-20045340. [Accessed: 05-Dec-2019].

10. M. D. Jackson, "Polarized Thinking: A Cognitive Distortion," Exploring your mind, 12-Feb-2019. [Online]. Available: https://exploringyourmind.com/polarized-thinking-cognitive-distortion/. [Accessed: 08-Dec-2019].

11. "Treating Loneliness: It's More Than Just Meeting Others," Psychology Today. [Online]. Available: https://www.psychologytoday.com/ca/blog/web-loneliness/201404/treating-loneliness-its-more-just-meeting-others. [Accessed: 17-Dec-2019].

12. J. Davis, "Mid-point mental health: why are we all feeling so flat?," Harper's BAZAAR, 08-Aug-2019. [Online]. Available:

https://www.harpersbazaar.com/uk/beauty/mind-body/a27416416/mid-point-mental-health-why-are-we-all-feeling-so-flat/. [Accessed: 07-Dec-2019].

13. Canadian Mental Health Association, "What's the difference between anxiety and stress?," What's the difference between anxiety and stress? — Here to Help. [Online]. Available: https://www.heretohelp.bc.ca/q-and-a/whats-the-difference-between-anxiety-and-stress. [Accessed: 09-Dec-2019].

14. W. Meek, "How Can Cognitive Distortions in GAD Change?," Verywell Mind, 16-Aug-2019. [Online]. Available: https://www.verywellmind.com/cognitive-distortions-and-anxiety-1393157. [Accessed: 17-Dec-2019].

15. Healthline, "11 Ways to Stop a Panic Attack". [Online]. Available: https://www.healthline.com/health/how-to-stop-a-panic-attack#breathe-deep [Accessed 10 Dec. 2019].

16. "Intents — Dialogflow Documentation — Google Cloud," Google. [Online]. Available: https://cloud.google.com/dialogflow/docs/intents-overview. [Accessed: 17-Dec-2019].

17. Figma, Inc, "Getting Started with Figma," Figma. [Online]. Available: https://help.figma.com/article/116-getting-started. [Accessed: 17-Dec-2019].

18. "GIMP," Wikipedia, 13-Dec-2019. [Online]. Available: https://en.wikipedia.org/wiki/GIMP. [Accessed: 17-Dec-2019].

19. "React (web framework)," Wikipedia, 15-Dec-2019. [Online]. Available: https://en.wikipedia.org/wiki/React_(web_framework). [Accessed: 17-Dec-2019].

20. "Firebase," Google. [Online]. Available: https://firebase.google.com/. [Accessed: 17-Dec-2019].

21. "zerorpc," npm. [Online]. Available: https://www.npmjs.com/package/zerorpc. [Accessed: 17-Dec-2019].

22. "Information retrieval document search using vector space model in R," Data Science Central. [Online]. Available: https://www.datasciencecentral.com/profiles/blogs/information-retrieval-document-search-using-vector-space-model-in. [Accessed: 10-Dec-2019].

23. "Cosine Similarity - Understanding the math and how it works? (with python)," Machine Learning Plus, 30-Oct-2018. [Online]. Available: https://www.machinelearningplus.com/nlp/cosine-similarity/. [Accessed: 10-Dec-2019].

24. "Edit Distance and Jaccard Distance Calculation with NLTK," GoTrained Python Tutorials, 10-Jun-2019. [Online]. Available: https://python.gotrained.com/nltk-edit-distance-jaccard-distance/. [Accessed: 09-Dec-2019].

25. "Source code for nltk.model.ngram," nltk.model.ngram - NLTK 3.0 documentation. [Online]. Available: http://www.nltk.org/_modules/nltk/model/ngram.html. [Accessed: 12-Dec-2019].

26. N. Jaiswal, "SequenceMatcher in Python," Medium, 09-Dec-2019. [Online]. Available: https://towardsdatascience.com/sequencematcher-in-python-6b1e6f3915fc. [Accessed: 17-Dec-2019].

27. E. Ma, "Word Distance between Word Embeddings," Medium, 06-Sep-2018. [Online]. Available: https://towardsdatascience.com/word-distance-between-word-embeddings-cc3e9cf1d632. [Accessed: 17-Dec-2019].

28. "gensim: topic modelling for humans," Machine learning consulting. [Online]. Available: https://radimrehurek.com/gensim/models/keyedvectors.html. [Accessed: 17-Dec-2019].

29. "Semantic Similarity using WordNet," Kaggle, 20-Apr-2017. [Online]. Available: https://www.kaggle.com/antriksh5235/semantic-similarity-using-wordnet. [Accessed: 17-Dec-2019].

30. S. Torres, A. Gelbukh, A. Mateos, M. Othon, and Unidad Profesional, "[PDF] Comparing Similarity Measures for Original WSD Lesk Algorithm: Semantic Scholar," [PDF] Comparing Similarity Measures for Original WSD Lesk Algorithm — Semantic Scholar, 01-Jan-1970. [Online]. Available: https://www.semanticscholar.org/paper/Comparing-Similarity-Measures-for-Original-WSD-Lesk-Torres-Gelbukh/651ee5def5cabff3cdf03b6c1a44c00aad9ef527. [Accessed: 17-Dec-2019].

31. R. Joshi, F. Ahmad, Joseph, A. Luna, Wayne, Arjun, Anonym, Marcelo, Alan, Neslihan, Daniela, Yujia, Radha, K. Ochieng, Gayatri, Adawat, Edu, R. Lawaniya, Adnen, P. Kaur, A. Das, and M. Berg, "Accuracy, Precision, Recall & F1 Score: Interpretation of Performance Measures," Exsilio Blog, 11-Nov-2016. [Online]. Available: https://blog.exsilio.com/all/accuracy-precision-recall-f1-score-interpretation-of-performance-measures/. [Accessed: 17-Dec-2019].

32. V. Luhaniwal, "Why better weight initialization is important in neural networks?," Medium, 07-May-2019. [Online]. Available: https://towardsdatascience.com/why-better-weight-initialization-is-important-in-neural-networks-ff9acf01026d. [Accessed: 17-Dec-2019].

33. J. Brownlee, "Why Initialize a Neural Network with Random Weights?," Machine Learning Mastery, 19-Aug-2019. [Online]. Available: https://machinelearningmastery.com/why-initialize-a-neural-network-with-random-weights/. [Accessed: 17-Dec-2019].

34. S. Yadav, "Weight Initialization Techniques in Neural Networks," Medium, 23-Nov-2019. [Online]. Available: https://towardsdatascience.com/weight-initialization-techniques-in-neural-networks-26c649eb3b78. [Accessed: 17-Dec-2019].

35. C.-F. Wang, "The Vanishing Gradient Problem," Medium, 08-Jan-2019. [Online]. Available: https://towardsdatascience.com/the-vanishing-gradient-problem-69bf08b15484. [Accessed: 17-Dec-2019].

36. G. DePersio, "What are the advantages of using a simple random sample to study a larger population?," Investopedia, 18-Nov-2019. [Online]. Available: https://www.investopedia.com/ask/answers/042915/what-are-advantages-using-simple-random-sample-study-larger-population.asp. [Accessed: 17-Dec-2019].

11 Appendix

Figure 10: Happy

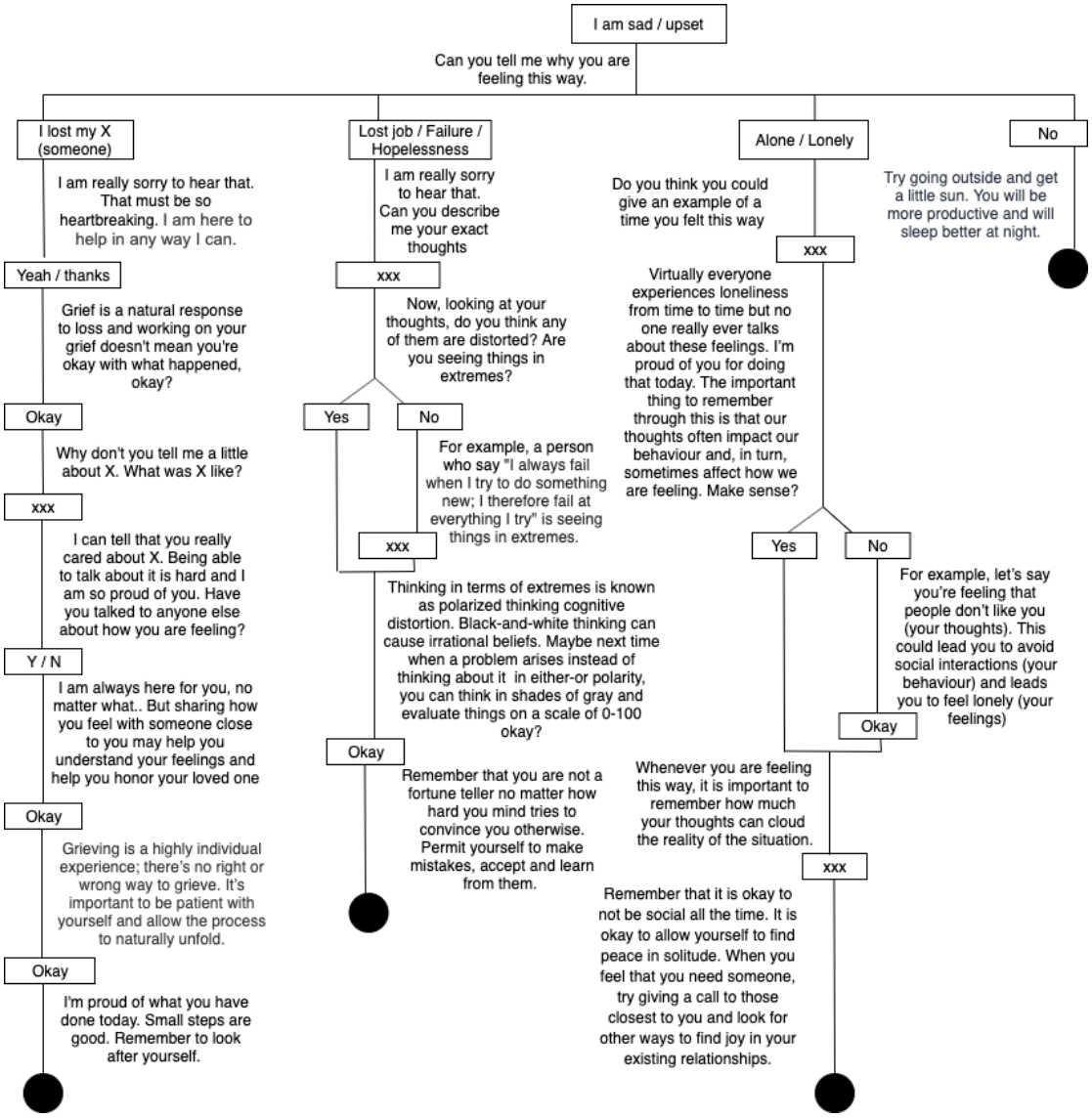

Figure 11: Sad Dialog Tree

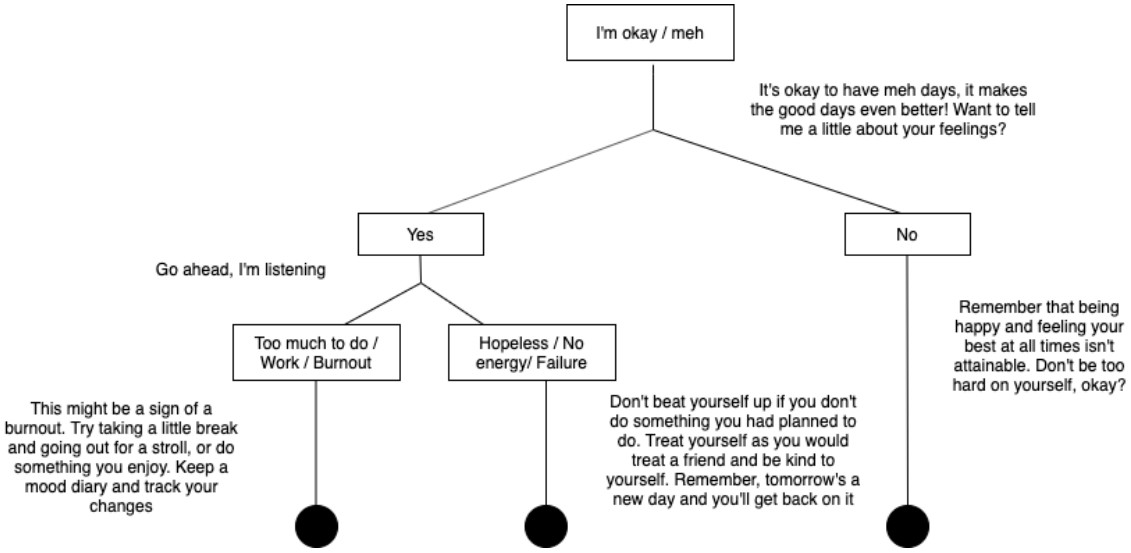

Figure 12: Okay Dialog Tree

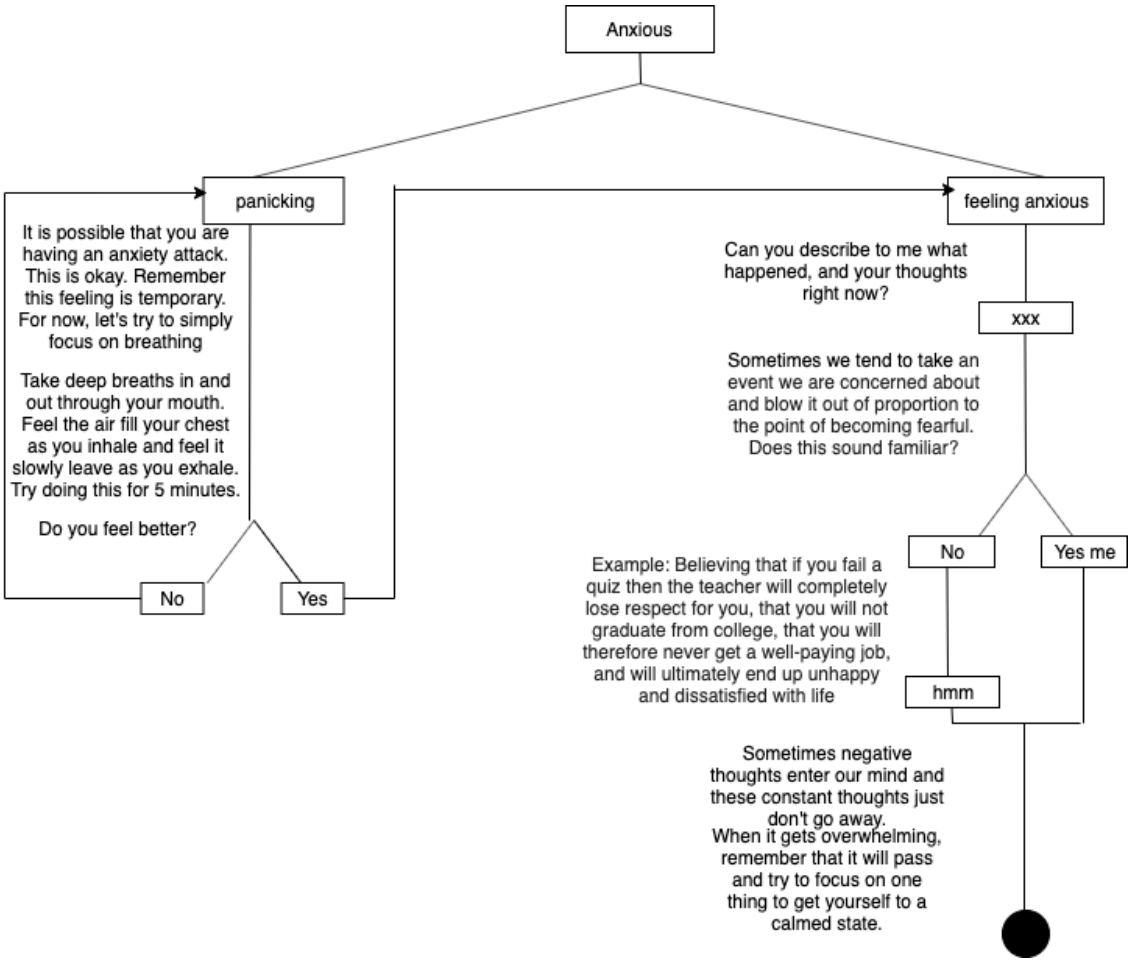

Figure 13: Anxious Dialog Tree

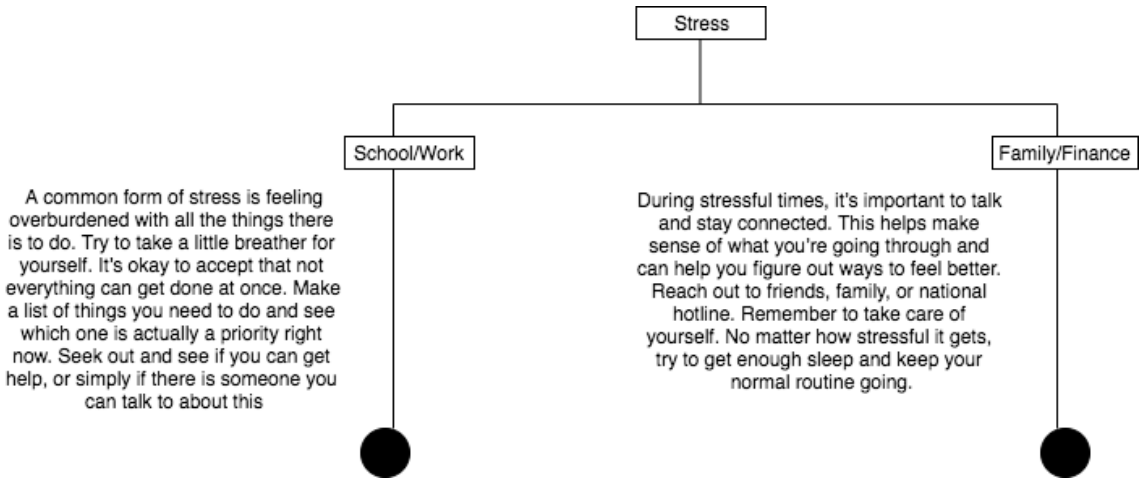

Figure 14: Stress Dialog Tree