Authors, Email Addresses, Course Sections:

Shatkratu Swarnkar [ss4092@scarletmail.rutgers.edu] **6**

Yuet Yue [yy655@scarletmail.rutgers.edu] **6**

Apsara Saraswat [as4451@scarletmail.rutgers.edu] **6**

Calvin Jude [cj500@scarletmail.rutgers.edu] **5**

Final Project Report

Professor Kulikowski

May 13, 2025

The Convolutional Conversationalist: Utilizing the GAD-7 to Test for Anxiety

Abstract

One of the most prevalent emotions that every participant within a particular field of study within an institution of higher education experiences is what our team believes to be anxiety. Additionally, our team believes that this trend has been even more apparent within the field of Computer Science, where an increasing amount of incoming admits pursue this field every year at Rutgers University, which in turn, creates an environment of higher competition within the current recession in the career market for Computer Science majors, specifically in the niche of software and cloud engineering.

Rationale and Goal(s)

While planning the initial project proposal, our group wanted to recreate what the major artificial intelligence-based pioneering companies have already created within the technological market today, Large Scale Language Models. Our group took inspiration from the success of OpenAI's ChatGPT, Anthropic's Claude, and Perplexity's PerplexityAI within today's technological sector, revolutionizing how different corporate industries across all sectors of the market operate, leveraging Large Language Models (LLMs) to automate and efficiently increase the rate of production without the need to incorporate more human employees. Our group also took inspiration from how these LLMs have created a new type of learning space for students across the world, from early education, such as elementary and middle school, along with intermediate education such as high school, into higher education, such as college (Bachelor's, Master's, Doctorate). In the current educational space, LLMs act as one of the best search engines in the world, as they have scraped data from all corners of the internet that is available for public use. Furthermore, LLMs are interactive, which means that students of all levels can utilize such models as a tutor, proofreader, and to some degree, a teacher. Through this, our group understood the broad implications of what an LLM could be, an employee and an educator, of course, to some degree. However, our group wanted to push the capabilities of artificial intelligence further. As stated above within the abstract, our group believes that there is currently a high level of anxiety within the community of students at Rutgers University who are studying Computer Science. The rationale of our project is to answer one simple question: If a LLM can act as an employee and an educator, to some degree, would it also be possible for a LLM to act as a therapist?

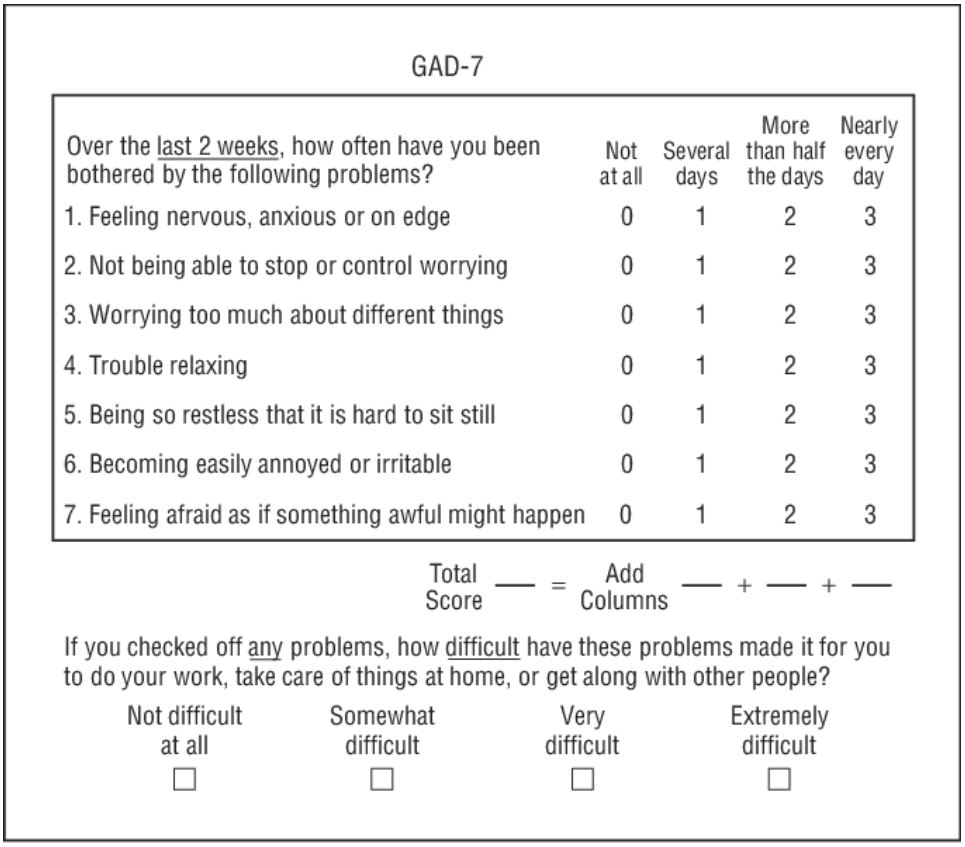
In the current medical practices within the United States, patients would have to pay a generous portion of their financial capital to speak with a therapist, as this is considered a treatment of the mind within the medical industry. However, there are pitfalls to this system as some students are discouraged from seeing a mental health professional due to, firstly, the cost of what a therapist or counselor charges on an hourly basis. Furthermore, this obstacle ties into the system of thought where students believe that they do not need to see a therapist and discourage themselves from doing so, even under the high levels of stress and anxiety of prospects. The goal of our final project for this course is to create a preliminary therapist that is easily accessible and can diagnose levels of anxiety based on the General Anxiety Disorder-7 (GAD-7) Test so that students can be more aware of their mental health. We also wanted to make this project free so that anyone can use it without dealing with the barrier of financial cost, along with the process of medical paperwork needed to be completed when one initially decides to go see a therapist.

Materials and Methods

The sources that were reviewed during the research phase of this final project are listed below in the Bibliography Section, where it is formatted in APA format. During the research phase of this project, we realized that LLMs take a lot of computational resources to run, as not only the computation needed for the creation of responses based on user input took a lot of computing power needed within the Graphics Processing Unit, it also needed a lot of storage space within what was needed in the Random Access Memory. To scale the project even further, our group realized that the LLM that we wanted to create also needed to memorize all previous user inputs along with the corresponding generation of sentences from the LLM. Since we were first planning to run our hypothetical LLM locally within our laptops, we realized that this process of storing not only the user inputs, but also the generated responses, would take a large amount of hard drive storage capacity, most likely in the terabytes as we knew we had to test this extensively before submission. Unfortunately, no one in our group had a high end desktop that would be considered possibly the minimum to be able to run such a project, therefore we sought to contact the Rutgers Office of Information Technology in regards to possibly using the Amarel Computer Clusters as we read online that researchers used the Office of Advanced Research Computing's Amarel Computer Cluster to reap results that would be approximate to what our group wanted to achieve. However, we have unfortunately been informed that only researchers, faculty, and staff are permitted to use such systems, therefore, we had to seek mentorship from our teaching assistant for the completion of this project, as his current field of study within his doctoral program corresponds with LLM interactions.

In regards to the mentorship of this project, our group utilized the knowledge that our teaching assistant, Gerasimos Chatzoudis, had in regards to the possible resources for artificial intelligence models available for public use without needing an extremely power computer with premium Graphics Processing Units, a large excess of Random Access Memory, along with an even larger capacity in hard drive storage space. Our teaching assistant recommended that we utilize what Hugging Face had in regards to the open models that were available for use by the general community of people who were interested in artificial intelligence. This was what we realized was the first deviation of our final project compared to our initial project proposal, where instead of building a convolution neural network from scratch and training the model to learn the most basic functions of responding to a user input, we would have to instead use a pretrained model that had to most basic functions of being able to take in a user input, responding to such user input with the proper context through sentence generation, and memorizing all previous user inputs and corresponding responses to ensure intractability when a user is typing out inputs to the model. Our team was first uneasy with this decision; however, realizing the time constraints given to us within this semester, building an LLM from scratch would be infeasible, therefore, the more plausible route would be to use a pretrained model instead.

The pretrained model our team decided to use was an older version of Llama, which was an open-source artificial intelligence model that Meta has made available for public use. Utilizing this, our team now needed to build off the pretrained model to fit our needs. Within the research phase, we understood that the foundation of the project also included a specified metric in which we needed to define levels of anxiety. No one on our team is a licensed therapist, nor a specialist within the field of psychology and cognitive science, therefore, we have decided that the GAD-7 would be the best metric to use as a preliminary assessment of one's anxiety levels. The figure below shows what the GAD-7 is and how it approximately calculates one's general anxiety from a scale defined within.



GAD-7

Over the last 2 weeks, how often have you been bothered by the following problems?	Not at all	Several days	More than half the days	Nearly every day
1. Feeling nervous, anxious or on edge	0	1	2	3
2. Not being able to stop or control worrying	0	1	2	3
3. Worrying too much about different things	0	1	2	3
4. Trouble relaxing	0	1	2	3
5. Being so restless that it is hard to sit still	0	1	2	3
6. Becoming easily annoyed or irritable	0	1	2	3
7. Feeling afraid as if something awful might happen	0	1	2	3

Total Score ___ = Add Columns ___ + ___ + ___

If you checked off any problems, how difficult have these problems made it for you to do your work, take care of things at home, or get along with other people?

Not difficult at all □ Somewhat difficult □ Very difficult □ Extremely difficult □

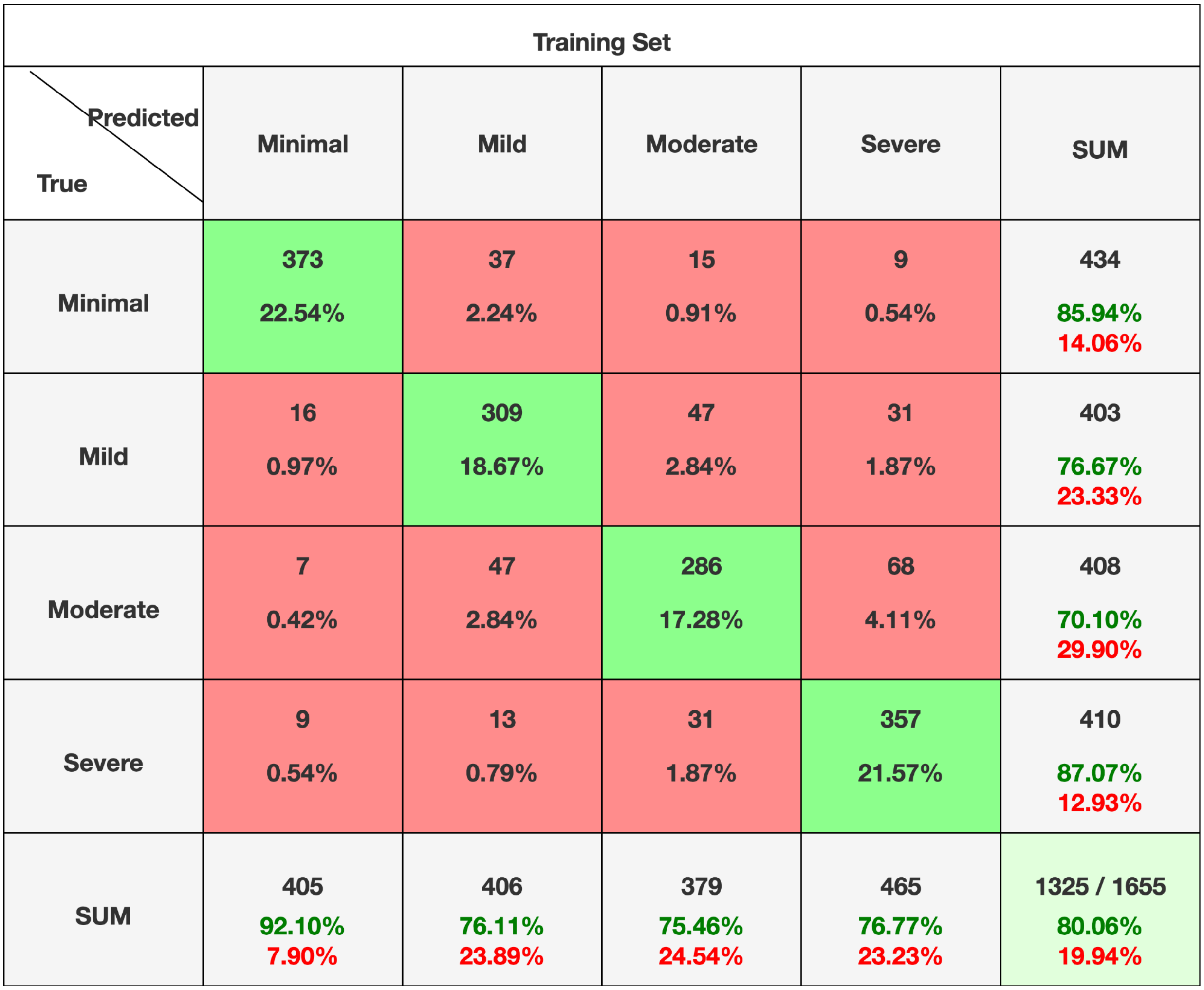
After we have decided upon the baseline assessment of what anxiety can be defined as, we reflected on such assessment through its categorical scores within the code of our project. Within our project, we first had a feature extractor that would take in each of the responses to each question illustrated within the GAD-7, which allowed us to assign scores to each of the questions listed. After that, we have the classifier file that would take the individual assessments from the feature extractor and combine them to create the final score. Here are the conditions in which we classified the specific anxiety level a user would have based on the GAD-7 test's metrics.

- A score below 4 would deem a user's anxiety level to be **MINIMAL**
- A score between 4 to 9 would deem a user's anxiety level to be **MILD**
- A score between 10 to 14 would deem a user's anxiety level to be **MODERATE**
- A score that is above 14 (15 or over) would deem a user's anxiety level to be **SEVERE**

As our group defined the scale in which what level of anxiety a user contained, we moved forward with the creation of a simple front-end UI page so that the user would be able to interact with the model instead of having to install our project code and respond to the now trained model within terminal using command line arguments, hence, our team was able to create a full-stack application that allowed for ease of use for the user so that the user would be able to preliminarily assess their anxiety levels.

The main roles of each of the authors within this project have been combined as we worked around multiple different aspects of this project, assisting each other when one of the members was stuck on a particular step in building such a project. The lead design of this project was mainly handled by Shatkratu Swarnkar and Yuet Yue, who first broke down the project into multiple steps and researched the foundational anxiety assessment, this being the GAD-7, for the assignment of anxiety categories to users based on their responses to the GAD-7 test's inquiries. They also kept in contact with TA Gerasimos Chatzoudis to obtain advice and possible directions to take during the building of this project, for example, which model to use and the range of parameters to use to be able to run this model on the group's machines. The principal implementation of this project was handled by Apsara Saraswat and Calvin Jude, where they handled the creation of the classifier methods along with the feature extractor methods, ensuring that the implementation of the GAD-7 test's ratings of anxiety were functional when a user would interact with the model in regards to assessing their anxiety levels preliminarily. The principal designer of this project was handled by Shatkratu Swarnkar, who created the front-end UI page so that users would not have to download our project code and interact with the trained model within the command line interface and finally, the writer of the polished paper, which is this final report, is Yuet Yue, and this report is proofread by all members of the group before its submission onto the canvas page for this course.

Results



Training Set

Predicted True	Minimal	Mild	Moderate	Severe	SUM
Minimal	373 22.54%	37 2.24%	15 0.91%	9 0.54%	434 85.94% 14.06%
Mild	16 0.97%	309 18.67%	47 2.84%	31 1.87%	403 76.67% 23.33%
Moderate	7 0.42%	47 2.84%	286 17.28%	68 4.11%	408 70.10% 29.90%
Severe	9 0.54%	13 0.79%	31 1.87%	357 21.57%	410 87.07% 12.93%
SUM	405 92.10% 7.90%	406 76.11% 23.89%	379 75.46% 24.54%	465 76.77% 23.23%	1325 / 1655 80.06% 19.94%

After the completion of the basic functionality of our trained model, our group moved on to the testing phase of our project. The testing methodology was simple; we conducted 1655 total inputs, where each of our group members replicated specific inputs that were catered towards a certain level of anxiety. If one were to take a look at the figure above, one can see that it is a confusion matrix that shows the accuracy of the anxiety levels that the trained model assigned to each response. If one were to read the "SUM" results that are structured into rows based on the results listed within each column, one can see that it represents the percentages that inform the reader about the prediction performance of the trained model. The prediction performance is based on the performance per predicted class, which standardizes for how accurately the model was able to assign the correct anxiety ranking based on the biased input of our group members, hence it helps answer the question of how correct the model was given a certain input catered towards a specific anxiety category. For example, the total times the model predicted the user to have a certain anxiety level was 405, and of course, the user input bias was catered to the **MINIMAL** level of anxiety. Within these 405 times that it assigns an anxiety level to a user, it only assigned the correct anxiety category 373 times, with the other assignments being incorrect. The green percentages represent the correct assignments of categorical anxiety, and the red percentages represent the error rates of the model's assignments. Looking into the "SUM" results that are structured in column form, which are taken from the reading of the individual tests of each row, the legend represents the percentages that indicate the actual class performance of the model. The actual class performance of the model represents the accuracy of the model in correctly identifying the specific category based on the "true" label being minimal, hence it is testing the ability of the model to be able to identify the user input to be biased within a certain manner.

Conclusions from Results

Based on the results listed in the previous section, we know that the model itself is not fully accurate, as it only has an 80.06% accuracy, where our group decided that being fully accurate would require at least a success rate of at least 90%. Even though our model is not fully accurate in regards to this, our group is happy with our results, given that this is the first time that we have tried to emulate an LLM, especially trying to train a pretrained model with arbitrary prompts that would determine something as complex as anxiety. However, our group also believes that even if we were not “fully” accurate, we were at least able to achieve results that would imply that our model is “highly” accurate. If there was more time given without the constraints of this course granting only half a semester to complete such a final project, our group would have been able to scale this project further to create a more accurate model that measures its accuracies through a higher level of testing, for example, if our group were to be able to generate over 10,000 inputs through a more detailed baseline reference of a test as that would allow for us to test even higher amounts of nuances to determining the anxiety levels of users so taht they can be assigned to proper anxiety categories.

Group Statement and Reflection

This project was very ambitious as our group tried to tackle one of the most revolutionary creations of the emergence of artificial intelligence, a Large Language Model. Throughout this project, our group learned a lot about the research phase, from mitigating the constraints that we had no control over, such as computational power and time, to finding solutions to alleviate such issues, which involved using a pre-trained model and adding training attributes of our own. More specifically, our group learned the following things: firstly, that the creation of an LLM, even the most bare-bones and simple LLM, would take a lot of time. Not only did we underestimate the amount of time it would take to create a convolutional neural network from scratch, but we also underestimated the computational power needed to have such a model be functional, especially since we had the intention of deploying the model for public use. In this particular regard, if our group had the opportunity of redoing this project from scratch, we would first find a method to utilize the vast amount of resources that the university offers in computational power to be able to compile and run such a resource-intensive project. However, the time constraints would be something that we would not be able to control, therefore, we believe that, due to the time constraints and if we were to redo this project from the start, we would end up using a pre-trained model again. Secondly, when looking at the GAD-7, our group knew that the GAD-7 is a generalized test, if given the chance to redo this project again, our group would try to find a more nuanced anxiety test or combine multiple anxiety tests to train our model with so that the model's assessment of anxiety would be more established. Building off of this, the final aspect that our group would have changed is that if we had a more nuanced version of an anxiety test to model our classifiers and feature extractors off of, our group would have been able to create more classifiers based off more prompts referenced off of a more nuanced anxiety test within the feature extractor.

Individual Statements from Authors

"Throughout this project, I deepened my full-stack web development skills by building a Flask-based frontend and backend that handle user authentication (via Flask-Login and Flask-Bcrypt) and persist data through SQLAlchemy. I structured and integrated a relational database to track questionnaire responses and model outputs, learning to manage environment variables securely with Python-dotenv. Although the core anxiety classifier and spaCy feature extractor were largely pre-existing, I gained valuable experience wiring those components into the application's logic. Most notably, I learned how to integrate and deploy a Llama-3.2-1 B-Instruct conversational model from Hugging Face, creating a seamless pipeline that connects user input, ML inference, and dynamic UI updates. Through this process, I solidified my ability to connect modern web interfaces with machine learning services and database systems in a maintainable, production-ready architecture." - **Shatkratu Swarnkar**

“Having the opportunity to build such a complex project, especially with having set high standards to somewhat compete with the global pioneering companies in artificial intelligence through LLMs, I have learned not only the pitfalls but also the requirements for being able to operate a complex system of artificial intelligence through a chatbot interface. During the research phase, I realized the computing power required to run such a project, such as needing an abnormal amount of VRAM within my machine's Graphics Processing Unit, an excessive amount of Random Access Memory, and even more storage space within my hard drive to run the now trained model locally without fail. Altering the parameters of the pre-trained model at first to fit the capabilities of my machine was also an attribute of LLMs that had to be learned along with the general functionality of how LLMs “memorize” the conversation that occurs between themselves and a user. The functionality of feature extraction, categorical classification, and interpreting the dataset obtained from rigorous testing was a focal point during the building and testing phase of the project.” - **Yuet Yue**

“I've learned how to select a proper dataset given certain requirements, how to clean and transform data (using the Python library pandas), train a logistic regression model for classification of data, and save and load this model for use (using the Python library joblib). Additionally, I've learned how to extract structured data from unstructured text in the user's input, particularly by using the Python library spaCy for named entity recognition, creating dictionaries to map qualitative data to quantitative classifications, and designing patterns found in the unstructured data. These patterns include handling numerically written and spelled-out numbers (using Python library word2number), expanding contractions in words for better parsing, and extracting important features such as age, sleep, life satisfaction score, etc. (using Python library re).” - **Apsara Saraswat**

“In this project, I developed an ML pipeline for a natural-language feature extraction that converts a free-form user input to structured psychological and behavioral metrics using SpaCy’s dependency parsing and named entity recognition and in combination with regex-based pattern matching and linguistic normalization (e.g., ‘I’m twenty-one’, ‘sleep about seven hours’) for better recalling on noisy or informal text. I also helped in refining a lightweight classifier for predicting anxiety/wellness’s. My focus was on feature importance analysis, as well as validation of generalizability of results over the user segments. Other than backend processing, I worked with the frontend team to establish a line of communication between the UI layer and our CNN model API endpoint. Such an end-to-end involvement exposed me to model-driven inference as well as full-stack application deployment and allowed me to learn a lot about natural language processing as well as connectivity, databases, flask, and hugging face.” - **Calvin Jude**

Project Repository: https://github.com/shicky1223/CS440_Final

Sources and References

Russell, S. J., & Norvig, P. (2020). *Artificial intelligence: A modern approach* (4th ed.). Prentice Hall.

Stack Overflow. (n.d.). *Stack Overflow*. Stack Exchange. https://stackoverflow.com

ChatGPT. (n.d.). *ChatGPT*. https://chatgpt.com/

Ray, P. P. (2023). ChatGPT: A comprehensive review on background, applications, key challenges, bias, ethics, limitations, and future scope. *Internet of Things and Cyber-Physical Systems, 3*, 121–154. https://doi.org/10.1016/j.iotcps.2023.04.003

OpenAI. (n.d.). *How ChatGPT and our foundation models are developed*. OpenAI Help Center. https://help.openai.com/en/articles/7842364-how-chatgpt-and-our-foundation-models-are-developed

AssemblyAI. (n.d.). *How ChatGPT actually works*. AssemblyAI. https://www.assemblyai.com/blog/how-chatgpt-actually-works/

Hugging Face. (n.d.). *Introduction - Hugging Face NLP course*. Hugging Face. https://huggingface.co/learn/nlp-course/en/chapter1/1

Henrythe9th. (n.d.). *AI crash course to help busy builders catch up to the public frontier of AI research in 2 weeks*. GitHub. https://github.com/henrythe9th/AI-Crash-Course

Paulk, R. (n.d.). *Codecrafters-io/build-your-own-X: Master programming by recreating your favorite technologies from scratch*. GitHub. https://github.com/codecrafters-io/build-your-own-x