UCL
SCHOOL OF
MANAGEMENT

MSIN0221: Natural Language Processing

Group Project Final Report: Group-16

Mental Health Support Chatbot: FIDATO

NAME	CANDIDATE NUMBER	STUDENT NUMBER
KHUSHI BANSAL	HCXN4	23098592
BING YANG	JTPW4	23095098
VARTIKA CHAUHAN	HNTM8	23189912

WORD COUNT	3837
PAGE COUNT	20

Table of Contents

Introduction

Mental well-being is integral to overall health, a fact brought into sharp relief by the COVID-19 pandemic's exacerbation of mental health challenges. The World Health Organization (WHO) reported that, prior to the pandemic, 12% of the global population was already dealing with mental disorders in 2019. This situation has dramatically worsened, with the WHO documenting a 27.6% increase in major depressive disorder cases and a 25.6% increase in anxiety disorders globally (World Health Organisation, 2022). The escalation has been particularly acute among individuals aged 20-35, who are at increased risk for suicidal and self-harming behaviours. (Global Burden of Disease (GBD), n.d.).

Despite their significance, mental health problems are frequently overlooked or misunderstood in daily contexts; neglect exacerbated in developing nations due to insufficient mental health services(World Health Organisation, 2022). The absence of awareness and intervention can lead to severe, long-lasting consequences. An example of this issue is Iran, showcasing one of the highest age-standardized rates of mental disorders worldwide. This highlights the pervasive challenge and underscores the imperative for global action. The statistics provided by the World Health Organisation (2022) and insights from the Global Burden of Disease study emphasise the urgent need for improved mental health awareness and services as the international community contends with and moves beyond the COVID-19 pandemic's impacts.

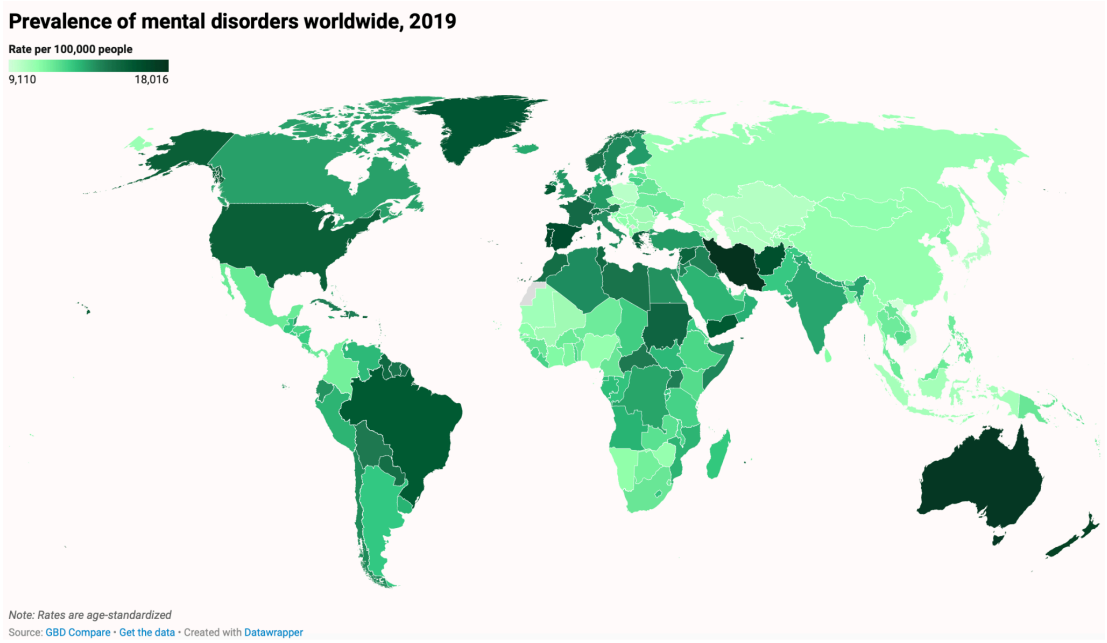


Figure (1)

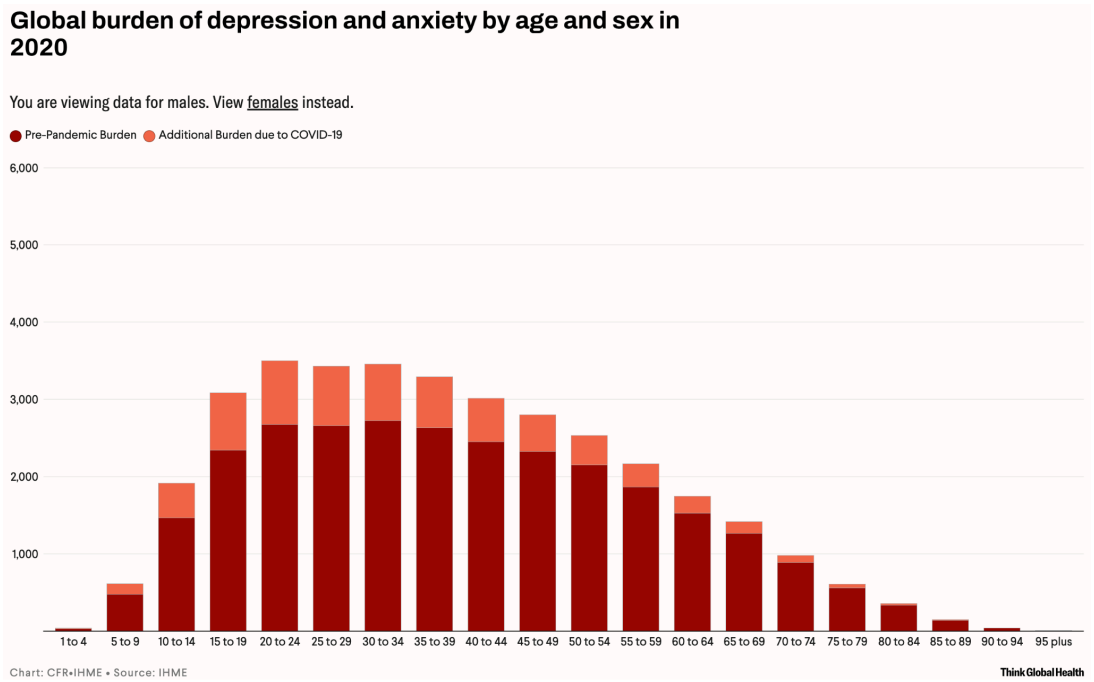


Figure (2) : Males

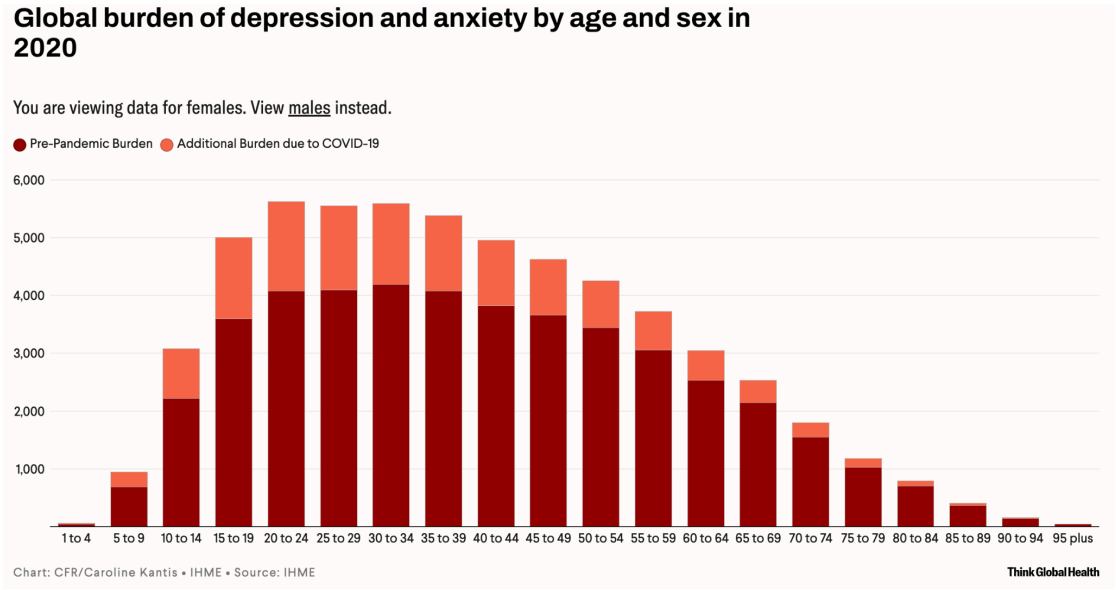


Figure (3): Females

The advent of Natural Language Processing (NLP) technology has revolutionised the field of mental health, offering new methods for providing early-stage consulting services. Leveraging the capabilities of computational linguistics and machine learning, NLP can analyse linguistic patterns and emotional expressions in text, offering substantial potential for identifying and supporting individuals facing mental health challenges. This innovation enables more accessible and more affordable access to mental health resources compared to traditional treatments like cognitive behavioural

therapy (CBT). While not a replacement for professional human intervention, NLP is a vital tool for early symptom detection and awareness, aiding in timely human follow-up.

Recognising the vital interplay between NLP and mental health, our team has initiated the development of a specialised mental health support chatbot. This project aims to devise an intelligent conversational agent adept at classifying and pinpointing five distinct mental health conditions using advanced text classification techniques. Furthermore, upon identifying specific mental health concerns, the chatbot is programmed to offer tailored suggestions, drawing from a comprehensive FAQ sheet to steer users towards appropriate assistance and further help. While the current iteration is a prototype constrained by data access and time limitations, our ambition is to refine and expand its capabilities.

Addressing the urgent need for early-stage mental health support, our chatbot endeavours to answer a critical question: "In situations where individuals are grappling with the initial stages of stress or anxiety, hesitant to reach out due to stigma, or simply seeking a quick, preliminary assessment before committing to professional therapy, what accessible solution can offer them immediate, confidential, and empathetic support to navigate their concerns and guide them towards the right resources?" Through this question, our chatbot aims to bridge the gap in mental health accessibility, offering a discreet, immediate, and understanding resource for those in need and guiding them towards proper mental health care and support.

Background and Literature Review

Target mental disorders

During the COVID-19 pandemic, several mental disorders have been particularly prevalent, including cognitive and attention deficits (commonly referred to as "brain fog"), anxiety, depression, substance use disorders, seizures, and suicidal behaviour (National Institutes of Health (NIH),2023). For our study, we have chosen to focus on anxiety, alcoholism, depression, and PTSD as our primary mental health conditions. These disorders have been selected based on their propensity to remain untreated during the early stages, which can lead to worsening conditions over time.

According to the NHS website, our selected five mental disorders and their symptoms are defined accordingly: (NHS, n.d.)

- Anxiety: a feeling of unease, such as worry or fear, that can be mild or severe.
- Alcoholism: an addiction to drinking alcohol, meaning in conditions where someone loses control over their drinking and has an excessive desire to drink.
- Depression: This is a common disorder, including symptoms of depressed mood or loss of interest, most of the time for at least two weeks, that interfere with daily activities.
- PTSD (post-traumatic stress disorder): often caused by very stressful, frightening or distressing events. It shows that someone usually relives the traumatic event through nightmares and flashbacks and may experience feelings of isolation, irritability and guilt.
- ADHD (Attention Deficit Hyperactivity Disorder): a condition that affects people's behaviour. People with ADHD can seem restless, may have trouble concentrating and may act on impulse.

Projected Scope of our Chatbot Fidato:

Fidato is planned as an innovative mental health chatbot that utilizes the BERT model for nuanced text classification. It will set a benchmark in identifying potential mental health issues through user interactions. Unlike traditional platforms that may inundate users with information, Fidato focuses on understanding and analyzing users' expressions and emotions, aiming to provide immediate, personalized support. This strategy enhances the user experience by creating an engaging, supportive environment conducive to mental health awareness and assistance.

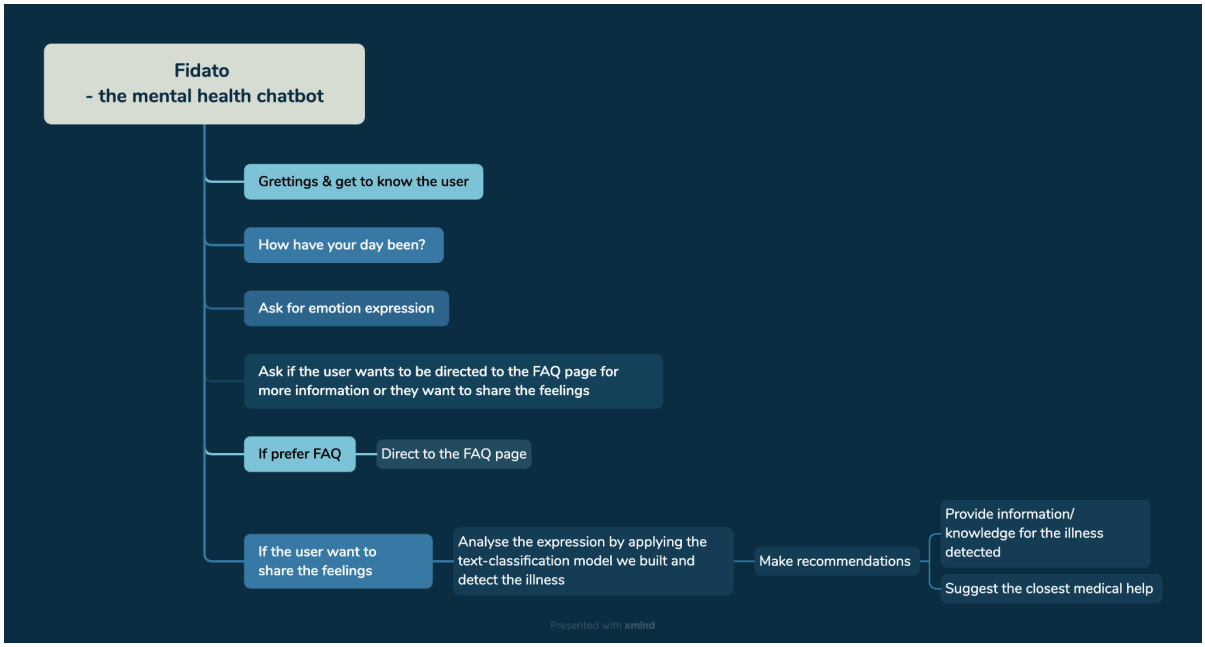
In the evolving landscape of mental health chatbots, characterized by the presence of over 41 diverse solutions (Denecke et al. (2021)), Fidato is positioned to carve out a unique niche. Inspired by established chatbots like Wysa and Woebot, which are recognized for integrating cognitive-behavioural techniques and fostering empathetic interactions, respectively, Fidato aspires to blend advanced machine learning with rule-based systems. This hybrid approach allows for continuous adaptation and personalized support based on user feedback and interactions, setting the stage for progressive sophistication.

Moreover, Fidato's web-based platform will ensure wider accessibility, distinguishing it from standalone software solutions and aiming to reach a broader audience seeking mental health support. While rapid-development chatbot solutions like personalized ChatGPT variants offer quick access, Fidato focuses on providing a comprehensive and

depth-oriented approach. The intention is to evolve over time, incorporating user feedback and the latest research to enhance its effectiveness and reach.

Our review of current literature, including studies like "A Mental Health Chatbot for Regulating Emotions (SERMO) - Concept and Usability Test" and others, reinforces the importance of chatbots like Fidato in the mental health domain. These digital solutions extend vital support channels, particularly to those reluctant to engage with traditional therapy settings. As Fidato continues to develop, we are committed to enhancing its sophistication and utility, drawing on best practices and emerging research to ensure it remains at the forefront of mental health support innovation. By maintaining a focus on user-centric development and continuous improvement, Fidato aims to be one of the sophisticated figures in the next generation of mental health support systems. Our vision for Fidato extends beyond initial interactions. We plan to continuously refine the platform, adapting to feedback and new mental health insights, ensuring Fidato remains a relevant, empathetic, and guiding force in online mental health support.

OUR CURRENT INITIAL VISION FOR THE BOT:



To interact with users and intervene in the potential illness in the early phase, we built a chatbot instead of just making a website entirely of information because a chatbot is accessible. With a chatbot, users can save time searching for information and match it with their symptoms themselves, given that they have already been through a lot. Our chatbot is designed to make users feel comfortable and detect some emotions/ symptoms underlying users' expressions that they might be unable to observe themselves.

Compared with other bots, our bots differ in the methods we used to build them. There are 41 different chatbots in the mental health market (Denecke et al., 2021, #). Most of them combine rule-based systems and AI-based techniques but work on stand-alone software, such as Wysa and WoeBot. Our bot used similar methods but worked only on the website.

In speaking of the method, we take Wysa as an example. It was initially a decision-tree-based model but has now evolved into one that can incorporate machine learning and rule-based systems. Its machine learning component allows Wysa to learn from user interactions, improving its responses over time and providing personalised support based on analysing user inputs and behaviours. Its rule-based component allows Wysa to apply evidence-based psychological techniques and interventions. However, we decided to make our chatbot accessible on the web because it is easier to access.

Data descriptions

Our dataset has been taken from Reddit posts from 28 subreddits, including 15 mental health support groups, covering 2018-2020. It contains text labelled the five mental disorders we've selected: ADHD, anxiety, depression, PTSD, and alcoholism. We have merged social and health anxiety into one anxiety category for easier handling and processing. The comments are from real-life posts, and they are indeed from the people who have the symptoms.

Methods

Text Classification Model

Initially, we developed a sophisticated text classification model using NLP to accurately categorize text into one of the five mental health disorders. This detailed account elaborates on the methodologies adopted, emphasizing the strategic decisions made during the model's development phase. (Appendix (1))

Project Setup and Dependency Management:

We began by preparing our Python environment and installing key libraries integral to machine learning and NLP tasks. These included TensorFlow for building and training neural network models, TensorFlow Hub for accessing pre-trained models, TensorFlow Text for processing textual data, and Pandas for data manipulation and analysis. Additionally, we used the Transformers library to leverage state-of-the-art NLP techniques, PyTorch as an alternative framework for deep learning tasks, the Datasets

library for handling NLP datasets effectively, and Scikit-learn for pre-processing and evaluation.

Data Handling and Preprocessing:

The dataset, comprising user-generated content from various subreddit forums, was categorized under five mental health conditions: ADHD, anxiety, depression, PTSD, and alcoholism. We employed Pandas for the initial data loading and preprocessing stages, addressing the dataset's inherent imbalances through downsampling strategies. We sampled down each category except alcoholism to the second lowest value, i.e., 6542 data points. This approach ensured balanced representation across all categories, thereby mitigating model bias toward more frequently occurring labels. Data cleaning processes were also implemented, removing irrelevant features and normalizing text data to prepare it for effective model training and analysis.

Rationale Behind Using BERT for Tokenization and Vectorization:

We chose the BERT model for its superior understanding of language context, which is critical for analyzing mental health-related texts. BERT's bidirectional approach captures linguistic nuances, which are essential for identifying subtle cues in mental health discourse. Its comprehensive pre-training on diverse data enhances its grasp of varied language patterns, which is vital for our project's goals. By fine-tuning BERT with our specific mental health dataset, we refine its capability to accurately classify complex mental health conditions. This methodology not only improves classification accuracy but also tailors the model to the unique challenges of mental health text analysis, supporting our aim to develop a refined, sensitive NLP tool for mental health support, thereby enhancing the effectiveness of our text classification system in the nuanced domain of mental health.

Model Architecture and Training Process:

After pre-processing, we employed a BERT-based architecture for our classification model. We chose BERT due to its proficiency in capturing complex language patterns and its proven track record in various NLP benchmarks. Our model architecture was designed to adapt the pre-trained BERT layers to our specific classification task, topped with a dense layer corresponding to our five target classes. We compiled our model using the Adam optimizer and categorical cross-entropy to cater to our multi-class classification problem. The training process was carefully monitored to balance efficiency and accuracy, ensuring the model learned effectively without overfitting.

Evaluation and Analytical Approach:

Upon training, we assessed our model's performance with a focus on generalization abilities, utilizing a separate test set for this purpose. We measured standard metrics such as accuracy and loss and delved deeper with a classification report providing precision, recall, and F1-scores for each class. This in-depth evaluation helped us understand the model's strengths and areas for improvement across different mental health conditions.

Throughout the model-building part of our project, our decisions, from data preprocessing to model selection and evaluation, were guided by the aim to develop an NLP model that not only performs well on standard metrics but also understands the complexities and variations in language used in the context of mental health. The choice of BERT, data balancing techniques, and evaluation metrics were all aligned with our goal to create a robust, sensitive, and accurate classification tool that can contribute meaningfully to mental health discourse analysis.

Integration of NLP-Based Text Classification into our Botpress Conversational Bot (Fidato)

Following the successful development of our text classification model tailored for mental health disorder identification, we opted for Botpress to construct an initial conversational bot prototype. Our selection was predicated on Botpress's reputation for efficiency and user-friendly platform, which significantly accelerates bot development.

Bot Interaction Design:

We named our bot “Fidato,” which means "a trustworthy friend" in Italian. This strategic choice aligns well with the bot's core requirements of comforting the user and providing a friendly, supportive interaction. This decision not only personalizes the bot but also sets the tone for the type of interaction users can expect: one based on trust, empathy, and confidentiality.

Fidato initiates dialogue with users through designed conversational pairs, aiming to cultivate a comfortable interaction space. By inquiring about users' day-to-day experiences or emotional states, Fidato strives to create an environment conducive to open communication. This initial engagement is vital for establishing a rapport, thereby encouraging users to express their thoughts and feelings more freely.

Informational and Analytical Transition:

After establishing initial rapport, Fidato offers users two primary pathways:

- **Redirecting to FAQ Page:** Users interested in learning about mental health are redirected to an FAQ page. This section serves educational purposes, providing information on common mental health conditions, their symptoms, and general guidance. Our goal is to implement the RAG (Retrieval Augmentation Generation) model for the FAQ page, although this component is currently under development.

- **Text Classification for Analysis and Emotional Support:** Users choosing to express their feelings activate Fidato's text classification capabilities. This feature allows the bot to analyze input text for patterns or keywords indicative of mental health conditions.

Response and Support:

Utilizing the insights gained from the text classification analysis, the Fidato extends a dual-faceted support system:

- **Educational Resource:** Fidato offers resources relevant to the identified mental health condition, aiming to enhance users' understanding of their situation and provide clarity on symptoms and coping mechanisms.

- **Professional Assistance:** Recognizing the limits of informational support, Fidato also suggests avenues for professional help, guiding users towards therapists or mental health services without imposing or pressuring them.

Integration Process of the Text Classification Model into the Bot Fidato

Our current focus is seamlessly incorporating the text classification model into our bot, Fidato. This effort aims to enable the bot to understand and analyze user inputs more profoundly, thus providing responses that are accurately tailored to the user's mental health context.

Technical Implementation:

The integration process involves several technical steps:

- **Model Embedding:** We are embedding the pre-trained text classification model into the Botpress environment. This requires configuring the bot's backend to communicate effectively with the model and ensure it can process user inputs through the model's framework.

- **Data Handling and Privacy:** Special attention is given to data handling and privacy concerns, ensuring all user interactions are managed securely and confidentially. User inputs are encrypted and anonymized before being processed by the model.

- **Response Mapping:** The outputs from the text classification model are mapped to corresponding responses and resources within the bot's framework. This mapping is crucial for providing relevant and helpful feedback based on the model's analysis of user sentiments and potential mental health issues.

Testing and Validation:

Comprehensive testing is underway to validate the integration:

- **Functional Testing:** We conduct thorough tests to ensure that the model's integration does not disrupt the bot's existing functionalities and that the bot correctly interprets and responds to user inputs.

- **Accuracy Validation:** The accuracy of the text classification model within the bot context is being assessed. This involves comparing the model's predictions based on user inputs against known outcomes to ensure reliability and precision.

- **User Experience Testing:** We will also evaluate the integration from a user experience perspective, ensuring that the bot's responses are appropriate, empathetic, and supportive, enhancing the overall user interaction.

Results and Evaluation

Model Performance:

The text classification model was thoroughly evaluated using the test dataset, achieving an impressive overall accuracy of approximately 86.95%. The accompanying detailed classification report highlighted the model's precision, recall, and F1-scores across various mental health categories. Notably, the model showed outstanding performance in correctly classifying texts associated with ADHD, with a precision and recall both at 0.90, and anxiety, showcasing a precision and recall around 0.88, reflecting its acute understanding of these specific conditions.

Despite its strengths, the model showed room for improvement, particularly in distinguishing between conditions with overlapping symptoms such as anxiety and depression, as well as alcoholism and depression. This was evident from the confusion matrix where instances of alcoholism were sometimes misclassified as anxiety or depression, indicating potential areas for refinement. The model’s balanced performance across categories—highlighted by precision and recall metrics—underscores its potential as an effective tool for textual mental health assessments. This balanced approach ensures that the model pays fair attention to all mental health conditions, making it a valuable asset in real-world applications for identifying and understanding mental health issues from textual data.

Bot Integration and Its Prospective Impact:

Integrating this refined text classification model into our conversational bot will significantly enhance its functionality. This advancement will transform the bot into an innovative digital mental health support platform, designed to engage users in meaningful dialogue, the bot provides a secure, informative, and supportive space, allowing individuals to discuss and understand their mental health issues without fear of judgment.

The bot's framework, empowered by the text classification model, will enable the automated analysis of user inputs to identify potential mental health concerns. Following this analysis, the bot will be able to offer personalized assistance, including educational resources and guidance on seeking professional help. This approach will not only aid users in understanding their own mental health but also direct them towards appropriate support services when necessary.

Error Analysis

Error Analysis for the Text Classification Model:

Based on the classification report and the confusion matrix, it is revealed that there are varied performances across different mental health conditions:

- **ADHD:** Accurate but confused with anxiety (172 cases) and depression (130 cases), suggesting overlapping symptoms or language similarities need addressing.
- **Alcoholism:** High precision but misses cases, with confusion mainly with anxiety (31 cases) and depression (29 cases), indicating overlapping language descriptors.
- **Anxiety:** Solid performance; however, misclassifies into depression (522 cases) and ADHD (135 cases), showing symptom or expression overlap.
- **Depression:** Good balance but confuses with anxiety (519 cases) and PTSD (73 cases), highlighting similar linguistic expressions.
- **PTSD:** Lowest accuracy, often misclassified as anxiety (66 cases) and depression (57 cases), suggesting a need for better differentiation in trauma-related language.

Error Analysis Recommendations:

1. Data Augmentation: Expand the dataset for underrepresented conditions, particularly PTSD, enhancing the model's ability to learn varied examples.

2. Contextual Analysis: Enhance the model’s capability in analyzing context around keywords to better differentiate conditions with overlapping symptoms, such as anxiety and depression.

3. Advanced Preprocessing: Implement refined text preprocessing techniques to more accurately distinguish between distinct linguistic features characteristic of different mental health conditions.

4. Continuous Training: Consistently update the model with fresh data and user feedback, allowing for ongoing refinement of its diagnostic accuracy and reduction of existing biases.

5. Cross-validation: Apply cross-validation methods to ensure the model's effectiveness and adaptability across a range of text samples and scenarios.

Error Analysis for the Bot Integration:

Resource Relevance: The bot may offer resources or suggestions that are not perfectly aligned with the user's specific condition or needs, particularly if the initial text analysis was flawed.

User Feedback Mechanisms: The current bot framework might lack robust mechanisms for users to provide immediate feedback on the relevance or helpfulness of the information provided, limiting opportunities for iterative improvement based on real-world usage.

Discussion

Possible ethical issues:

Although the mental health chatbot can bring huge opportunities for providing valuable help to people who are suffering from mental difficulties, it could also bring potential ethical issues to the table at the same time.

With the chatbot we made, the people we are targeting are the young people who generally have more exposure to digital solutions for their mental disorders. For instance, 81% of users of 2 or 3 chatbots in Facebook Messenger are 18- to 24 years old (Kretzschmar et al., 2019, #). According to a survey of young people's opinions on the use of chatbots in mental health support, they generally express their concerns about who has access to the confidential personal information exposed in the conversation, how the efficacy of the chatbot is based on the evidence given, and how efficiently the chatbot can protect users from emergency occasions.
To answer the concerns raised, we had a user-centric approach with a solid commitment to ethical standards.

- **Transparent data usage policy.** We ensure users are aware that their inputs are analysed to provide personalised support and that they retain control over their data, which means that they can withdraw anytime.

- **Integration of Evidence-Based Practices.** Ensure the chatbot's responses and interventions are grounded in evidence-based psychological practices, such as CBT, Dialectical Behavior Therapy (DBT), and mindfulness techniques. Regularly update the chatbot's knowledge base with the latest research findings to maintain its effectiveness.

- **Escalation Procedures.** Establish clear escalation protocols for emergencies. This could include direct referrals to mental health professionals or crisis intervention teams, ensuring users receive the support they need promptly.

Failures and Comebacks:
Initially, our chatbot project encountered difficulties with implementing the RAG model due to resource and time limitations. Attempts to elevate the bot's sophistication through IBM Watson, Azure Framework, and RASA also faced constraints. Presently, our efforts are concentrated on surpassing these barriers while also integrating the text classification model into the bot. We are committed to developing the chatbot further, aiming for a more professional and sophisticated tool. Despite early challenges, we are dedicated to refining the bot's functionality and enhancing user interactions in future developments.

Future Steps and Enhancements:

After completing and validating our text classification model's integration with the conversational bot, we aim to deploy it for broader user engagement. Our approach includes continuous monitoring and updates based on user feedback to meet diverse mental health support needs effectively.

We plan to enhance the bot's accuracy and user experience by integrating the RAG model into the FAQ section, improving response relevance and precision. Additionally, we aim to make our bot more inclusive by incorporating image and voice recognition features, enhancing accessibility for users with disabilities. These improvements allow users to interact with the bot through various formats beyond text, making it a more versatile and user-friendly mental health support tool. Our goal is continually refining these features, ensuring the bot serves as a comprehensive and accessible resource for those seeking mental health assistance.

References

Denecke, K., Vaaheesan, S., & Arulnathan, A. (2021). A Mental Health Chatbot for Regulating Emotions (SERMO) - Concept and Usability Test. IEEE, 9(3), 1170 - 1182. https://ieeexplore-ieee-org.libproxy.ucl.ac.uk/document/9000924/authors

Global Burden of Disease (GBD). (n.d.). Global Burden of Disease study - Mental health. Institute for Health Metrics and Evaluation. Retrieved March 11, 2024, from https://www.healthdata.org/research-analysis/health-risks-issues/mental-health

Kretzschmar, K., Tyroll, H., Pavarini, G., Manzini, A., Singh, I., & NeurOx Young People's Advisory Group. (2019). Can Your Phone Be Your Therapist? Young People's Ethical Perspectives on the Use of Fully Automated Conversational Agents (Chatbots) in Mental Health Support. Biomed Inform Insights, 11, 1-9. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6402067/pdf/10.1177_1178222619829083.pdf

National Institutes of Health (NIH). (2023, September 28). Mental Health | NIH COVID-19 Research. National Institutes of Health COVID-19 Research. Retrieved March 11, 2024, from https://covid19.nih.gov/covid-19-topics/mental-health

NHS. (n.d.). The NHS website - NHS. Retrieved March 20, 2024, from https://www.nhs.uk/

World Health Organisation. (2022, March 2). COVID-19 pandemic triggers 25% increase in prevalence of anxiety and depression worldwide. World Health Organization (WHO). Retrieved March 11, 2024, from https://www.who.int/news/item/02-03-2022-covid-19-pandemic-triggers-25-increase-in-prevalence-of-anxiety-and-depression-worldwide

World Health Organisation. (2022, March 2). Mental Health and COVID-19: Early evidence of the pandemic's impact. World Health Organisation. Retrieved March 11, 2024, from https://www.who.int/publications/i/item/WHO-2019-nCoV-Sci_Brief-Mental_health-2022.1

Dataset:

Low, D. M., Rumker, L., Torous, J., Cecchi, G., Ghosh, S. S., & Talkar, T. (2020). Natural Language Processing Reveals Vulnerable Mental Health Support Groups and Heightened Health Anxiety on Reddit During COVID-19: Observational Study. Journal of medical Internet research, 22(10), e22635.

Libraries:

The model was developed using TensorFlow and TensorFlow Hub (Abadi et al., 2016).
Tokenization was performed using the BERT tokenizer provided by the Hugging Face's Transformers library (Wolf et al., 2020).
The dataset was split into training and test sets using the train_test_split function from Scikit-Learn (Pedregosa et al., 2011).

Appendix:

1) Mental Health Chatbot final.ipynb

```
[ ] # !pip uninstall -y tensorflow tensorflow-hub tensorflow-text keras tensorboard ml-dtypes
    !pip install --no-cache-dir tensorflow-hub==0.15
    !pip install tensorflow-text==2.15
    !pip install tensorflow==2.15
```

```
[ ] !pip install pandas transformers
    !pip install transformers
    !pip install torch
    !pip install datasets
    !pip install scikit-learn
    import tensorflow as tf
    import tensorflow_hub as hub
    import torch
    from transformers import BertTokenizer, BertForSequenceClassification
    from torch.utils.data import DataLoader, RandomSampler, SequentialSampler, TensorDataset
    from sklearn.model_selection import train_test_split
    from sklearn.metrics import accuracy_score
    import numpy as np
    import pandas as pd
    from transformers import Trainer, TrainingArguments
    from datasets import Dataset
    from sklearn.preprocessing import LabelEncoder
    import tensorflow_text as text
    from tensorflow.keras.utils import to_categorical
```

Data Handling And Preprocessing

```
# Replace 'your_excel_file.xlsx' with the actual name of your uploaded Excel file
file_name = 'MERGED MENTAL H DATASET (5).xlsx'

# Read the Excel file
df = pd.read_excel(file_name)

# Display the first 5 rows of the DataFrame
print(df.head())
```

```
  subreddit       date                                               post  \
0      adhd 2020-01-01  ADHD gets worse at night High school senior he...
1      adhd 2020-01-01  The First Step of a multi-step task Perhaps yo...
2      adhd 2020-01-01  I have been carrying around Play-Doh for years...
3      adhd 2020-01-01  How do you keep from getting bored in a relati...
4      adhd 2020-01-01  I need entertain Hello I used to play games li...

  Post/Pre
0     post
1     post
2     post
3     post
4     post
```

```
[ ] #check the distribution of labels
    df['subreddit'].value_counts()
```

```
anxiety       58568
depression    55887
adhd          30298
ptsd           6542
alcoholism     4515
Name: subreddit, dtype: int64
```

```
[ ] #DOWNSIZING DATA
    min_samples = 6542  # Define the number of samples to match the minority class

    # Replace 'df_ham_downsampled' with a list of downsampled DataFrames for each subreddit
    # We use the same 'min_samples' and 'random_state' for consistency
    subreddits = ['depression', 'anxiety', 'adhd', 'ptsd']

    # Dictionary comprehension to downsample each subreddit DataFrame
    dfs_downsampled = {subreddit: df[df.subreddit == subreddit].sample(min_samples, random_state=2022) for subreddit in subreddits}

    # If you want to access one of them, for example, the downsampled DataFrame for 'depression':
    df_depression_downsampled = dfs_downsampled['depression']
    df_anxiety_downsampled = dfs_downsampled['anxiety']
    df_adhd_downsampled = dfs_downsampled['adhd']
    df_ptsd_downsampled = dfs_downsampled['ptsd']
    df_alcoholism_downsampled = df[df.subreddit == 'alcoholism']  # No downsampling applied
```

```
[ ] df_balanced = pd.concat([df_depression_downsampled, df_anxiety_downsampled,df_adhd_downsampled, df_ptsd_downsampled, df_alcoholism_downsampled])
    df_balanced.shape
```

```
(30683, 4)
```

```
[ ] df_balanced['subreddit'].value_counts()
```

```
depression    6542
anxiety       6542
adhd          6542
ptsd          6542
alcoholism    4515
Name: subreddit, dtype: int64
```

BERT for Tokenization and Vectorization

```
[ ] from transformers import BertTokenizer
    import torch

    # Initialize the BERT tokenizer
    tokenizer = BertTokenizer.from_pretrained('bert-base-uncased')

    # Define the max length for BERT
    max_length = 128  # You can adjust this according to the needs and constraints of your model

    # Function to tokenize and encode the dataset
    def encode_texts(texts):
        input_ids = []
        attention_masks = []

        for text in texts:
            encoded_dict = tokenizer.encode_plus(
                text,                      # Text to encode
                add_special_tokens=True,   # Add '[CLS]' and '[SEP]'
                max_length=max_length,     # Pad or truncate
                pad_to_max_length=True,    # Pad all to `max_length` if necessary
                return_attention_mask=True,  # Construct attn. masks
                return_tensors='pt',       # Return pytorch tensors
            )

            # Add the encoded sentence to the list
            input_ids.append(encoded_dict['input_ids'])
            # And its attention mask (differentiating padding from non-padding)
            attention_masks.append(encoded_dict['attention_mask'])

        # Convert lists to tensors
        input_ids = torch.cat(input_ids, dim=0)
        attention_masks = torch.cat(attention_masks, dim=0)

        return input_ids, attention_masks
```

```
    # Preprocess the 'post' column
    posts = df['post'].values  # Extract text data to preprocess
    input_ids, attention_masks = encode_texts(posts)
```

```
tokenizer_config.json: 100% 48.0/48.0 [00:00<00:00, 2.09kB/s]
vocab.txt: 100% 232k/232k [00:00<00:00, 3.83MB/s]
tokenizer.json: 100% 466k/466k [00:00<00:00, 11.2MB/s]
config.json: 100% 570/570 [00:00<00:00, 13.4kB/s]
Truncation was not explicitly activated but `max_length` is provided a specific value, please use `truncation=True` to explicitly truncate examples to max l
/usr/local/lib/python3.10/dist-packages/transformers/tokenization_utils_base.py:2645: FutureWarning: The `pad_to_max_length` argument is deprecated and will
  warnings.warn(
```

Model Architecture and Training Process

```
[20] # Parameters
     BATCH_SIZE = 16
     EPOCHS = 3

     # Load and preprocess dataset
     # df = pd.read_csv('your_large_dataset.csv')  # Load your dataset
     posts = df['post'].values
     subreddits = df['subreddit'].values

     # Encode labels
     label_encoder = LabelEncoder()
     subreddits_encoded = label_encoder.fit_transform(subreddits)
     subreddits_encoded = to_categorical(subreddits_encoded)

     # Split dataset
     train_posts, test_posts, train_labels, test_labels = train_test_split(posts, subreddits_encoded, test_size=0.1)

     # Convert to tf.data.Dataset for efficient loading
     AUTOTUNE = tf.data.experimental.AUTOTUNE
     train_data = tf.data.Dataset.from_tensor_slices((train_posts, train_labels))
     train_data = train_data.shuffle(buffer_size=len(train_posts)).batch(BATCH_SIZE).prefetch(buffer_size=AUTOTUNE)
     test_data = tf.data.Dataset.from_tensor_slices((test_posts, test_labels)).batch(BATCH_SIZE).prefetch(buffer_size=AUTOTUNE)

     # BERT layers
     bert_preprocess_model_url = 'https://tfhub.dev/tensorflow/bert_en_uncased_preprocess/3'
     bert_model_url = 'https://tfhub.dev/tensorflow/small_bert/bert_en_uncased_L-2_H-128_A-2/2'

     # Build the model
     text_input = tf.keras.layers.Input(shape=(), dtype=tf.string, name='text')
     preprocessor = hub.KerasLayer(bert_preprocess_model_url, name='preprocessing')
     encoder_inputs = preprocessor(text_input)
     encoder = hub.KerasLayer(bert_model_url, trainable=True, name='BERT_encoder')
     outputs = encoder(encoder_inputs)
     net = outputs['pooled_output']
     net = tf.keras.layers.Dropout(0.1)(net)
```

```
[20] net = tf.keras.layers.Dense(subreddits_encoded.shape[1], activation='softmax', name='classifier')(net)
     model = tf.keras.Model(text_input, net)

     # Compile the model
     model.compile(optimizer=tf.keras.optimizers.Adam(learning_rate=5e-5), loss='categorical_crossentropy', metrics=['accuracy'])

     # Train the model
     model.fit(train_data, epochs=EPOCHS, validation_data=test_data)

     # Evaluate the model
     model.evaluate(test_data)
```

```
Epoch 1/3
8765/8765 [==============================] - 2866s 326ms/step - loss: 0.5125 - accuracy: 0.8139 - val_loss: 0.3867 - val_accuracy: 0.8652
Epoch 2/3
8765/8765 [==============================] - 2807s 320ms/step - loss: 0.3831 - accuracy: 0.8642 - val_loss: 0.3677 - val_accuracy: 0.8699
Epoch 3/3
8765/8765 [==============================] - 2815s 321ms/step - loss: 0.3407 - accuracy: 0.8797 - val_loss: 0.3670 - val_accuracy: 0.8738
974/974 [==============================] - 87s 89ms/step - loss: 0.3670 - accuracy: 0.8738
[0.3670276701450348, 0.8737565279006958]
```

```
[21] loss, accuracy = model.evaluate(test_data)
     print(f'Test Loss: {loss}')
     print(f'Test Accuracy: {accuracy}')
```

```
974/974 [==============================] - 135s 138ms/step - loss: 0.3670 - accuracy: 0.8738
Test Loss: 0.3670276701450348
Test Accuracy: 0.8737565279006958
```

```
predictions = model.predict(test_data)
predicted_labels = np.argmax(predictions, axis=1)
```

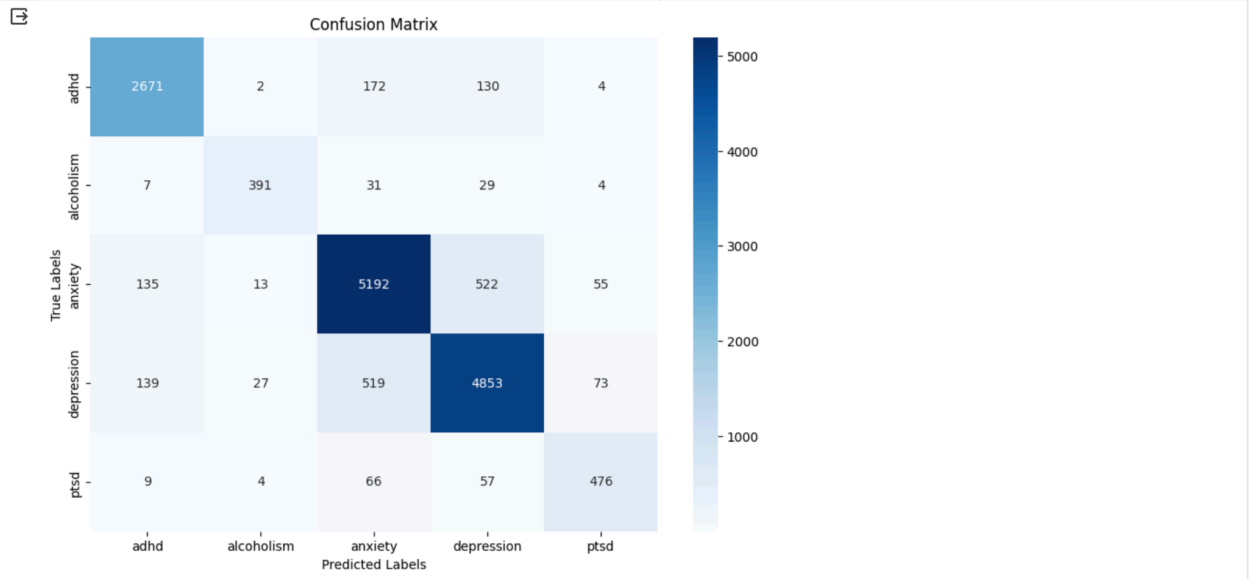
```
974/974 [==============================] - 86s 88ms/step
```

Error Analysis

```
[14] from sklearn.metrics import classification_report

     # Convert test labels from one-hot encoding to integers
     test_labels_integers = np.argmax(test_labels, axis=1)

     # Convert predicted labels back to original subreddit labels
     predicted_subreddits = label_encoder.inverse_transform(predicted_labels)
     true_subreddits = label_encoder.inverse_transform(test_labels_integers)

     # Generate and print the classification report
     print(classification_report(true_subreddits, predicted_subreddits, target_names=label_encoder.classes_))
```

```
                precision    recall  f1-score   support

          adhd       0.90      0.90      0.90      2979
    alcoholism       0.89      0.85      0.87       462
       anxiety       0.87      0.88      0.87      5917
    depression       0.87      0.86      0.87      5611
          ptsd       0.78      0.78      0.78       612

      accuracy                           0.87     15581
     macro avg       0.86      0.85      0.86     15581
  weighted avg       0.87      0.87      0.87     15581
```

```
[15] from sklearn.metrics import confusion_matrix
     import seaborn as sns
     import matplotlib.pyplot as plt

     # Assuming true_subreddits and predicted_subreddits are your actual and predicted labels respectively
     # You may need to run your prediction and label encoding steps again if they're not stored

     # Compute the confusion matrix
     cm = confusion_matrix(true_subreddits, predicted_subreddits, labels=label_encoder.classes_)

     # Plot the confusion matrix using Seaborn heatmap
     plt.figure(figsize=(10, 7))
     sns.heatmap(cm, annot=True, fmt='d', cmap='Blues', xticklabels=label_encoder.classes_, yticklabels=label_encoder.classes_)
     plt.title('Confusion Matrix')
     plt.xlabel('Predicted Labels')
     plt.ylabel('True Labels')
     plt.show()
```



Testing

[16]
```
#PTSD COMMENT
new_posts = ["anyone have any advice on projecting? i told my counselor about how one of my biggest issues is projecting trauma onto other people and she o
processed_posts = tf.data.Dataset.from_tensor_slices((new_posts, [0]))  # Dummy labels
processed_posts = processed_posts.batch(BATCH_SIZE).prefetch(buffer_size=AUTOTUNE)

# Predicting new texts
new_predictions = model.predict(processed_posts)
new_predicted_labels = np.argmax(new_predictions, axis=1)
new_predicted_labels = label_encoder.inverse_transform(new_predicted_labels)  # Convert back to original labels
print(new_predicted_labels)
```

```
1/1 [==============================] - 0s 416ms/step
['ptsd']
```

[19]
```
#ADHD COMMENT
new_posts = ["English assignment So i have to do this English assignment about how certain groups are misrepresented in the media, i hyper focused it in on
processed_posts = tf.data.Dataset.from_tensor_slices((new_posts, [0]))  # Dummy labels
processed_posts = processed_posts.batch(BATCH_SIZE).prefetch(buffer_size=AUTOTUNE)

# Predicting new texts
new_predictions = model.predict(processed_posts)
new_predicted_labels = np.argmax(new_predictions, axis=1)
new_predicted_labels = label_encoder.inverse_transform(new_predicted_labels)  # Convert back to original labels
print(new_predicted_labels)
```

```
1/1 [==============================] - 0s 25ms/step
['adhd']
```

[18]
```
#ALCOHOLISM COMMENT
new_posts = ["Looking for answers So this is new to me. I,Âôm going to be brief. And if this isn,Âôt the right forum for it please redirect me where I shou
processed_posts = tf.data.Dataset.from_tensor_slices((new_posts, [0]))  # Dummy labels
processed_posts = processed_posts.batch(BATCH_SIZE).prefetch(buffer_size=AUTOTUNE)

# Predicting new texts
new_predictions = model.predict(processed_posts)
new_predicted_labels = np.argmax(new_predictions, axis=1)
new_predicted_labels = label_encoder.inverse_transform(new_predicted_labels)  # Convert back to original labels
print(new_predicted_labels)
```

```
1/1 [==============================] - 0s 21ms/step
['alcoholism']
```