



ISSN: 2814-1709

CTICTR 3(1): 76 - 93 (June 2024)

Received: 00-04-2024

Accepted: 00-06-2024

<https://doi.org/10.61867/pcub.v2i1a.060>

A Large Language Model to Compare Human and Machine Empathy for Work-Life Balance-Induced Depression

Israel Onyedikachi Nnaji

Ernest E. Onuiri

nnaji0102@pg.babcock.edu.ng

School of Computing, Babcock University, Ilishan-Remo, Ogun State, Nigeria.

A Large Language Model to Compare Human and Machine Empathy for Work-Life Balance-Induced Depression

Israel O. Nnaji^{a*}, Ernest E., Onuirib

^{a,b} School of Computing, Babcock University, Ilishan-Remo, Ogun State, Nigeria.

^annaji0102@pg.babcock.edu.ng

^bonuirie@babcock.edu.ng

Abstract

Work-life balance is a critical aspect of well-being, and its disruption can lead to various negative outcomes, including depression. It has been estimated that many people extend work activities beyond their stipulated work hours in Nigeria, also bearing the impact of the steady economic challenges, its effect on the living conditions of an average citizen, these conditions alone may lead to several forms of depression and work-life balanced induced depression is an example. Empathy, both from humans and machines, plays significant roles in supporting individuals experiencing work-life balance-induced depression. This research develops a large language model that compares human and machine empathy in addressing work-life balance-induced depression. The study employs a mixed-methods approach, incorporating both quantitative and qualitative data. A comprehensive review provides a foundation for understanding the underlying factors and potential interventions related to work-life balance-induced depression. Then, a large language model is developed from fine-tuning existing GPT-3 davinci model, leveraging advanced natural language processing techniques. The model is trained on a dataset comprising real-life scenarios related to work-life balance-induced depression, personal experiences, challenges, and coping mechanisms. Both human participants and the large language model are presented with these scenarios and asked to provide empathetic responses. Evaluation of the model's performance on the generated empathetic statements involved the use of perplexity, accuracy and F1-score as metrics. The findings of this research contribute to a better understanding of the potential role of machine empathy in addressing work-life balance-induced depression.

Keywords: work-life balance, empathy, depression, large language model, predictive model

1. Introduction

Natural Language Generation (NLG), an approach to Natural Language Processing (NLP) as a branch of Artificial Intelligence (AI) enables continuous improvements in the technologies behind Human-Computer Interaction (HCI) and cognitive psychology where software or devices understands inputs from humans made in the form of sentences or speech provides as output the necessary or related information required with embedded human centeredness expressed in the form of artificial empathy [1].

The recent accomplishment and area witnessing these transformations is in the application of Large Language Models (LLM) or predictive models trained using self-supervised learning on large quantities of unlabeled data that has produced successful applications such as the *ChatGPT* made available by the *OpenAI* team that has a lot of rave reviews now as well other related applications in that manner [2]. In the bid to build trust or more reliance on these applications, responses generated during conversation must be void of unnecessary offensive languages, bias, or discriminations. Furthermore, the conversations must express emotional intelligence, and empathy when required. This should include any other form of human attributes that may address some of the current challenges of Large Language Models[2].

One of the major benefits that can be derived from utilizing the capabilities of LLM is by providing applications that uses *prompts* [3] to generate responses that improves conversations in near human-like fashion. The application of this kind can be useful to address the issues of people who find it difficult to strike the balance between work and their ability to maintain their obligations at home. This obviously create a condition regarded as situational depression under work-life balance category as one of the silent challenges inhibiting the productivity of these individuals beset with such factors that increases their conditions daily. There exist few means to address this ever-growing condition. According to [4] it is estimated that many people work 50 hours or more per week in Nigeria bearing the impact of economic challenges and its effect on the living conditions of an average citizen. This condition is a trigger for persons already showing symptoms of work-life balance induced depression making this work beneficial to them as a medium to express themselves using generated data infused with NLP via LLMs to provide updated empathetic conversations that can improve their lives.

This paper outlines the ways responses from LLM applications can be beneficial to this category of people. It begins with comparing the answers to the questions asked to a human psychotherapist during a physical session with selected persons experiencing work-life balance induced depression with the answers generated by the developed LLM all in the bid to ascertain the level of artificial empathy embedded in the response. This approach with the use of LLM evaluation metrics increases the reliance on Language Models in their ability to engage in meaningful conversations that can improve the wellbeing of people suffering from work life balance induced depression.

2.0 Related Work

Today, the innovations borne from the application of NLP techniques have extended beyond just translating texts from one language to another but spans across both individual and enterprise purposes. Organizations [5] now use NLP for several automated tasks including processing, analyzing

and archiving large documents, running chatbots for customer service automation, analyzing customer feedbacks from marketing insights, classification and extraction of texts from both structured and unstructured data for content moderation, improving searches and are even added to enhance Search Engine Optimization (SEO) for faster search results[6].

Classified as a type of NLP model is Large Language Model (LLM) which is trained on massive amount of data [7] with the ability to handle growing demand for machines to handle complex language tasks, including translation, summarization, information retrieval, and conversational interactions [8]. Several researchers have discussed the evolution of Large Language Models but [2] provides the evolution of Large Language Models right from 1950 when humans have explored the mastery of language intelligence by machines. This paper also sheds more light on the effect of scaling language models for better performance and discusses the concept of emergent abilities, which are the unexpected behavior observed due to the increase of the model's data size. This improved performance of LLM hinged on the concept of model scaling [8] (increased data size, tokens, and parameters) involves the usage of huge computational resources at a very expensive cost. [9] posits that the cost of delivering a model with only 1.5 billion parameters costs \$1.6 million; however, this innovation has revolutionized the performance of text-generating models.

Due to the high cost of building language models from the scratch, most applications are developed by fine-tuning existing models [10] which is basically machine-learning technique that involves making conscious adjustments to a pre-trained model to achieve improved performance using the dataset peculiar to the application specifications [11]. [12] investigates the transferability of pre-trained language models on artificial datasets and demonstrated the limitations of pre-trained language models and emphasizes the importance of fine-tuning on task-specific datasets.

Apart from working with specific datasets, the adjustments also includes modifying the existing parameters of the pre-trained model to be fine-tuned and [13] proposed an approach for fine-tuning called "Sparse Rational Activation Pruning and re-parameterization" (SRAP-REP). This approach reduces number of parameters without compromising performance. The author argues that other methods that require many parameters may make the model impractical for some applications. In the aspect of reduced number of parameters, [14] propose a modular fine-tuning approach that is computationally efficient, which improves the performance of NLP models by separating the model into encoder and classifier modules. Both [14] and [11] use The General Language Understanding Evaluation (GLUE) as the benchmark for the collection of resources for training, evaluating, and analyzing Natural Language Understanding (NLU) systems.

[15] discusses the application of LLM in the mental health domain by leveraging LLM for mental health prediction via online text-data, the paper also produces a fine-tuned model 'Mental-Alpaca' as the output that outperforms GPT-3.5 (25 times bigger) by 16.7% on balanced accuracy. The combination of parameter efficient fine-tuning and deep learning techniques are used in [16] to provide a model that assists mental health care providers with an assessment of depression and this model also outperforms all previously published methods.

To adequately determine the performance of any fine-tuned model, several papers [17][16][18] implement large language metrics such as Confusion Matrix, Perplexity, Accuracy and F1-Score but to extend the evaluation of LLM in determining the level of empathy from the responses generated from using them for work-life balanced induced depression as the core of this research paper, [19] proposes a benchmark for evaluating empathy in language models using a new dataset regarded as the *Empathetic Dialogues* consisting of over 25,000 dialogues between two speakers annotates with

empathy scores using several metrics to evaluate the empathy such as Perplexity, Accuracy, etc. Though in [20], the paper acknowledges that measuring empathy in conversations can be challenging, therefore proposing an automatic method for evaluation that reduces the need for human evaluation using the EMP-EVAL approach that incorporates the influence of Emotion, Cognitive and Emotional empathy.

However, from this review of literatures which provides work across language models, mentioning both empathetic and fine-tuned models with their varying evaluations and benchmarks, there is no work on large language models that addresses the work-life balance induced depression. Therefore, there is need to present a fine-tuned model that is adequately evaluated using defined large language model evaluation metrics to address the issues arising from kind of depression.

3.0 Methodology

The objective of developing a large language model involves several steps, focusing on comparing human and machine empathy for work-life balance-induced depression and the ability to comprehend and respond appropriately within the context of related conversations. The process begins with fine-tuning the GPT-3 model and integrating it into the OpenAI playground page. Next, a dataset is created from conversations (Questions and Answers) between individuals experiencing symptoms of work-life balance depression and a professional psychologist. This dataset is then transformed into prompt-completion pairs. Subsequently, a role-playing task is implemented for prompt-engineering to simulate the persona of a virtual psychologist, generating suitable responses to questions posed by these individuals.

This approach aims to accumulate responses from both real psychologists during their sessions. These responses are then evaluated to determine the level of artificial empathy embedded in the responses generated by the virtual psychologist.

To assess the generated responses, the existing GPT-3 'text-davinci-003' model is initially fine-tuned using the dataset compiled from the aforementioned conversations. The dataset is converted into prompt-completion pairs, a process facilitated by the Python code snippet depicted in Figure 3.1. Within the 'process_yaml_file' function, the conversations are represented as data stored in a 'yaml_file', which is then read into the 'data' variable. This data is subsequently passed as a parameter into the 'generate_prompt_completion_pairs' function to initiate the generation of prompt-completion pairs. Finally, the resulting pairs are exported as jsonl, serving as the output format for model data.

```

Generate prompt and completion pairs from the work_life_balance_depression dataset and exported as
prompt_completion_pairs_prepared.jsonl

[5] def generate_prompt_completion_pairs(data):
    pairs = []
    for conversation in data['conversations']:
        for i in range(len(conversation) - 1):
            prompt = conversation[i]
            completion = conversation[i + 1]
            pairs.append({"prompt": prompt, "completion": completion})
    return pairs

def process_yaml_file(file_path):
    with open(file_path, 'r') as file:
        data = yaml.safe_load(file)
        pairs = generate_prompt_completion_pairs(data)
    return pairs

def export_to_json(prompt_completion_pairs, output_file):
    with open(output_file, 'w') as file:
        json.dump(prompt_completion_pairs, file, indent=4)

yaml_file = 'work_life_balance_depression.yaml'
json_output_file = 'prompt_completion_pairs.json'

pairs = process_yaml_file(yaml_file)
export_to_json(pairs, json_output_file)

```

Figure 1: Code snippet for the generation of prompt-completion pairs.

After this process, Figure 3.2 illustrates the automated data preparation process, demonstrating the utilization of the Command Line Interface (CLI) command executed within Jupyter Notebook as the Integration Development Environment for this research endeavor.

```

!openai tools fine_tunes.prepare_data -f 'prompt_completion_pairs.json'

```

Analyzing...

- Your file contains 67 prompt-completion pairs. In general, we recommend having at least a few hundred examples. We've found that performance tends to 1
- Your data does not contain a common separator at the end of your prompts. Having a separator string appended to the end of the prompt makes it clearer
- Your data does not contain a common ending at the end of your completions. Having a common ending string appended to the end of the completion makes it
- The completion should start with a whitespace character (` `). This tends to produce better results due to the tokenization we use. See <https://platform>

Based on the analysis we will perform the following actions:

- [Recommended] Add a suffix separator ` ->` to all prompts [Y/n]: Y
- [Recommended] Add a suffix ending `\n` to all completions [Y/n]: Y
- [Recommended] Add a whitespace character to the beginning of the completion [Y/n]: Y

Your data will be written to a new JSONL file. Proceed [Y/n]: Y

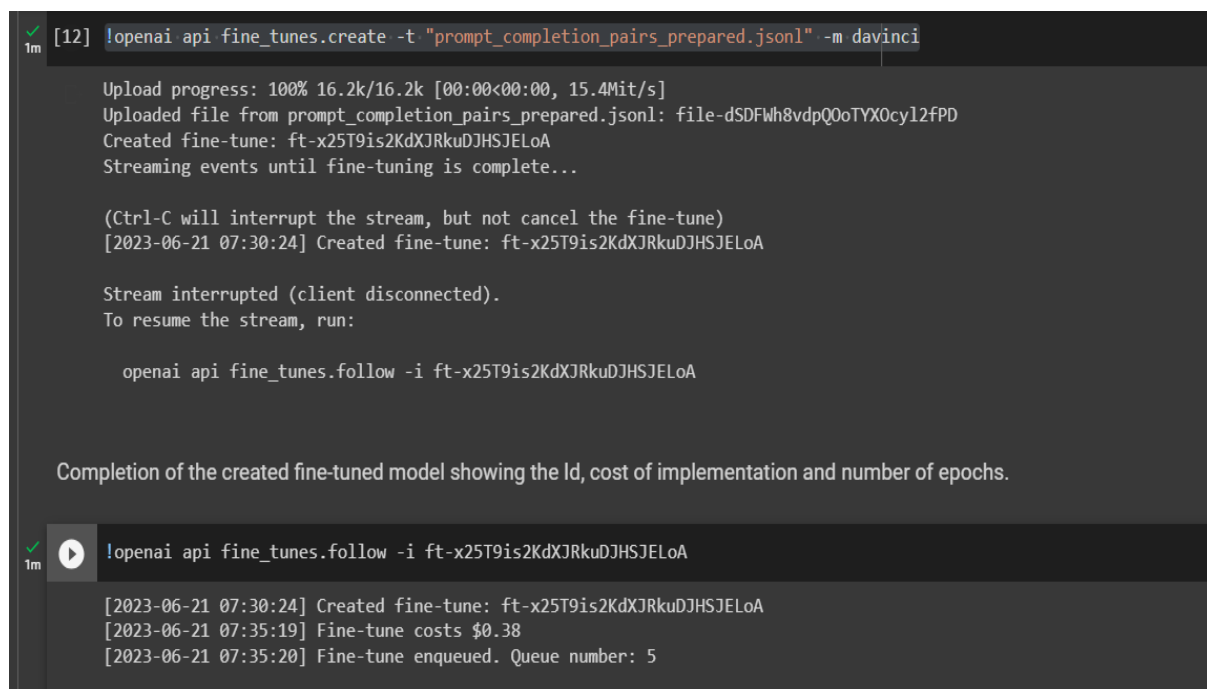
Wrote modified file to `prompt_completion_pairs_prepared.jsonl`
Feel free to take a look!

Now use that file when fine-tuning:
> openai api fine_tunes.create -t "prompt_completion_pairs_prepared.jsonl"

After you've fine-tuned a model, remember that your prompt has to end with the indicator string ` ->` for the model to start generating completions, rath
Once your model starts training, it'll approximately take 3.36 minutes to train a `curie` model, and less for `ada` and `babbage`. Queue will approximate

Figure 2: Data preparation process

Multiple base models of the GPT-3 are available, distinguished by the number of parameters and unique computational resources [21]. The recognized models include Ada, Babbage, Curie, and Davinci. In this paper, we focus on fine-tuning the 'Davinci' model, as it is considered the most capable for efficiently handling text-generation and question-and-answer tasks with higher quality[21]. Figure 3.3 outlines the CLI command for creating a fine-tuned model using a prompt-completion pair output file named 'prompt_completion_pairs_prepared.jsonl', based on the 'Davinci' model. The figure concludes with the creation of the model, indicating its ID, implementation cost, and the number of epochs. Additionally, Figure 3.4 presents the metadata of the fine-tuned model, identified as 'davinci:ft-personal-2023-06-21-07-48-30'.



```
[12] !openai api fine_tunes.create -t "prompt_completion_pairs_prepared.jsonl" -m davinci

Upload progress: 100% 16.2k/16.2k [00:00<00:00, 15.4Mit/s]
Uploaded file from prompt_completion_pairs_prepared.jsonl: file-dSDFWh8vdpQ0oTYX0cy12fPD
Created fine-tune: ft-x25T9is2KdXJRkuDJHSJELoA
Streaming events until fine-tuning is complete...

(ctrl-C will interrupt the stream, but not cancel the fine-tune)
[2023-06-21 07:30:24] Created fine-tune: ft-x25T9is2KdXJRkuDJHSJELoA

Stream interrupted (client disconnected).
To resume the stream, run:

openai api fine_tunes.follow -i ft-x25T9is2KdXJRkuDJHSJELoA

Completion of the created fine-tuned model showing the Id, cost of implementation and number of epochs.

!openai api fine_tunes.follow -i ft-x25T9is2KdXJRkuDJHSJELoA

[2023-06-21 07:30:24] Created fine-tune: ft-x25T9is2KdXJRkuDJHSJELoA
[2023-06-21 07:35:19] Fine-tune costs $0.38
[2023-06-21 07:35:20] Fine-tune enqueued. Queue number: 5
```

Figure 3: Implementation showing the utilization of the prompt-completion pairs to create the fine-tuned model.

```

{
  "object": "list",
  "data": [
    {
      "object": "fine-tune",
      "id": "ft-x25T9is2KdXJRkuDJHSJELoA",
      "hyperparams": {
        "n_epochs": 4,
        "batch_size": 1,
        "prompt_loss_weight": 0.01,
        "learning_rate_multiplier": 0.1
      },
      "organization_id": "org-oyC0Um9RliSxS21fVAbSDUbM",
      "model": "davinci",
      "training_files": [
        {
          "object": "file",
          "id": "file-dSDFWh8vdpQ0oTYX0cy12fPPD",
          "purpose": "fine-tune",
          "filename": "prompt_completion_pairs_prepared.jsonl",
          "bytes": 16166,
          "created_at": 1687332624,
          "status": "processed",
          "status_details": null
        }
      ],
      "validation_files": [],
      "result_files": [
        {
          "object": "file",
          "filename": "prompt_completion_pairs_prepared.jsonl",
          "bytes": 16166,
          "created_at": 1687332624,
          "status": "processed",
          "status_details": null
        }
      ],
      "validation_files": [],
      "result_files": [
        {
          "object": "file",
          "id": "file-u6i8hUBAVj2cnsMTMoEO0n5f5",
          "purpose": "fine-tune-results",
          "filename": "compiled_results.csv",
          "bytes": 13936,
          "created_at": 1687333711,
          "status": "processed",
          "status_details": null
        }
      ],
      "created_at": 1687332624,
      "updated_at": 1687333712,
      "status": "succeeded",
      "fine_tuned_model": "davinci:ft-personal-2023-06-21-07-48-30"
    }
  ]
}

```

Figure 4: Metadata of the fine-tuned model

The performance comparison between the fine-tuned model 'davinci:ft-personal-2023-06-21-07-48-30' and 'text-davinci-003' was conducted based on their perplexity, accuracy, and F1-score, which are the LLM evaluation metrics for this study. The results of this analysis are presented in Section 4.0.

In addition to assessing the LLM evaluation metrics, further evaluation was conducted to determine if the fine-tuned model exhibits any form of artificial empathy in work-life balanced interactions. This assessment involved analyzing the generated responses for expressions of empathy, which can be

inferred by detecting emotions in the responses. According to [20], emotions play a crucial role in predicting empathy, as understanding, and identifying others' emotions are fundamental aspects of empathy. To evaluate empathy levels in the responses, Text emotion detection was conducted utilizing the pre-fine-tuned Cardiffnlp/Twitter-RoBERTa-Base-Emotion model [22] along with the GoEmotions emotion labels [23]. Subsequently, the obtained emotion labels and scores were graphically depicted, as demonstrated in Figure 3.5 below.

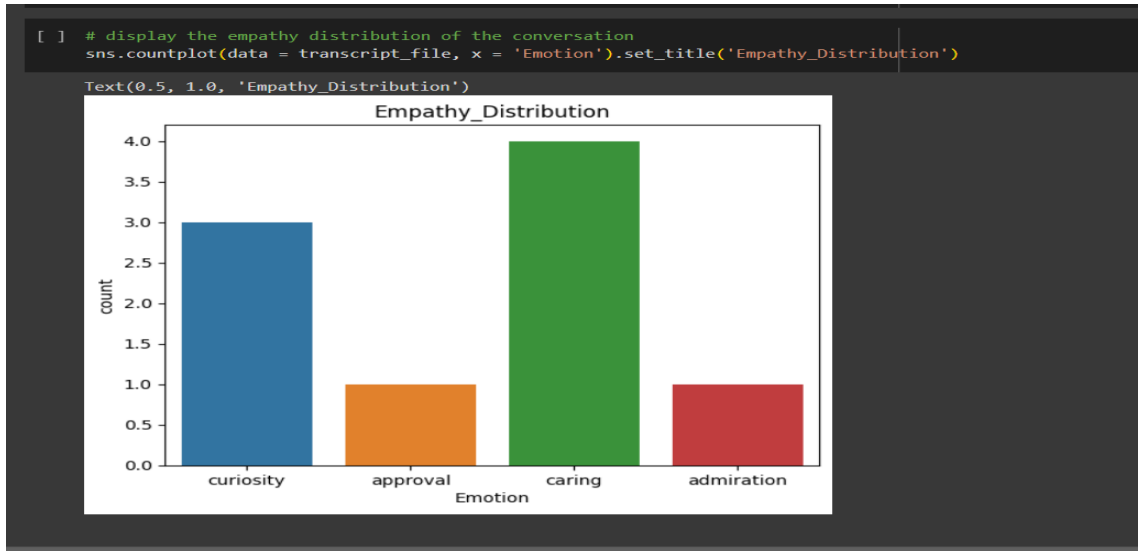


Figure 5: Empathy distribution chart to capture emotion labels and scores.

Utilizing the Python seaborn library, the graphical representation depicts the count or empathy score on the 'y' axis and the predicted emotion label on the 'x' axis. Figure 3.5 illustrates the end-to-end process for employing a large language model to compare human and machine empathy for work-life balance-induced depression, serving as the proposed technique for implementation. This methodology finds support in [20] due to the potential subjectivity and susceptibility to inconsistent results or errors associated with human evaluations in this scenario.

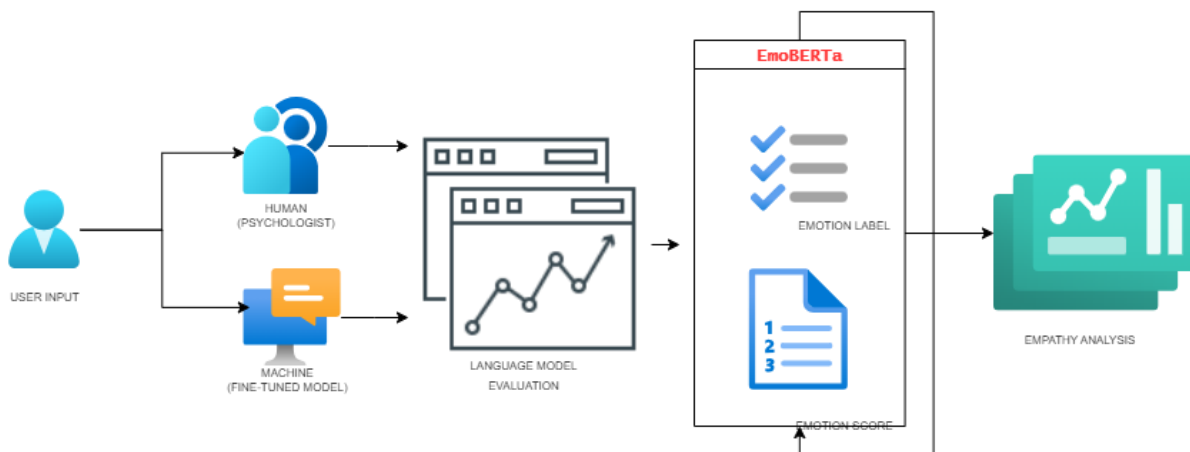


Figure 6: Diagram to illustrate the research design.

3.1 Dataset and collection procedure

The data collection procedure, serving as the primary source of data, involved the participation of anonymous individuals experiencing work-life depression triggered by factors such as rising living costs, stagnant salaries, extended working hours (including weekends), and increased domestic responsibilities, particularly for women managing childcare alongside their jobs. These individuals engaged in weekly 20-minute sessions with a clinical psychologist over a span of two weeks. Prior to these sessions, participants were sent consent forms outlining the study's purpose, procedures, and a confidentiality clause assuring the privacy of their conversations. It was emphasized that their identities would remain undisclosed in any ensuing reports or publications.

The qualitative research methodology utilized live sessions between the clinical psychologist and anonymous individuals displaying symptoms of work-life depression, aligning with the targeted triggers identified for this study. Table 1 provides descriptions of these triggers, while transcripts of the conversations were formatted as comma-separated value (.CSV) files, featuring two columns: 'therapist' and 'patient'. These transcripts were subsequently employed in the empathy evaluation processes.

3.2 Ethical considerations

The data collection process described earlier conforms to data privacy laws, thereby adhering to a key ethical consideration. It ensures anonymity for both the individuals who responded to the questionnaire and the psychologist involved in the study. This anonymity is maintained throughout the development of this work, supported by a consent form specifically designed for this research. Table 1.1 presents information on these individuals, including their ages, genders, current job descriptions, and locations, as well as the primary triggers and associated symptoms of work-life balance depression.

Table 1: Profile of persons experiencing work-life balance depression.

S/N	Name	Age	Sex	Job Description	Location	Trigger	Symptoms
1	Anonymous	34	M	Enterprise Architect	Lagos	More responsibilities at work causing longer work hours without pay	<ul style="list-style-type: none"> • Low motivation. • Indifference. • Substance abuse.
2	Anonymous	28	F	Quality	Lagos	Presence of a	<ul style="list-style-type: none"> • Low

				Assurance Engineer		new baby	<ul style="list-style-type: none"> productivity Poor coordination Fear of job loss
3	Anonymous	26	F	Customer Service Representative / Student	Lagos	Academic expectations.	<ul style="list-style-type: none"> Poor coordination Continuous tiredness and fatigue. Trouble sleeping.
4	Anonymous	31	M	Software Engineer / MSc student in Nigeria	Lagos	Handling two jobs within the same time zone.	<ul style="list-style-type: none"> Continuous tiredness and fatigue. Fear of job loss.
5	Anonymous	27	M	Transport worker	Lagos	Long work hours and financial pressure for incoming baby.	<ul style="list-style-type: none"> Substance abuse. Increased fatigue
6	Anonymous	35	M	MSc Student/Worker	UK	Increase number of bills to pay with limited number of work hours.	<ul style="list-style-type: none"> Low productivity Poor coordination
7	Anonymous	45	M	Banker	Lagos	Mid-life crisis and financial expectations from family.	<ul style="list-style-type: none"> Low motivation. Low productivity.

4. Fine-tune model evaluation using large language model metrics.

To ascertain the perplexity, a measure indicating how effectively the fine-tuned model predicts user prompts and responds with empathy-laden texts, the calculation involves computing the average log probability of the test data, followed by exponentiation. This process aims to generate either a lower or higher perplexity

```
[ ] def generate_predictions(prompt):
    response = openai.Completion.create(
        engine='davinci:ft-personal-2023-06-21-07-48-30',
        prompt=prompt,
        max_tokens=150, # Adjust the value as per your requirements
        temperature=0.7 # Adjust the temperature as needed
    )
    return response.choices[0].text.strip()

[ ] from transformers import OpenAIGPTLMHeadModel, OpenAIGPTTokenizer
import torch

def compute_perplexity(predictions, dataset):
    tokenizer = OpenAIGPTTokenizer.from_pretrained('openai-gpt') # Adjust the model as per your fine-tuned model
    if tokenizer.pad_token is None:
        tokenizer.add_special_tokens({'pad_token': '[PAD]'})
    model = OpenAIGPTLMHeadModel.from_pretrained('openai-gpt')

    tokenized_dataset = tokenizer.batch_encode_plus(dataset, return_tensors='pt', padding=True, truncation=True)
    tokenized_predictions = tokenizer.batch_encode_plus(predictions, return_tensors='pt', padding=True, truncation=True)

    input_ids = tokenized_dataset['input_ids']
    target_ids = tokenized_dataset['input_ids']
    with torch.no_grad():
        logits = model(input_ids=input_ids, labels=target_ids).logits
        perplexity = torch.nn.functional.cross_entropy(logits.view(-1, logits.shape[-1]), target_ids.view(-1))

    return perplexity.item()
```

Figure 7: Implementation to calculate Perplexity.

Utilizing the Davinci model from the GPT-3 series, particularly 'text-davinci-003', as a comparative model alongside the Evaluation Dataset (ED) and Prompt (P), the compilation and scores for these metrics are presented below within the provided code snippet. The distinctions between the models are delineated in the corresponding tables.

Evaluation Dataset (ED): Recently, I've been having frequent panic attacks, and it's significantly impacting my ability to function. I'm struggling to control my racing thoughts and rapid heartbeat. How can I effectively cope with this level of anxiety?

Prompt (P): I constantly feel overwhelmed and anxious. It's interfering with my daily activities, work responsibilities, and relationships. I require assistance in comprehending and managing these intense emotions.

Table 2: comparative result showing the Perplexity scores of fine-tuned model 'davinci:ft-personal-2023-06-21-07-48-30' against GPT-3 'text-davinci-003' model.

S/N	Model	Prompt	Evaluation dataset	Perplexity
1.	davinci:ft-personal-2023-06-	P	ED	9.996932029724121

	21-07-48-30			
2.	text-davinci-003	P	ED	10.312542915344238

Next is accuracy, which measures the percentage of correctly predicted tokens or labels compared to the total number of tokens or labels in the evaluation dataset. Accuracy is commonly employed for named entity recognition and is necessary for detecting emotion labels and assessing the alignment of generated responses with identified labels. This implementation is depicted in the snippet provided in Figure 4.1 below.

```
[14] def calculate_accuracy(predictions, labels):
    correct_predictions = 0
    total_predictions = len(predictions)

    for pred, label in zip(predictions, labels):
        if pred == label:
            correct_predictions += 1

    accuracy = correct_predictions / total_predictions
    return accuracy

# Example usage:
evaluation_data = [...] # List of evaluation examples
expected_labels = [...] # List of expected labels for evaluation examples
predicted_labels = [...] # List of predicted labels by the fine-tuned model

accuracy = calculate_accuracy(predicted_labels, expected_labels)
print(f"Accuracy: {accuracy}")
```

Figure 8: Implementation to calculate Accuracy.

Evaluation Dataset (ED) = ["I've been feeling very anxious and overwhelmed lately.",
 "My mood has been consistently low for the past few weeks.",
 "I'm having trouble sleeping and often experience racing thoughts.",
 "I've lost interest in activities that I used to enjoy.",
 "I find it hard to concentrate and make decisions."
]

Expected Labels (EL) = ["Anxiety", "Depression", "Insomnia", "Lack of Interest", "Difficulty Concentrating"]

Predicted Labels (PL) = ["Anxiety", "Sadness", "Insomnia", "Lack of Interest", "Anxiety"]

Table 3: comparative result showing the accuracy scores of fine-tuned model ‘davinci:ft-personal-2023-06-21-07-48-30’ against GPT-3 ‘text-davinci-003’ model.

Model	Evaluation_data	Expected_labels	Predicted labels	Accuracy
davinci:ft-personal-2023-06-21-07-48-30	ED	EL	PL	0.8
text-davinci-003	ED	EL	PL	0.6

The F1-Score, the final metric, integrates recall and precision to generate a unified score, representing the harmonic mean of these two metrics. In mathematical terms, it combines recall and precision. Precision quantifies the ratio of accurately predicted positive instances to all predicted positive instances, while recall gauges the ratio of correctly predicted positive instances to all actual positive instances. Substituting the sample dataset with the evaluation dataset, the method for computing perplexity, accuracy, and F1-Score, along with the table presenting these metric scores, is presented.

```

from sklearn.metrics import f1_score

# Assuming you have the ground truth labels and predicted labels
ground_truth = ['...', '...', '...', '...', '...']
predicted_labels = ['...', '...', '...', '...', '...']

# Calculate the F1-Score
f1 = f1_score(ground_truth, predicted_labels, average='weighted')

# Print the F1-Score
print("F1-Score:", f1)

```

Figure 9: Implementation to calculate F1-Score

Ground Truth (GT) = ["Depression", "Anxiety", "Sadness", "Difficulty Concentrating"]
 Predicted Labels (PL) = ["Depression", "Anxiety", "Insomnia", "Lack of Interest"]

Table 4: comparative result showing the F1-Scores of fine-tuned model ‘davinci:ft-personal-2023-06-21-07-48-30’ against GPT-3 ‘text-davinci-003’ model.

Model	Ground_truth	Predicted_truth	F1-Score
davinci:ft-personal-2023-06-21-07-48-30	GT	PL	0.65
text-davinci-003	GT	PL	0.5

4.1 Evaluation of artificial empathy using the ‘cardiffnlp/twitter-roberta-base-emotion’ model.

Figure 10:

```

import torch
from transformers import AutoTokenizer, AutoModelForSequenceClassification

def detect_emotions(input_text):
    # Load EmoBERTa model and tokenizer
    model_name = "cardiffnlp/twitter-roberta-base-emotion"
    tokenizer = AutoTokenizer.from_pretrained(model_name)
    model = AutoModelForSequenceClassification.from_pretrained(model_name)

    # Tokenize input text
    encoded_input = tokenizer.encode_plus(
        input_text,
        add_special_tokens=True,
        truncation=True,
        padding="max_length",
        max_length=128,
        return_tensors="pt"
    )

    # Perform emotion prediction
    with torch.no_grad():
        logits = model(**encoded_input).logits
        probabilities = torch.softmax(logits, dim=1).squeeze()

    # Define GoEmotions labels
    labels = [
        "admiration", "amusement", "anger", "annoyance", "approval", "caring", "confusion",
        "curiosity", "desire", "disappointment", "disapproval", "disgust", "embarrassment",
        "excitement", "fear", "gratitude", "grief", "joy", "love", "nervousness", "optimism",
        "pride", "realization", "relief", "remorse", "sadness", "surprise", "neutral"
    ]

    # Map predicted labels to GoEmotions labels
    emotions = {label: score.item() for label, score in zip(labels, probabilities)}

    return emotions

# Example usage:
input_text = '''I understand how frustrated you are with your life and how much you want to get a job right now, but I guess listening to me would cheer you up in the m
Human: Okay so how can you help? I can use an empathy method to understand your feelings and communicate with you.'''
emotion_scores = detect_emotions(input_text)
print(emotion_scores)

{'admiration': 0.11623817682266235, 'amusement': 0.017891982570290565, 'anger': 0.1366683840751648, 'annoyance': 0.7292014360427856}
    
```

Implementation showing the detection of emotion with labels and scores.

4.2 Discussion of Results

The decision to fine-tune the GPT-3 model into a large language model for comparing human and machine empathy regarding work-life balance-induced depression depends on the objectives of the study. Its performance can be assessed using large language evaluation metrics, comparing them with the results of 'text-davinci-003' in terms of Perplexity, Accuracy, and F1-score. Additionally, the evaluation considers the presence of artificial empathy in the form of emotion labels and scores. Previous studies have indicated that human evaluations may yield inconsistent results and errors [20], influencing the choice of evaluation methods in this research. While comparing with existing studies would have been ideal, the relevant literature primarily focuses on sentiment analysis to identify depression in tweets [24][25][26], with some exceptions like [16], which presents a depression severity assessment tool for mental health care providers. Due to the expense of fine-tuning models using OpenAI tools and the volume of data required for this study, emergent abilities associated with fine-tuning a model with a massive amount of data for model scaling [8] were not observed, presenting a limitation of this study. However, evaluation using defined LLM metrics on the fine-tuned model indicates that 'davinci:ft-personal-2023-06-21-07-48-30' outperforms GPT-3 'text-davinci-003' in terms of Perplexity, Accuracy, and F1-Score, as depicted in Table 4.1, 4.2, and 4.3. Furthermore, the presence of artificial empathy through emotion labels and scores is evident, as shown in Figure 4.4. This underscores the potential of fine-tuned models to generate empathetic responses to questions posed by individuals experiencing work-life balance-induced depression.

5.0 Conclusion

This research paper explores the utilization of a fine-tuned large language model in generating empathetic responses for individuals experiencing depression induced by work-life imbalance and assesses the model's performance. The evaluation involves assessing the fine-tuned model's performance using evaluation metrics such as Perplexity, Accuracy, and F1-Score, in comparison to a GPT-3 model. Furthermore, the evaluation extends to identifying emotion labels and scores by subjecting generated responses to an emotion detection model [3]. Results indicate that the fine-tuned model surpasses the performance of the parent GPT-3 model. Additionally, analysis from the emotion detection model reveals the presence of recognizable emotion labels within the responses.

However, limitations were encountered during the study. These include the scarcity of existing literature on evaluating work-life balanced depression using Large Language Models, the high cost associated with model fine-tuning, and the unavailability of datasets used as secondary data for this study. These limitations hindered the ability to detect emergent abilities typically identified during fine-tuning a model on a large volume of data.

Despite these challenges, the study's findings offer valuable insights into the potential of Large Language Models for addressing work-life balance-induced depression. This research contributes to current and future endeavors in this field, offering potential benefits to individuals coping with such challenges, thereby enhancing their overall quality of life.

References

- [1] C. Dong *et al.*, “A Survey of Natural Language Generation,” *ACM Comput. Surv.*, vol. 55, no. 8, pp. 1–38, 2023, doi: 10.1145/3554727.
- [2] W. X. Zhao *et al.*, “A Survey of Large Language Models,” pp. 1–51, 2023, [Online]. Available: <http://arxiv.org/abs/2303.18223>
- [3] M. U. Hadi *et al.*, “Large Language Models: A Comprehensive Survey of its Large Language Models: A Comprehensive Survey of its Applications Challenges, Limitations, and Future Prospects,” *TechRxiv*, 2023, doi: 10.36227/techrxiv.23589741.v3.
- [4] B. Aderounmu, “Work, family life, and current economic situation (I) - Businessday NG,” Lagos. [Online]. Available: <https://businessday.ng/opinion/article/work-family-life-and-current-economic-situation-i/>
- [5] P. M. Mah, I. Skalna, and J. Muzam, “Natural Language Processing and Artificial Intelligence for Enterprise Management in the Era of Industry 4.0,” *Appl. Sci.*, vol. 12, no. 18, 2022, doi: 10.3390/app12189207.
- [6] B. Dickson, “What is Natural Language Processing (NLP)?,” *PC Magazine*. pp. 122–127, 2020. [Online]. Available: https://www.sas.com/en_gb/insights/analytics/what-is-natural-language-processing-nlp.html%0Ahttp://search.ebscohost.com/login.aspx?direct=true&db=asx&AN=141517497&site=eds-live
- [7] D. Zan *et al.*, “Large Language Models Meet NL2Code : A Survey,” vol. 1, pp. 7443–7464, 2023.
- [8] H. Naveed, A. U. Khan, S. Qiu, and M. Saqib, “A Comprehensive Overview of Large Language Models,” pp. 1–37.
- [9] K. Wiggers, “The emerging types of language models and why they matter,” *TechCrunch*. 2022. [Online]. Available: techcrunch.com/2022/04/28/the-emerging-types-of-language-models-and-why-they-matter/
- [10] X. Han *et al.*, “Pre-trained models : Past , present and future,” *AI Open*, vol. 2, no. June 2021, pp. 225–250, 2023, doi: 10.1016/j.aiopen.2021.08.002.
- [11] C. Sun, X. Qiu, Y. Xu, and X. Huang, “How to Fine-Tune BERT for Text Classification?,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 11856 LNAI, no. 2, pp. 194–206, 2019, doi: 10.1007/978-3-030-32381-3_16.
- [12] C. H. Chiang and H. Y. Lee, “On the Transferability of Pre-trained Language Models: A Study from Artificial Datasets,” *Proc. 36th AAAI Conf. Artif. Intell. AAAI 2022*, vol. 36, pp. 10518–10525, 2022, doi: 10.1609/aaai.v36i10.21295.
- [13] N. Ding *et al.*, “Parameter-efficient fine-tuning of large-scale pre-trained language models,” *Nat. Mach. Intell.*, vol. 5, no. March, 2023, doi: 10.1038/s42256-023-00626-4.
- [14] S. Ruder, J. Pfeiffer, and I. Vulić, “Modular and Parameter-Efficient Fine-Tuning for NLP Models,” *EMNLP 2022 - 2022 Conf. Empir. Methods Nat. Lang. Process. Tutor. Abstr.*, pp. 23–29, 2022.
- [15] B. T. Atmaja and A. Sasou, “Leveraging Pre-Trained Acoustic Feature Extractor For Affective Vocal Bursts Tasks,” in *Proceedings of 2022 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference, APSIPA ASC 2022*, Institute of Electrical and Electronics Engineers Inc., 2022, pp. 1412–1417. doi: 10.23919/APSIPAASC55919.2022.9980083.
- [16] C. Lau, X. Zhu, and W. Chan, “Automatic depression severity assessment with deep learning using parameter-efficient tuning,” no. 1.
- [17] I. C. Obagbuwa, S. Danster, and O. C. Chibaya, “Supervised machine learning models for depression sentiment analysis,” *Front. Artif. Intell.*, vol. 6, 2023, doi: 10.3389/frai.2023.1230649.
- [18] Y. Chang *et al.*, “A Survey on Evaluation of Large Language Models,” pp. 1–23, 2023, [Online]. Available: <http://arxiv.org/abs/2307.03109>
- [19] J. Chen, S. Yang, J. Xiong, and Y. Xiong, “An effective emotion tendency perception model in empathic dialogue,” *PLoS One*, vol. 18, no. 3 March, pp. 1–16, 2023, doi: 10.1371/journal.pone.0282926.
- [20] B. Amjad, M. Zeeshan, and M. O. Beg, “EMP-EVAL: A Framework for Measuring Empathy in Open Domain Dialogues,” 2023, [Online]. Available: <http://arxiv.org/abs/2301.12510>

- [21] L. Floridi and M. Chiriatti, "GPT-3: Its Nature, Scope, Limits, and Consequences," *Minds Mach.*, vol. 30, no. 4, pp. 681–694, 2020, doi: 10.1007/s11023-020-09548-1.
- [22] F. Barbieri, J. Camacho-Collados, L. Neves, and L. Espinosa-Anke, "TWEETEVAL: Unified benchmark and comparative evaluation for tweet classification," *Find. Assoc. Comput. Linguist. Find. ACL EMNLP 2020*, pp. 1644–1650, 2020, doi: 10.18653/v1/2020.findings-emnlp.148.
- [23] "GoEmotions: A Dataset of Fine-Grained Emotions | Papers With Code." [Online]. Available: <https://paperswithcode.com/paper/goemotions-a-dataset-of-fine-grained-emotions#code>
- [24] R. Safa, P. Bayat, and L. Moghtader, *Automatic detection of depression symptoms in twitter using multimodal analysis*, vol. 78, no. 4. Springer US, 2022. doi: 10.1007/s11227-021-04040-8.
- [25] J. J. Stephen, P. Prabu, and J. J. Stephen, "Detecting the magnitude of depression in Twitter users using sentiment analysis," vol. 9, no. 4, pp. 3247–3255, 2019, doi: 10.11591/ijece.v9i4.pp3247-3255.
- [26] J. Pool-cen, H. Carlos-martínez, and G. Hernández-chan, "Detection of Depression-Related Tweets in Mexico Using Crosslingual Schemes and Knowledge Distillation," pp. 1–18, 2023.