A Clinical Text Corpus for Emotion and Mental Health Classification Using ClinicalBERT

Alysson Hana A. Perez
Computer Studies and Engineering
Jose Rizal University
Mandaluyong City, Philippines
alyssonhana.perez@my.jru.edu

***Abstract*—The increasing cases of psychiatric problems highlight the need to have reliable digital tools that can identify cases of emotional distress in textual interaction. In this study, an automated emotion and sentiment classification system is presented, using the ClinicalBERT transformer architecture, which is trained on large clinical corpora. The system aims to detect the affective conditions, such as anxiety, depression, and stress, derived out of the self-report, clinically mimetic textual statements. Using domain-specific contextual embeddings, ClinicalBERT achieves the highest level of emotion-detection accuracy as compared to generic models like BERT-base. The dataset is based on the corpus of Sentiment Analysis on Mental Health on Kaggle [13] and went through preprocessing steps, including tokenization, normalization, and encoding, before fine-tuning. The model performance is considered on the basis of accuracy, F1-score and confusion-matrix analysis. This is in harmony with United Nations Sustainable Development Goal 3 (Good Health and Well-Being) [14], because it will make it easier to identify mental health issues early and promote the adoption of AI-assisted clinical decision support. A model that is interpretable and data-driven and helps clinicians diagnose the signs of psychological distress early on, reduce diagnostic bias, and improve patient outcomes through timely interventions is expected [5], [6], [15], [16].**

Keywords—ClinicalBERT, emotion classification, mental health, natural language processing (NLP), transformer models, sentiment analysis, Sustainable Development Goal (SDG) 3.

I. INTRODUCTION

Textual data emotion and mental-health identification analysis has been an area of considerable research focus in natural language processing (NLP) and clinical informatics in the past several years. Perceptiveness and contextual contingency of human expression makes emotion recognition in text other than positive/negative sentiment particularly challenging to specific emotion recognition (e.g., anxiety, fear, sadness) [1], [2]. It is this prospective to early intervention, monitoring, and individualized care that is displayed to be promising in the clinical or mental-health contexts is the automatic categorization of the statements made by the patient into such categories as anxiety, confusion, restlessness, or other affective conditions [3].

The study is consistent with the United Nations Sustainable Development Goal (SDG) 3: Good Health and Well-Being, and more specifically with Target 3.4, that states the focus on mental health and well-being, the promotion of which should come at the cost of a reduction in premature deaths caused by non-communicable diseases by prevention and treatment [14]. The suggested ClinicalBERT-based model will fit into this international project by implementing artificial intelligence to support mental health assessment and early detection. The study contributes to the ability to intervene in time and makes clinical decisions with data, as it is based on automated emotion and sentiment detection of patient-like text. AI implementation into mental health analysis is in accordance with the World Health Organization Comprehensive Mental Health Action Plan 20132030, which recommends the use of digital technologies to empower mental health systems and access to services [15]. Moreover, the work can be used to address the call to find scalable, equitable, and sustainable mental health solutions in the whole world since it limits diagnostic bias and enhances detection accuracy [16].

The data set in question consists of short self-reporting data in the form of statements, and it is labeled as one of the statuses (e.g. Anxiety), thus, offering supervised classification data of textual emotional or mental-state identification of text. Since the text has been patient-like or self-reported (as it is not just a structured clinical note), the data falls between the vernacular of mental-health expression, and the vernacular of clinical documentation. One can also learn and test models, which strive to extrapolate the underlying emotion/psychological condition given solely the linguistic clues and such text-label pairs [4].

The BERT architecture based on transformer ClinicalBERT version in the form of the clinical/biomedical text occupies an especially appropriate niche in this assignment. The ClinicalBERT has been trained on large clinical note corpora such as the MIMIC-III dataset and has been useful in clinical language pattern modellings [5], [6]. Indicatively, Huang et al. [7] set ClinicalBERT to be superior among the discharge summary based hospital readmission prediction models. Such cases are a benefit of this domain-specific pre-training whose textual inputs are written in the language of clinical or health-related nature,

and with or without self-understandable or self-report wordings [8].

The use of ClinicalBERT in regards to the given data will allow using pre-trained models of clinical language and further optimize the model retrieved to meet the particular problem of emotional or mental-state classification. The acumen of the clinical and behavioural language of the model can lead to the optimal discrimination over a general sentiment-analysis model because the phrases in the dataset refer to mental-health issues (e.g., trouble sleeping, restless heart, confused mind), which the sentiment-analysis model can carry out [9]. Besides, the fine-tuning paradigm enables the model to self-report the less strict nature of the data, and retain the underlying clinical terminology during pre-training [10].

Regarding the methodological aspect, there are a number of significant things that may be considered when training the dataset to the models. Some preprocessing (tokenization, cleaning, missing values) and balancing of classes might be necessary to reduce the under-represented emotional states. This is because the entries of the text are finite and, therefore, model configuration (sequence length, truncation strategy, batch size) must be configured to the context to maintain the text. Multi-class evaluation measures, like accuracy, F1-score and AUC are used in the evaluation. Such are in correspondence with the available literature on text emotion detection and classification [1], [4], [11].

The dataset may also be applicable to practice in the domain of mental-health surveillance, sentiment surveillance on the population level in a healthcare institution, and automatic text triage of patient-reports. In the digital level of mental-health services, e.g. a trained model based on this data might automatically identify the usages that can denote the manifestation of anxiety or distress to be followed up by a human in a timely manner [12]. The existence of such systems can be useful to clinicians and researchers when it comes to their perception of the mental-health trends of populations and enhancing patient outcomes.

The overall excitement of this information in self-report brief statements involving emotional or psychological condition is better befitting the domain-sensitive model like ClinicalBERT. The work of text that is clinically relevant, labelled emotions, and domain specific modeling provides a solid basis on which the research of the automated sentiment and emotion analysis in mental-health settings can be based. This information may be an invaluable contribution to the production of explainable, clinically intelligent NLP systems where a systematic preprocessing, classification, and reorganization of the models is carried out.

Fine-tuning large pretrained language models offers a principled way to transfer the syntactic and semantic priors learned from massive unlabeled corpora into our specific classification task, yielding large gains over training from scratch or hand-engineered TF–IDF features. In practice, task-specific fine-tuning of Transformer encoders such as BERT consistently improves macro-F1 on text classification benchmarks by aligning the model's contextual representations with the target label space [17], [18]. For code-switched Filipino–English inputs, multilingual encoders like **XLM-RoBERTa** provide robust subword representations and cross-lingual generalization, which have been shown to help downstream sentiment tasks in low-resource or mixed-language settings [19]. Moreover, **domain adaptation**—either via domain-adaptive pretraining or careful task-level fine-tuning—improves robustness to in-domain vocabulary and style shifts [20]. Complementing this, Filipino-prior encoders such as **RoBERTa-Tagalog** use subword vocabularies tailored to Tagalog morphology and orthography, which can reduce out-of-vocabulary effects and stabilize optimization for local-language sentiment datasets [21]. These findings collectively motivate our fine-tuning setup and the inclusion of both multilingual and Filipino-prior backbones in our experiments.

II. Problem Statement

Misdiagnosis or underdiagnosis of mental-health conditions, including anxiety, depression, or other associated emotional disorders, is one of the most important problems in clinical practice. This is a problem in that a clinical assessment is highly dependent on the human subjective interpretation of verbal or written reports of a patient. Patients can talk about their symptoms vaguely, emotionally or non-medically e.g. when someone says I am restless or having trouble sleeping or I am so confused in my head as depicted in the dataset. The absence of systematic tools that analyze these subtle emotional statements may cause healthcare professionals to miss vital clues about the psychological issues, which results in a false or slow diagnosis. The issue is particularly important in the mental health sphere, where language patterns can be used as the first and the most readily available indicator of distress in the background [1], [3], [11].

To cope with it, the project will focus on the creation of the automated sentiment and emotion analysis system with the help of ClinicalBERT, which will be trained on the labeled statements of the dataset. With this model, clinicians might use it as a decision-support tool to determine if the patient-like text includes a pattern of anxiety or emotional disturbance so that there may be an additional, objective analysis [5], [6], [9]. This would not substitute doctors, but this system would act as an aid, as it would point at possible emotional red flags in the stories of patients. Finally, the proposed project aims at alleviating the presence of diagnostic bias, as well as enhancing the capability of the timely detection of mental-health issues by converting the qualitative patient utterances into quantifiable, factual knowledge [7], [8], [10], [12].

III. Research Objectives

The main goal of the project would be to develop a ClinicalBERT based emotion and sentiment analysis model, which would be trained on the labeled clinical-like statements of the dataset comprising of text statements, which would allow healthcare professionals to identify a mental-health condition, such as anxiety, in the initial and correct manner. The project will implement the principles of natural language processing (NLP) and deep learning and convert unstructured patient expressions into measurable indicators of emotions that can be used as a second-level diagnostic instrument among clinicians.

In order to accomplish this general objective, the project will strive to obtain the following specific objectives:

To clean, **tokenize** and encode the text column (statement) and a label column (status) of the data, so that the data can be trained on a model and that there is no noise or inconsistencies in the data.

To optimize the **ClinicalBERT** model on the ready dataset so that it could be capable of classifying the emotional or psychological state based on the patient-like statements. To assess the predictive reliability of the model, the accuracy, F1-score, and a confusion matrix analysis will be used as quantitative measures to determine the performance of the model.

To assess the **interpretability** and clinical relevance of the model by assessing the words or phrases that most significantly determine the classification outcome and give transparency and explainable AI to a healthcare application.

In order to illustrate the potential of the model as a clinical decision-support tool by suggesting how it can help healthcare professionals detect the signs of early emotional distress, thus avoiding diagnostic bias and ensuring a better outcome of treatment.

IV. Scope of Work

A. Overview

The current project is concerned with the design of an automated system to identify sentiments and emotions based on ClinicalBERT and is trained on a collection of text statements, with labels of emotional states, including Anxiety. The dataset (Combined Data.csv) is a collection of patient like expressions related to mental and emotional conditions, which serve as the basis to train and test the model. The main goal of the work is to decrease the possible misdiagnosis or underdiagnosis of mental-health assessment by providing a machine learning-based decision-support system to clinicians. The system is going to investigate the unstructured text input and categorize them into emotional entities in order to aid in early detection and clinical interpretation. The project is limited to processing and analysis of text-based data exclusively without the inclusion of such other modalities as speech, facial expressions, or physiological indicators. It focuses on the development, evaluation, and comparison of models in the context of sentiment and emotion detection in tasks based on transformer-based models, in particular, ClinicalBERT.

B. Key Activities

The project itself will start with the dataset preparation and exploration to make sure that the text and labels are properly formatted and cleaned and structured to be used in training. Preprocessing of the data will involve, tokenization of the data with the tokenizer of ClinicalBERT, text normalization and converting the emotional labels to numerical values. Once the data has been prepared, fine-tuning and development of the model will be done based on the ClinicalBERT architecture with hyperparameters like learning rate, batch size, and training epochs being optimized on the dataset. To compare the performance of ClinicalBERT and a general-purpose model, including BERT-base, the effectiveness of domain-specific pretraining in dealing with clinical and emotional language will be evaluated. Evaluation and validation will then be conducted by determining the accuracy and precision as well as recall and F1-score figures and calculating the confusion matrices to establish patterns of misclassification. Interpretability analysis will also be conducted in order to indicate important linguistic clues that determine the classification outcomes. Lastly, the project will come to an end with documentation and demonstration, which will involve demonstration of the implementation of how the patient-like text inputs can be categorized into emotional groups with comprehensive reporting of the results, limitation of the method, and possible clinical implication.

V. Methodology

A. Data Collection & Sources

The data utilized in this paper was gathered in the form of the collection of text entries of a patient-like character on Kaggle [13], where people describe a certain emotional and psychological state. In this project, the data was combined and made into a CSV file titled Combined Data.csv, which had the number of columns totaling [3] and [insert total numbers of records, e.g. 5,000 or your records] records in it. The columns are an index field (Unnamed: 0), a text field named statement, with an expression that is written by the user, say, trouble sleeping, or confused mind and a categorical label field status, as in Anxiety. To eliminate inconsistencies and to provide a standard structure that was to be fine-tuned to ClinicalBERT, the data was collected and preprocessed. As the main input of the sentiment and emotion analysis model, this dataset allows supervised learning and distinguishing mental-health-related textual data into emotionally relevant clinically significant categories.

B. Data Preprocessing

Prior to analysis, the dataset will undergo extensive preprocessing to ensure its quality and suitability for machine learning.

1. Data Cleaning

The first step in preprocessing the dataset involves conducting a thorough inspection and cleaning of the data to ensure accuracy and consistency. The dataset (Combined Data.csv) is examined for missing, null, or duplicate entries within both the statement and status columns. Records containing empty text fields or undefined emotion labels are removed to maintain data quality. Duplicate statements are also eliminated to prevent bias during model training. Furthermore, all class labels under the status column are standardized to a consistent format by converting them to uniform case and trimming unnecessary spaces. This step ensures that emotional categories such as Anxiety, Depression, and Stress are correctly represented and ready for supervised learning.

2. Text Normalization

The text normalization is then carried out in order to uniformize the structure and format of input statements. The text in the statement column is turned to lower case to maintain uniformity in the entries although thought is taken to ensure that no meaning of the entry is lost in case of the use of capitalization to bring out the emphasis or feelings. Unnecessary punctuations, special characters, and extra white space are eliminated, yet the punctuation marks that can have emotional implications (e.g. exclamation points or question marks) are retained. There is expansion of certain types of contractions like can't or I am to cannot and I am respectively to facilitate uniformity of tokens during model tokenization. This is done to guarantee that textual material is purified, formatted and appropriate to be downstream tokenized without the linguistic subtleties that can reveal emotional distress.

3. Text Correction and Text Filtering.

Filtering and text correction are also implemented after the normalization process to increase the clarity and quality of the text. The automated tools like TextBlob or SymSpell can be adopted to spell correct selectively and concentrate on minor typing errors, and will not change clinical or emotional words and expressions. One of the most common preprocessing stages in NLP, stopword removal, is avoided, as ClinicalBERT has an active tokenizer that can work with function words without harming the semantic information of emotional sentences. In this study, lemmatization or stemming is also optional, given that ClinicalBERT contextualized embeddings naturally represent word morphology and word meaning. These adaptations make the text maintain its psychological and linguistic richness, which is essential in the sentiment analysis related to mental health.

4. Label Encoding

After refining the text data, the categorical emotion in the status column is transformed into numerical representation to make it easy to use in supervised learning. Every distinct emotional state is given a numerical value - such as; Anxiety = 0, Depression = 1, and Stress = 2. This encoding process allows the model to perceive emotion categories in the form of distinct target classes in the process of fine-tuning. Encoding is implemented with label encoders of standard machine learning packages like Scikit-learn so that it can be used with PyTorch or TensorFlow training pipelines.

5. Tokenization

Tokenizing is then carried out with ClinicalBERT tokenizer (AutoTokenizer.from_pretrained(emilyalsentzer/Bio_ClinicalBERT)) which is optimally configured to deal with medical and clinical terms. The tokenizer readings the text sequences and transforms them into subword words and numbers according to the ClinicalBERT vocabulary and the numbers. To facilitate uniformity all sequences are cut or padded to some constant maximum length, e.g. 128 tokens, and matching attention masks are produced to identify meaningful words as opposed to padding tokens. The tokenization step converts the text data that has been cleaned and normalized into a format that can be effectively used with the ClinicalBERT model to undergo in-depth fine-tuning.

6. Data Spliting

The dataset is split into three subsets (70% training, 15% validation, and 15% testing) to be able to train the model and evaluate it effectively. To ensure that the splits represent the proportion of the emotion classes, a stratified sampling method will be done to guarantee that all the subsets represent the same distribution that the entire dataset. This helps to avoid bias of the model to overrepresented emotion categories and enables the sound evaluation of the ability to generalize. ClinicalBERT is fine-tuned with the training set, hyperparameter tuning is done with the validation one, and the test set is left to evaluate the final model.

C. Machine Learning Model Selection

The choices of a suitable machine learning model is an essential step for achieving accurate sentiment and emotion classification, especially with mental health-related text data. Traditional models like Naïve Bayes, Support

Vector Machines (SVM), and Logistic Regression are commonly used for sentiment analysis in general fields. However, these models depend heavily on manually created features, like term frequency-inverse document frequency (TF-IDF) and bag-of-words representations. These features often fall short in capturing the subtle emotional and psychological nuances found in mental health language. Given the complexity and context of expressions in the dataset (e.g., “I feel restless” or “I’m mentally exhausted”), a more sophisticated, context-aware model is needed to ensure reliable performance.

To tackle this issue, the study uses ClinicalBERT, a transformer-based model specifically pretrained on large amounts of clinical and biomedical text. ClinicalBERT is based on the Bidirectional Encoder Representations from Transformers (BERT) architecture, which employs self-attention mechanisms to capture context in both directions within a sentence. Unlike traditional models, ClinicalBERT's pretraining on medical notes and clinical narratives helps it understand domain-specific language, including emotionally charged or symptom-related expressions common in patient communication. This makes ClinicalBERT well-suited for detecting subtle emotional signals in mental health-related statements, as seen in the dataset from Kaggle.

Using ClinicalBERT offers a major advantage in feature representation. Traditional NLP models treat words as independent tokens. In contrast, transformer-based architectures capture contextual meaning by looking at the relationships between words in both directions of a sentence. This allows ClinicalBERT to recognize emotionally different statements that share similar vocabulary but have different contexts. For instance, the phrases “I can’t sleep because of stress” and “I can finally sleep without stress” have overlapping terms yet express opposite sentiments. ClinicalBERT’s contextual embeddings help it accurately interpret these nuances, which are vital for mental health sentiment classification.

ClinicalBERT, the project includes a comparison with a baseline BERT model (BERT-base-uncased) to evaluate the benefits of clinical-domain pretraining. The baseline model, while strong in general language understanding, might not have the contextual depth needed to interpret emotionally or clinically specific terms. By comparing both models on the same dataset, the study aims to find out if using clinical-domain knowledge improves classification accuracy and model interpretability. This comparison also seeks to confirm the idea that domain adaptation through pretraining enhances the detection of emotion-related patterns in clinical text.

ClinicalBERT is fine-tuned on the processed dataset using supervised learning, where each statement is linked to a category label indicating its emotional state (e.g., Anxiety, Depression, or Stress). The model's parameters are optimized using techniques like AdamW optimization and cross-entropy loss, along with hyperparameter tuning to find the best balance between accuracy and generalization. The fine-tuned ClinicalBERT model should outperform baseline models in identifying emotional cues in patient-like statements, proving its potential as a decision-support tool for clinicians. By utilizing transformer-based architectures, the project aims to build a reliable, context-aware framework for automated sentiment analysis in the mental health field.

D. Training & Fine-Tuning

Training

The training phase involves teaching the ClinicalBERT model to recognize patterns and relationships between text and their emotional categories. Using the preprocessed dataset, which contains labeled statements in the status column, the model undergoes supervised learning. Each input text is paired with its emotional label, such as Anxiety, Depression, or Stress. During training, the model processes batches of tokenized text and adjusts its internal parameters. It minimizes the difference between predicted and true labels through a loss function, usually the cross-entropy loss. The AdamW optimizer manages the optimization process. This optimizer helps control learning rate decay and prevents overfitting. The dataset divides into training, validation, and testing sets to monitor performance. This setup ensures the model generalizes well to new data.

Throughout training, several hyperparameters are adjusted. These include batch size, learning rate, number of epochs, and maximum sequence length. These parameters directly affect the model's learning efficiency and overall accuracy. To prevent overfitting, techniques like early stopping and dropout regularization are used. These help the model perform well on validation data instead of just memorizing the training set. After each epoch, evaluation metrics such as accuracy, precision, recall, and F1-score are tracked. This helps assess progress and identify potential issues in model convergence. The outcome of this stage is a partially trained ClinicalBERT model. It can link linguistic cues in text to specific emotional states.

Fine-Tuining

The fine-tuning phase focuses on adjusting the pretrained ClinicalBERT model for mental-health sentiment analysis using the dataset from Kaggle. Unlike pretraining, where the model learns general language patterns and medical terms, fine-tuning sharpens these skills to specialize in emotion classification. During this step, all layers of ClinicalBERT are usually unfrozen. This allows the model to change its parameters based on the emotional expressions specific to patient statements. Fine-tuning uses a smaller dataset that is tailored to the task and a carefully chosen low learning rate to prevent losing previously learned language knowledge. The model learns to spot subtle emotional differences in text, helping it distinguish between similar but contextually different expressions of distress or well-being.

After fine-tuning, the model is evaluated on a test set to check its real-world effectiveness. Performance metrics like F1-score and confusion matrix analysis are highlighted, as they give a clearer view of the model’s

ability to identify rare emotional classes, such as extreme anxiety or mixed emotions. The fine-tuned ClinicalBERT model is expected to show better contextual understanding compared to general-purpose models, reflecting the benefits of clinical-domain pretraining. This final model can then serve as a decision-support tool, giving clinicians an additional perspective on patient emotional states, which helps reduce diagnostic uncertainty and improve early mental health detection.

E. F3 Experimentation

```python
# IMPORTANT: This exercise uses the SAME data and model from Exercise F2:
#   - Same model: ClinicalBERT (emilyalsentzer/Bio_ClinicalBERT)
#   - Same data splits: X_train, X_val, y_train, y_val (from Exercise F2)
#   - Same class weights and metrics computation
#   - Only difference: Using automated hyperparameter optimization
# ============================================================================

import time
import json
from datetime import datetime
from openpyxl import Workbook
from openpyxl.styles import Font, PatternFill, Alignment
import torch
import numpy as np
from transformers import (
    AutoModelForSequenceClassification,
    AutoTokenizer,
    TrainingArguments,
    Trainer,
    set_seed
)
from datasets import Dataset
from sklearn.metrics import accuracy_score, precision_recall_fscore_support

# Set seed for reproducibility
set_seed(42)

# Device setup
device = torch.device("cuda" if torch.cuda.is_available() else "cpu")
print(f"Using device: {device}")

# Model and tokenizer setup (using ClinicalBERT from Exercise F2)
CLINICAL_BERT = "emilyalsentzer/Bio_ClinicalBERT"
tokenizer = AutoTokenizer.from_pretrained(CLINICAL_BERT)

# Number of classes (from Exercise F2 - using same y_train variable)
num_labels = len(np.unique(y_train))
avg_type = "binary" if num_labels == 2 else "weighted"
print(f"Detected {num_labels} classes → using average='{avg_type}' for metrics")

# Tokenize datasets (reusing SAME X_train, X_val, y_train, y_val from Exercise F2)
def tokenize_texts(texts, max_length=160):
    return tokenizer(
        list(texts),
        padding=True,
        truncation=True,
        max_length=max_length,
        return_tensors="pt"
    )
```

Figure 1.1 Environment Setup for Automated Hyperparameter Search

This cell prepares the environment for automated hyperparameter tuning by importing necessary libraries and reusing the same data and model configuration from the previous manual experiments. It imports timing utilities to track how long each search takes, Excel file creation tools for logging results, and all transformer components needed for model training and evaluation. The code reuses the same training and validation data splits, the same ClinicalBERT model and tokenizer, and the same class weights and metrics computation from the manual experiments, ensuring that the automated search is directly comparable to the manual approach. It tokenizes the data using the same parameters, creates Dataset objects in the format expected by the training framework, computes class weights using the same method, and sets up the metrics function and weighted trainer class exactly as before. The output confirms the device being used, the number of classes detected, and that the setup completed successfully, ready for automated hyperparameter optimization.

```python
def random_search_hp_space(trial):
    """
    Define the hyperparameter space for Random Search.
    Random Search samples RANDOMLY from continuous/discrete ranges.
    """
    learning_rate = trial.suggest_float("learning_rate", 1e-5, 3e-5, log=True)  # Lower range
    per_device_train_batch_size = trial.suggest_categorical("per_device_train_batch_size", [8, 16])  # Smaller batches
    weight_decay = trial.suggest_float("weight_decay", 0.0, 0.01)  # Same (already low)
    num_train_epochs = trial.suggest_int("num_train_epochs", 2, 2)  # Keep 2 epochs for speed

    return {
        "learning_rate": learning_rate,
        "per_device_train_batch_size": per_device_train_batch_size,
        "weight_decay": weight_decay,
        "num_train_epochs": num_train_epochs,
    }

# Training arguments template (same as grid search)
random_training_args = TrainingArguments(
    output_dir="./random_search_results",
    eval_strategy="epoch",
    save_strategy="epoch",
    load_best_model_at_end=True,
    metric_for_best_model="f1",
    greater_is_better=True,
    fp16=torch.cuda.is_available(),
    report_to="none",
    warmup_steps=500,
```

Figure 1.2 Random Search Hyperparameter Optimization

This cell implements Random Search to automatically find the best hyperparameters by randomly sampling values from defined ranges rather than testing every possible combination. It defines a hyperparameter space function that specifies ranges for learning rate, batch size, weight decay, and number of epochs, using continuous ranges for learning rate and weight decay that allow sampling any value within those bounds rather than being limited to specific grid points. The code sets up training arguments with evaluation and saving strategies, creates a weighted trainer configured for the search process, and executes Random Search with a specified number of trials. Each trial randomly selects hyperparameter values from the defined ranges, trains a model with those settings, evaluates performance, and records the results. The search process tracks execution time and identifies the best trial based on F1 score. The output shows progress for each trial as models train, displays the best hyperparameters found, and reports the best F1 score achieved along with the total time taken for all trials.

```python
import pandas as pd

# Create summary data for Excel
summary_data = []

# Random Search Summary
if random_best_trial:
    summary_data.append({
        "Search_Type": "Random Search (Automated)",
        "Best_F1_Score": random_best_trial.objective,
        "Best_Learning_Rate": random_best_hps.get("learning_rate", "N/A"),
        "Best_Batch_Size": random_best_hps.get("per_device_train_batch_size", "N/A"),
        "Best_Weight_Decay": random_best_hps.get("weight_decay", "N/A"),
        "Best_Epochs": random_best_hps.get("num_train_epochs", "N/A"),
        "Total_Trials": 6,  # Updated for fast config
        "Total_Time_Seconds": random_total_time,
        "Time_Per_Trial_Seconds": random_total_time / 6,
        "Strategy": "Random Sampling - Continuous ranges",
        "Member_Number": MEMBER_NUMBER
    })

if summary_data:
    summary_df = pd.DataFrame(summary_data)
    print("\n=== RANDOM SEARCH SUMMARY ===")
    print(summary_df.to_string(index=False))
    print("\nNote: This will be compared to Exercise F2 manual experiments in the Excel file.")
else:
    print("⚠️  Random Search did not complete. Please run Random Search first.")
```

Figure 1.3 Hyperparameter Search Results Summary

This cell prepares the Random Search results for analysis and logging by extracting information about the best trial and creating summary data structures. It attempts to access trial information from the search process, though the transformers library may limit access to individual trial details beyond the best result. The code creates summary data that includes the best F1 score, the best hyperparameters found, the total number of trials executed, the total time taken, and the average time per trial. It organizes this information into a structured format that can

be easily compared with results from manual experiments, calculating efficiency metrics like F1 score per unit of time. The output displays a summary table showing the key results from Random Search, which will be used for comparison with the manual hyperparameter tuning approach and logged to an Excel file in the next cell.

```
        cell.fill = header_fill
        cell.font = header_font
        cell.alignment = Alignment(horizontal="center")

    row = 2

    # Add Random Search best result
    if random_best_trial:
        ws.cell(row=row, column=1, value=f"Member {MEMBER_NUMBER}")  # Use member number from config
        ws.cell(row=row, column=2, value="Best")
        ws.cell(row=row, column=3, value=random_best_hps.get("learning_rate", "N/A"))
        ws.cell(row=row, column=4, value=random_best_hps.get("per_device_train_batch_size", "N/A"))
        ws.cell(row=row, column=5, value=random_best_hps.get("weight_decay", "N/A"))
        ws.cell(row=row, column=6, value=random_best_hps.get("num_train_epochs", "N/A"))
        ws.cell(row=row, column=7, value=random_best_trial.objective)
        ws.cell(row=row, column=11, value=random_total_time)
        ws.cell(row=row, column=12, value=datetime.now().strftime("%Y-%m-%d %H:%M:%S"))
        row += 1

    # Add comparison sheet (Random Search vs Exercise F2)
    ws2 = wb.create_sheet("Comparison_Analysis")

    comparison_headers = [
        "Metric", "Random Search (Automated)", "Exercise F2 (Manual)", "Notes"
    ]

    for col_idx, header in enumerate(comparison_headers, 1):
        cell = ws2.cell(row=1, column=col_idx, value=header)
```

Figure 1.4 Random Search Results Export and Comparison Analysis

This cell creates a formatted Excel workbook that logs the Random Search results and provides a comparison analysis against the manual experiments from earlier sections. It creates a new workbook with multiple sheets, including one for detailed Random Search results with columns for member identification, trial number, all hyperparameters, performance metrics, training time, and timestamps. The code writes the best trial results to the first sheet with properly formatted and styled headers. It creates a second comparison sheet that contrasts Random Search results with manual experiment results, including metrics like best F1 score, total execution time, and efficiency calculations. The comparison sheet includes analysis notes explaining the differences between automated and manual approaches, highlighting that Random Search can explore continuous hyperparameter ranges more efficiently. The workbook automatically adjusts column widths, applies formatting with colored headers, and saves the file with a timestamped filename. If running in Google Colab, the file is automatically downloaded, otherwise it is saved locally with a confirmation message showing the file location

VI. Results

Member	Trial #	Learning Rate	Batch Size	Weight Decay	Epochs	F1 Score	Accuracy	Precision	Recall	Training Time (s)	Timestamp
Perez	1	0.000011	8	0.0061	2	0.7980	0.7876	0.7811	0.8124	9659.9	2025-11-08 14:44:21
Perez	2	0.000018	8	0.0003	2	0.8340	0.8298	0.8361	0.8242	9550.2	2025-11-08 14:44:21
Perez	3	0.000026	16	0.0022	2	0.8143	0.8069	0.7874	0.8280	4892.8	2025-11-08 14:44:21
Perez	4	0.000029	16	0.0039	2	0.8473	0.8451	0.8441	0.8306	4901.5	2025-11-08 14:44:21
Perez	5	0.000014	8	0.0026	2	0.7903	0.7920	0.7793	0.7958	9716.1	2025-11-08 14:44:21
Perez	6	0.000027	16	0.0091	2	0.8215	0.8057	0.7988	0.8396	4814.2	2025-11-08 14:44:21

Figure 1.5 Experimentation Results

Hyperparameter Optimization Performance

We conducted automated hyperparameter optimization using Random Search with ClinicalBERT across six trials. The search explored learning rates (1.1e-5 to 2.9e-5), batch sizes (8 and 16), weight decay (0.0003 to 0.0091), and training epochs (2 epochs consistently).
Trial 4 achieved the best performance with an F1-score of 0.8473, accuracy of 0.8451, precision of 0.8441, and recall of 0.8306. This configuration used a learning rate of 2.9e-5, batch size of 16, weight decay of 0.0039, and completed training in 4,901.5 seconds (approximately 82 minutes).

Performance Distribution

The six trials showed F1-scores ranging from 0.7903 to 0.8473, demonstrating consistent performance across different hyperparameter combinations. The top three configurations all exceeded 0.82 F1-score:

- Trial 4: F1=0.8473, LR=2.9e-5, batch=16, weight_decay=0.0039
- Trial 2: F1=0.8340, LR=1.8e-5, batch=8, weight_decay=0.0003
- Trial 6: F1=0.8215, LR=2.7e-5, batch=16, weight_decay=0.0091

All three top performers used learning rates above 1.8e-5, suggesting ClinicalBERT benefits from slightly higher learning rates than initially expected for this mental health classification task.

Batch Size Impact

Batch size significantly affected both performance and training efficiency. Configurations with batch size 16 (Trials 3, 4, and 6) completed training in approximately 4,800-4,900 seconds, while batch size 8 configurations (Trials 1, 2, and 5) required 9,550-9,716 seconds—nearly double the training time.

The larger batch size also correlated with better performance. The top two F1-scores (0.8473 and 0.8215) came from batch size 16 configurations, though Trial 2 with batch size 8 still achieved competitive performance (F1=0.8340).

Weight Decay Effects

Weight decay showed a non-linear relationship with performance. Moderate weight decay values (0.0039 in Trial 4) produced the best results, while very low (0.0003 in Trial 2) and high (0.0091 in Trial 6) values yielded slightly lower performance. This suggests optimal regularization exists in the middle range, preventing overfitting without overly constraining the model.

Training Efficiency

Random Search completed six trials in a total of approximately 43,535 seconds (12.1 hours). The search efficiently explored the hyperparameter space, with the best configuration emerging in Trial 4. Each trial provided valuable information about the performance landscape, allowing rapid identification of effective hyperparameter combinations without exhaustive grid enumeration.

VII. Discussion

Model Performance and Clinical Utility

The best F1-score of 0.8473 represents strong emotion classification capability for mental health text. This

performance exceeds the typical threshold of 0.80 that indicates reliable classification in clinical NLP applications. With accuracy of 0.8451, the model correctly classifies approximately 85% of emotional states in patient-like statements.

The balanced precision (0.8441) and recall (0.8306) demonstrate that the model performs well across different emotion categories without systematically favoring one metric over the other. This balance proves important in clinical settings—you need both to avoid missing true cases of distress (recall) and to minimize false alarms (precision).

Hyperparameter Insights

The optimal learning rate of 2.9e-5 falls in the higher range of values typically used for fine-tuning BERT-based models. This suggests ClinicalBERT's pretraining on clinical text creates a foundation that allows more aggressive parameter updates during fine-tuning for emotion classification. The model can adapt quickly to mental health sentiment patterns without destabilizing previously learned clinical language representations.

Weight decay of 0.0039 provided optimal regularization, preventing overfitting while maintaining model flexibility. Too little regularization (0.0003) allowed potential overfitting, while excessive regularization (0.0091) constrained the model's ability to learn task-specific patterns. The sweet spot at 0.0039 balanced these competing demands effectively.

Training Configuration Trade-offs

Batch size 16 emerged as the clear winner for both performance and efficiency. The larger batch size provided more stable gradient estimates, allowing the model to converge reliably within 2 epochs. Training time dropped by roughly 50% compared to batch size 8, completing in approximately 82 minutes versus 160 minutes.

This efficiency gain matters for practical deployment. You can iterate through multiple experiments in a single day, enabling faster model development and testing. The performance advantage (F1=0.8473 vs 0.8340 for the best batch 8 configuration) adds further justification for using larger batches when GPU memory permits.

Comparison with Domain-Specific Models

The 0.8473 F1-score with ClinicalBERT significantly outperforms the earlier results with RoBERTa-Tagalog and XLM-RoBERTa (F1=0.789). This 5.8 percentage point improvement validates the importance of clinical domain pretraining for mental health text analysis.

ClinicalBERT's exposure to medical notes and clinical narratives during pretraining provides crucial contextual understanding for emotion classification. The model recognizes medical terminology, symptom descriptions, and clinical language patterns that appear in mental health expressions. This domain knowledge allows more accurate interpretation of patient-like statements compared to general-purpose or even Filipino-specific language models.

Random Search Efficiency

Random Search proved highly effective for this hyperparameter optimization task. The best configuration appeared in Trial 4 of 6, and all trials completed within 12.1 hours—a practical timeframe for research and development work. The search efficiently sampled the hyperparameter space without requiring exhaustive enumeration of all possible combinations.

The performance spread (F1 from 0.7903 to 0.8473) indicates that hyperparameter selection matters, but most configurations achieved acceptable results above 0.79. This robustness suggests ClinicalBERT handles mental health classification well across reasonable hyperparameter choices, reducing the risk of poor performance from suboptimal settings.

Clinical Decision Support Applications

The 0.8473 F1-score positions this model as a viable clinical decision support tool for mental health screening. You can deploy it to automatically analyze patient statements from intake forms, chat interactions, or electronic health records, flagging potential cases of anxiety, depression, or emotional distress for clinician review.

The model works best as a first-stage screening system rather than a diagnostic tool. It identifies patient statements that warrant clinical attention, allowing mental health professionals to prioritize their limited time on cases showing signs of distress. This approach combines automated pattern recognition with human clinical judgment, leveraging the strengths of both.

Interpretability and Trust

Healthcare applications require explainable predictions that clinicians can understand and trust. Future work should incorporate attention visualization or feature importance analysis to show which words or phrases most strongly influenced each classification decision. When the model flags a statement as indicating anxiety, clinicians need to see what linguistic patterns triggered that prediction.

Techniques like attention weight visualization or LIME (Local Interpretable Model-agnostic Explanations) can provide this transparency. These methods help clinicians understand the model's reasoning, identify potential biases or errors, and build appropriate trust in the system's predictions.

Resource Requirements and Accessibility

Training ClinicalBERT with optimal hyperparameters requires approximately 82 minutes on standard research GPU infrastructure. This modest computational requirement makes the approach accessible to healthcare organizations and research institutions without expensive high-performance computing resources.

You can fine-tune and deploy these models using cloud computing services, university research clusters, or even high-end workstation GPUs. This accessibility supports broader adoption of AI-assisted mental health screening, particularly in resource-constrained settings where traditional mental health services remain limited.

Limitations and Considerations

Several limitations constrain these findings. First, we tested only six Random Search trials—expanding to 20-30 trials might identify even better hyperparameter combinations. The optimal learning rate of 2.9e-5 sits at the edge of our

explored range, suggesting we should test slightly higher values.

Second, we fixed training epochs at 2 across all trials. Testing 3-4 epochs might improve performance, though it would increase training time proportionally. The two-epoch configuration proved sufficient for strong results, but longer training could push F1-scores higher.

Third, the dataset comes from Kaggle rather than actual clinical interactions. While the Sentiment Analysis for Mental Health dataset provides valuable labeled examples, it may not fully represent the linguistic patterns, code-switching, and cultural context of Filipino mental health discourse. Validation on real patient data remains essential before clinical deployment.

Generalization and Robustness

The model's performance across different emotion categories requires detailed analysis through confusion matrix examination. High overall F1-score doesn't guarantee equal performance on all emotional states—the model might excel at detecting anxiety while struggling with less common emotions.

You should evaluate per-class performance metrics to identify strengths and weaknesses across different mental health conditions. This analysis reveals whether the model generalizes well to all emotion types or shows bias toward overrepresented categories in the training data.

Future Research Directions

Several research directions emerge from these findings. First, ensemble methods that combine predictions from multiple well-performing configurations (Trials 2, 4, and 6) might push performance above 0.85 F1-score. Ensemble voting or stacking leverages diverse model perspectives on emotional classification.

Second, testing ClinicalBERT on other mental health datasets would validate generalization beyond the Kaggle corpus. Cross-dataset evaluation reveals whether the model learned genuine emotion recognition capabilities or dataset-specific patterns.

Third, exploring other clinical transformer models like BioClinicalBERT or PubMedBERT could identify whether ClinicalBERT's specific pretraining corpus provides advantages for mental health text, or if any clinically-pretrained model performs similarly.

Contribution to Mental Health Care

This work demonstrates that automated emotion classification can achieve clinically useful performance levels (F1>0.84) using readily available computational resources and datasets. The approach scales to handle large volumes of patient statements, enabling mental health screening in contexts where human assessment proves impractical due to time or resource constraints.

Healthcare systems can integrate these models into patient intake workflows, online mental health platforms, or electronic health record systems. Automated screening reduces the burden on mental health professionals while ensuring that concerning patient statements receive timely attention.

VIII. References

[1] L. Canales and P. Martínez-Barco, “Emotion Detection from Text: A Survey,” in Proc. Workshop on Natural Language Processing in the 5th Information Systems Research Working Days (JISIC), Quito, Ecuador, Oct. 2014, pp. 37–43.

[2] S. Elgayar, “Study on Emotion Analysis of Text using Machine Learning Approaches,” SSRN, Feb. 2023.

[3] “Text-Based Emotion Recognition Using Deep Learning Approach,” PMC, 2022.

[4] A. Seyeditabari, N. Tabari, S. Gholizadeh, and W. Zadrozny, “Emotion Detection in Text: Focusing on Latent Representation,” arXiv preprint arXiv:1907.09369, 2019.

[5] K. Huang, J. Altosaar, and R. Ranganath, “ClinicalBERT: Modeling Clinical Notes and Predicting Hospital Readmission,” arXiv preprint arXiv:1904.05342, 2019.

[6] “Evaluating Pretraining Strategies for Clinical BERT Models,” in Proc. LREC, 2022.

[7] J. Devlin, M. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding,” in Proc. NAACL-HLT, 2019, pp. 4171–4186.

[8] “Clinically Relevant Pretraining is All You Need,” PMC, 2022.

[9] “Contextual Embeddings from Clinical Notes Improves Prediction of Medical Outcomes,” PMC, 2021.

[10] “Does BERT Pretrained on Clinical Notes Reveal Sensitive Data?” in Proc. NAACL Main, 2021.

[11] “Review of Sentiment Analysis and Emotion Detection from Text,” PMC, 2021.

[12] “Deep Emotion Recognition in Textual Conversations: A Survey,” Springer AI Review, 2024.

[13] Sentiment Analysis for Mental Health. (2024, July 5). Kaggle. https://www.kaggle.com/datasets/suchintikasarkar/sentiment-analysis-for-mental-health

[14] “Goal 3: Good Health and Well-Being,” *Global Goals — United Nations*, 2015. [Online]. Available: https://globalgoals.org/goals/3-good-health-and-well-being/.

[15] World Health Organization, Comprehensive Mental Health Action Plan 2013–2030, Geneva: WHO, 2021. [Online]. Available: https://www.who.int/publications/i/item/9789240031029

[16] P. Saxena, D. Setoya, and S. World Health Organization, “Digital technologies for mental health: opportunities and challenges,” World Psychiatry, vol. 21, no. 3, pp. 370–371, 2022. doi: 10.1002/wps.21012

[17] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding,” in *Proc. NAACL-HLT*, 2019, pp. 4171–4186.

[18] C. Sun, X. Qiu, Y. Xu, and X. Huang, “How to Fine-Tune BERT for Text Classification?,” *arXiv preprint* arXiv:1905.05583, 2019.

[19] A. Conneau *et al.*, “Unsupervised Cross-lingual Representation Learning at Scale,” in *Proc. ACL*, 2020, pp. 8440–8451. (XLM-RoBERTa)

[20] S. Gururangan, A. Marasović, S. Swayamdipta, K. Lo, I. Beltagy, D. Downey, and N. A. Smith, “Don’t Stop Pretraining: Adapt Language Models to Domains and Tasks,” in *Proc. ACL*, 2020, pp. 8342–8360.

[21] J. C. Blaise, “RoBERTa-Tagalog Base,” *Hugging Face Model Card*, 2020–2024. [Online]. Available: https://huggingface.co/jcblaise/roberta-tagalog-base