Utilizing Transformer for Emotional Understanding on Chinese Mental-Health Dataset

by

Mingyu Du
B.Sc., University of Victoria, 2018

A Report Submitted in Partial Fulfillment of the
Requirements for the Degree of

MASTER OF ENGINEERING

in the Department of Electrical and Computer Engineering
University of Victoria
BC, Canada

© Mingyu Du, 2025
University of Victoria

All rights reserved. This project may not be reproduced in whole or in part, by photocopying or other means, without the permission of the author.

Utilizing Transformer for Emotional Understanding on Chinese Mental-Health Dataset

by

Mingyu Du
B.Sc., University of Victoria, 2018

Supervisory Committee

Dr. Xiaodai Dong, Faculty Supervisor
(Department of Electrical and Computer Engineering)

Dr. Hong-Chuan Yang, Committee Member
(Department of Electrical and Computer Engineering)

ABSTRACT

The rapid development of large language models has demonstrated successful performance in various areas. In terms of mental health, large language models exhibit the capability to understand emotional feeling to some extent. However, research in the mental health field requires a broad range of interdisciplinary knowledge and is often constrained by limited resources. This project focuses on the analysis of sentiment in conversational texts using large language models and investigating the model performances. By comparing 8 different open source models, the project demonstrates the outstanding performance of hfl/chinese-roberta-wwm-ext in emotional understanding using the mental health dataset released by Tongji University.

Contents

List of Tables

List of Figures

ACKNOWLEDGEMENTS

I am deeply grateful to my mentor, Dr. Xiaodai Dong, for her exceptional guidance and unwavering support throughout this research endeavor. Her expertise, insightful feedback, and continuous encouragement have been invaluable in shaping the direction and outcomes of this study. Her unwavering commitment to my academic growth and professional development has been truly inspiring.

I am indebted to helpful folks for their generous allocation of time and resources, their willingness to share their wealth of knowledge, and their unwavering dedication to pushing me to new heights. Their help has not only enriched the quality of this research but has also had a profound impact on my personal and intellectual growth. I am truly fortunate to have had the privilege of working under their guidance.

Moreover, Thanks to OpenAI because I used Chatgpt to help me correct grammar and spelling mistakes.

Chapter 1

Introduction

Large language models (LLMs) are revolutionizing the mental health field with their advanced natural language processing capabilities. These AI-driven tools can understand and generate human-like text, offering benefits such as early detection of mental health issues, digital interventions, and clinical support. They are being utilized for diagnostics, therapy, and enhancing patient engagement, demonstrating effectiveness in providing accessible and destigmatized eHealth services. LLMs have the potential to analyze patient data, summarize therapy sessions, and aid in complex diagnosis, thus saving significant time and effort. Furthermore, they can offer personalized guidance, self-assessment of symptoms, and support in treatment decisions. However, the use of LLMs in mental health also presents challenges, including data privacy, the need for clinical validation, and the risk of biases in the training data that could lead to unfair treatment. Despite these concerns, LLMs show promising potential in advancing mental health care, emphasizing the need for continued research and development in this area.

1.1 Transformer and Bert

Transformer is a type of deep learning model introduced in the paper Attention is All You Need by Vaswani et al [1]. They adopt a mechanism called self-attention to weigh the importance of different words in a sentence, allowing the model to understand context more effectively. This architecture is highly parallelizable and has become the foundation for many state-of-the-art natural language processing (NLP) models. BERT, developed by Google, is a specific implementation of the Transformer architecture. It is designed to understand the context of a word in a sentence by looking at the position before and after

the target word to enhance bidirectional relation. This makes BERT particularly powerful for tasks like sentiment analysis, where understanding the nuance and context of words is crucial. BERT can be fine-tuned on sentiment analysis datasets to classify the sentiment of text as positive, negative, or neutral.

1.2 Outline of Report

This project studies the sentiment analysis of conversational texts, and addresses challenges related to dataset bias, cultural issues, and limited understanding, because the dataset is based on Chinese context, which is different from commonly researched and tested English dataset, and the project examines different model's capability about emotional understanding on the target dataset.

The structure of the report is that the Chapter 1 introduces about project background and techniques applied in this project and the Chapter 2 offers the details regarding the fundamentals of Transformer, which is the core system of this project. The Chapter 3 is the expriment part that consists of proble definition, alogorithm, dataset and LLM training, etc. The Chapter 4 is for conclusion and future work.

1.3 Literature Review

1.3.1 Overview

Generally, the literature review is conducted by following the PRISMA framework, which is introduced and employed in paper [2]. The primary objective is to conduct a systematic and transparent review of LLM research papers across the domain of mental health. This process aims not only to discover the existing progress but also to facilitate reciprocal learning and strengths sharing. In this part, the following key questions are addressed:

1. What can be identified as Background and Foundation in the research approaches within the domain?

2. What are the profound progress of research in the prior studies?

3. What are the identical gaps and how this project can augment and improve to address?

1.3.2 Background and Foundation

AI and large language models (LLMs) are involved in the mental health domain because they address the growing need for accessible mental health interventions, particularly considering the global shortage of mental health professionals and barriers faced by individuals seeking traditional therapy [3]. Especially during the Covid-19 pandemic, the situation could be much worse as the review [4] noted. A similar viewpoint is also supported in the reviews [5, 6].

Regarding prior works, review [7] examined 53 studies evaluating 41 different chatbots, focusing on their primary functions, such as therapy, training, and screening for mental health issues, with a particular emphasis on depression and autism. The review categorized chatbots based on several criteria, including purpose, response mechanisms, and target disorders. Another review [8] discusses the various applications of chatbots, such as in prevention, treatment, and post-treatment support for mental health issues. For instance, study [9] explores Woebots [10] usability, acceptability, and preliminary effectiveness in reducing problematic substance use among adults aged 18-65. A similar trial, for application called Tess, is conducted as well [11]. And other representative applications like Wysa [12] are introduced to the world as typical Chatbots for having conversation with people suffering from mental heath issues. A specific example is shown in paper [13] that develops a program featuring a virtual agent capable of engaging users in a multi-topic conversation and giving real-time feedback on nonverbal cues while analyzing spoken language and facial expressions.

Findings in [14] reveal that a significant portion of participants acknowledges the benefits of chatbots in mental healthcare, with 65% agreeing on their positive impact and 79% believing they could empower clients in managing their health. Persisting challenges regarding emotional comprehension and sensitive issues remain significant barriers [14]. Despite the progress made by chatbots utilizing traditional AI algorithms, the review in [15] calls for further research with standardized outcome measures, emphasizing the need for more rigorous studies to thoroughly evaluate the effectiveness of conversational agents in mental health treatment. Furthermore, research in [16] advocates for incorporating user perspectives in developing chatbots, a concern also applied to conversational agents (CAs).

With the advent of large language models (LLMs) like ChatGPT, there is significant potential to enhance text-centric multimodal sentiment analysis tasks [17]. More than that, the idea named Conversational Agents, referred to as CA throughout this paper, is applied in mental health domain. CA is a specific role of LLM in a dialogue that is driven by an

identical purpose, which is either to support a potential user with what is in need or to play a role of assistant in the mental health clinical workflow. To discover the potential capability of LLM to address mental-health related issues [18], not only the publicly available LLMs, such as ChatGPT achieved final diagnosis accuracy of 76.9% (95% CI, 67.8%-86.1%) in findings from a study of 36 clinical vignettes [19], but also the pretrained language models are utilized in development for aiding mental health practitioners [20].

1.3.3 Prior Studies

There have been innovative achievements proposed. Emotional understanding or emotional intelligence and so on, is the key to empower a solution regardless of the application scenarios. In therapy process, which is about enhancing emotional understanding or incorporating emotional intelligence to support users in need, or in other words, the target audience are those who are reluctant to seek mental health advice due to stigmatization. For example, a virtual assistant (VA) is designed to provide initial mental health support, particularly for individuals suffering from major depressive disorder (MDD) [21]. This kind of support in essence is to put in empathy tones in the response to help ease negative emotion. Also, a combination of different method is adapted to addressee the gap regarding emotional understanding [22]. For example, focusing on accurately interpreting emotional cues in speech to generate contextually appropriate responses [23]. In addition, an interesting perspective [24] is to summarize psychotherapy discussions during psychotherapy sessions. Similarly, yet different, in another paper, the authors aim to create a more effective summarization tool that can facilitate quicker and more accurate counselor responses, ultimately benefiting users seeking mental health support in online communities [25]. The paper proposes incorporating a curriculum learning technique, inspired by human teaching methods, to enhance the training process by progressively introducing easier to harder examples.

Another purpose is to perform predictions, in different scenarios, like early detection regarding depression, post-traumatic-stress-disorder (referred to PTSD throughout this paper), anxiety, etc. For example, the paper [20] presents a model that performs early detection of mental disorders and suicidal ideation from social content, which can help effectively prevent suicide. Additionally, the paper states that conversational agents (CAs) can be personalized and optimally implemented in clinical practice to enhance treatment outcomes and support mental health professionals [26]. Another example is an interactive AI-based tool [27] is particularly beneficial for less experienced counselors, augmenting human capabilities by employing advanced natural language generation techniques to

diagnose counseling needs and provide tailored example responses, which is generating personalized suggestions based on specific counseling strategies in mental health support, aiming for better outcomes for those seeking help online. Besides, the document [28] focuses on improving early detection methods and overall mental health understandings by utilizing transformer-based architectures (BERT and RoBERTa) alongside BiLSTM neural networks for effective multiclass prediction (such as anxiety, ADHD, bipolar disorder, PTSD, depression, etc.) to analyze and identify a diverse set of linguistic features, that most effectively signify mental health conditions, sourced from Reddit posts [28]. Lastly, an outstanding achievement made in paper [29], demonstrates that task-adaptive tokenization significantly enhances generation performance while using up to 60% fewer tokens than traditional methods. This innovative approach not only enriches the linguistic capabilities of existing language models but also represents a meaningful advancement in the accessibility and effectiveness of mental health support through text generation technologies.

The third purpose is to develop evaluation methods to assess the understanding capability of LLMs. For example, it is critical for LLM effective and ethically responsible deployment in real-world scenarios of mental health [30]. And another outstanding example is this paper [31], which introduce the EmoBench, that provided a structured framework based on psychological theories, proposing a comprehensive definition of machine EI that includes Emotional Understanding (EU) and Emotional Application (EA).

One perspective suggests that prompt engineering can contribute to improving emotional understanding in LLMs. Paper [32] introduces EmotionPrompt, a method that combines original prompts with additional emotional stimuli to assess LLMs emotional understanding. Another perspective, which is stated in paper [33], that considers facial expression may not be authentic and privacy issues. The authors introduce a novel annotated dataset called Non-Facial Body Language (NFBL) for emotion analysis in long-sequential, de-identified videos, named EALD, by collecting and processing athletes post-match interview sequences. In addition to the above findings, a profound innovation is proposed in paper [34]. The authors propose the openCHA framework, an open-source solution that integrates external resources such as knowledge bases and data sources, enabling CHAs to provide tailored healthcare responses.

1.3.4 Research Gaps

There are also common research challenges alongside prior progress, as Table 1.1 generally categorized and stated:

Table 1.1: Challenges discovered in literature review

Challenges	Description
Dataset Bias	(1) It involves unbalanced classes in dataset [20]. (2) Datasets are typically presented in English. For example, the datasets predominantly consist of response from English speakers, which raises concerns about cultural differences in expressing depression [22]. (3) Small-size dataset reduces the ability to draw firm conclusions about the reliability and validity of the findings.
Gender Bias	Societal biases influence the perception of mental health conditions [35].
Culture Issues	This arises from multilingual factors, which cause differing emotional understandings across cultures [20].
Limited Understandings	(1)It is about limitations in the LLMs ability to understand or respond appropriately. Paper [22] specifically noted that this limitation may undermine the robustness of emotional and semantic understanding required for accurate mental health detection. This is a challenge as it reduces the efficacy and effectiveness of conversational agents (CAs). (2) LLMs often rely on patterns in the data rather than a deep understanding of emotions, leading to significant gaps between LLMs and human emotional intelligence [31]. In addition, it could result in unnecessary interruptions during conversation, and it puts negative impact on users experience, especially when users are in desire to talk and for attention from listeners.

User experience	(1) This issue arises from the user control mechanism. For example, participants expressed a desire for more control over the intervention, including the ability to tailor session lengths and content to their needs [36]. (2) Another reason is is the absence of personalization. Conversations that are not tailored to users individual needs or circumstances can result in reduced engagement. [37].
Concerns on Psychological Comorbidities	It means that assorted symptoms are very common in individuals with mental health issues [36]. In other words, depression and anxiety could concurrently persist. And LLM applications usually address by response or prediction on one symptom.
Response generation problems	(1) This limitation means it does not yet incorporate a text-to-speech model, which would enhance the user experience by allowing the system to communicate in spoken form rather than just text [23]. (2) Another limitation is about generation speed. The speed of generation could be a problem. According to study [21], task-adaptive tokenization did not significantly improve generation speed in English, possibly due to the structural differences between the English vocabulary and character-based languages like Chinese.
Reasoning techniques	The study primarily used chain-of-thought reasoning, which may not fully utilize other reasoning techniques that could be more effective in emotional scenarios [31].
Inherent Vulnerabilities of Large Language Models (LLMs)	Common issues in LLMs, such as hallucination (the generation of incorrect or fabricated information) and sensitivity to prompts, which could undermine its performance on emotional tasks [38].
Interaction Limitations	Issues related to the threshold for recognizing user utterances led to frequent interruptions by the robot, which could disrupt interactions and affect user experience negatively [28].

Cooperation issue between multiple components	The performance of a single system or solution may compromise because one component relies on the quality of the output of another one. For instance, the effectiveness of both the Motivational Response Generator (MRG) and the Empathetic Rewriting Framework (ERF) heavily depends on contextual understanding. If the system cannot accurately capture or interpret context, the responses may fall short [39]. There is another specific and detailed example in paper [40] that stated Employing the ROUGE-1 score explicitly as a reward is highly prone to the collapse, and as a consequence, the text generation deteriorates through a repetition of similar phrases.

Chapter 2

Fundamentals of Transformer

2.1 Overeview

Before diving into the details of project techniques for emotional understanding tasks, it is necessary to introduce Transformer and pre-trained models. First, they are crucial and profound work in the prior studies. Second, detailed explanation of Transformer and Bert can be helpful in understanding the adopted techniques of this project. Therefore, the following paragraphs are mainly oriented on the working principles of Transformer and Google-Bert.

2.2 Working Principle of Transformer

The Transformer is a model architecture for processing sequence data (such as text, audio, and time series) in deep learning. It can capture remote dependencies in parallel. Its core innovation is the Self-Attention Mechanism, which enables the model to handle global dependencies in sequences without relying on recursive structures.

Originally proposed by Vaswani et al. in their 2017 paper Attention is All You Need, the Transformer differs from traditional recurrent neural networks (RNNs) and long short-term memory networks (LSTMs) by using Self-Attention instead of sequential processing. This design allows the Transformer to process data more efficiently in parallel, especially with long sequences.

In the evolution of natural language processing (NLP), long short-term memory (LSTM) and recurrent neural networks (RNN) have played crucial roles in capturing temporal dependencies in sequence data. However, as task complexity increases, these models

face challenges, particularly in dealing with long-distance dependencies and parallelization.

Key challenges remain as follows:

Table 2.1: Challenges in previous solutions

Challenges	Definition
Sequential modeling	Sequential data (text, time series, audio, etc.) can be captured dependencies at different locations in the sequence to better understand the context. This is useful for tasks such as machine translation, text generation, sentiment analysis, etc.
Parallel computing	Parallel computing, which means that it can be efficiently accelerated on modern hardware. Compared to sequence models such as RNN and CNN, it is easier to perform efficient training and inference on hardware such as GPU and TPU. (Because scores can be calculated in parallel in self-attention)
Long-range dependency	Traditional recurrent neural networks (RNN) may face the problem of gradient disappearance or gradient explosion when dealing with long sequences. Long-range dependencies can be better handled because it does not need to process the input sequence sequentially.

2.2.1 Transformer Architecture

The Transformer model consists of two main components: an Encoder and a Decoder. Each of these components is made up of multiple identical layers stacked on top of each other. In the original paper, both the encoder and decoder have 6 layers, but this can be extended to any number of layers (N). The model architecture is illustarted in Figure 2.1.

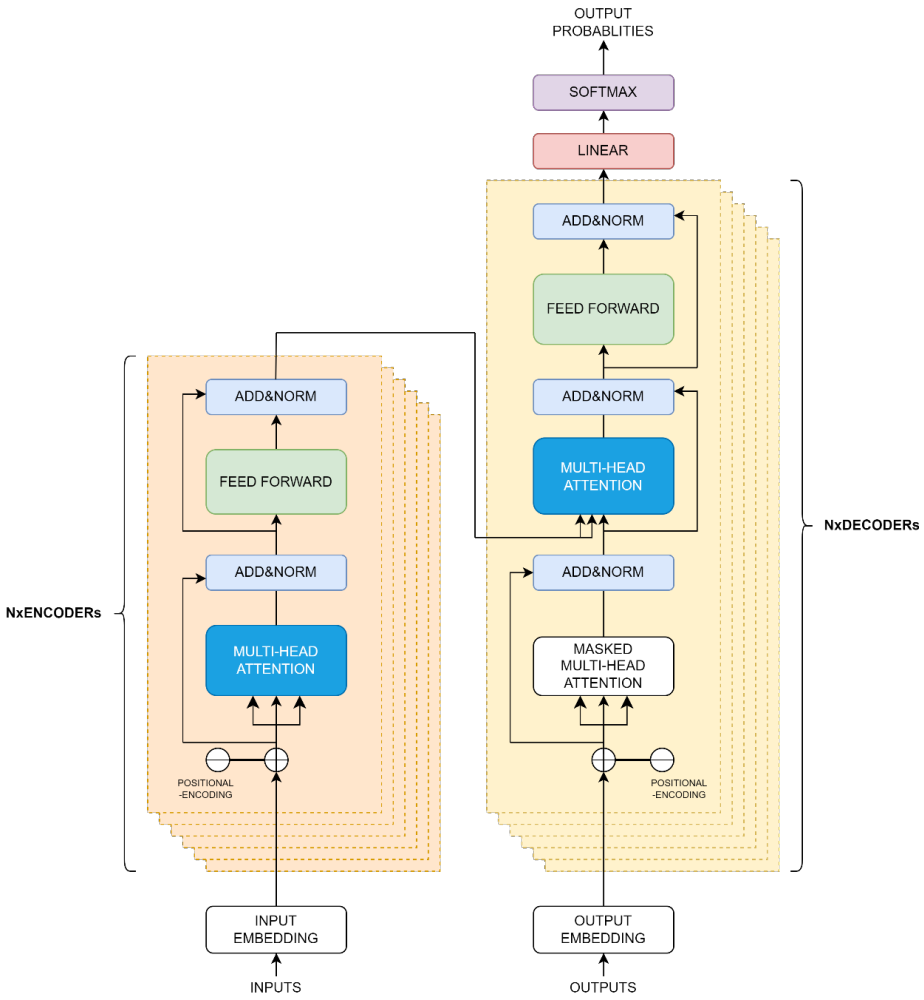


Figure 2.1: Architecture.

Before diving into the details of Transformer, there are some basic concepts required to be grasped. They are explained in the following sections.

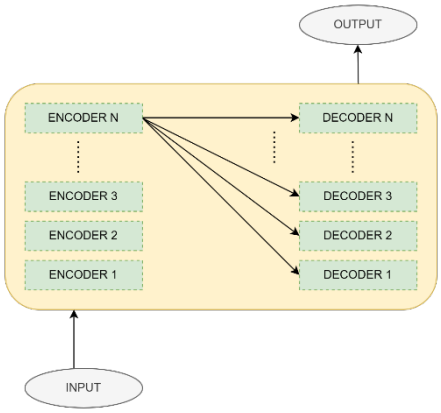


Figure 2.2: Illustrations on layers.

2.2.2 Self-attention

Self-attention helps the model process each word in the input sequence by considering all the words in the sequence, enhancing the encoding of each word. During processing, the self-attention mechanism integrates the understanding of all relevant words into the word we are processing.

Initially, consider revisiting the concept of self-attention. Given an input sequence "Awesome Transformer", wherein x1 and x2 denote the word embeddings corresponding to "Awesome" and "Transformer" respectively, which have been augmented with positional encodings. Subsequently, these augmented word vectors are transformed into the Query (Q), Key (K), and Value (V) vectors necessary for computing the Attention weights. This transformation is facilitated through the application of three distinct weight matrices, denoted as WQ, WK, and WV, which operate to project the input representations into their respective attention roles. In practical application, each sample, specifically every sequence dataset, is input as a matrix format. As depicted in the referenced figure, the matrix X comprises word vectors for the terms "Awesome" and "Transformer". This matrix undergoes transformation to yield Query (Q), Keys (K), and Values (V). Given the assumption that individual word vectors are of 512 dimensions, the initial matrix X consequently has dimensions (2,512). The transformation matrices WQ, WK, and WV, used in this process, each possess dimensions (512,64). Consequently, the resultant Query, Keys, and Values matrices all attain dimensions of (2,64). The subsequent procedure entails in order to compute the Attention weights.

Step 1: Begin by assessing the pairwise affinity between each element in the sequence. Recalling from prior discourse, the dot product methodology serves this purpose effec-

tively, entailing an element-wise multiplication of vectors from Q with their counterparts in K.

Step 2: Subsequently, normalize these affinity scores across each input sequence entry to mitigate potential gradient instability during training phases. The applied normalization scheme is rooted in scaling by the square root of the key dimension.

Step 3: The normalized scores are then transformed into a probability distribution between [0,1] using the softmax function to emphasize the relationships among words. Consequently, the transformed Score matrix becomes a probabilistic representation, still maintaining a certain size but now with values distributed between [0,1], which is indicative of probabilities.

Step 4: Finally, the computed attention probabilities are utilized to weight the corresponding Value vectors, effectuating a weighted summation that underscores significant input features. This is realized through a dot product between the SoftMax scores and V. Given V's dimensions as (2,64), the resultant matrix multiplication ((2,2) *(2,64)) yields the final output Z, a (2,64) dimensional matrix embodying the contextualized information extracted via the attention mechanism.

2.2.3 Masked Mutil-head attention

The Transformer model employs the Multi-Head Attention mechanism as a core component in both its encoder and decoder. In the decoder, this mechanism is extended with an additional masking step to control the flow of information during training. Masking serves to restrict attention to specific parts of the input by nullifying the influence of certain tokens on the models parameter updates. This ensures the model learns from appropriate context and maintains structural integrity in sequence processing. Table 2.2 offers more details.

Table 2.2: Different approaches for Mask methods

Mask Methods	Approach Detail
Padding Mask	A padding mask is used in LLM training to ignore padded tokens added for aligning sequence lengths within a batch. Sequences often vary in length, so shorter ones are padded with special tokens to match the longest. These padding tokens hold no meaning and can distort learning if not masked. The padding mask assigns 0s to padding positions and 1s to valid tokens, guiding the model to attend only to relevant input. It plays a key role in attention mechanismsespecially in encoder-decoder architecturesby maintaining focus and ensuring consistent training across batched inputs.
Sequence Mask	A sequence mask, or causal mask, is essential for autoregressive language models that predict tokens based only on preceding context. It ensures the model cannot access future tokens during training, maintaining the natural left-to-right flow of language. This mask is a triangular matrix that blocks attention to future positions by setting them to negative infinity before softmax. It's mainly used in decoder layers to prevent data leakage and ensure temporal consistency. Without it, the model could cheat by seeing answers prematurely, undermining performance in text generation and related tasks.

2.2.4 Encoder side

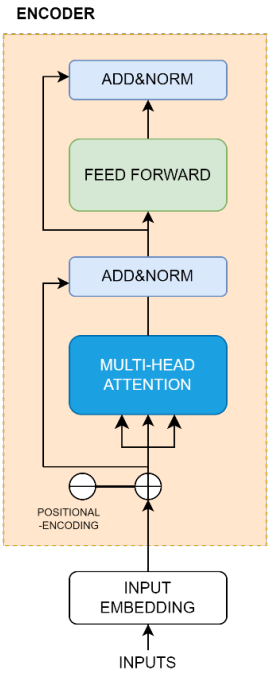


Figure 2.3: Encoder architecture.

The encoder module is essential in the Transformer model, converting input sequences, such as text sentences, into high-dimensional embeddings that capture rich context. It consists of several identical layers, each with two main parts: a multi-head self-attention system and a feedforward neural network. Residual connections and layer normalization are added to every layer to improve stability and efficiency.

The specific workflow is explained as the following:

1. Input Embedding

In the initial phase of the encoding process, each lexical unit in the input sequence undergoes transformation into a high-dimensional word embedding. This procedure initiates at the fundamental tier of the encoder architecture, wherein individual input tokens—representing either whole words or sub-word units—are mapped to numerical vectors via an embedding layer. This embedding mechanism encapsulates the semantic nuances of the tokens, thereby effectuating their conversion into fixed-length vectors of 512 dimensions, facilitating subsequent computational analyses [1].

2. Positional Encoding

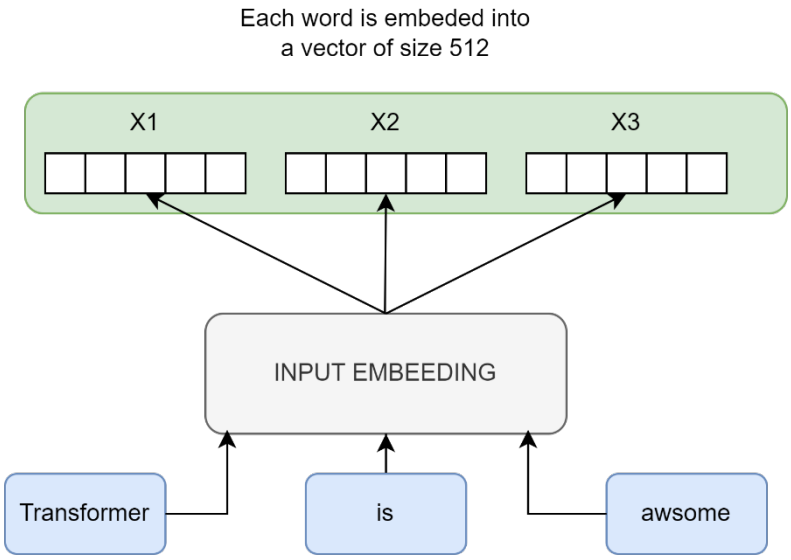


Figure 2.4: Input embedding.

Unlike Recurrent Neural Networks (RNNs), Transformer architectures do not have an inherent mechanism to capture the sequential order of tokens. To mitigate this limitation, positional encoding is incorporated into the input embeddings, thereby empowering the model with knowledge pertaining to the relative position of each token within the sequence. A novel approach was proposed by researchers, involving the utilization of sinusoidal and cosinusoidal functions to generate positional encodings. That is:

$$PE_{(pos,2i)} = \sin\left(\frac{pos}{10000^{2i/d_{model}}}\right)$$
$$PE_{(pos,2i+1)} = \cos\left(\frac{pos}{10000^{2i/d_{model}}}\right)$$

where pos is the position, i is the dimension index, and d_{model} is the embedding dimension. The positional encoding is added to the word vector so that the representation of each word includes positional information.

3. Stack of Encoder Layers

The Transformer encoder consists of multiple identical layers, typically 6 layers in the original model. Each layer has two key sublayers:

- **Multi-head Attention Mechanism**: This is the core component of the Transformer. Each word is encoded based on its relationship to other words in the sequence. The

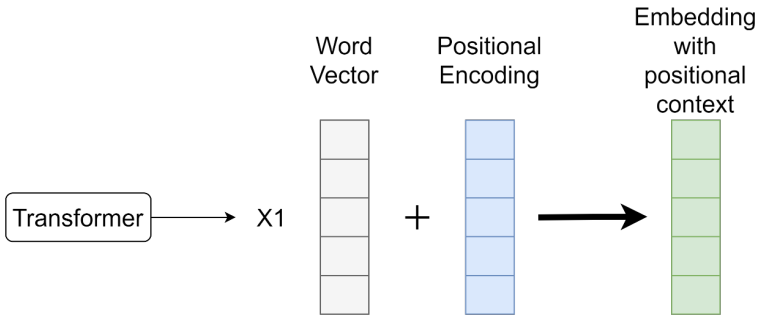


Figure 2.5: Positional encoding.

multi-head mechanism allows the model to efficiently focus on different contexts.

- **Feedforward Neural Network**: The attention output of each word is further processed by a feedforward neural network. This network computes the input for each position independently.

To improve stability and performance, residual connections are applied in each sublayer, followed by layer normalization. This ensures that information flows smoothly through the network while maintaining the integrity of the data.

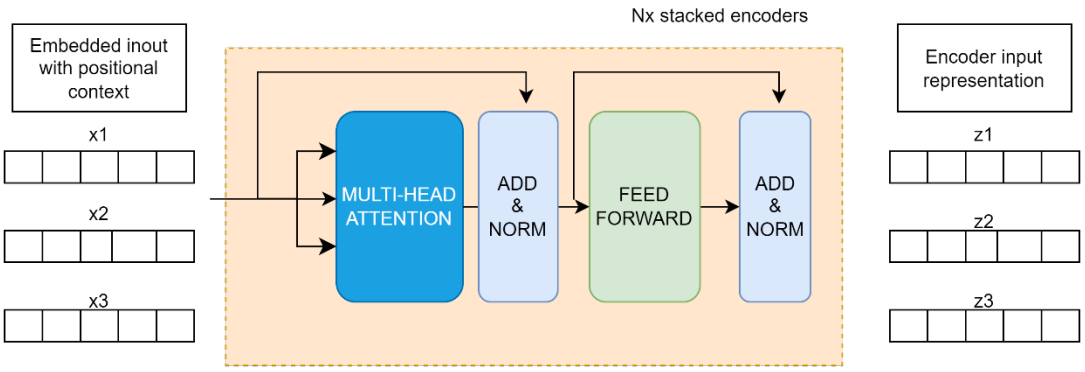


Figure 2.6: Details in encoder layer.

1) Multi-Head Attention Mechanism

The Multi-Head Attention mechanism is a key component of the Transformer architecture, enhancing the models expressiveness and enabling it to capture diverse features within the input sequence. This mechanism operates by concurrently directing attention to various segments of the input through several independent attention

heads. Subsequently, it integrates the information gathered from these multiple perspectives to produce a more comprehensive and nuanced representation.

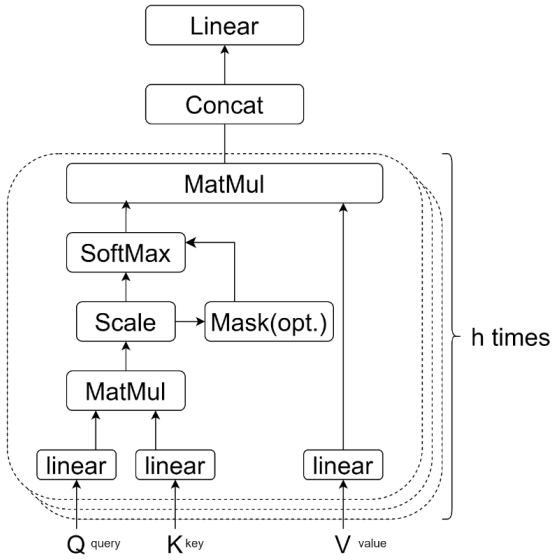


Figure 2.7: Mutil-head attention mechanism.

As in Figure 2.7, the multi-head attention process can be illustrated that linear transformation facilitates the generation of multiple attention heads: Each word vector in the input sequence undergoes three distinct linear transformations in order to produce the corresponding Query, Key, and Value vectors. The parameters for these transformations are independently defined for each attention head.

If the model has h headers, the dimensions of the query, key, and value vectors for each header are usually set to $\frac{1}{h}$ of the number of total dimensions, so that the output of all headers still maintains the original dimension after being concatenated together.

Each head independently executes the self-attention mechanism, which involves calculating the correlation between the query and the key to produce attention scores. These scores are subsequently utilized to perform a weighted summation of the values, thereby generating the output specifically to each head.

In summary, the workflow of mathematical calculation is as follows:

Step 1 Query (matrix) $*$ Keys (matrix) = Scores (matrix)

Step 2 $\frac{\text{Scores}}{\sqrt{d_k}}$ = Scaled scores where d_k is the dimension of the key vector

Step 3 SoftMax (Scaled scores) = Attention weights

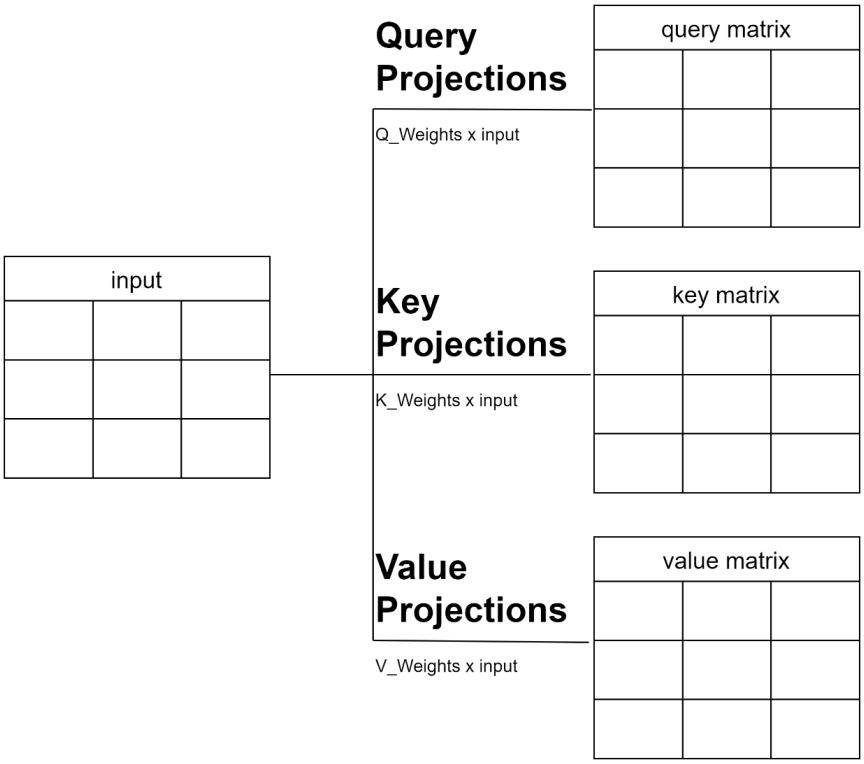


Figure 2.8: input projection.

Step 4 Attention weights $*$ values = output

The original paper proposes the formula:

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \tag{2.1}$$

2) Layer Normalization

In the encoder architecture, a normalization step is applied after each sublayer. Furthermore, residual connections are utilized, where the input of each sublayer is added to its output. This approach effectively addresses the issue of vanishing gradients, thereby enabling the successful training of deeper models by preserving critical information as it propagates through the network.

$$Output = LayerNorm(x + SubLayer(x)) \tag{2.2}$$

Where $SubLayer(x)$ is the output result after sublayer operation, such as multi-head attention or feedforward network on input.

3) Feedforward neural network

Feedforward neural networks typically consist of two fully connected layers with a nonlinear activation function. This architecture is employed to execute further nonlinear transformations on the representation at each location. Notably, these operations are conducted independently at each position, and the parameters are not shared across different locations.

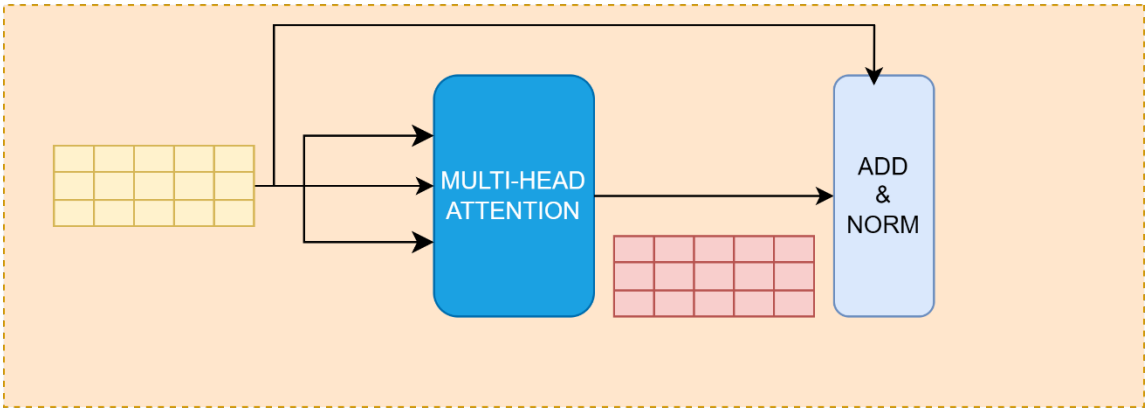


Figure 2.9: Normalization.

4. Output of the encoder

The output from the final encoder layer is a sequence of vectors, each capturing a sophisticated contextual representation of the input. These vectors subsequently serve as input to the decoder within the Transformer architecture, where they are instrumental in directing the decoding process. This precise and detailed encoding mechanism facilitates the decoder's ability to accurately concentrate on pertinent segments of the input during tasks such as translation or text generation.

2.2.5 Decoder side

The architecture of the decoder mirrors that of the encoder. Each layer within the decoder is: a multi-head self-attention mechanism designed to process the input to the decoder, a multi-head attention mechanism aimed at attending to the output from the encoder, and a feedforward neural network. The decoder architecture is:

1. Input Embedding

At the start of the decoder phase, the process mirrors that of the encoder. The input first passes through an embedding layer.

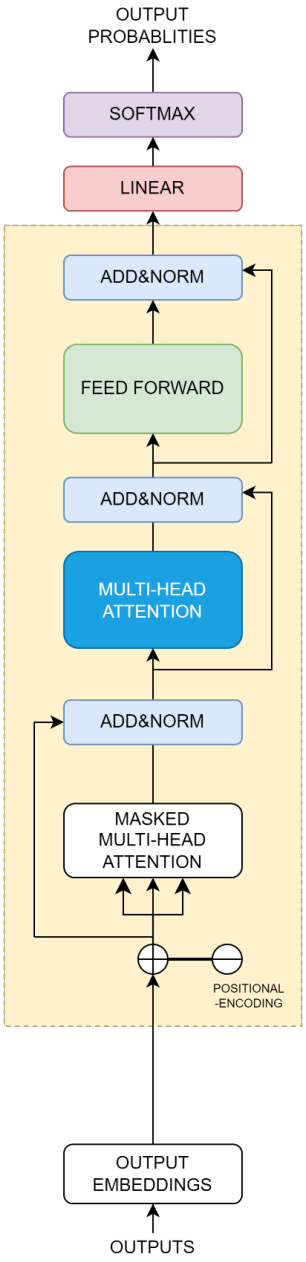


Figure 2.10: Decoder Architecture

2. Positional Encoding

Following the embedding, as in the encoder, the input passes through the positional encoding layer. This sequence is designed to produce positional embeddings. These positional embeddings are then channeled into the first multi-head attention layer of the decoder, where the attention scores specific to the decoders input are computed.

3. Stack of Decoder Layers

The decoder consists of a series of identical layers, with the original Transformer model featuring six layers. Each layer comprises three primary sub-components:

1) Masked multi-head self-attention sublayer

In the conventional multi-head attention mechanism, each Query at a given position performs a dot product with all Keys to compute the attention scores, which are then used to weight and sum the corresponding Values, producing the final output. However, in the context of the decoder, where the generation of a sequence must not be influenced by future tokens, a Masking mechanism is employed. This mechanism ensures that during the calculation of attention scores, future positions are masked, effectively setting their scores to negative infinity. Consequently, after applying the SoftMax function, these positions receive a zero weight. For instance, when computing the attention for the word “are," the masking process guarantees that the subsequent words like "you" is not in the sequence.

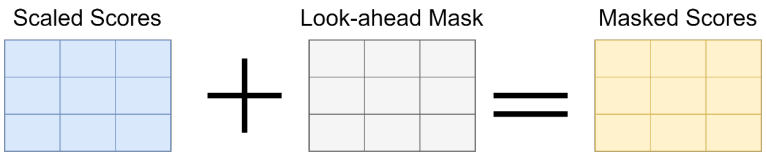


Figure 2.11: Mask process.

2) Multi-head attention sublayer

The multi-head attention sublayer is a critical component of the Transformer’s decoder, facilitating the effective focus on the input sequence (the encoder’s output) while generating the target sequence. Essentially, this sublayer compares the current decoder input (Query) with the encoder outputs (Keys and Values) to generate a new representation. This mechanism allows the decoder to dynamically emphasize different parts of the input sequence as needed, thereby integrating relevant information from the source into the generated output.

3) Feedforward neural network

Similar to the counterpart in the encoder, each decoder layer includes a fully connected feed-forward network, applied to each position separately and identically.

4) Linear Classifier and SoftMax Function for Output Probability

The journey of data through the Transformer model culminates in the final linear layer, which functions as a classifier. The size of this classifier corresponds to the total number of categories represented, specifically the number of words in the vocabulary. For instance, in

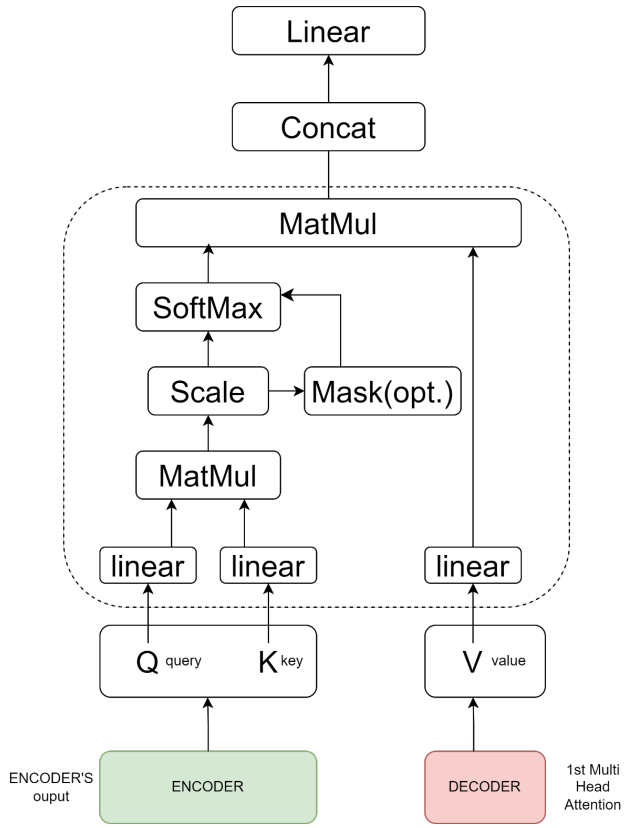


Figure 2.12: Decoder's Mutil-head attention mechanism.

a scenario with 1,000 distinct categories corresponding to 1,000 different words, the output of the classifier will be an array containing 1,000 elements. This output is then fed into the SoftMax layer, which converts it into a series of probability scores, each ranging from 0 to 1. The highest value among these probability scores is crucial, as its corresponding index indicates the model's prediction for the next word in the sequence.

4. Output of the decoder

The output of the final layer is converted into a predicted sequence, typically via a linear layer followed by a SoftMax layer to generate probabilities across the vocabulary. The decoder integrates the newly generated output into its growing list of inputs and continues the decoding process. This cycle repeats until the model predicts a specific token, signaling completion. The token with the highest predicted probability is designated as the concluding class, typically represented by an end token. Notably, the decoder is not restricted to a single layer; it can be configured with multiple layers, each building upon the input received from the encoder and its preceding layers. This multi-layered architecture

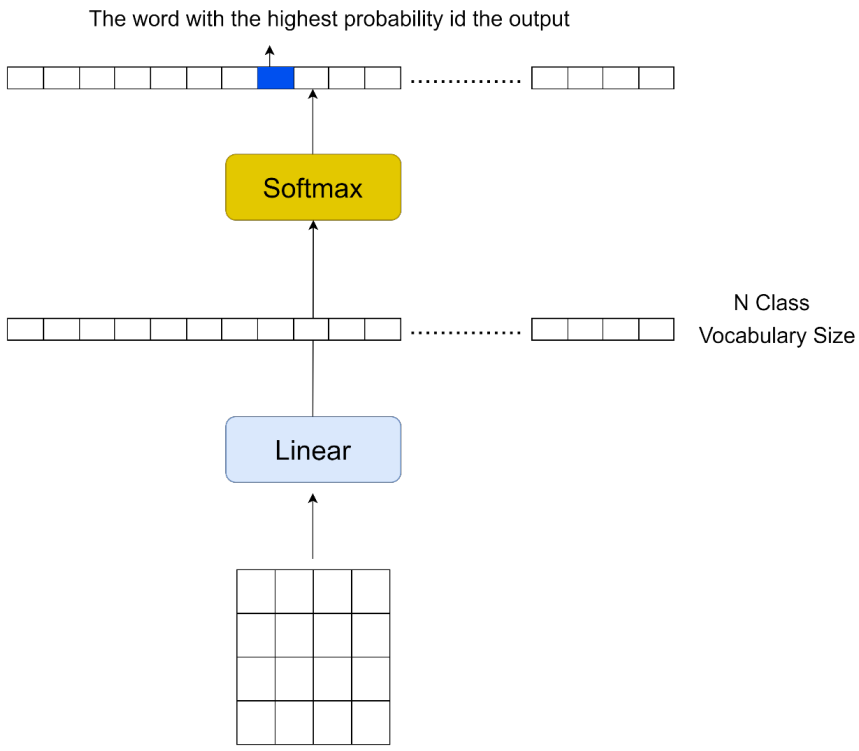


Figure 2.13: Output probability.

enables the model to diversify its focus and extract varying attention patterns across its attention heads. Such an approach significantly enhances the model's predictive capabilities by fostering a more nuanced understanding of different attention combinations.

2.3 Introduction of Bert

The introduction of Googles BERT in 2018, an open-source natural language processing (NLP) framework, marked a pivotal shift in NLP through its innovative use of bidirectional training. This approach significantly enhances the models ability to make context-dependent predictions, surpassing traditional unidirectional methods. BERT's ability to comprehend contextual information from both preceding and succeeding words enables it to excel in complex tasks such as question answering and disambiguating linguistic nuances, thereby surpassing the performance of its predecessors. At the core of its architecture are Transformer models, which enable dynamic, adaptive connections between input and output elements, enhancing BERTs efficacy and versatility in language tasks.

2.3.1 Model description

BERT is a transformer-based model pre-trained on an extensive corpus of English linguistic data through a self-supervision mechanism. The distinguishing feature of this pre-training is its unsupervised nature; the model learns from unannotated raw text corpora, leveraging publicly accessible data without requiring manual labeling. The pre-training process incorporates a dual-objective strategy, automatically generating both input sequences and corresponding labels from the text corpus, which facilitates a robust understanding of linguistic context.

2.3.2 Intended uses and limitations

You can use the raw model for either masked language modeling or next sentence prediction, but it's mostly intended to be fine-tuned on a downstream task. See the model hub to look for fine-tuned versions of a task that interests you. Note that this model is primarily aimed at being fine-tuned on tasks that use the whole sentence (potentially masked) to make decisions, such as sequence classification, token classification or question answering. For tasks such as text generation you should look at model like GPT2. Even if the training data used for this model could be characterized as fairly neutral, this model can have biased predictions. The bias will also affect all fine-tuned versions of this model [41].

2.3.3 Training data

The BERT model was pretrained on BookCorpus, a dataset consisting of 11,038 unpublished books and English Wikipedia (excluding lists, tables and headers) [41].

2.3.4 Training procedure

The training procedure can be found in the Table 2.4.

Table 2.3: Training procedure.

Preprocessing	Preprocessing details can be referred to https://huggingface.co/google-bert/bert-base-uncased.
Pretraining	Pretraining details are introduced in https://huggingface.co/google-bert/bert-base-uncased.

2.3.5 Evaluation results

When fine-tuned on downstream tasks, this model achieves the following results [41]:

Table 2.4: Glue test results.

Task	MNLI-(m/mm)	QQP	QNLI	SST-2	CoLA	STS-B	MRPC	RTE	Average
Score	84.6/83.4	71.2	90.5	93.5	52.1	85.8	88.9	66.4	79.6

2.3.6 Core architecture and functionality

Recurrent and convolutional neural networks use sequential computation to generate predictions. That is, they can predict which word will follow a sequence of given words once trained on huge datasets. In that sense, they were considered unidirectional or context-free algorithms.

In contrast, transformer-powered models like BERT, which also use the encoder-decoder architecture, are bidirectional because they predict words based on both previous and following words. This is achieved through the self-attention mechanism, a layer incorporated in both the encoder and the decoder. The goal of the attention layer is to capture the contextual relationships existing between different words in the input sentence.

Today, there are many versions of pre-trained BERT, but in the original paper, Google trained two versions of BERT: BERTbase and BERTlarge, with different neural architectures. BERTbase was developed with 12 transformer layers, 12 attention layers, and 110 million parameters, while BERTlarge used 24 transformer layers, 16 attention layers, and 340 million parameters. As expected, BERTlarge outperformed its smaller brother in accuracy tests.

Chapter 3

Sentiment Analysis by Transformer

3.1 Problem Definition

Our problem is a binary classification task, which is determining the emotion of an individual's statement that is postive or negative. The defintion is simplified and illustrated in Problem 1.

Problem 1.
Input: Context information $C = \{c1, c2, ..., cn\}$
Output: 0 or 1, where 0 for negative sentiment and 1 for postive sentiment

3.2 Alogorithm for solution

However, one LLM model does not have the "real" capability as a true human has. What a LLM can accomplish is to predict the next word that has the highest possibility to appear in the position of a sentence. Therefore, to address the problem, we can put the word [positive] and [negative] at the end of sentence and make the LLM to predict the word. Because the EATD-Corpus dataset has the labeled sentence with positive or negative, the expriment can be conducted by simply masking the last word and make the LLM predict.

Algorithm 1: Solution to address the problem

Input: Text Information

Output: Prediction Results

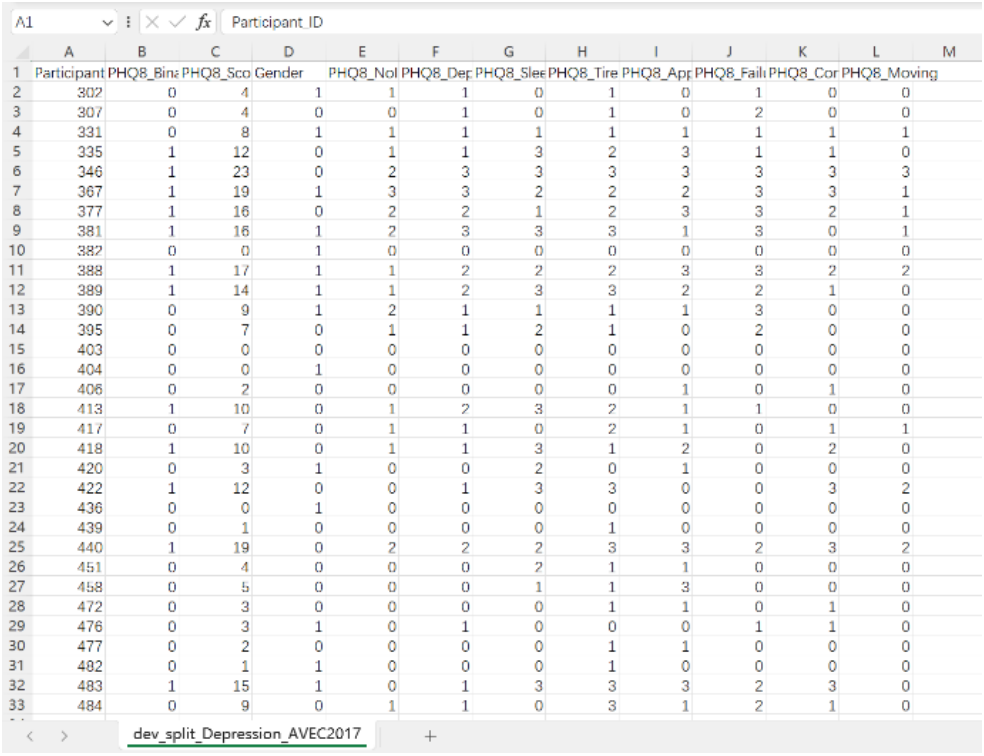
Data: Dataset $EATD-Corpus$

Step 1: Dataset Preprocessing
if *Positive text is found* **then**
    put 1 into the first column
    put the corresponding text in the second column
else if *Negative text is found* **then**
    put 0 into the first column
    put the corresponding text in the second column
Step 2: Fine-tune Stage
Step 2.1: Load a pre-trained model using AutoModelForSequenceClassification
Step 2.2: Prepare the dataset using the load_dataset function
Step 2.3: Tokenize the data with a tokenizer matching a selected model
Step 2.4: Define training arguments with Training Arguments
Step 2.5: Initialize a Trainer with your model, data, and arguments
Step 2.6: Call trainer.train() to start fine-tuning
Step 3: Prediction Stage
while *Masked word* **do**
    Predict 1 or 0 according to the highest probability score

3.3 Data Quality and Dataset Availability

There are some sources of data collected from participants of testing projects. For example, the DAIC-WOZ Dataset (https://dcapswoz.ict.usc.edu/) is from University of Southern California. However, most of them request potential users to register an account and charge for download.

There are two concerns for the data quality. Firstly, the form of data is different across various projects. For example, the DAIC-WOZ dataset, which is revealed in Fig. 3.1, only contains the test results from psychological exam and other information, such as gender, which is irrelevant to the purpose of this project. In addition, the DAIC-WOZ dataset gives many psychological test scores that is also irrevelent to the goal of this project.



	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Participant	PHQ8_Bina	PHQ8_Sco	Gender	PHQ8_Nol	PHQ8_Dep	PHQ8_Slee	PHQ8_Tire	PHQ8_App	PHQ8_Fail	PHQ8_Cor	PHQ8_Moving	
2	302	0	4	1	1	1	0	1	0	1	0	0	
3	307	0	4	0	0	1	0	1	0	2	0	0	
4	331	0	8	1	1	1	1	1	1	1	1	1	
5	335	1	12	0	1	1	3	2	3	1	1	0	
6	346	1	23	0	2	3	3	3	3	3	3	3	
7	367	1	19	1	3	3	2	2	2	3	3	1	
8	377	1	16	0	2	2	1	2	3	3	2	1	
9	381	1	16	1	2	3	3	3	1	3	0	1	
10	382	0	0	1	0	0	0	0	0	0	0	0	
11	388	1	17	1	1	2	2	2	3	3	2	2	
12	389	1	14	1	1	2	3	3	2	2	1	0	
13	390	0	9	1	2	1	1	1	1	3	0	0	
14	395	0	7	0	1	1	2	1	0	2	0	0	
15	403	0	0	0	0	0	0	0	0	0	0	0	
16	404	0	0	1	0	0	0	0	0	0	0	0	
17	406	0	2	0	0	0	0	0	1	0	1	0	
18	413	1	10	0	1	2	3	2	1	1	0	0	
19	417	0	7	0	1	1	0	2	1	0	1	1	
20	418	1	10	0	1	1	3	1	2	0	2	0	
21	420	0	3	1	0	0	2	0	1	0	0	0	
22	422	1	12	0	0	1	3	3	0	0	3	2	
23	436	0	0	1	0	0	0	0	0	0	0	0	
24	439	0	1	0	0	0	0	1	0	0	0	0	
25	440	1	19	0	2	2	2	3	3	2	3	2	
26	451	0	4	0	0	0	2	1	1	0	0	0	
27	458	0	5	0	0	0	1	1	3	0	0	0	
28	472	0	3	0	0	0	0	1	1	0	1	0	
29	476	0	3	1	0	1	0	0	0	1	1	0	
30	477	0	2	0	0	0	0	1	1	0	0	0	
31	482	0	1	1	0	0	0	1	0	0	0	0	
32	483	1	15	1	0	1	3	3	3	2	3	0	
33	484	0	9	0	1	1	0	3	1	2	1	0	

Figure 3.1: Different data form.

Secondly, the dataset adopted in this project contains information needed. It includes the translated text from original audio of participants. However, it may contain unexpected errors such as missing text from audio, which may due to the failure sampling by the electronic device. For example,in Fig. 3.2, the t_56 folder has a txt labeled with positive, which contains chinese characters, that is Architecture in English, has barely any sense because it does not convey any sentiment implication.

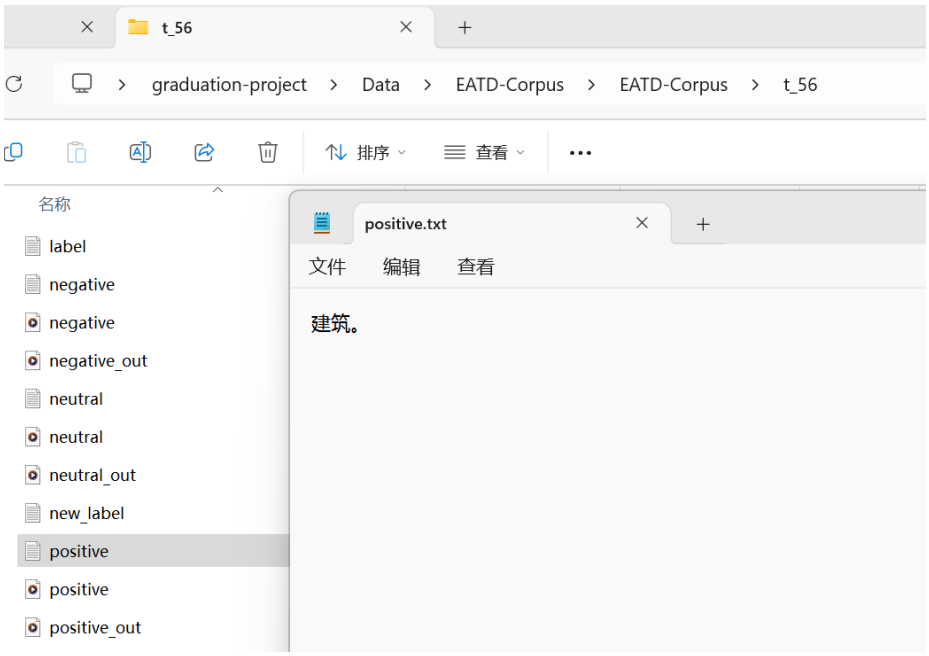


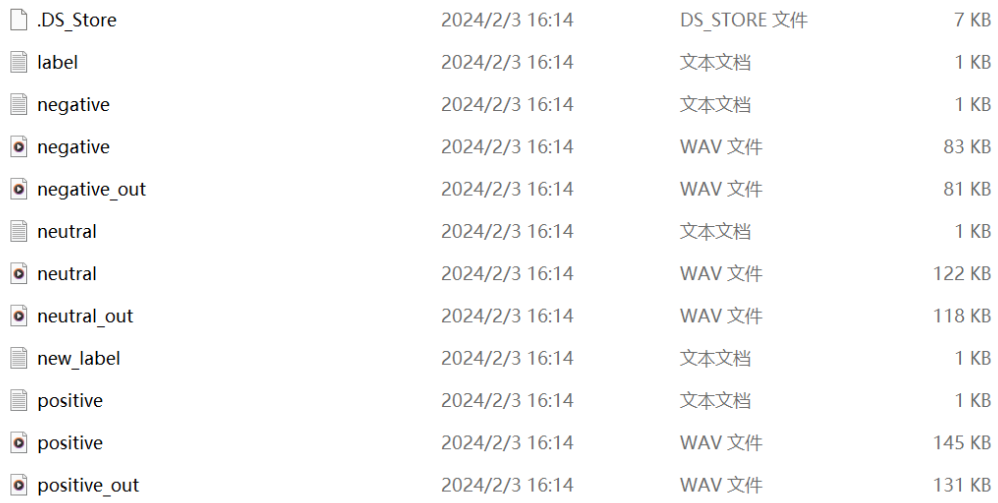
Figure 3.2: Selected data form.

3.4 Dataset Description

The dataset is available at https://huggingface.co/datasets/jimregan/eatd_corpus. It came from an experiment conducted at Tongji University in Shanghai. The source is https://github.com/speechandlanguageprocessing/ICASSP2022-Depression. The dataset contains 162 folders that are 83 folders for training and 79 folders for validation. Each folder contains positive, neutral and negative information collected from candidates. They were categorized for the original paper. In my project, I collected the positive and negative information from the original dataset for sentiment analysis.

3.5 Data Processing

All the context data is stored into txt files and it is convenient to put them into a csv file so that experiment can be done appropriately. Also, the situation is simplified because the original data has catgorized the statement of participants into three categories that are positive, neutral and negative. This project sets the goal as to find out a potential individual is in negative mood or not. Therefore, the neutral information is not within the problem domain. And the context is put into the review column and the corresponding flag 1 or 0 is put into label column in the csv file.



.DS_Store	2024/2/3 16:14	DS_STORE 文件	7 KB
label	2024/2/3 16:14	文本文档	1 KB
negative	2024/2/3 16:14	文本文档	1 KB
negative	2024/2/3 16:14	WAV 文件	83 KB
negative_out	2024/2/3 16:14	WAV 文件	81 KB
neutral	2024/2/3 16:14	文本文档	1 KB
neutral	2024/2/3 16:14	WAV 文件	122 KB
neutral_out	2024/2/3 16:14	WAV 文件	118 KB
new_label	2024/2/3 16:14	文本文档	1 KB
positive	2024/2/3 16:14	文本文档	1 KB
positive	2024/2/3 16:14	WAV 文件	145 KB
positive_out	2024/2/3 16:14	WAV 文件	131 KB

Figure 3.3: Data in each folder.

1	放松的时候喜欢听音乐、读书、休息、漫步、旅行，原因是可以放松自己，不会思考其他任何问题。
0	目前来看没有什么后悔过的事情。
1	人们都很友好，环境也很好，然后有很多可以发展的机会。
0	我有时候会失眠，然后失眠的时候就会比较焦虑，然后也比较烦躁。
1	上次应该是1月20号吧期末考试刚考完那天，因为压力特别大，考完之后就很放松，也很开心。
0	是一门课成绩出了之后觉得很不开心，因为本来以为那门课的成绩是肯定没有问题的，没想到这学期那门课的成绩确实最差。
1	我的爸爸妈妈他们在我遇到困难的过程当中会给予我最大的鼓励，努力，让我不再消极和低沉，然后从负面情绪的影响当中走出来，我觉得这是非常重要的。
0	大概两三天前，是在焦虑毕业以后，一个是博士毕业，再一个是博士毕业，该如何找工作的问题。
1	他写代码很厉害。
0	在玩桌游的时候。然后他太菜了。
1	喜欢家乡的食物。
0	最后悔的应该是我奶奶，没有好好的去多和她交流，多回家去看看她。
1	最近出去旅游了一趟，玩得还算挺开心的。
0	睡觉或者听歌。
1	我最好的朋友叫做约翰张，他是我的老板，他经常有一些奇思妙想，有各种各样的idea，他都会和我讨论，让我去实现。当我有问题的时候我都会去找他，他都会很耐心的帮我解答。有时候我的论文写得不好，他会认真帮我修改一遍、两遍三遍，甚至
1	点2点3点。4点。邮件我他，他早上7点的时候都会准时的回答我邮件里的问题，令我十分感动，所以他是我的最好的朋友。
0	我上一次感到不开心，生气。懊恼。大概是去年11月初的时候，当时我有投一次国际会议，那时候国际会议出了审稿结果，啊，审稿的结果非常不好啊，有三个审稿人，一个质疑我的贡献不足，一个人认为我的方法提升了不多，另外一个人觉得我我
1	喜欢看电影，因为看电影可以让一个人安静下来。
0	当我生气时我会尽量从令我生气的场景中抽离出来，保持冷静，然后再去回顾整个事件，来平息自己的情绪。
1	在我放松的时候我喜欢去健身房锻炼身体，因为锻炼可以让我保持良好的身材，也能让我更加的放松自己。

Figure 3.4: Extracted data from folders.

As shown in Fig. 3.3, every folder contains one individual's text and voice data. The folder also contains labels sucha s postive, neutral and negative. Then all the respective information are extracted with a simple python code and put them into one csv file that has 324 lines of text data, which is shown in Fig. 3.4. I put 1 and 0 in the first column to respectively represent postive and negative messages from text files shown in Fig. 3.3.

3.6 Methodology Overview

In general, sentiment analysis task takes advantage of fill-mask pre-trained models that are trained to understand context by covering words and let the mode to predict which word is

best-fit in the missing one. There are multiple steps to go through in the coding experiment. A workflow is presented as following:

(1) Model Selection:

- The code uses a pre-trained model from the **transformers** library. This model is already trained on a large corpus of text and can be fine-tuned for specific tasks like sequence classification.

(2) Tokenization:

- The **AutoTokenizer** is used to convert text data into tokens that the model can understand. This involves breaking down the text into smaller units and converting them into numerical representations.

(3) Data Preparation

- The dataset is loaded and filtered to ensure that only relevant data (i.e., reviews) is used.
- The dataset is split into training and testing sets to evaluate the models performance. 80% of the dataset is prepared for training while the remaining is for performance evaluation.

(4) Tokenization Function:

- The **process_function** tokenizes the text data and aligns it with the corresponding labels (i.e., the target classes for classification).

(5) Evaluation Metric:

- The **accuracy_metric** is loaded to evaluate the model's performance. The compute metrics function calculates the accuracy of the models predictions.

(6) Training Arguments:

- The **TrainingArguments** define the parameters for training, such as learning rate, batch size, number of epochs, and evaluation strategy.

(7) Trainer:

- The **Trainer** class from the **transformers** library is used to handle the training and evaluation process. It takes care of the training loop, evaluation, and saving the best model based on the specified metric (accuracy in this case).

(8) Training and Evaluation:

- The **trainer.train()** method fine-tunes the pre-trained model on the provided dataset.
- The **trainer.evaluate()** method evaluates the models performance on the test set.

By combining these components, the code can fine-tune a pre-trained model for the specific task of binary classification.

3.7 Training Arguments

Table 3.1: Details about Training arguments

Arguments name	value
learning_rate	2e-5
per_device_train_batch_size	32
per_device_eval_batch_size	128
num_train_epochs	5
weight_decay	0.01
output_dir	model_for_seqclassification
logging_steps	10
evaluation_strategy	epoch
save_strategy	epoch
load_best_model_at_end	True
metric_for_best_model	accuracy
fp16	True

Table 3.1 shows the traing arguments apoted for this project. The training aruguments can be changed. For example, num_train_ epochs could be less or more than 5 times. But due to the limit of computing resources, trials can not be conducted as many times as expected.

3.8 Model Performance

Table 3.2 and 3.3 present the accuracy of each model over ten training epochs. The overall performance of the models varied significantly, largely influenced by their pre-training corpora and architectural choices in relation to the nature of the downstream task, which is sentiment analysis. A striking observation is the superior performance of models specifically pre-trained on Chinese language data ("hflchinese-roberta-wwm-ext", "bert-base-chinese") and a strong multilingual model ("xlm-roberta-base"). This strongly suggests the task involves processing Chinese text, where specialized or relevantly trained models possess a distinct advantage by aligning better with the linguistic characteristics of the input.

Table 3.2: Accuracy Results

Model Name	Epochs and Results				
	1	2	3	4	5
Distilbert/distilbert-base-uncased	0.369231	0.400000	0.523077	0.569231	0.738462
Google-bert/bert-base-uncased	0.738462	0.461538	0.723077	0.661538	0.661538
FacebookAI/roberta-base	0.369231	0.369231	0.446154	0.400000	0.723077
FacebookAI/xlm-roberta-base	0.846154	0.892308	0.923077	0.923077	0.969231
FacebookAI/xlm-roberta-large	0.584615	0.446154	0.400000	0.461538	0.430769
Google-bert/bert-base-multilingual-cased	0.430769	0.630769	0.769231	0.830769	0.815385
Hfl/chinese-roberta-wwm-ext	0.892308	0.923077	0.953846	0.969231	0.969231
Google-bert/bert-base-chinese	0.861538	0.907692	0.938462	0.907692	0.938462

Table 3.3: Accuracy Results(Cont.)

Model Name	Epochs and Results				
	6	7	8	9	10
Distilbert/distilbert-base-uncased	0.738462	0.738462	0.738462	0.738462	0.738462
Google-bert/bert-base-uncased	0.661538	0.784615	0.723077	0.738462	0.738462
FacebookAI/roberta-base	0.769231	0.692308	0.753846	0.738462	0.769231
FacebookAI/xlm-roberta-base	0.923077	0.953846	0.938462	0.938462	0.938462
FacebookAI/xlm-roberta-large	0.369231	0.476923	0.338462	0.569231	0.507692
Google-bert/bert-base-multilingual-cased	0.815385	0.830769	0.846154	0.846154	0.830769
Hfl/chinese-roberta-wwm-ext	0.984615	0.984615	0.984615	0.953846	0.953846
Google-bert/bert-base-chinese	0.938462	0.953846	0.953846	0.953846	0.953846

Models like hfl/chinese-roberta-wwm-ext (a RoBERTa variant with Whole Word Masking tailored for Chinese) and xlm-roberta-base demonstrated not only high accuracy but also efficient learning, achieving low validation losses. This highlights the comprehensive multilingual coverage when tackling specific language tasks.

Conversely, general-purpose models pre-trained primarily on English (e.g., bert-base-uncased,roberta-base) exhibited moderate performance, adequate but not competitive with the specialized models. The bert-base-multilingual-cased model showed better adaptation than the English-only models, reinforcing the value of multilingual pre-training, yet it did not reach the efficacy of the Chinese-specific or the top-performing xlm-roberta-base model. A significant anomaly was the xlm-roberta-large model, which, despite its larger capacity, severely underperformed. It emphasizes that increased model size is not an advantage.

Finally, the performance of distilbert-base-uncased illustrates the classic trade-off between computational efficiency and predictive power. As a distilled model, it achieved the fastest evaluation runtime and highest throughput, albeit with lower accuracy than the top-tier models. This positions it as a viable option when resource constraints or latency requirements are paramount.

More details can be found in the Appendix.

3.9 Discussion of Training Results of 8 Models

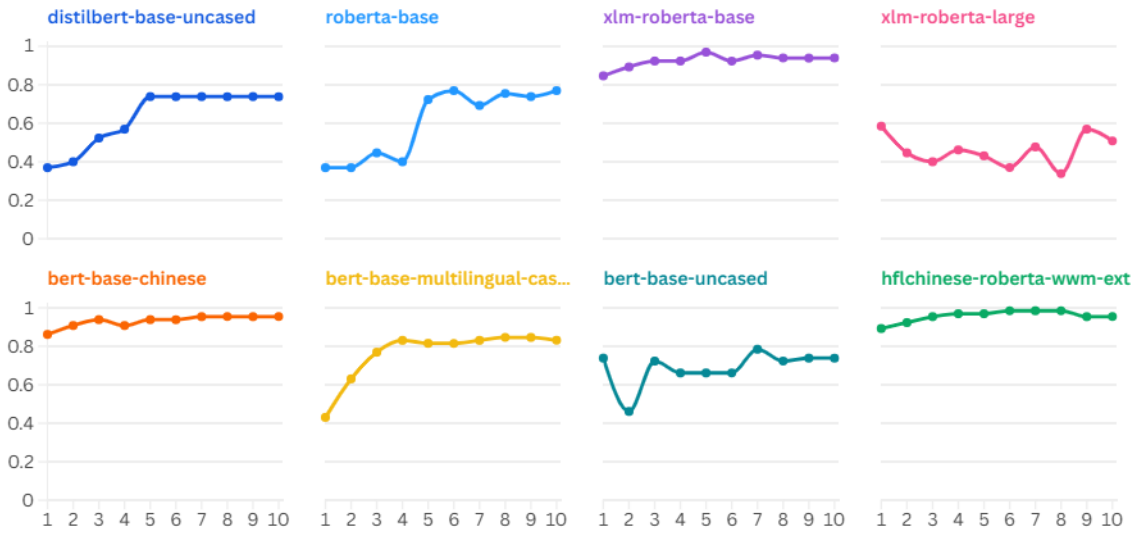


Figure 3.5: Training results.

The training phase, conducted over 10 epochs, provided insights into each model's learning behavior, convergence speed, and potential for generalization or over-fitting. Please refer to Appendix for the details.

- hflchinese-roberta-wwm-ext: This model demonstrated exceptional learning from the outset. Starting with a strong baseline accuracy (0.892308) and a validation loss of 0.426358 in epoch 1.

- xlm-roberta-base: It began with an impressive accuracy of 0.846154 and validation loss of 0.623017. Training loss dropped significantly to 0.057600 by epoch 10. The model achieved its best validation loss (0.159480) and accuracy (0.969231) at epoch 5. Similar to "hflchinese-roberta-wwm-ext", after epoch 5, the validation loss showed a tendency to increase (e.g., 0.272580 by epoch 10).

- bert-base-chinese: Showed strong learning capabilities, particularly in fitting the training data. Initial accuracy was high (0.861538) with a validation loss of 0.460596. Validation loss stabilized around 0.20 after epoch 6 (e.g., 0.203522 at epoch 8, 0.206050 at epoch 10), while accuracy peaked at 0.953846 from epoch 7 onwards.

- bert-base-multilingual-cased: This model displayed a steady learning pace. It started with a modest accuracy of 0.430769 and validation loss of 0.717619. Training loss significantly decreased to 0.012800 by epoch 10. Validation loss consistently improved, reaching its minimum of 0.352257 at epoch 8, where accuracy also peaked at 0.846154.

- bert-base-uncased: Demonstrated moderate but somewhat slow learning. It began with an accuracy of 0.738462 and validation loss of 0.675924. Training loss reduced to 0.455400 by epoch 10. Validation loss generally decreased, reaching its minimum of 0.473919 at epoch 10, with accuracy peaking at 0.784615 at epoch 7 but slightly declining to 0.738462 by epoch 10.

- distilbert-base-uncased: As a lighter model, it showed consistent improvement from a lower starting point (accuracy 0.369231, validation loss 0.724159). Training loss decreased to 0.323200. Validation accuracy at 0.738462 from epoch 5 through epoch 9, dropping slightly in the reported final evaluation which uses the model from epoch 10 (0.738462 was the accuracy for epoch 10's validation run). The validation loss reached its minimum of 0.532546 at epoch 9.

- roberta-base: This model started with an accuracy of 0.369231 and validation loss of 0.714438. Training loss was reduced to 0.325200 by epoch 10. A notable event was a spike in validation loss to 0.877224 and a dip in accuracy to 0.400000 at epoch 4, after which it recovered. Accuracy peaked at 0.769231 at epochs 6 and 10 (validation run), with the validation loss for epoch 6 being 0.585097.

- xlm-roberta-large: This model struggled significantly throughout the training process. It started with a validation accuracy of 0.584615 and loss of 0.693299. Unlike other models, its training loss remained exceptionally high, only decreasing from 0.71 to 0.681400 by epoch 10. Validation loss and accuracy showed no consistent improvement, with accuracy often below its initial value (e.g., 0.338462 at epoch 8).

3.10 Evaluation Results of 8 Models

The final evaluation metrics, captured after 10 epochs of training, provide a definitive snapshot of each model's performance on unseen data and its computational efficiency. Please refer to Appendix for the details.

Figure 3.6 reveals that these results clearly establish "hflchinese-roberta-wwm-ext" as the top-performing model, achieving near-perfect accuracy and the lowest loss. "xlm-roberta-base" and "bert-base-chinese" also deliver excellent results, forming a distinct top tier. Distilbert-base-uncased stands out for its superior speed and efficiency, processing data significantly faster than all other models. The "base" sized models generally cluster together in terms of runtime and throughput, with xlm-roberta-large being a considerable outlier due to its larger architecture, resulting in the longest runtime and lowest processing speed. This highlights the computational cost associated with larger models, which, in this particular instance, did not translate to improved predictive performance.

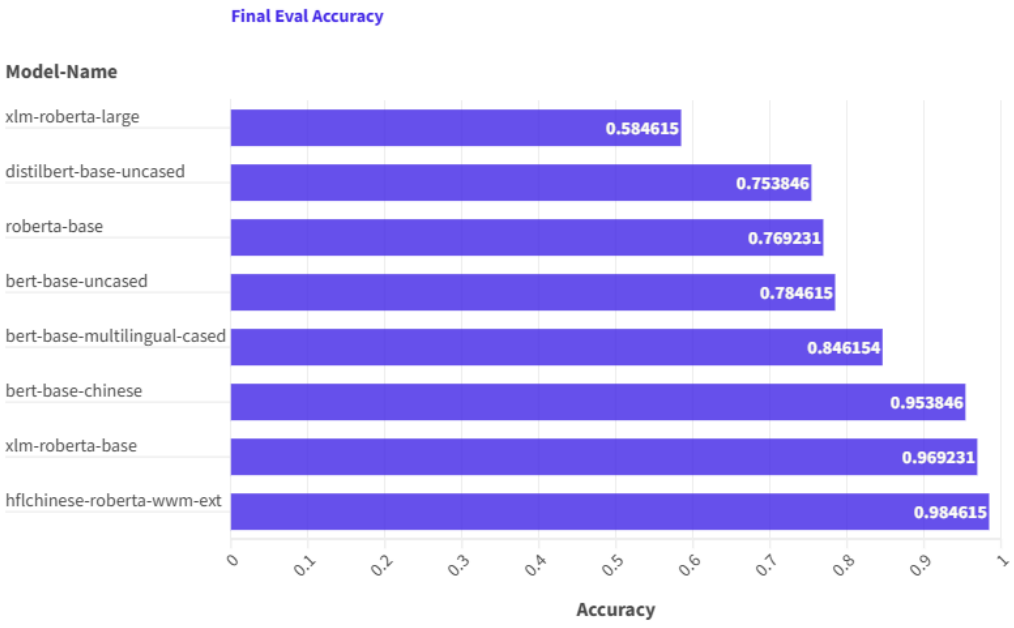


Figure 3.6: Evaluation results of 8 models.

3.11 Project Constraints

Due to the essence of fill-mask models, there are some constraints in this project output. Further explanations are in Table 3.3.

The accuracy issue is noticeable in the results. Because of the small size of test dataset, the evaluation about accuracy is remarkably close to the best result of training model. This issue can be resolved by a different larger size dataset, in which way, the accuracy for the test dataset will be slightly different from the best result of training model.

Table 3.4: Constraints summary

Ambiguity and Irrelevance	They may not always be suitable for tasks requiring high precision or understanding of the input text. It has been observed that ambiguous or irrelevant information may be generated.
Data Biases	Like other language models, fill-mask models can perpetuate harmful stereotypes and biases present in the data they were trained on. The model learns from the trained data, but if that data contains biases, the model will likely reflect those biases in its predictions.
Lack of Specificity	Fill-mask models are designed to predict the most-likely word to fill a given mask, based on the context provided. However, in many real-world scenarios, there may be multiple valid words that could fill the mask, and the model might not always choose the one that is most appropriate for the specific context.
Syntax and Grammar	Applying transformations to text datasets is not straightforward because they can disrupt syntax, grammatical correctness, and even alter the meaning of the original text.

Chapter 4

Conclusions and Future works

This project has explored multiple pre-trained models that demonstrate a range of accuracy in the sentiment analysis of conversational texts in Chinese. With the same training arguments and datasets, the lowest accuracy can be just around 60% while the best result can be over 90%. To be specific, the best result regarding accuracy is about 98.46% with the pre-trained model Hfl/chinese-roberta-wwm-ext. Although there are some practical limitations and project constraints, the accuracy statistics are encouraging, and it offers insight into the future works.

Given the current circumstances, an AI agent for detecting sentiment can be developed. Not only it can collect voice to perform a sentiment analysis, but also it can scan an individuals face based on the computer vison techniques. To summarize, it is possible to construct a sophisticated mutilmodal dataset and develop an LLM agent. In addition, to further digest the value of the dataset, it is promising to develop an AI agent that can generate response to any questions that a participant may ask in the research domain of Medical Question Answering.

Appendix A

Recorded Experiment Results with respective model name

A.1 Distilbert/distilbert-base-uncased

[90/90 01:26, Epoch 10/10]

Epoch	Training Loss	Validation Loss	Accuracy
1	No log	0.724159	0.369231
2	0.696900	0.723693	0.400000
3	0.665200	0.706423	0.523077
4	0.657600	0.681735	0.569231
5	0.638700	0.614558	0.738462
6	0.544000	0.557963	0.738462
7	0.495200	0.603405	0.738462
8	0.413600	0.550448	0.738462
9	0.385900	0.532546	0.753846
10	0.323200	0.549021	0.738462

[1/1 : < :]

```
{'eval_loss': 0.532545804977417,
 'eval_accuracy': 0.7538461538461538,
 'eval_runtime': 0.0483,
 'eval_samples_per_second': 1344.481,
 'eval_steps_per_second': 20.684,
 'epoch': 10.0}
```

A.2 FacebookAI/roberta-base-eval

[90/90 02:57, Epoch 10/10]

Epoch	Training Loss	Validation Loss	Accuracy
1	No log	0.714438	0.369231
2	0.679000	0.768757	0.369231
3	0.657400	0.691132	0.446154
4	0.689800	0.877224	0.400000
5	0.672500	0.590796	0.723077
6	0.588400	0.585097	0.769231
7	0.485000	0.636602	0.692308
8	0.406000	0.574343	0.753846
9	0.374200	0.634921	0.738462
10	0.325200	0.603292	0.769231

[1/1 : < :]

```
{'eval_loss': 0.585097074508667,
 'eval_accuracy': 0.7692307692307693,
 'eval_runtime': 0.0794,
 'eval_samples_per_second': 818.892,
 'eval_steps_per_second': 12.598,
 'epoch': 10.0}
```

A.3 FacebookAI/xlm-roberta-base

[90/90 07:32, Epoch 10/10]

Epoch	Training Loss	Validation Loss	Accuracy
1	No log	0.623017	0.846154
2	0.671600	0.471606	0.892308
3	0.572500	0.296113	0.923077
4	0.359600	0.219454	0.923077
5	0.154200	0.159480	0.969231
6	0.140300	0.305770	0.923077
7	0.065200	0.204412	0.953846
8	0.090400	0.271379	0.938462
9	0.090500	0.258956	0.938462
10	0.057600	0.272580	0.938462

[1/1 : < :]

```
{'eval_loss': 0.15947960317134857,
 'eval_accuracy': 0.9692307692307692,
 'eval_runtime': 0.0795,
 'eval_samples_per_second': 818.091,
 'eval_steps_per_second': 12.586,
 'epoch': 10.0}
```

A.4 FacebookAI/xlm-roberta-large

[90/90 15:58, Epoch 10/10]

Epoch	Training Loss	Validation Loss	Accuracy
1	No log	0.693299	0.584615
2	0.715300	0.708286	0.446154
3	0.686300	0.720775	0.400000
4	0.716500	0.697266	0.461538
5	0.701800	0.730172	0.430769
6	0.681200	0.745587	0.369231
7	0.723000	0.714746	0.476923
8	0.688500	0.710104	0.338462
9	0.700200	0.686839	0.569231
10	0.681400	0.679872	0.507692

[1/1 : < :]

```
{'eval_loss': 0.6932992935180664,
 'eval_accuracy': 0.5846153846153846,
 'eval_runtime': 0.2185,
 'eval_samples_per_second': 297.52,
 'eval_steps_per_second': 4.577,
 'epoch': 10.0}
```

A.5 google-bert/bert-base-chinese

[90/90 02:06, Epoch 10/10]

Epoch	Training Loss	Validation Loss	Accuracy
1	No log	0.460596	0.861538
2	0.597200	0.245093	0.907692
3	0.285600	0.169836	0.938462
4	0.118700	0.174151	0.907692
5	0.065400	0.180907	0.938462
6	0.027200	0.214658	0.938462
7	0.012800	0.205452	0.953846
8	0.007600	0.203522	0.953846
9	0.006700	0.204050	0.953846
10	0.005100	0.206050	0.953846

[1/1 : < :]

```
{'eval_loss': 0.20545195043087006,
 'eval_accuracy': 0.9538461538461539,
 'eval_runtime': 0.085,
 'eval_samples_per_second': 764.987,
 'eval_steps_per_second': 11.769,
 'epoch': 10.0}
```

A.6 google-bert/bert-base-multilingual-cased

[90/90 03:50, Epoch 10/10]

Epoch	Training Loss	Validation Loss	Accuracy
1	No log	0.717619	0.430769
2	0.709400	0.634619	0.630769
3	0.613100	0.491957	0.769231
4	0.423500	0.381147	0.830769
5	0.253900	0.438335	0.815385
6	0.106200	0.393300	0.815385
7	0.074100	0.369595	0.830769
8	0.026200	0.352257	0.846154
9	0.023800	0.366508	0.846154
10	0.012800	0.357884	0.830769

[1/1 : < :]

```
{'eval_loss': 0.35225674510002136,
 'eval_accuracy': 0.8461538461538461,
 'eval_runtime': 0.0799,
 'eval_samples_per_second': 813.427,
 'eval_steps_per_second': 12.514,
 'epoch': 10.0}
```

A.7 google-bert/bert-base-uncased

[90/90 02:28, Epoch 10/10]

Epoch	Training Loss	Validation Loss	Accuracy
1	No log	0.675924	0.738462
2	0.697000	0.685442	0.461538
3	0.698600	0.621995	0.723077
4	0.673100	0.602990	0.661538
5	0.621500	0.657067	0.661538
6	0.633600	0.590807	0.661538
7	0.546200	0.508162	0.784615
8	0.525600	0.502246	0.723077
9	0.486900	0.474987	0.738462
10	0.455400	0.473919	0.738462

[1/1 : < :]

```
{'eval_loss': 0.5081617832183838,
 'eval_accuracy': 0.7846153846153846,
 'eval_runtime': 0.0935,
 'eval_samples_per_second': 695.536,
 'eval_steps_per_second': 10.701,
 'epoch': 10.0}
```

A.8 hfl/chinese-roberta-wwm-ext

[90/90 01:37, Epoch 10/10]

Epoch	Training Loss	Validation Loss	Accuracy
1	No log	0.426358	0.892308
2	0.569200	0.250230	0.923077
3	0.302200	0.175016	0.953846
4	0.158200	0.118355	0.969231
5	0.075800	0.111239	0.969231
6	0.034300	0.109685	0.984615
7	0.019600	0.102801	0.984615
8	0.012200	0.106846	0.984615
9	0.010200	0.113611	0.953846
10	0.008300	0.114567	0.953846

[1/1 : < :]

```
{'eval_loss': 0.10968469828367233,
 'eval_accuracy': 0.9846153846153847,
 'eval_runtime': 0.0804,
 'eval_samples_per_second': 808.273,
 'eval_steps_per_second': 12.435,
 'epoch': 10.0}
```

Appendix B

Code

```
!pip install transformers==4.21.0 datasets evaluate
!wget https://raw.githubusercontent.com/dumingyu/EATD-Corpus-dataset
/main/all.csv
import evaluate
from datasets import load_dataset
from transformers import AutoModelForSequenceClassification
from transformers import AutoTokenizer
from transformers import DataCollatorWithPadding
from transformers import TrainingArguments
from transformers import Trainer
# 'GBK' to 'UTF-8'
with open('all.csv', 'r', encoding='GBK') as file:
content = file.read()
with open('all.csv', 'w', encoding='UTF-8') as file:
file.write(content)
# load
dataset = load_dataset('csv', data_files='all.csv')
dataset = dataset.filter(lambda x: x["review"] is not None)
dataset
```

```
datasets = dataset["train"].train_test_split(0.2)
datasets
#model_name = "hfl/chinese-roberta-wwm-ext"
#model_name = "google-bert/bert-base-chinese"
#model_name = "google-bert/bert-base-multilingual-cased"
#model_name = "google-bert/bert-base-uncased"
model_name = "FacebookAI/roberta-base"
#model_name = "FacebookAI/xlm-roberta-base"
#model_name = "FacebookAI/xlm-roberta-large"
#model_name = "distilbert/distilbert-base-uncased"
tokenizer = AutoTokenizer.from_pretrained(model_name, num_labels=2)
tokenizer
def process_function(examples):
tokenized_examples = tokenizer(examples["review"], max_length=64,
    truncation=True)
tokenized_examples["labels"] = examples["label"]
return tokenized_examples
tokenized_datasets = datasets.map(process_function, batched=True)
tokenized_datasets
accuracy_metric = evaluate.load("accuracy")
accuracy_metric
def compute_metrics(eval_pred):
predictions, labels = eval_pred
predictions = predictions.argmax(axis=-1)
return accuracy_metric.compute(predictions=predictions,
    references=labels)
model = AutoModelForSequenceClassification.from_pretrained(model_name,
    num_labels=2)
args = TrainingArguments(
```

```
learning_rate=2e-5,
per_device_train_batch_size=32,
per_device_eval_batch_size=128,
num_train_epochs=10,
weight_decay=0.01,
output_dir="model_for_seqclassification",
logging_steps=10,
evaluation_strategy = "epoch",
save_strategy = "epoch",
load_best_model_at_end=True,
metric_for_best_model="accuracy",
fp16=True,
)
trainer = Trainer(
model,
args=args,
train_dataset=tokenized_datasets["train"],
eval_dataset=tokenized_datasets["test"],
tokenizer=tokenizer,
compute_metrics=compute_metrics,
data_collator=DataCollatorWithPadding(tokenizer=tokenizer)
)
trainer.train()
trainer.evaluate()
```

Bibliography

[1] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, (Red Hook, NY, USA), pp. 6000–6010, Curran Associates Inc., 2017.

[2] Y. M. Cho, S. Rai, L. Ungar, J. Sedoc, and S. Guntuku, "An integrative survey on mental health conversational agents to bridge computer science and medical perspectives," in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing* (H. Bouamor, J. Pino, and K. Bali, eds.), (Singapore), pp. 11346–11369, Association for Computational Linguistics, 2023.

[3] G. Dosovitsky, B. S. Pineda, N. C. Jacobson, C. Chang, and E. L. Bunge, "Artificial intelligence chatbot for depression: Descriptive study of usage," *JMIR Formative Research*, vol. 4, no. 11, p. e17065, 2020.

[4] E. M. Boucher, N. R. Harake, H. E. Ward, S. E. Stoeckl, J. Vargas, J. Minkel, A. C. Parks, and R. Zilca, "Artificially intelligent chatbots in digital mental health interventions: A review," *Expert Review of Medical Devices*, vol. 18, no. sup1, pp. 37–49, 2021.

[5] A. N. Vaidyam, D. Linggonegoro, and J. Torous, "Changes to the psychiatric chatbot landscape: A systematic review of conversational agents in serious mental illness: Changements du paysage psychiatrique des chatbots: Une revue systématique des agents conversationnels dans la maladie mentale sérieuse," *The Canadian Journal of Psychiatry*, vol. 66, no. 4, pp. 339–348, 2021.

[6] A. Ahmed, A. Hassan, S. Aziz, A. A. Abd-alrazaq, N. Ali, M. Alzubaidi, D. Al-Thani, B. Elhusein, M. A. Siddig, M. Ahmed, and M. Househ, "Chatbot features for

anxiety and depression: A scoping review," *Health Informatics Journal*, vol. 29, no. 1, p. 14604582221146719, 2023.

[7] A. A. Abd-alrazaq, M. Alajlani, A. A. Alalwan, B. M. Bewick, P. Gardner, and M. Househ, "An overview of the features of chatbots in mental health: A scoping review," *International Journal of Medical Informatics*, vol. 132, p. 103978, 2019.

[8] E. Bendig, B. Erb, L. Schulze-Thuesing, and H. Baumeister, "The next generation: Chatbots in clinical psychology and psychotherapy to foster mental health – a scoping review," *Verhaltenstherapie*, vol. 32, no. Suppl. 1, pp. 64–76, 2019.

[9] J. J. Prochaska, E. A. Vogel, A. Chieng, M. Kendra, M. Baiocchi, S. Pajarito, and A. Robinson, "A therapeutic relational agent for reducing problematic substance use (woebot): Development and usability study," *Journal of Medical Internet Research*, vol. 23, no. 3, p. e24850, 2021.

[10] K. K. Fitzpatrick, A. Darcy, and M. Vierhile, "Delivering cognitive behavior therapy to young adults with symptoms of depression and anxiety using a fully automated conversational agent (woebot): A randomized controlled trial," *JMIR Mental Health*, vol. 4, no. 2, p. e7785, 2017.

[11] R. Fulmer, A. Joerin, B. Gentile, L. Lakerink, and M. Rauws, "Using psychological artificial intelligence (tess) to relieve symptoms of depression and anxiety: Randomized controlled trial," *JMIR Mental Health*, vol. 5, no. 4, p. e9782, 2018.

[12] B. Inkster, S. Sarda, and V. Subramanian, "An empathy-driven, conversational artificial intelligence agent (wysa) for digital mental well-being: Real-world data evaluation mixed-methods study," *JMIR mHealth and uHealth*, vol. 6, no. 11, p. e12106, 2018.

[13] M. R. Ali, S. Z. Razavi, R. Langevin, A. Al Mamun, B. Kane, R. Rawassizadeh, L. K. Schubert, and E. Hoque, "A virtual conversational agent for teens with autism spectrum disorder: Experimental results and design lessons," in *Proceedings of the 20th ACM International Conference on Intelligent Virtual Agents*, IVA '20, (New York, NY, USA), pp. 1–8, Association for Computing Machinery, 2020.

[14] C. Sweeney, C. Potts, E. Ennis, R. Bond, M. D. Mulvenna, S. O'neill, M. Malcolm, L. Kuosmanen, C. Kostenius, A. Vakaloudis, G. Mcconvey, R. Turkington, D. Hanna,

H. Nieminen, A.-K. Vartiainen, A. Robertson, and M. F. Mctear, "Can chatbots help support a person's mental health? perceptions and views from mental healthcare professionals and experts," *ACM Trans. Comput. Healthcare*, vol. 2, no. 3, pp. 25:1–25:15, 2021.

[15] A. N. Vaidyam, H. Wisniewski, J. D. Halamka, M. S. Kashavan, and J. B. Torous, "Chatbots and conversational agents in mental health: A review of the psychiatric landscape," *The Canadian Journal of Psychiatry*, vol. 64, no. 7, pp. 456–464, 2019.

[16] T. Koulouri, R. D. Macredie, and D. Olakitan, "Chatbots to support young adults' mental health: An exploratory study of acceptability," *ACM Trans. Interact. Intell. Syst.*, vol. 12, no. 2, pp. 11:1–11:39, 2022.

[17] H. Yang, Y. Zhao, Y. Wu, S. Wang, T. Zheng, H. Zhang, Z. Ma, W. Che, and B. Qin, "Large language models meet text-centric multimodal sentiment analysis: A survey," 2024.

[18] J. Li, A. Dada, B. Puladi, J. Kleesiek, and J. Egger, "Chatgpt in healthcare: A taxonomy and systematic review," *Computer Methods and Programs in Biomedicine*, vol. 245, p. 108013, 2024.

[19] A. Rao, M. Pang, J. Kim, M. Kamineni, W. Lie, A. K. Prasad, A. Landman, K. Dreyer, and M. D. Succi, "Assessing the utility of chatgpt throughout the entire clinical workflow: Development and usability study," *Journal of Medical Internet Research*, vol. 25, no. 1, p. e48659, 2023.

[20] S. Ji, T. Zhang, L. Ansari, J. Fu, P. Tiwari, and E. Cambria, "Mentalbert: Publicly available pretrained language models for mental healthcare," in *Proceedings of the Thirteenth Language Resources and Evaluation Conference* (N. Calzolari, F. Béchet, P. Blache, K. Choukri, C. Cieri, T. Declerck, S. Goggi, H. Isahara, B. Maegaard, J. Mariani, H. Mazo, J. Odijk, and S. Piperidis, eds.), (Marseille, France), pp. 7184–7190, European Language Resources Association, 2022.

[21] K. Yang, S. Ji, T. Zhang, Q. Xie, Z. Kuang, and S. Ananiadou, "Towards interpretable mental health analysis with large language models," in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing* (H. Bouamor, J. Pino, and K. Bali, eds.), (Singapore), pp. 6056–6077, Association for Computational Linguistics, 2023.

[22] H. Chua, A. Caines, and H. Yannakoudakis, "A unified framework for cross-domain and cross-task learning of mental health conditions," in *Proceedings of the Second Workshop on NLP for Positive Impact (NLP4PI)* (L. Biester, D. Demszky, Z. Jin, M. Sachan, J. Tetreault, S. Wilson, L. Xiao, and J. Zhao, eds.), (Abu Dhabi, United Arab Emirates (Hybrid)), pp. 1–14, Association for Computational Linguistics, 2022.

[23] H. Xue, Y. Liang, B. Mu, S. Zhang, M. Chen, Q. Chen, and L. Xie, "E-chat: Emotion-sensitive spoken dialogue system with large language models," 2024.

[24] A. Srivastava, T. Suresh, S. Peregrine, Lord, M. S. Akhtar, and T. Chakraborty, "Counseling summarization using mental health knowledge guided utterance filtering," 2022.

[25] S. Sotudeh, N. Goharian, H. Deilamsalehy, and F. Dernoncourt, "Curriculum-guided abstractive summarization for mental health online posts," in *Proceedings of the 13th International Workshop on Health Text Mining and Information Analysis (LOUHI)* (A. Lavelli, E. Holderness, A. Jimeno Yepes, A.-L. Minard, J. Pustejovsky, and F. Rinaldi, eds.), (Abu Dhabi, United Arab Emirates (Hybrid)), pp. 148–153, Association for Computational Linguistics, 2022.

[26] S. Provoost, H. M. Lau, J. Ruwaard, and H. Riper, "Embodied conversational agents in clinical psychology: A scoping review," *Journal of Medical Internet Research*, vol. 19, no. 5, p. e6553, 2017.

[27] S.-L. Hsu, R. S. Shah, P. Senthil, Z. Ashktorab, C. Dugan, W. Geyer, and D. Yang, "Helping the helper: Supporting peer counselors via ai-empowered practice and feedback," 2023.

[28] S. Zanwar, D. Wiechmann, Y. Qiao, and E. Kerz, "Exploring hybrid and ensemble models for multiclass prediction of mental health status on social media," 2022.

[29] S. Liu, N. Deng, S. Sabour, Y. Jia, M. Huang, and R. Mihalcea, "Task-adaptive tokenization: Enhancing long-form text generation efficacy in mental health and beyond," in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing* (H. Bouamor, J. Pino, and K. Bali, eds.), (Singapore), pp. 15264–15281, Association for Computational Linguistics, 2023.

[30] X. Wang, X. Li, Z. Yin, Y. Wu, and J. Liu, "Emotional intelligence of large language models," *Journal of Pacific Rim Psychology*, vol. 17, p. 18344909231213958, 2023.

[31] S. Sabour, S. Liu, Z. Zhang, J. Liu, J. Zhou, A. Sunaryo, T. Lee, R. Mihalcea, and M. Huang, "Emobench: Evaluating the emotional intelligence of large language models," in *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (L.-W. Ku, A. Martins, and V. Srikumar, eds.), (Bangkok, Thailand), pp. 5986–6004, Association for Computational Linguistics, 2024.

[32] C. Li, J. Wang, Y. Zhang, K. Zhu, W. Hou, J. Lian, F. Luo, Q. Yang, and X. Xie, "Large language models understand and can be enhanced by emotional stimuli," 2023.

[33] D. Li, X. Liu, B. Xing, B. Xia, Y. Zong, B. Wen, and H. Kälviäinen, "Eald-mllm: Emotion analysis in long-sequential and de-identity videos with multi-modal large language model," 2024.

[34] M. Abbasian, I. Azimi, A. M. Rahmani, and R. Jain, "Conversational health agents: A personalized llm-powered agent framework," 2024.

[35] I. Lin, L. Njoo, A. Field, A. Sharma, K. Reinecke, T. Althoff, and Y. Tsvetkov, "Gendered mental health stigma in masked language models," in *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing* (Y. Goldberg, Z. Kozareva, and Y. Zhang, eds.), (Abu Dhabi, United Arab Emirates), pp. 2152–2170, Association for Computational Linguistics, 2022.

[36] H. Gaffney, W. Mansell, and S. Tai, "Conversational agents in the treatment of mental health problems: Mixed-method systematic review," *JMIR Mental Health*, vol. 6, no. 10, p. e14166, 2019.

[37] A. A. Abd-Alrazaq, M. Alajlani, N. Ali, K. Denecke, B. M. Bewick, and M. Househ, "Perceptions and opinions of patients about mental health chatbots: Scoping review," *Journal of Medical Internet Research*, vol. 23, no. 1, p. e17828, 2021.

[38] Q. Yang, M. Ye, and B. Du, "Emollm: Multimodal emotional understanding meets large language models," 2024.

[39] T. Saha, V. Gakhreja, A. S. Das, S. Chakraborty, and S. Saha, "Towards motivational and empathetic response generation in online mental health support," in *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '22, (New York, NY, USA), pp. 2650–2656, Association for Computing Machinery, 2022.

[40] A. Srivastava, I. Pandey, M. S. Akhtar, and T. Chakraborty, "Response-act guided reinforced dialogue generation for mental health counseling," in *Proceedings of the ACM Web Conference 2023*, WWW '23, (New York, NY, USA), pp. 1118–1129, Association for Computing Machinery, 2023.

[41] "Google bert/bert base uncased." https://huggingface.co/google-bert/bert-base-uncased, 2024.

[42] "Residual neural network," *Wikipedia*, 2024.

[43] M. Arjmand, F. Nouraei, I. Steenstra, and T. Bickmore, "Empathic grounding: Explorations using multimodal interaction and large language models with conversational agents," 2024.

[44] "Connected papers | find and explore academic papers." https://www.connectedpapers.com/.

[45] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)* (J. Burstein, C. Doran, and T. Solorio, eds.), (Minneapolis, Minnesota), pp. 4171–4186, Association for Computational Linguistics, 2019.