I Couldn't Find a Therapist so I Built One Using BERT, Part I – the report

ABSTRACT

In this paper, I explore the mechanics of training a computer empathy. "Empathy" is operationally defined in this project as the ability to classify a subject's written anecdote and accurately predict the underlying condition from which the subject's experiences are derived. I apply machine learning's NLP (natural language processing) technique called a "multi-classification task" to achieve this, and experiment with several ways to improve the predictive accuracy of the models I build. My model for part 1 of this paper yielded an evaluation accuracy of 72.58% (3 epochs at learning rate of $2e^{-5}$) the test dataset without any fine tuning, lending support for the hypothesis that it is possible to teach computers empathy as operationally defined. In part 2 of the paper, I put this result to a more rigorous test and fine tune the parameters to improve the accuracy.

INTRODUCTION

Opening

There is no reason why a person should not be able to feel understood by another individual willing to engage in real talk, but that is precisely the problem, isn't it? The invention of internet and social media led to a proliferation of constant chatter yet it feels as though there are fewer listeners than ever. How can this be?

The answer I found is relatively simple; chatter is not a form of communication I would classify as real talk. Real talks allow us to become both the storyteller, and the listener. Stories are not chatter. Stories carry power to move people—chatter does not. Mediums do exist, however, where real talk happens almost exclusively. Among them are blog pages, forums, and therapists. I decided to leverage these sources to teach a computer empathy—a critical skill for a great listener.

As we approach a world with an increasing shortage of listeners who can make us feel understood, I believe that an artificial intelligence can offer emotional companionship, and help distressed storytellers connect with therapists who are skilled in navigating the client's story genre, as classified by the AI.

Why I Built My Own "Therapist"

24 isn't typically an age one might consider "old" (hopefully), but 14 years of searching for answers can feels like an eternity.

When someone suffers from mental illnesses [someone like me], they tend to ask a lot of

questions. The vast majority of those questions turn out to be clever logical traps aimed to validate our deepest fears. Therefore, while they're obviously not the most useful thoughts, they are so compelling that I, along with my brethren, are drawn to them like moths to an electrifying death rave.

zap! zap zap! zap…!

I admit I am no wiser than a moth, which is exactly why we need *real* mental health professionals.

One umbrella category within this vast profession are therapists. They guide us towards asking better questions—questions that lead to answers about our pasts, and our traumas. They help us realize things in ways that we alone would never have thought. They teach us of our self-worthiness of love and guides us towards an exit to our great mazes. No machine can do that; at least not yet, and for a long time.

However, the issue remains that mental healthcare is widely inaccessible. There are cultural barriers, stigma, language barriers, and costs that make mental healthcare a luxury for most. My personal experiences can attest to all of those factors. I couldn't afford therapy when I was younger due to my "complaints" being hushed and swept under the rug of Korean mental health stigma. I couldn't afford one in adulthood because I neither had the money nor insurance, and ironically, I couldn't afford one after being employed because I had no time. It was 2019 when I was finally prescribed medication— and I consider myself *very* lucky.

Figure 1
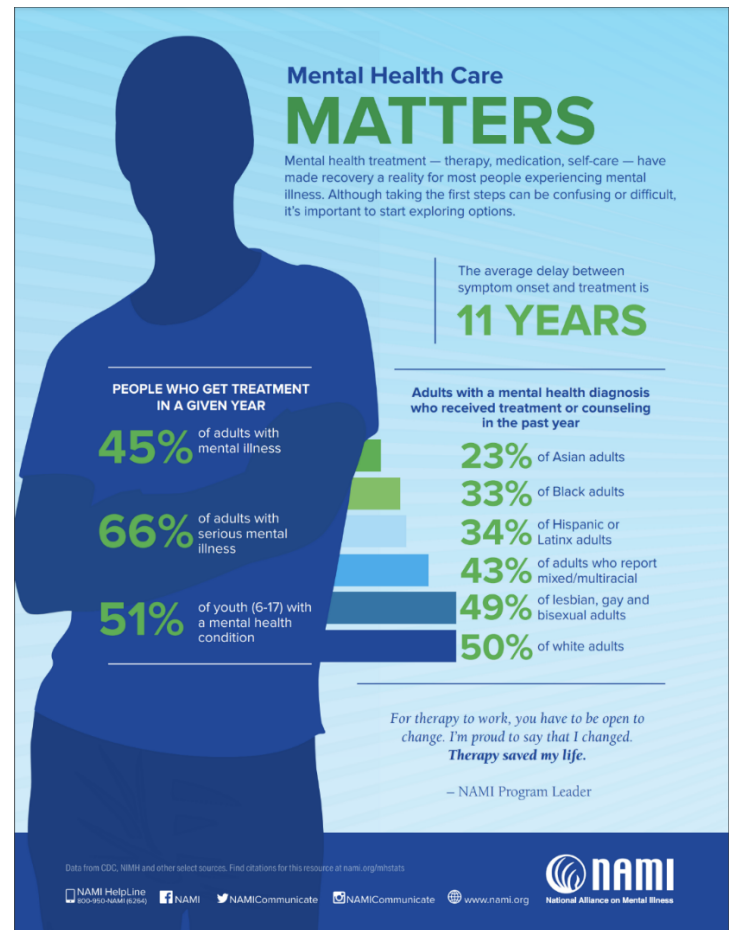
This project began as an attempt at trying to understand myself better. I didn't build something that can take the place of a therapist, and I wasn't trying to. I built a **classifier** that can potentially recognize the illnesses I have based on patterns in my writing. My thinking was, that by understanding which "disorders" are detected [and to what probability] in my writing patterns, I would have some useful insights. Maybe they could then help me identify my own cognitive distortions? Or so would be my hope. After all, we need to understand where our issues are to clue into the "why's" and the "how's"—and what we can do about it. Furthermore, I believed in

the utility this project potentially has to offer for people who relate to my circumstances.

METHODS

How I Built it.

I used Google's **BERT** to build my classifier. BERT stands for **B**idirectional **E**ncoder **R**epresentations from **T**ransformers. Bidirectionality indicates an approach to machine language learning models where the machine captures contextual ques by observing the input text sequence from left to right, and from right to left, simultaneously. Furthermore, BERT utilizes a technique called the Masked Language Model (MLM) through which it enables deep bidirectional learning. The idea behind it is BERT will [*MASK*] tokens at random (tokens being tokenized text from the input), and will attempt to predict words that belong in them.

Because the nature of this project is to receive a text input and predict a label for that input, the appropriate task here is a **multi-classification task.** Multi-classifiers differ from **binary-classification** tasks in that the questions we are asking can't be answered as "yes", or a "no". Rather, multi-classifiers are acceptable for tasks that has numerous potential answers. I used six labels to build my classifier.

I pulled the data I needed from Reddit, circumventing a common issue where highly task-specific data simply doesn't exist, or requires high level legal authorization to access.

Data Extraction

I picked six disorders relating to trauma and depression from reddit with robust community engagement. From those subreddits I chose top, rising, new, and hot as categories to filter the available data. I selected the "body" column from those categories because it contained the most textual information compared to other columns such as: title, number of comments, and number of upvotes. The following list is the subreddits I chose to compare.

- **r/depression**
- **r/ptsd**—*post-traumatic stress disorder*
- **r/cptsd**—*complex post-traumatic stress disorder*
- **r/bpd**—*borderline personality disorder*
- **r/bipolar**
- **r/dissociation**.

In particular, it would be interesting to observe whether BERT would be capable of distinguishing between **PTSD** (*post-traumatic stress disorder*) and **CPTSD** (*complex post-traumatic stress disorder*). The **DSM-V** (*Diagnostic and Statistical Manual for Mental Disorders*) does not recognize CPTSD as its own diagnosis, while the **ICD-11** (*International Classification of Diseases*) does. The difference between the two manuals is that the former is published and maintained by the American Psychological Association, while the latter is produced by the World Health Organization.

I used **PRAW** to extract data from the subreddits. To extract data from other subreddits, I interchanged 'depression' from the "reddit.subreddit('depression')" line with other disorders. This left me with six datasets, with roughly 1000 rows each.

However, the cumulative of the six datasets would only have 6000 rows which is quite limited. I thus utilized two of the four categories specified earlier for the training step: new, and rising. Thankfully, these categories helped double each dataset resulting in a final dataset of roughly 11,000 rows.

While hot and top posts made a tempting case to train the model on, I went for a many-to-one approach in my data selection. This is because top and hot posts are determined by the number of upvotes and engagements on a post (number of comments, views, and etc.), and I conjectured this would mean while the post may resonate with the general sentiment of the community, an everyday post that does not gain as much attention risk being misrepresented by the masses. Thus, if I chose to pursue training the model based on hot and top posts, it would be the same as learning a stereotype and classifying people based on a generalization. This is a typical (and documented) issue within the APA, and it is a pitfall I'd like to avoid.

___

BERT expects three .tsv file inputs to train and test from:

- **train.tsv**
- **test.tsv**
- **dev.tsv**

The .tsv format stands for *tab separated value.* I used sklearn's **train_test_split** module to create the train, test, and dev datasets at size .01 (1% test data from the master dataset).

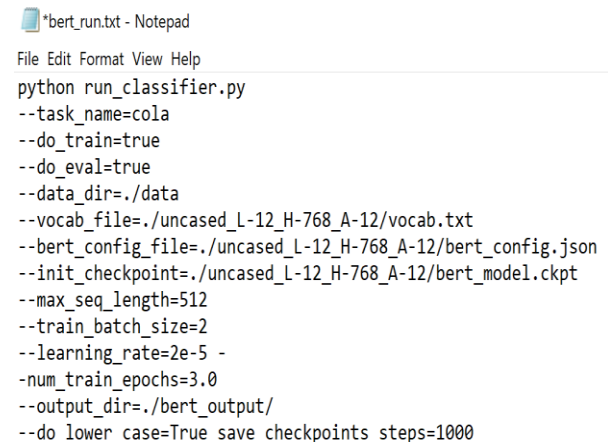The **train.tsv** is formatted by containing the following columns:

- **Column 0:** A unique id for each row.
- **Column 1:** A label for the data we want to classify, as an integer.
- **Column 2:** A 'throwaway' column with a random alphabet. I chose 'q'… just because.
- **Column 3:** A column with *untokenized* text we want to classify.

The **test.tsv** file simply requires a unique id column, and a corresponding text.

The three files are then loaded into the BERT directory's **data** file.

Setting the Parameters…

Figure 2

```
*bert_run.txt - Notepad
File Edit Format View Help
python run_classifier.py
--task_name=cola
--do_train=true
--do_eval=true
--data_dir=./data
--vocab_file=./uncased_L-12_H-768_A-12/vocab.txt
--bert_config_file=./uncased_L-12_H-768_A-12/bert_config.json
--init_checkpoint=./uncased_L-12_H-768_A-12/bert_model.ckpt
--max_seq_length=512
--train_batch_size=2
--learning_rate=2e-5 -
-num_train_epochs=3.0
--output_dir=./bert_output/
--do_lower_case=True save_checkpoints_steps=1000
```

The parameter below was used to train the first model.

- Init_checkpoint = uncased_L_12_H-768_A12/bert_model.ckpt

The pretrained model specified in this parameter is a pre-trained model which was used to program BERT and it contains a large corpus of English text from various books and Wikipedia. This general purpose model is able to be used for a large variety

of tasks, and is thus suitable as a base to train my first model.

I set the epoch to 3. A number of epoch(s) is equivalent to the number of times BERT runs through the entire dataset to update weights during training. In order to avoid over-fitting the data, roughly 3-5 epochs are generally recommended based on the "elbow point" approach, although this can vary depending on the data.

Much of BERT is written in pre-Tensorflow 2.0, and I consequently experienced pesky compatibility issues when running. In my attempt to rectify the issue, I tried installing a lower version of Tensorflow by calling:

- conda install Tensorflow==1.15.0

from the conda virtual environment. The command prompt then returned an error indicating that Python 3.8 is incompatible with that version of Tensorflow. I removed the entire environment in vengeful spite and started over with another—this time with Python 3.7, and Tensorflow 1.15.0

The training process took roughly 4 hours on my device, equipped with **NVIDIA GeForce GTX 1070**. I adjusted the batch size down to 2, instead of the default (32) due to the limited memory on a GPU. The model will train faster with a greater batch size, but will receive an **OOM (out of memory)** error if the hardware does not have enough memory to support it.
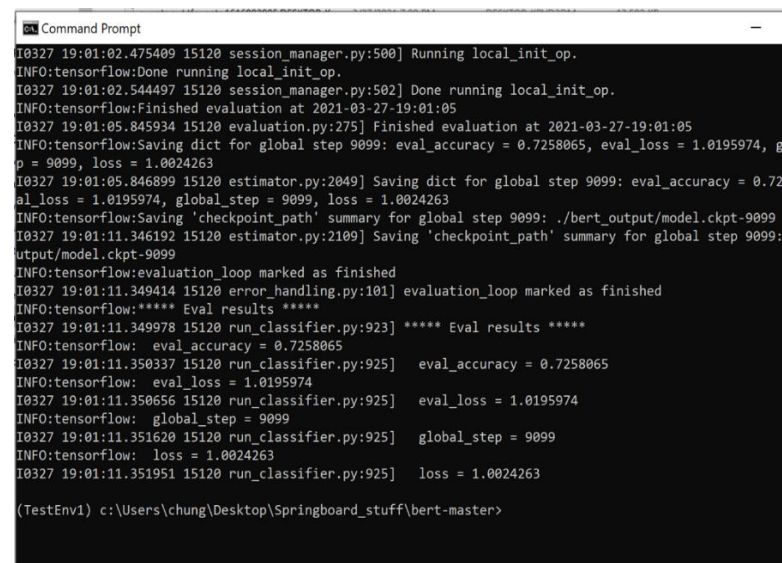

RESULTS

BERT finally finished running, ending with the global step at 9099.

Figure 3
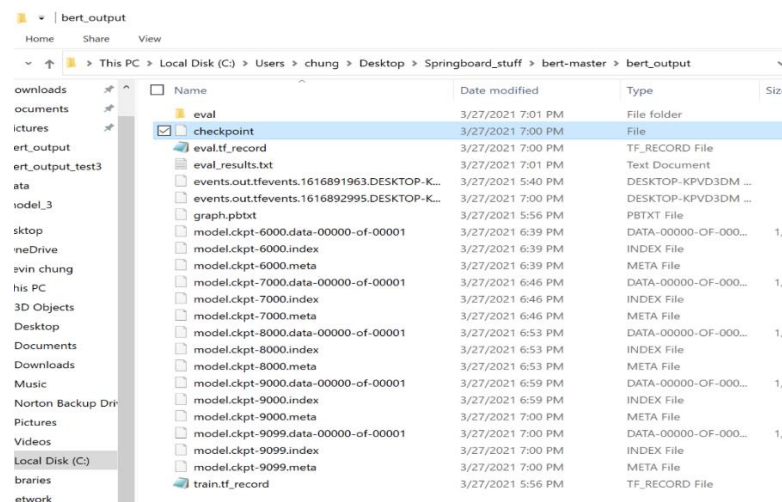
CMD output:

Figure 4

The model achieved an evaluation accuracy of 72.58% based on the test data with a loss of ~1.019. Unlike the evaluation accuracy, the loss is not a percentage. It is calculated by weighing the performance of predictions for each example, and uses the train / test dataset for its metrics. With a low loss number, I am convinced that the model predicted fairly well provided a limited dataset.

Predictions

I tried really hard to be ruthless to my creation.

I wanted to push my model and evaluate just how far it can reach, so I gathered a completely brand new dataset from reddit to test on. The new dataset had roughly the same number of rows, and kept the same labels.

Time to run the prediction with the following parameters. You will notice I omitted the "—do_train = True" line, and instead entered "—do_predict".

Figure 5

```
python run_classifier.py
--task_name=cola
--do_predict=true
--data_dir=./data
--vocab_file=./uncased_L-12_H-768_A-1
--bert_config_file=./uncased_L-12_H-7
--init_checkpoint=./model/model.ckpt-
--max_seq_length=512
--output_dir=./bert_output/
```

To run a prediction loop from my model, I made sure to initialize from the checkpoint with the highest value. The checkpoint files are the "model.ckpt-9099[…]" files sitting in the bert-output folder; you only need to specify up to the checkpoint number. Once BERT finished the prediction loop, I got the following in my output directory:

Figure 6
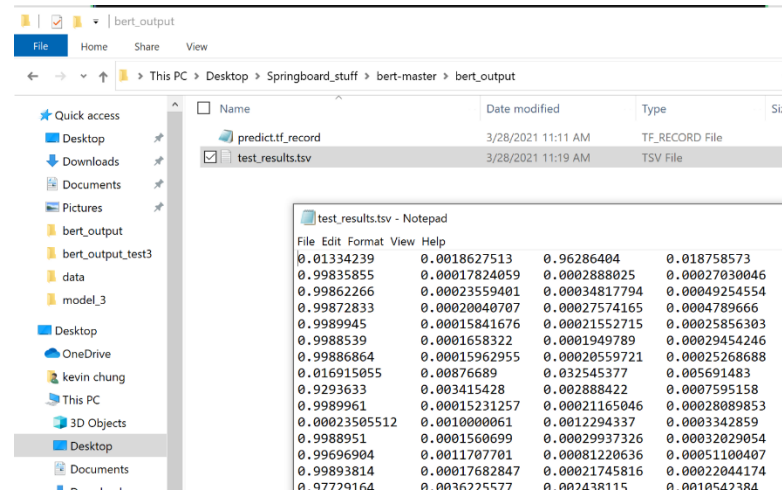
Each column of numbers represent the probability vectors that an example text belongs to a label.
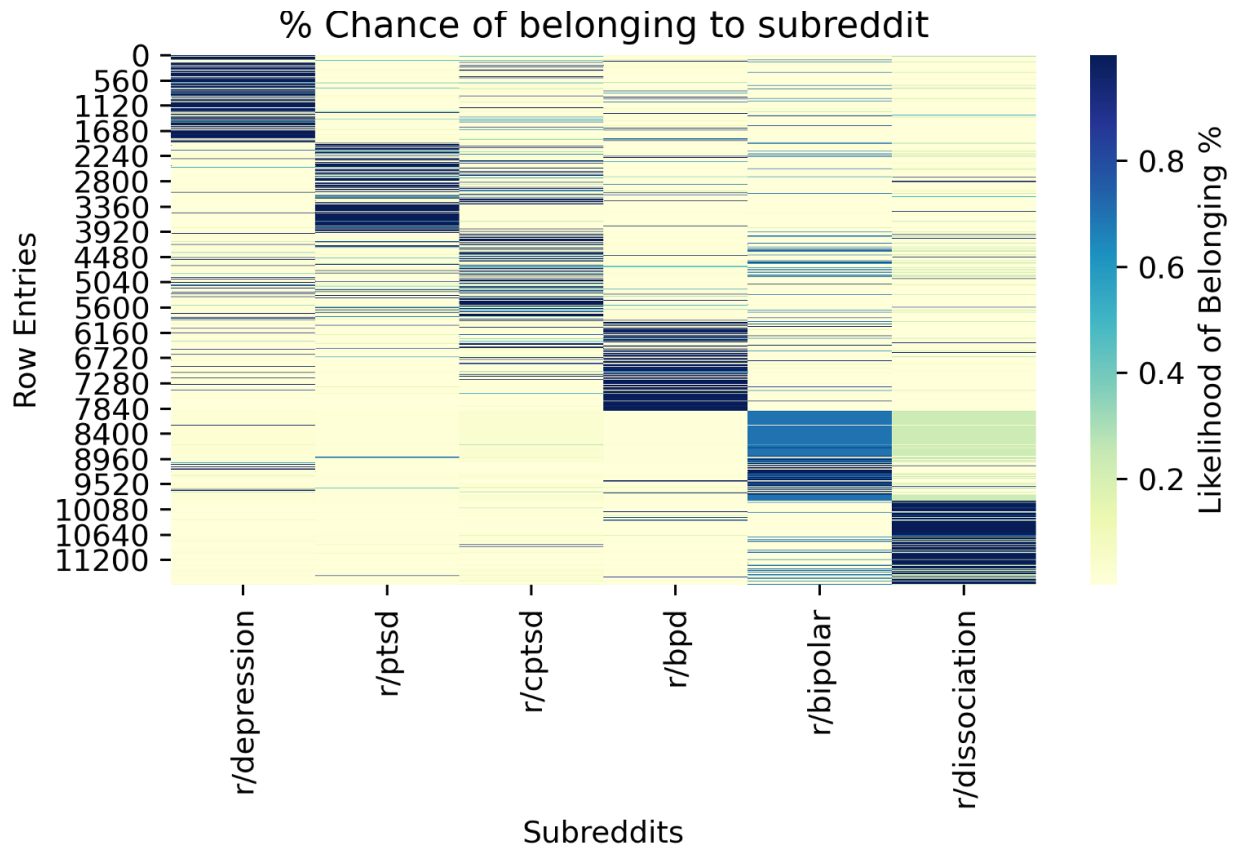
Revisiting the order of my labels:

- **Column 0** = r/depression
- **Column 1** = r/ptsd
- **Column 2** = r/cptsd
- **Column 3** = r/bpd
- **Column 4** = r/bipolar
- **Column 5** = r/dissociation

Next, I visualized the results in a few different ways. I reformat this document to improve the readability of the images.

Visualizations

I used **seaborn** to generate heatmaps. The heatmaps are useful for visualizing a high-level view of the model's overall performance. The general trend demonstrates high-probabilities of text belonging to each label descend as the row number goes up, like a staircase. The accuracy is about as expected at the level I shown in the evaluation output. I was pleasantly surprised to see that each label was clearly distinguished from one another. This is an interesting outcome for several reasons:

- The data was pulled from communities that may or may not have been diagnosed by a professional, introducing potential for mislabeled data. In spite of that the model achieved surprisingly decent accuracy. However, this risk may have been mitigated by having enough data via the **Central Limit Theorem.** Meaning, we may not see as good of a result with less data to predict from.

- BERT was able to find unique features from each label based on the writing patterns of individuals posting on the subreddits. This alludes to the potentiality of unique "thought" patterns being captured from each label—lending support for my hypothesis that this is possible at all. Of course, there are plenty of risks for confounding variables, but we can eliminate most of them by refining the experiment in future iterations. (More in part II).

- PTSD and CPTSD labels were significantly distinguished from one another, lending evidence to support ICD-11's decision to include CPTSD as an independent diagnosis. This was an especially powerful and validating realization for me, as someone who struggles with CPTSD.

Figure 8

correlations heatmap
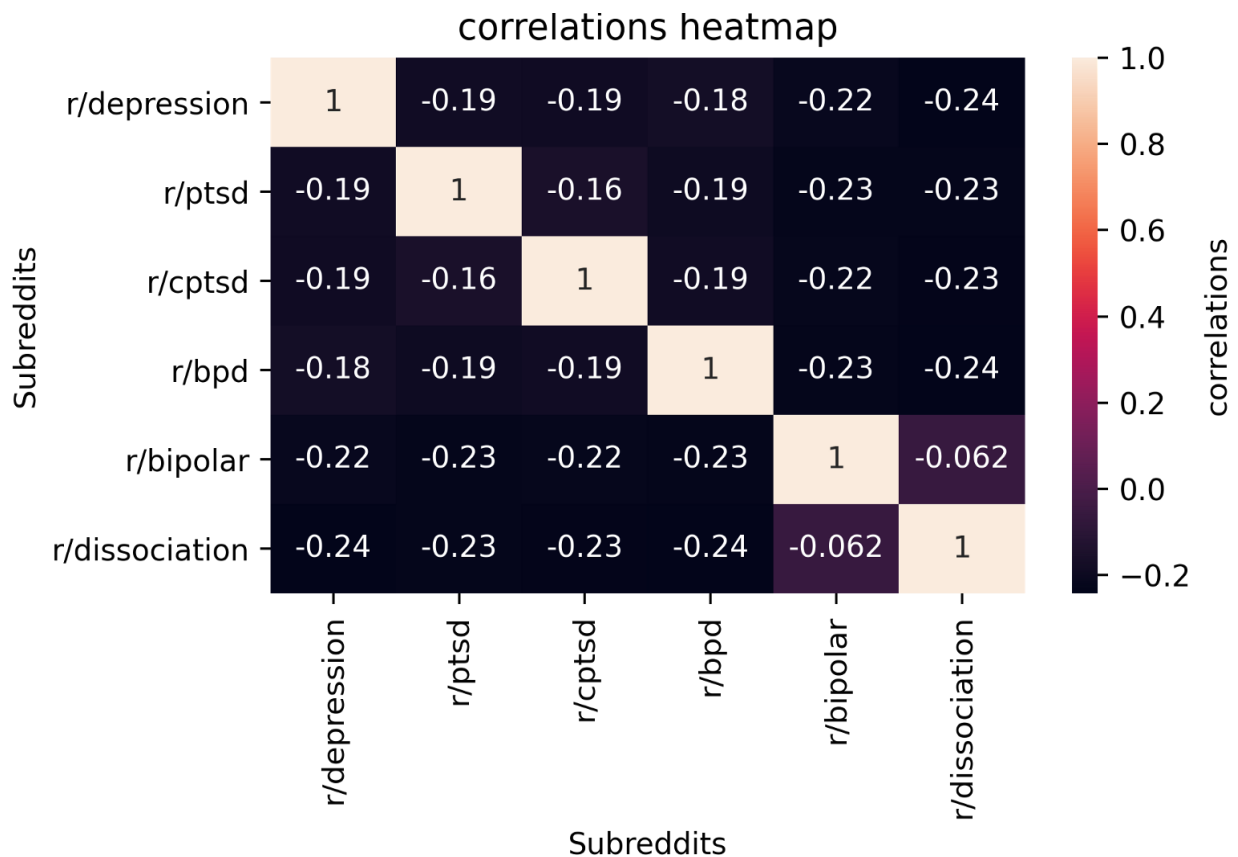
The above is a correlations heatmap generated by seaborn. Although the negative correlations appear exaggerated due to the coloring of the heatmap, it does achieve negative correlations across the board to distinguish each label from another with confidence. At the simplest interpretation, this graph demonstrates that BERT is capable of capturing the subtle differences between the features in each label.

These heatmaps tell a story about the macro trends in the prediction results. In order to obtain a closer view, it is necessary to evaluate how effective the model was for each label. Since I knew the exact number of rows were in each label, I generated the following graphs with ease using **matpoltlib**.
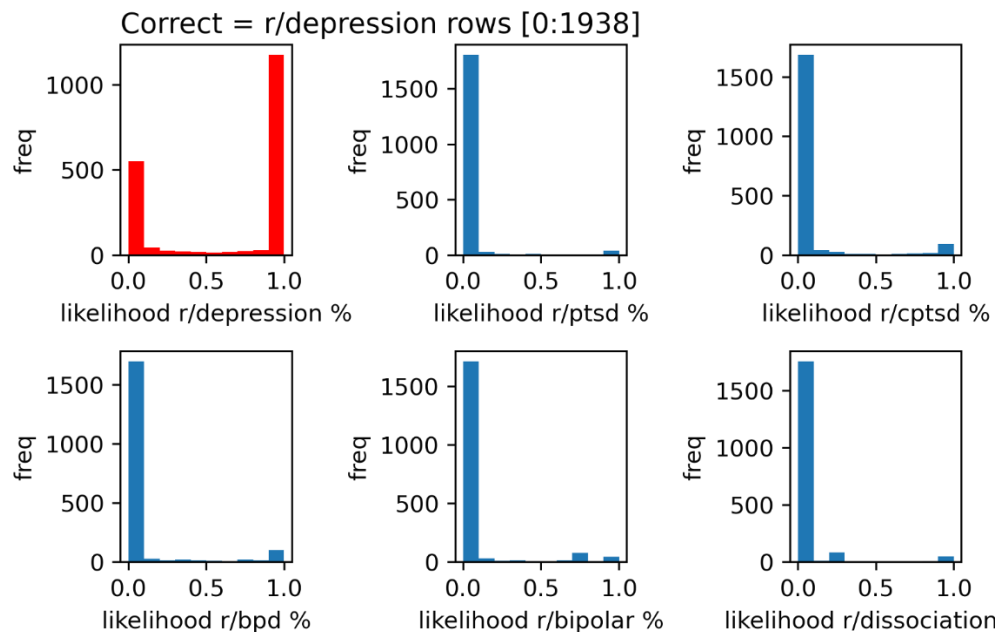

Figure 9
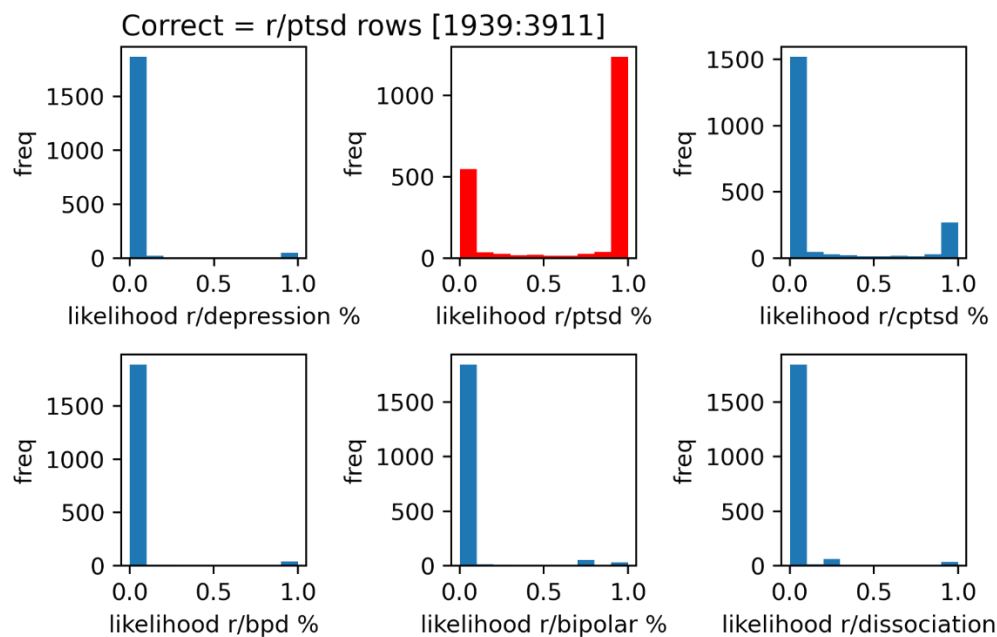
Depression was the correct answer.

Figure 10

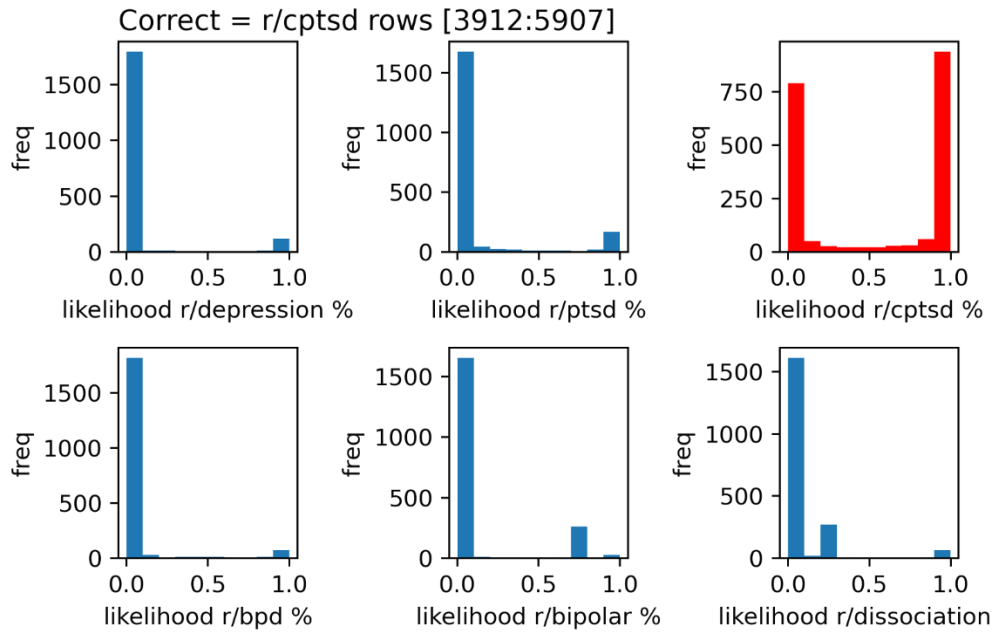PTSD was the correct answer.

Figure 11

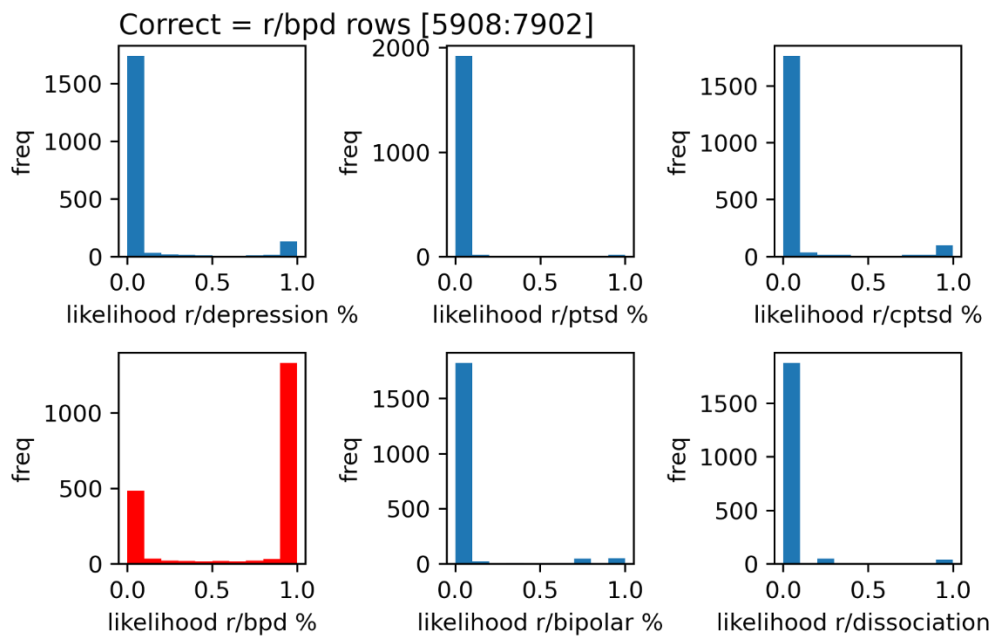CPTSD was the correct answer.

Figure 12

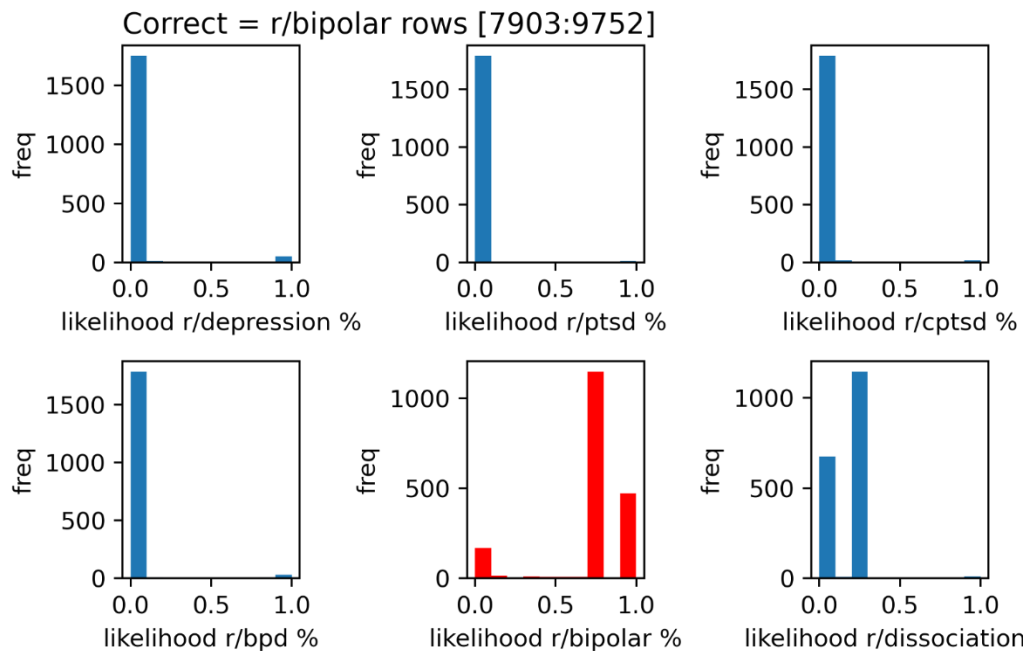Borderline Personality Disorder was the correct answer.
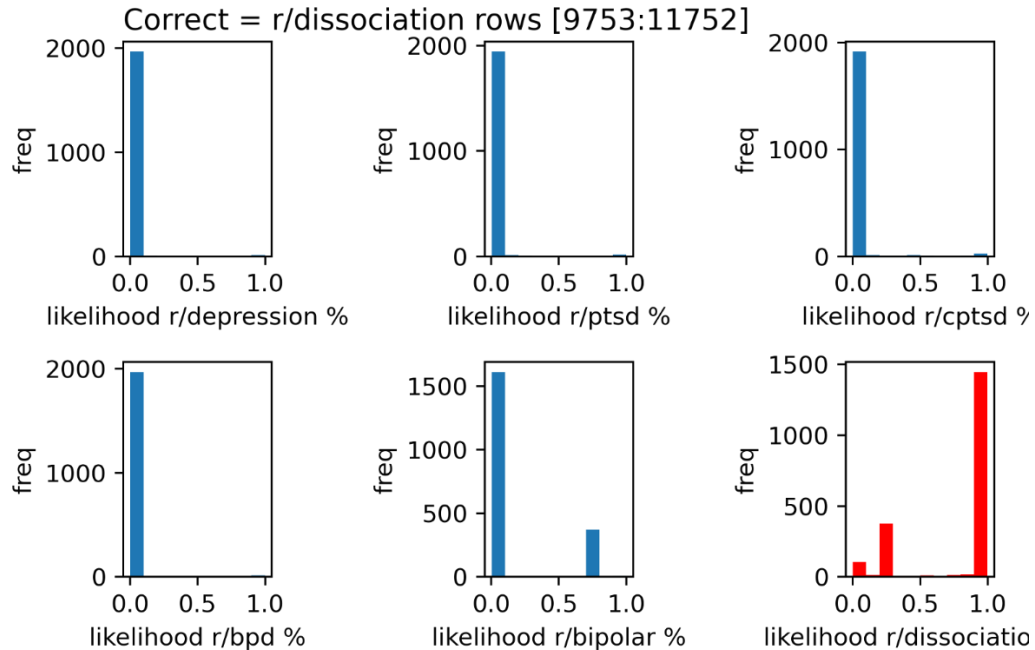
Figure 13

Bipolar was the correct answer.

Figure 14

Dissociation was the correct answer.

DISCUSSION

One interesting effect from the way this experiment was designed, is that I am able to view the probability vectors for all six labels for a single example text. This means I am able to view all six probabilities that an input text belongs to each label. In a winner-takes-all approach, I would only be able to view that my model predicted each label correctly from their respective positions on the dataset. One way to interpret the ramifications is that the displayed accuracy may not reflect the actual accuracy of the model, given that mental illnesses can develop comorbidities and remain on spectrums.

This could also explain the reasons for why CPTSD and PTSD were more closely related than others when seen under the histogram for each respective label (*see figures 10 & 11*). For example, when the correct answer was the PTSD label, there were both fair amounts of incorrect answers and correct answers from both PTSD and CPTSD compared to other labels. While this relationship is not demonstrated easily provided the negative correlations, it is nevertheless an interesting effect to note.

With that said, there are drawbacks to this dataset in many ways. One obvious one is that CPTSD has more null values in its rows than others. This is due to a community specific behavior of writing only titles, and not the body columns when posting. I purposely left this feature in the dataset and did not attempt to fill the rows with more content in this iteration of the experiment because this could be an important feature that is specific to this community compared to others.

A major fault in the current design is that it also lacks a label which serves as a control variable. In the future, when the model displays more competent results, I plan to include a column of random texts from several sources in the future to emulate examples of casual conversation instead of anecdotes regarding specific experiences. I'd be curious to know how this would influence the outcome.

There is also the possibility that the features extracted from this dataset reflect platform-specific behaviors. In fact, it is likely the case that different online communities have different rules and cultures for posting. This is a significant hypothesis worth investigating in and of itself. I explore this hypothesis further in the part 2 of the paper where I draw data from other sources and put it to the test against test data from cross platforms.

While there is much more to be discussed about the dataset and the ensuing results, the fact remains that there are more questions than answers at this point. However, it is of personal opinion and vindication that the results of this experiment show promise. Therefore, the possibility of an empathetic artificial intelligence seems likely, albeit it cannot be proven at this time.

Concluding Thoughts…

I struggled with debilitating mental illnesses for a majority of my life. Relatively speaking, 24 years hasn't been a very long life; however, experiencing failure after failures, missing opportunities, self-

sabotaging support systems and close friendships, and the unending numbness to it all made those years very long lived. The journey to recovery ahead feels even longer, but I am diligently taking my steps.

Moments in life can be frustrating, painful and unfair, and sometimes these sentiments haunt us long beyond the expiration of those terrible memories. However, what we often forget first are moments of our strength and resilience precisely during those times. It's a pity we don't always see the light in ourselves when they shine brightest, but their effects are definitely observed in contemplations of gratitude for the present moment. My experiences led me to study psychology in university, to work and keep different jobs, and they eventually became inspirations to build this project. Friends, family, Nintendogs— these are things to be grateful for.

This project is my story, and my attempt at turning the negatives in my life to something positive for others. There is something powerful about storytelling in the way that it inspires both the storyteller and the listener. I am grateful for the stories I found on reddit.

But admittedly, not many of us are actually lucky to have a listener when it really counts.

It's why services like the Suicide Prevention Lifeline **(800) 273–8255** and the Crisis Text Line (**Text HELLO to 741741**) exists. I often think about times I wished for someone to talk to besides a wall and a bottle of whiskey—surrogates for an audience and a mic, I suppose.

How emotionally transformative would it have been to simply know, someone besides myself *knows* me? Would it even be possible to vocalize these experiences? I can't peer inside the mind of someone's past, and neither could one see mine. But it is possible to see reflections of ourselves and of others when we write honestly. And what if the paper could speak back to us? If the wall is no longer just a wall, and the mic doesn't get you drunk?

I hope to progress this project to an extent that people may have regular access to a "listener" who understands, and can help us understand our stories better in the way data spells. Life is full of struggle and terrible tribulations, but they are not without resilience and eventual triumph. People struggling with mental illnesses experience overwhelming isolation and loneliness, yet their stories are full of transformative gems. Their stories deserves to be heard and understood. First, perhaps by an A.I. "therapist" you spill your guts to. Then, by a real therapist your A.I. friend connects you to, for when real therapists aren't accessible.