# A COMPARATIVE STUDY OF PROMPTING AND FINE-TUNING FOR BINARY TEXT CLASSIFICATION OF SUSTAINABLE DEVELOPMENT GOALS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

MERT ATAY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
COMPUTER ENGINEERING

AUGUST 2024

Approval of the thesis:

**A COMPARATIVE STUDY OF PROMPTING AND FINE-TUNING FOR BINARY TEXT CLASSIFICATION OF SUSTAINABLE DEVELOPMENT GOALS**

submitted by **MERT ATAY** in partial fulfillment of the requirements for the degree of **Master of Science in Computer Engineering Department, Middle East Technical University** by,

Prof. Dr. Naci Emre Altun
Dean, Graduate School of **Natural and Applied Sciences** _____

Prof. Dr. Halit Oğuztüzün
Head of Department, **Computer Engineering** _____

Assoc. Prof. Dr. Pelin Angın Ülküer
Supervisor, **Computer Engineering, METU** _____

**Examining Committee Members:**

Prof. Dr. Pınar Karagöz
Computer Engineering, METU _____

Assoc. Prof. Dr. Pelin Angın Ülküer
Computer Engineering, METU _____

Assist. Prof. Dr. Hakan Emekci
Graduate School, Applied Data Science, TED University _____

Date:19.08.2024

**I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.**

Name, Surname: Mert Atay

Signature :

# ABSTRACT

## A COMPARATIVE STUDY OF PROMPTING AND FINE-TUNING FOR BINARY TEXT CLASSIFICATION OF SUSTAINABLE DEVELOPMENT GOALS

Atay, Mert
M.S., Department of Computer Engineering
Supervisor: Assoc. Prof. Dr. Pelin Angın Ülküer

August 2024, 58 pages

The announcement of the 17 Sustainable Development Goals (SDGs) in 2015 by the United Nations has higlighted the need for automated document classification aligned with these global goals, as the volume of documents continues to increase. Current research focuses on traditional methods for automatic SDG classification, with Fine-Tuning being the latest approach, but neglects recent advancements in Large Language Models (LLMs) which excel in diverse tasks through Prompting or Prompt Engineering.

This research compares the performance of binary text classification for SDG relevancy using state-of-the-art LLMs ChatGPT and Gemini via Prompting techniques, including Zero-Shot and Few-Shot Prompting with varying context levels, against traditional Fine-Tuning applied to Transformer-based models BERT and its resource-efficient variant DistilBERT, which serve as benchmarks.

Based on comparative analysis, Fine-Tuning Transformer-based models outperforms Prompting with LLMs in SDG binary text classification, which aligns with existing research findings. Zero-Shot Prompting shows superior results for both models, with

performance improving as the context level of the Prompt increases. In contrast, both models exhibit lower performance with Few-Shot Prompting, with ChatGPT notably experiencing a significant decline. Increasing the number of examples in Few-Shot Prompts does not lead to improved performance for either model. Among Fine-Tuned models, DistilBERT and BERT perform similarly in SDG binary text classification, with DistilBERT being particularly advantageous due to its computational efficiency. However, LLMs have the potential to demonstrate improved text classification performance with the development of more capable models and effective Prompting techniques.

Keywords: Large Language Models, Fine-Tuning, Prompting, Prompt Engineering, Binary Text Classification, Sustainable Development Goals

# ÖZ

## SÜRDÜRÜLEBİLİR KALKINMA AMAÇLARININ İKİLİ METİN SINIFLANDIRMASINDA İSTEMLEME VE İNCE-AYAR ÜZERİNE KARŞILAŞTIRMALI BİR ÇALIŞMA

Atay, Mert
Yüksek Lisans, Bilgisayar Mühendisliği Bölümü
Tez Yöneticisi: Doç. Dr. Pelin Angın Ülküer

Ağustos 2024 , 58 sayfa

Birleşmiş Milletler'in 2015 yılında 17 Sürdürülebilir Kalkınma Amaçlarını (SDG) açıklaması ve ilgili belge hacminin artmaya devam etmesi nedeniyle bu küresel hedeflere uygun otomatik belge sınıflandırma ihtiyacı önem kazanmıştır. Mevcut araştırmalar, otomatik SDG sınıflandırması için geleneksel yöntemlere odaklanmakta olup, son yaklaşım olarak İnce-Ayar (Fine-Tuning) üzerine yoğunlaşmaktadır ve çeşitli görevlerde İstemleme (Prompting) veya İstem Mühendisliği (Prompt Engineering) yoluyla üstün etkinlik gösteren Büyük Dil Modellerindeki (LLM) gelişmeleri ihmal etmektedir.

Bu araştırma, en son teknoloji LLM'ler olan ChatGPT ve Gemini'nin, değişen bağlam düzeyleriyle Sıfır-Atışlı ve Az-Atışlı İstemleme teknikleri kullanarak, SDG ilişiği için ikili metin sınıflandırma performansını, kıyas noktası olarak hizmet veren Transformer-tabanlı modeller BERT ve kaynak verimli varyantı DistilBERT'e uygulanan geleneksel İnce-Ayar yöntemi ile karşılaştırmaktadır.

Karşılaştırmalı analize göre, İnce-Ayarlı Transformer-tabanlı modeller, ikili SDG metin sınıflandırmasında, LLM'ler ile İstemleme yönteminden daha iyi performans göstermektedir ve bu durum mevcut araştırma bulgularıyla örtüşmektedir. Sıfır-Atışlı İstemleme, her iki model için de üstün sonuçlar gösterirken, alınan performans İstem'in (Prompt) bağlam düzeyi arttıkça iyileşmektedir. Buna karşın, Az-Atışlı İstemleme ile her iki model de daha düşük performans sergilemektedir; özellikle ChatGPT belirgin bir düşüş yaşamaktadır. Az-Atışlı İstemleme'de örnek sayısının artırılması, her iki model için de herhangi bir performans iyileşmesine yol açmamaktadır. İnce-Ayarlı modeller arasında, DistilBERT ve BERT, ikili SDG metin sınıflandırmasında benzer performans gösterirken, DistilBERT hesaplama verimliliği nedeniyle özellikle avantajlıdır. Ancak, LLM'ler daha yetkin modellerin ve etkili İstemleme yöntemlerinin geliştirilmesiyle, metin sınıflandırmada iyileşme potansiyeli göstermektedir.

Anahtar Kelimeler: Büyük Dil Modelleri, İnce-Ayar, İstemleme, İstem Mühendisliği, İkili Metin Sınıflandırma, Sürdürülebilir Kalkınma Amaçları

This thesis is dedicated to my family, whose unwavering support made it possible.

## ACKNOWLEDGMENTS

I would like to express my gratitude to my advisor, Assoc. Prof. Dr. Pelin Angın Ülküer, for her guidance and support throughout this research. I am also thankful to my committee members, Prof. Dr. Pınar Karagöz and Assist. Prof. Dr. Hakan Emekci, for their insightful feedback and encouragement.

I extend my appreciation to all the faculty and staff of Middle East Technical University and the Computer Engineering Department, whose teachings and dedication have profoundly shaped my knowledge and greatly contributed to this thesis.

Finally, I am deeply grateful to my family and friends for their unwavering support and encouragement, which have been my motivation throughout this process.

# TABLE OF CONTENTS

ABSTRACT . . . . . v

ÖZ . . . . . vii

ACKNOWLEDGMENTS . . . . . x

TABLE OF CONTENTS . . . . . xi

LIST OF TABLES . . . . . xiv

LIST OF FIGURES . . . . . xvi

LIST OF ABBREVIATIONS . . . . . xvii

CHAPTERS

1 INTRODUCTION . . . . . 1

1.1 Motivation and Problem Definition . . . . . 1

1.2 Proposed Methods and Models . . . . . 2

1.3 Contributions and Novelties . . . . . 3

1.4 The Outline of the Thesis . . . . . 4

2 LITERATURE REVIEW . . . . . 5

2.1 Foundations . . . . . 5

2.1.1 Prompting . . . . . 5

2.1.2 Fine-Tuning . . . . . 6

2.2 Prompting for Text Classification . . . . . 6

2.3 Prompting vs. Fine-Tuning for Text Classification . . . . . 7
2.4 Surveys on Large Language Models . . . . . 9
2.5 Automated Classification of Sustainable Development Goals . . . . . 9
3 PRELIMINARIES . . . . . 11
3.1 Sustainable Development Goals . . . . . 11
3.2 Large Language Models . . . . . 13
3.2.1 Tokenization . . . . . 13
3.2.2 Attention Mechanism . . . . . 13
3.2.3 Activation Functions . . . . . 14
3.2.4 Layer Normalization . . . . . 14
3.2.5 Transformer Architecture . . . . . 14
3.2.6 BERT . . . . . 15
3.2.7 DistilBERT . . . . . 16
3.2.8 ChatGPT . . . . . 16
3.2.9 Gemini . . . . . 18
3.3 Prompting (Prompt Engineering) . . . . . 19
3.3.1 Zero-Shot Prompting . . . . . 19
3.3.2 Few-shot Prompting . . . . . 20
3.4 Evaluating Binary Text Classifiers . . . . . 20
3.4.1 Binary Cross Entropy Loss . . . . . 20
3.4.2 Accuracy . . . . . 21
3.4.3 Precision . . . . . 21
3.4.4 Recall . . . . . 22

3.4.5 F1 Score . . . 22

4 METHODOLOGY . . . 23

4.1 Dataset and Processing . . . 23

4.1.1 OSDG Community Dataset . . . 23

4.1.2 Binary Dataset Preparation . . . 24

4.2 Prompt Design . . . 25

4.2.1 Zero-Shot Prompts . . . 26

4.2.2 Few-Shot Prompts . . . 29

4.3 Binary Text Classification of SDG Relevance Using ChatGPT and Gemini with Prompting . . . 29

4.4 Fine-Tuning DistilBERT and BERT for Binary Text Classification of SDG Relevance . . . 30

5 RESULTS . . . 33

5.1 Evaluation of Prompting for Binary Text Classification of SDGs . . . 33

5.1.1 Zero-Shot Prompting Results . . . 33

5.1.2 Few-Shot Prompting Results . . . 36

5.2 Evaluation of Fine-Tuning for Binary Text Classification of SDGs . . . 39

6 DISCUSSION . . . 45

6.1 Future Work . . . 46

7 CONCLUSIONS . . . 47

REFERENCES . . . 49

APPENDICES

# LIST OF TABLES

TABLES

Table 3.1 Confusion Matrix for Binary Classification . . . . . 20

Table 4.1 OSDG-CD Features . . . . . 23

Table 4.2 SDG Labels Distribution . . . . . 24

Table 4.3 Breakdown of Information Components Across Context Levels in Zero-Shot Prompting . . . . . 26

Table 5.1 Evaluation of Zero-Shot Prompting Combined Across All SDGs and Context Levels . . . . . 33

Table 5.2 Evaluation of Zero-Shot Prompting Across Context Levels . . . . . 34

Table 5.3 Evaluation of Zero-Shot Prompting Across SDGs . . . . . 35

Table 5.4 Evaluation of Few-Shot Prompting Combined Across All SDGs and N-shots . . . . . 36

Table 5.5 Evaluation of Few-Shot Prompting Across N-shot . . . . . 37

Table 5.6 Evaluation of Few-Shot Prompting Across SDGs . . . . . 38

Table 5.7 Optimal Hyperparameters of DistilBERT Models . . . . . 39

Table 5.8 Optimal Hyperparameters of BERT Base Models . . . . . 40

Table 5.9 Average Performance of DistilBERT and BERT Models on Binary Text Classification of SDGs . . . . . 41

Table 5.10 Evaluation of DistilBERT Models . . . . . 42

Table 5.11 Evaluation of BERT Base Models . . . . . 43

Table 5.12 Overall Performance Comparison of Prompting and Fine-Tuning Approaches . . . . . 43

Table A1 ChatGPT Zero-shot Results Across SDGs and Context Levels . . . . 55

Table A2 Gemini Zero-shot Results Across SDGs and Context Levels . . . . . 56

Table A3 ChatGPT Few-shot Results Across SDGs and Context Levels . . . . 57

Table A4 Gemini Few-shot Results Across SDGs and Context Levels . . . . . 58

# LIST OF FIGURES

FIGURES

Figure 3.1 The 17 Sustainable Development Goals. Source: [1] . . . . . 12

Figure 3.2 Training steps of ChatGPT. Source: [2] . . . . . 17

Figure 3.3 Gemini models process mixed inputs (text, image, audio, video) and generate responses with integrated text and images. Source: [3] . . 18

Figure 4.1 Prompt-Based Binary Text Classification Pipeline for SDG Relevancy . . . . . 26

## LIST OF ABBREVIATIONS

| | |
|---|---|
| SDG | Sustainable Development Goal |
| NLP | Natural Language Processing |
| LLM | Large Language Model |
| ICL | In-Context Learning |

# CHAPTER 1

# INTRODUCTION

## 1.1 Motivation and Problem Definition

In 2015, the United Nations announced the 17 Sustainable Development Goals (SDGs), addressing a range of global issues including poverty, hunger, inequality, and climate change. SDGs are adopted by all member countries and they serve as a universal call to protect the planet and ensure that all people can enjoy peace and prosperity.

SDGs are gaining increasing importance and are being incorporated into the core strategies of governments, organizations and businesses. Corporate Social Responsibility (CSR) has emerged as a form of international private business self-regulation that introduces the alignment with SDGs which is particularly important. The CSR concept requires companies to now also have a positive impact on global problems. This shift signifies the growing relevance of the SDG in corporate activities and public communication.

As different domains increasingly include SDGs, it becomes necessary for experts to have automated ways of classifying documents based on how they align with these goals, especially given the growth in document volumes. In addition, this challenge includes timely and accurate assessments which would be useful in informing decision-making, reporting as well as strategic planning.

Automated solutions can address this challenge by providing scalable solutions to manage and analyze the growing corpus of SDG-related documents. They can allow efficient tracking of progress, improve transparency levels, and help firms meet their CSR objectives by ensuring that their operations are consistently aligned with SDGs.

Furthermore, automated solutions can identify gaps and opportunities for improvement, thus driving more effective and impactful contributions to global sustainability goals.

Natural Language Processing (NLP), Machine Learning, and Deep Learning methodologies provide solutions to automate document classification. However, current literature on automatic classification of SDGs primarily focuses on traditional methods, with the latest approach being Fine-Tuning and does not levereage the recent advancements in Large Language Models (LLMs). With the advancements in LLMs like ChatGPT [4] from OpenAI and Gemini [5] from Google, new chat-based text generation models are emerging. These models enable textual interactions and can be utilized across a spectrum of tasks—from everyday activities such as writing emails to technical tasks like coding, data annotation, and importantly, text classification.

These interactions with the models are known as Prompting or Prompt Engineering, and they play a crucial role not only in defining the task for the model but also in obtaining optimal results and outputs. The models are highly sensitive to even minor changes in the Prompt text, highlighting the importance of carefully crafting Prompts to achieve desired outcomes efficiently. Consequently, researchers are keen on exploring Prompting and the capabilities of LLMs across various tasks, including text classification and comparing their performance with traditional methodologies such as Fine-Tuning.

## 1.2 Proposed Methods and Models

This research aims to compare the performance of binary text classification for SDG relevancy using the latest LLMs via Prompting and traditional Fine-Tuning methods on Transformer-based models.

For comparative analysis, two prominent LLMs are employed for Prompting-based classification: ChatGPT and Gemini. Two Prompting approaches are utilized: Zero-Shot Prompting with various context levels for the binary SDG classification task, and Few-Shot Prompting with varying numbers of data examples.

For Fine-Tuning, one of the foundational models, BERT, and its resource-efficient variant, DistilBERT, are employed, along with grid search-based hyperparameter optimization, to serve as benchmarks for evaluating SDG classification performance using the available dataset.

Research utilizes the OSDG-CD [6][7] dataset, which consists of publicly available text excerpts labeled according to their relevance to one of the 17 SDGs. Each excerpt also includes an agreement score reflecting the volunteers' consensus on the suggested label, framing the task as a binary classification problem.

### 1.3 Contributions and Novelties

Our contributions are as follows:

- We conduct a comparative analysis of Prompting versus Fine-Tuning for binary text classification, focusing on SDG relevancy.

- We present a detailed comparison of Zero-Shot Prompting across various context levels and Few-Shot Prompting with varying numbers of examples for SDG relevancy classification.

- We compare binary text classification capabilities of state-of-the-art large language models ChatGPT and Gemini using Prompting, within the context of SDG relevancy.

- We evaluate binary text classification performance on SDG relevancy using foundational Transformer-based models DistilBERT and BERT. Specifically, leveraging DistilBERT allows us to assess the performance of a resource-efficient model on this task.

- We deliver highly accurate Fine-Tuned models and Prompt designs for LLMs in binary text classification task, specifically focused on SDG relevancy.

## 1.4 The Outline of the Thesis

The first chapter introduces the research, outlining its motivation and problem definition. It discusses the methods and models employed and highlights the research's contributions and innovations. Chapter 2 presents a literature review, offering an overview of existing research and identifying areas that require further study. Chapter 3 provides preliminary background information on the methodologies and technologies used, facilitating a better understanding of the research and its context. Chapter 4 offers a detailed description of the research methodology. Chapter 5 presents and analyzes the comparative results in detail. It draws conclusions based on these findings, which are further summarized in Chapter 6, along with an overview of the research and its outcomes.

# CHAPTER 2

# LITERATURE REVIEW

Following recent advancements in Generative AI and Large Language Models (LLMs), the exploration of their capabilities and use cases has become a focal point of active research. Among the most prominent design patterns in Generative AI are Fine-Tuning and Prompting [8] also known as Prompt Engineering. Researchers are particularly interested in exploring Prompting and its application in adopting Generative AI across various tasks, including text classification, and comparing its performance with more traditional methodologies such as Fine-Tuning.

## 2.1 Foundations

### 2.1.1 Prompting

Radford et al. [9], Brown et al. [10], and Dai et al. [11] have conducted foundational research on employing Prompt Engineering for performing multiple tasks with LLMs. Radford et al. [9] demonstrated that Language Models can learn common NLP tasks without any supervised training, paving the way for Zero-Shot Prompting. Brown et al. [10] introduced GPT-3 model and showed that, without gradient updates or Fine-Tuning, GPT-3 can achieve high performance on numerous NLP tasks with Few-Shot demonstrations through textual interaction with the model, i.e., Few-Shot Prompting. Dai et al. [11] provided an explanation of the mechanism behind the In-Context Learning capability of LLMs, which is optimization based on meta-gradients. They supported their explanation by comparing the behaviors of In-Context Learning and Fine-Tuning in real-world applications, hence explaining the validity of Zero-Shot and Few-Shot Prompting.

### 2.1.2 Fine-Tuning

Howard and Ruder [12] laid the foundations for employing LLMs for Fine-Tuning, proposing the groundbreaking Transfer Learning method Universal Language Model Fine-tuning (ULMFiT), which made Fine-Tuning accessible for NLP tasks. Vaswani et al. [13] introduced a simple yet revolutionary Transformer architecture based solely on the attention mechanism. This architecture demonstrated strong generalization capabilities, even with limited data, across various English tasks. Their work laid the foundation for Language Models that remain state-of-the-art to this day. Following these advancements, Devlin et al. [14] introduced BERT, a Pre-trained Language Model based on the Transformer architecture. BERT significantly simplified the process of Fine-Tuning for a wide range of NLP tasks.

## 2.2 Prompting for Text Classification

Exploring the capabilities of Prompting is an active research topics among scholars and there has been recent studies focusing on the application of Prompt Engineering for text classification across diverse domains. Salinas and Morstatter [15] have delved into Prompting, investigating how variations in the structure of a Prompt can influence decisions made by LLMs. Their study demonstrated that even minor alterations, such as adding an extra space character at the end of a Prompt, can significantly impact the answer generated by an LLM, highlighting the importance of careful crafting of Prompts for any downstream task.

Parizi et al. [16] conducted a comparative analysis of different Prompt Engineering methodologies for text classification within the legal domain. They found that certain methodologies do not effectively transfer to the legal domain, especially when dealing with complexities such as lengthy documents and domain-specific language. This underscores the potential limitations of current Prompt Engineering approaches and emphasizes how dataset characteristics and domain-specific nuances can limit the performance of Prompt-based usage of LLMs.

Fatemi et al. [17] investigated the performance of the GPT model and Prompting

in Zero-Shot multi-class text classification specifically within the journalism domain. They demonstrated strong performance, underscoring the effectiveness of LLMs for automated text classification in domain-specific applications.

Loukas et al. [18] explored the capabilities of LLMs in Few-Shot text classification within the financial domain, demonstrating effective reduction in operational costs compared to traditional Few-Shot approaches, making LLMs useful for resource limited scenarios.

Gunes and Florczak [19] focused on the policy document domain, employing LLMs to automatically classify policy documents. They emphasized the limitations of relying solely on LLMs without any human expert intervention and noted improved accuracy with increased human effort.

Dao [20] conducted a performance comparison of LLMs in the English education domain using the VNHSGE English dataset. The results revealed an intriguing finding: LLMs outperformed Vietnamese students in English proficiency, highlighting their potential for teaching and learning the English language at the high school level.

### 2.3 Prompting vs. Fine-Tuning for Text Classification

With recent advancements in LLMs and Prompting techniques, researchers are increasingly interested in comparing traditional Fine-Tuning methodologies with Prompting. Several studies have explored these two approaches across various domains.

Chae and Davidson [21] employed Zero-Shot Prompting to leverage LLMs for text classification. They compared these results with those obtained from traditional machine learning and deep learning approaches to text classification, demonstrating the effectiveness of LLMs as Zero-Shot text classifiers.

Trajanov et al. [22] compared the text classification performance of ChatGPT and Fine-Tuned BERT models within the climate domain. They highlight that Fine-Tuned models outperform LLMs for tasks related to climate domain.

Clavié et al. [23] conducted a comparison between Prompt Engineering, includ-

ing various aspects of Prompting, and traditional methodologies for text classification in the practical context of job type classification. Their findings indicated that LLMs achieve best performance with well-designed Prompts compared with traditional methodologies. Furthermore, they corroborated the findings of Salinas and Morstatter [15], emphasizing that minor adjustments to Prompts can yield significantly different results, highlighting the critical role of Prompt structure in achieving high performance.

Bosley et al. [24] conducted a comparison between Fine-Tuning smaller LLMs and In-Context Learning, specifically Prompt Engineering with larger LLMs, for text classification within the political science domain. Their preliminary findings suggest that given the availability of labeled data, Fine-Tuning continues to outperform prompting in text classification tasks.

Yu et al. [25] focused on comparing the performance of open-source, closed-source, and traditional LLMs in text classification tasks, providing insights into the effectiveness of prompting with larger models versus Fine-Tuning with smaller models. Their findings suggest that Fine-Tuning smaller LLMs can match or even outperform prompting with larger LLMs in these tasks.

Wang et al. [26] conducted a comparative study between In-Context Learning (ICL) and supervised learning, exploring various label perturbations to explain ICL's learning mechanisms. Their findings underscore that the quality of groundtruth labels significantly influences ICL performance, whereas data imbalance has negligible impact. Moreover, they emphasize that ICL achieves increasingly competitive performance relative to supervised learning as model size gets larger.

Eisenstadt et al. [27] compared the performance of Fine-Tuning and Prompt Engineering in text classification, focusing on sales call transcription classification. They highlight that Fine-Tuned models, tailored to the specific task, outperform LLMs in text classification within this context.

Qiu and Jin [28] compared Prompting with Fine-Tuning using ChatGPT and a Fine-Tuned BERT model in the development of intelligent design support systems. Their study reveals that ChatGPT achieves comparable performance in sentence-level clas-

sification but encounters challenges with shorter sequences. Additionally, they highlight a significant performance boost for ChatGPT when Few-Shot Prompting is employed.

## 2.4 Surveys on Large Language Models

With recent advancements in the field, LLMs have become a focal point of research interest. In addition to studies on prompt engineering and comparisons with Fine-Tuning, several survey papers have been published on LLMs. These surveys offer comprehensive overviews of the models, their capabilities, and areas for improvement.

Surveys by Raiaan et al. [29], Hadi et al. [30], Kalyan [31], and Pahune and Chandrasekharan [32] provide valuable insights into the current state of research in the realm of LLMs.

## 2.5 Automated Classification of Sustainable Development Goals

One of the foundational works in boosting automated text classification for Sustainable Development Goals (SDGs) is the OSDG Community Dataset [7]. This dataset comprises over 42,000 publicly available text excerpts labeled by more than 1,000 volunteers, aligning each text with relevant SDGs. The availability of such labeled data has paved the way for employing machine learning-based automated text classification for SDGs.

Following the publishing of OSDG Community dataset, Pukelis et al. [33] introduced an open-source tool, OSDG, designed to identify relevant SDGs within a given text.

Similarly, leveraging the OSDG dataset, Angın et al. [34] investigated the Fine-Tuning of BERT and RoBERTa models for classifying the relevance of sustainability reports to SDGs.

Rodríguez Medina [35] curated their own dataset to evaluate and compare different deep learning methodologies for automated text classification related to SDGs. Their

findings highlight that BERT achieved the highest performance, while support vector machines and logistic regression classifiers showed comparable results with the added advantage of computational efficiency.

Hajikhani and Suominen [36] trained traditional machine learning models to classify patents based on their relevance to SDGs, viewing patents as indicators of innovation. Leveraging these models, they conducted an analysis using patent samples from the European Patent Office (EPO) to explore how patents and recent innovations align with SDGs.

Guisiano et al. [37] developed SDG-Meter, a Fine-Tuned BERT model specifically tailored for multilabel classification of texts based on their relevance to the SDGs.

# CHAPTER 3

# PRELIMINARIES

## 3.1 Sustainable Development Goals

In 2015, the United Nations announced the 17 Sustainable Development Goals (SDGs) as part of the The 2030 Agenda for Sustainable Development, addressing a range of global issues including poverty, hunger, inequality, and climate change. SDGs are adopted by all member countries and they serve as a universal call to protect the planet and ensure that all people can enjoy peace and prosperity.

Each SDG addresses a distinct global issue, as illustrated in Figure 3.1. Furthermore, each SDG includes targets and indicators developed to track the progress toward achieving the respective global goal. Targets signify a specific objective aimed at advancing sustainable development on a global scale. Indicators refer to measurable parameters or metrics utilized for monitoring and evaluating the progress towards these targets. A Target and the respective indicators of SDG 7 can be seen as examples.

**Examples of Targets and Indicators for SDG 7: Affordable and Clean Energy**

**Target 7.1**: By 2030, ensure universal access to affordable, reliable and modern energy services.
**Indicator 7.1.1**: Proportion of population with access to electricity.
**Indicator 7.1.2**: Proportion of population with primary reliance on clean fuels and technology.


Figure 3.1: The 17 Sustainable Development Goals. Source: [1]

## 3.2 Large Language Models

This section covers the foundational concepts that are essential for understanding the workings of LLMs. For further details on LLMs, please refer to surveys by Raiaan et al. [29], Hadi et al. [30], Kalyan [31], and Pahune and Chandrasekharan [32]. These surveys provide a comprehensive overview of the mechanisms, applications, and limitations of LLMs.

### 3.2.1 Tokenization

Tokenization is a critical text preprocessing step in LLMs, which involves breaking down a sequence of text into individual units called "tokens". Based on the implementation of the model, tokens can be characters, symbols, or words, and there are various tokenization algorithms tailored for specific tasks.

### 3.2.2 Attention Mechanism

The Attention Mechanism represents a pivotal advancement in machine learning, paving the way for the foundational Transformer architecture. It aims to enhance a machine learning model's performance by enabling it to selectively focus on different parts or tokens of the input and output sequences, thereby capturing their importance and relevance. This is achieved by dynamically assigning varying weights to distinct tokens within a sequence. During prediction, the mechanism then weighs output tokens according to their relevance to the input sequence. There are various attention mechanisms in the literature, but Transformers utilize the Self-Attention Mechanism, which focuses on a single sequence at a time. This mechanism enables the sequence to learn from itself, rather than assessing the relevance of an output sequence to the input sequence directly.

### 3.2.3 Activation Functions

Activation functions are essential components of neural networks, integral to the architecture of LLMs. They compute the weighted sum of a "neuron," also known as a "perceptron," which models the functioning of a biological neuron using a mathematical function. Afterwards, by adding bias value into this sum Activation Functions determine whether the neuron should activate. Activation Functions are crucial as they introduce non-linearity into neuron outputs, thereby enhancing the network's ability to learn complex tasks. Commonly adopted activation functions include the sigmoid function, the Rectified Linear Unit (ReLU) function, and the hyperbolic tangent (tanh) function.

### 3.2.4 Layer Normalization

Layer Normalization is a foundational technique employed in deep learning to ensure a stable training process and enhance the performance of neural networks, which are building blocks of LLMs. It computes the mean and standard deviation across each instance across all features, effectively addressing the internal covariate shift (ICS) problem. Layer Normalization is widely used in prominent LLMs such as GPT-3 and BERT to achieve effective training.

### 3.2.5 Transformer Architecture

Transformer architecture is the main building block of LLMs. It is a deep learning architecture proposed by Vaswani et al. [13] which is based on the Attention Mechanism, demonstrating strong generalization capabilities even with limited data. Currently, most state-of-the-art LLMs such as ChatGPT and Gemini are built based on the Transformer architecture.

Transformer architecture only relies on self-attention mechanism for computing input and output representations, unlike previous architectures that rely on RNNs or convolutions. Transformer architecture incorporates an Encoder-Decoder model. Encoder processes the input sequence to produce "hidden states" or matrix representations of

the input, while the Decoder uses these positional matrix representations to generate the output iteratively. Encoder-Decoder model in Transformers consist of multiple stacked layers having the same structure. Input sequence is processed through each Encoder layer sequentially. Similarly, In Decoder, input comes from the last Encoder layer and the previous Decoder layer. This layered structure allows the Transformer to effectively capture complex relationships in data and maintain a consistent processing approach.

### 3.2.6 BERT

Bidirectional Encoder Representations from Transformers (BERT) [14] is a language model based on the Transformer architecture. It is a groundbreaking model as it enables efficient fine-tuning for a wide range of NLP tasks. BERT model has two sizes: BERT-Base with 12 layers, each with a hidden size of 768 and 12 self-attention heads (110 million parameters) and BERT-Large with 24 layers with a hidden size of 1024 and 16 self-attention heads (340 million parameters).

BERT is pre-trained on a large corpus of textual data, comprising BookCorpus (800 million words) and English Wikipedia (2,500 million words). It is trained using two primary tasks: Masked Language Modeling (MLM) and Next Sentence Prediction (NSP). In MLM task, random portions of input tokens are masked, and the model predicts these masked tokens to train its underlying representations. NSP task, as the name suggests, involves predicting whether two given sentences logically follow one another, helping the model capture contextual relationships between pairs of sentences.

Fine-tuning BERT utilizes its underlying Self-Attention Mechanism to adapt the pre-trained model for downstream tasks. This involves presenting task-specific input-output pairs to the model and adjusting all parameters during training. Compared to pre-training, Fine-Tuning is significantly more computationally efficient, enabling optimal results and widespread applicability of the model across various NLP tasks.

### 3.2.7 DistilBERT

DistilBERT [38] is a distilled version of BERT, designed to be smaller and faster while having much of BERT's language understanding capabilities. It uses a combination of knowledge distillation techniques and a more compact architecture, resulting in both a 40% reduction in size compared to BERT-base and a 60% increase in processing speed, while retaining 97% of BERT's language understanding capability. This makes DistilBERT highly efficient for a range of natural language processing tasks, including text classification.

DistilBERT maintains the fundamental architecture of BERT but simplifies it significantly by removing token-type embeddings and the pooler. Additionally, it reduces the number of layers by half compared to the original BERT model. The model is distilled using large batch sizes and gradient accumulation techniques, discarding Next Sentence Prediction (NSP) task, which follows the advancements in BERT training methods [39].

### 3.2.8 ChatGPT

Generative Pre-trained Transformer (GPT) is an advanced large language model developed by OpenAI, renowned for its ability to generate coherent and contextually relevant text [40]. GPT models achieve this through extensive pre-training on large-scale textual datasets, followed by Fine-Tuning for specific tasks such as text classification and question-answering.

OpenAI's work in AI research has led to the development of transformative models including GPT-3, GPT-3.5 and GPT-4. Following the success of GPT-3, OpenAI furthered its research to develop ChatGPT [4], specifically designed for conversational applications. ChatGPT showcases improved abilities in contextual understanding, coherence and generating responses to earlier models. It builds upon the foundations, i.e., fine-tuned from, established by GPT-3, GPT-3.5, and the latest iteration, GPT-4, continually refining its architecture to improve conversational capabilities.

The GPT-3.5 model employs 13 Transformer blocks, each featuring 12 attention

heads and 768 hidden units to process sequences of tokens. In addition to its core Transformer architecture, GPT-3.5 integrates crucial components such as layer normalization, residual connections, and positional embeddings. These elements stabilize the training process and enhance performance across diverse language tasks, including text generation, comprehension, and translation. This architecture positions GPT-3.5 as a robust method for modeling natural language sequences, consistently showing state-of-the-art performance in various linguistic applications. [40]

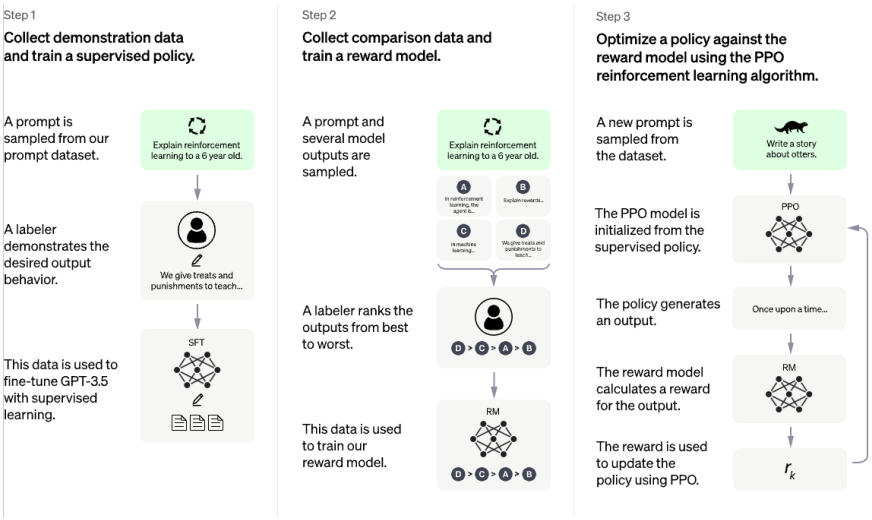


Figure 3.2: Training steps of ChatGPT. Source: [2]

ChatGPT, which was Fine-Tuned from the GPT-3.5 series, was trained using Reinforcement Learning from Human Feedback (RLHF), a method similar to its counterpart InstructGPT but with changes in data collection [2]. Initially, supervised Fine-Tuning was employed by OpenAI, where human AI trainers creaft dialogues in both human user and AI assistant roles. Then, Trainers with model-generated suggestions are used to aid in formulating responses, resulting in a dataset tailored for dialogue format.

In reinforcement learning, a reward model is implemented to assess the quality ranking of two or more model responses. This model utilizes data derived from interactions between AI trainers and the chatbot, where the chatbot asks the trainers to rank alternative model outputs. These ranked responses are then employed in Proximal Policy Optimization to further fine-tune the model, undergoing multiple iterations of refinement.

For further details on ChatGPT, including its background, applications, key challenges, bias, ethics, limitations, and future scope, the following reference is recommended [40].

### 3.2.9 Gemini

Gemini [3], developed by Google, is designed as a versatile family of models tailored for multimodal tasks including text, image, audio, and video processing (Figure 3.3). Gemini model family is designed to perform well in both general tasks that involve various data types and specialized tasks that require domain-specific reasoning. Gemini 1.0 offers three model sizes: Ultra for handling complex tasks, Pro for scalable performance, and Nano optimized for use on devices like smartphones and small computers.

Gemini models extend Transformer decoders with enhanced architectures and optimizations tailored for stable training at scale and efficient inference on Google's Tensor Processing Units (TPU). The models are designed to process inputs with up to 32k context length and utilize efficient attention mechanisms, such as Multi-Query Attention.

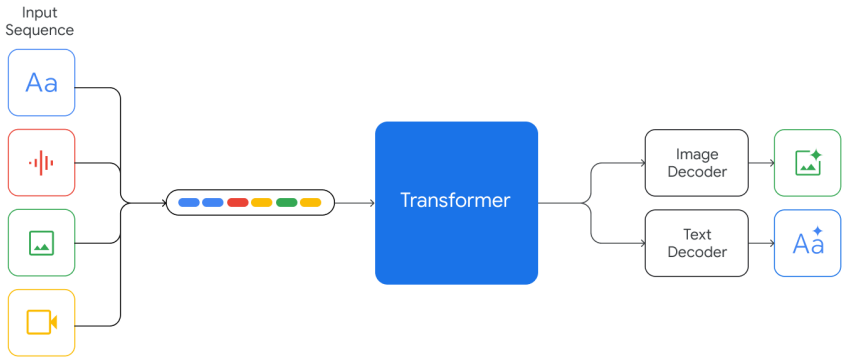


Figure 3.3: Gemini models process mixed inputs (text, image, audio, video) and generate responses with integrated text and images. Source: [3]

Gemini models are trained on a diverse dataset sourced from web documents, books, and code, incorporating image, audio, and video inputs. After training on a representative sample of this corpus, the SentencePiece tokenizer [41] is employed to

improve vocabulary inference and optimize tokenization efficiency. Gemini also employs advancements in training algorithms, datasets, and infrastructure. Pro model uses scalable infrastructure and optimized learning algorithms, enabling accelerated pre-training over a few weeks compared to the larger Ultra model. Nano model employs distillation techniques to develop compact models tailored for on-device applications such as summarization and reading comprehension.

Finally, for quality assurance, heuristic rules and model-based classifiers are used to ensure the reliability and safety of Gemini models, aligning them with Google's stringent policies.

## 3.3 Prompting (Prompt Engineering)

Prompting [42] or Prompt Engineering involves creating task-specific instructions, referred to as "Prompts" which guide the outputs of pre-trained LLMs without altering their underlying parameters. This approach is pivotal for enhancing the adaptability of LLMs across various tasks and domains, offering a flexible alternative to traditional methods that often require extensive retraining or Fine-Tuning for downstream tasks.

Nowadays, Prompt Engineering includes a wide range of techniques, from foundational Zero-Shot and Few-Shot Prompting to more complex methods like Chain Of Code Prompting. For an in-depth taxonomy and further details of each technique, the following survey is recommended [42].

### 3.3.1 Zero-Shot Prompting

Zero-Shot prompting [9] is a method revolves around crafting carefully constructed Prompts to guide the model towards performing new tasks. Zero-Shot Prompting involves incorporating task descriptions directly within the prompt, thereby eliminating the requirement for labeled training datasets. The model then utilizes its pre-existing knowledge to generate predictions for the new task solely based on the information provided in the prompt. This innovative technique demonstrates how well LLMs can handle a wide range of tasks with very little guidance.

### 3.3.2 Few-shot Prompting

Few-Shot prompting [10] involves providing a small set of input-output examples to help models understand and perform a new task. Even with minimal high-quality examples, models demonstrate improved performance on challenging tasks. However, integrating these examples to Prompts requires additional tokens, which can be problematic for longer text inputs. Moreover, the selection and structure of Prompt examples play a crucial role in getting optimal results from the models. Nonetheless, Few-Shot Prompting significantly enhances the capabilities of pre-trained LLMs like GPT-3 in tackling various complex tasks.

## 3.4 Evaluating Binary Text Classifiers

Loss metric Binary Cross Entropy Loss and metrics Precision, Recall, F1 Score and Accuracy are commonly used to evaluate a binary classifier's performance. During detailed breakdown of the metrics, please refer to the Confusion Matrix shown in Table 3.1, which summarizes the classifier's predictions against actual classes (positive and negative) fur further clarification.

Table 3.1: Confusion Matrix for Binary Classification

| **Actual / Predicted** | **Positive (1)** | **Negative (0)** |
|---|---|---|
| **Positive (1)** | True Positive (TP) | False Negative (FN) |
| **Negative (0)** | False Positive (FP) | True Negative (TN) |

### 3.4.1 Binary Cross Entropy Loss

Binary Cross Entropy Loss, also known as Log Loss, is a loss function used to evaluate the prediction error of binary classifiers in Machine Learning. The loss value increases as the prediction probability diverges from the groundtruth label. It is calculated by:

$$\text{BCELoss} = -\frac{1}{N}\sum_{i=1}^{N}\left[y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)\right]$$

where:

- $N$ is the total number of dataset samples,
- $y_i$ is the true label of the $i$-th sample ($y_i \in \{0, 1\}$),
- $\hat{y}_i$ is the predicted probability of the $i$-th sample belonging to positive class (class 1).

### 3.4.2 Accuracy

Accuracy measures the overall correctness of the classifier. It is calculated by the number of correct predictions devided by the total number of predictions. While Accuracy is a straightforward metric, it is not suitable for measuring the performance of a classifier, especially in cases of imbalanced datasets where it can overlook the true performance due to its tendency to prioritize majority classes over minority ones.

$$\text{Accuracy} = \frac{\text{True Positives} + \text{True Negatives}}{\text{Total Predictions}}$$

### 3.4.3 Precision

Precision metric is calculated by the number of true positives divided by the total number of positive predictions (true positives and false positives). Precision can be seen as a metric focusing on quality where it tells us about accuracy of correctly predicted positive instances.

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}$$

### 3.4.4 Recall

Recall metric is calculated by the number of positive samples correctly classified as positive divided by the total number of positive samples. It can be seen as a metric focusing on quantity where it shows the models ability to identify the positive instances.

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$$

### 3.4.5 F1 Score

F1 score is calculated as the harmonic mean of Precision and Recall metrics. It provides a scoring which balances Precision and Recall.

$$\text{F1 Score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

# CHAPTER 4

# METHODOLOGY

## 4.1 Dataset and Processing

### 4.1.1 OSDG Community Dataset

As the dataset, OSDG Community Dataset (OSDG-CD) Version 2024.01 [7] is used. The dataset consists of publicly available collection of text excerpts, validated by a diverse global community of volunteers from numerous countries through the OSDG Community Platform (OSDG-CP) [6]. Each text excerpt was assessed by volunteers for alignment with one of the United Nations Sustainable Development Goals (SDGs) by deciding whether to accept or reject the suggested SDG label. This highlights the dataset's emphasis on binary classification.

Table 4.1: OSDG-CD Features

| Feature Name | Description |
|---|---|
| doi | Digital Object Identifier of the original document |
| text_id | Unique text identifier |
| text | Text excerpt from the document |
| sdg | The SDG the text is validated against |
| labels_negative | The number of volunteers who rejected the suggested SDG label |
| labels_positive | The number of volunteers who accepted the suggested SDG label |
| agreement | Agreement score based on the formula: $agreement = \frac{\lvert labels\_positive - labels\_negative \rvert}{labels\_positive + labels\_negative}$ |

### 4.1.2 Binary Dataset Preparation

A series of processing steps were applied to prepare the dataset for binary text classification. Initially, entries with an agreement score greater than 0.5 were retained to ensure the reliability of the ground-truth data and eliminate ambiguous labels. The distribution of data across 16 SDG labels before and after applying the criterion based on agreement score is as follows:

Table 4.2: SDG Labels Distribution

| SDG | Data Count (Total) | Data Count (agreement > 0.5) | Pos Samples | Neg Samples | Total Samples |
|---|---|---|---|---|---|
| 1 | 2734 | 1611 | 705 | 705 | 1410 |
| 2 | 2457 | 1235 | 705 | 705 | 1410 |
| 3 | 2689 | 2078 | 705 | 705 | 1410 |
| 4 | 3740 | 2861 | 705 | 705 | 1410 |
| 5 | 4338 | 3270 | 705 | 705 | 1410 |
| 6 | 2815 | 1650 | 705 | 705 | 1410 |
| 7 | 3048 | 2307 | 705 | 705 | 1410 |
| 8 | 1509 | 1003 | 705 | 705 | 1410 |
| 9 | 2510 | 1763 | 705 | 705 | 1410 |
| 10 | 2244 | 1221 | 705 | 705 | 1410 |
| 11 | 2277 | 1491 | 705 | 705 | 1410 |
| 12 | 1108 | 706 | 705 | 705 | 1410 |
| 13 | 2102 | 1462 | 705 | 705 | 1410 |
| 14 | 1141 | 894 | 705 | 705 | 1410 |
| 15 | 2472 | 1544 | 705 | 705 | 1410 |
| 16 | 5451 | 3308 | 705 | 705 | 1410 |
| Total: | 42,635 | 28,404 | 11,280 | 11,280 | 22,560 |

The labels distribution indicates an imbalance within the dataset. To avoid introducing bias towards SDGs with more number of examples, the same number of positive and negative samples are taken from each SDG. Specifically, as many positive samples as the SDG with the fewest examples (SDG 12 has 706 examples, with 705 being a multiple of 15 to ensure an equal distribution of negative samples) are selected. For SDGs with a larger number of positive samples, the samples are selected randomly.

As the next processing step to create the binary dataset, an equal number of data were sampled from each remaining SDG label to serve as negative examples. Negative

examples from each each SDG were selected randomly. To elaborate, there are 705 positive examples for each given SDG, where the text excerpt is relevant. In that case, 705 negative examples are drawn from the remaining 15 SDGs, resulting in 47 examples being sampled from each remaining SDG. Subsequently, the labels in each SDG's binary dataset were assigned "1" for positive examples and "0" for negative examples.

As the final step, each SDG binary dataset was divided into training, validation, and test sets with an 80:10:10% ratio respectively. A random seed was utilized to ensure the reproducibility of the data splits and the selection of negative examples.

The average character length of text excerpts in the binary dataset (with 1,410 examples selected from each SDG) is 610 characters, and no additional preprocessing has been performed. For BERT models, which have a maximum limit of 512 tokens, truncation is applied to texts longer than 512 tokens.[1]

## 4.2 Prompt Design

In the context of LLMs, a "Prompt" refers to a specific input provided to guide the model in generating a desired output, structured to elicit responses tailored to particular tasks or formats. Crafting well-structured prompts is essential for achieving accurate and high-quality responses from an LLM. During prompt design process, guidelines and documentation from OpenAI [43] and Google [44] were followed. The structure of parametric Prompts were based on open-source libraries Scikit-LLM [45] and Promptify [46].

Prompts were also designed to guide the LLMs to respond in the anticipated output format. For binary classification, the expected output is solely the classification result in integer format: 1 denoting relevance of the input text excerpt to the given SDG, while 0 denoting irrelevance.

[1] Note that BERT models use a subword tokenizer (WordPiece), where one token roughly corresponds to one word rather than one character.

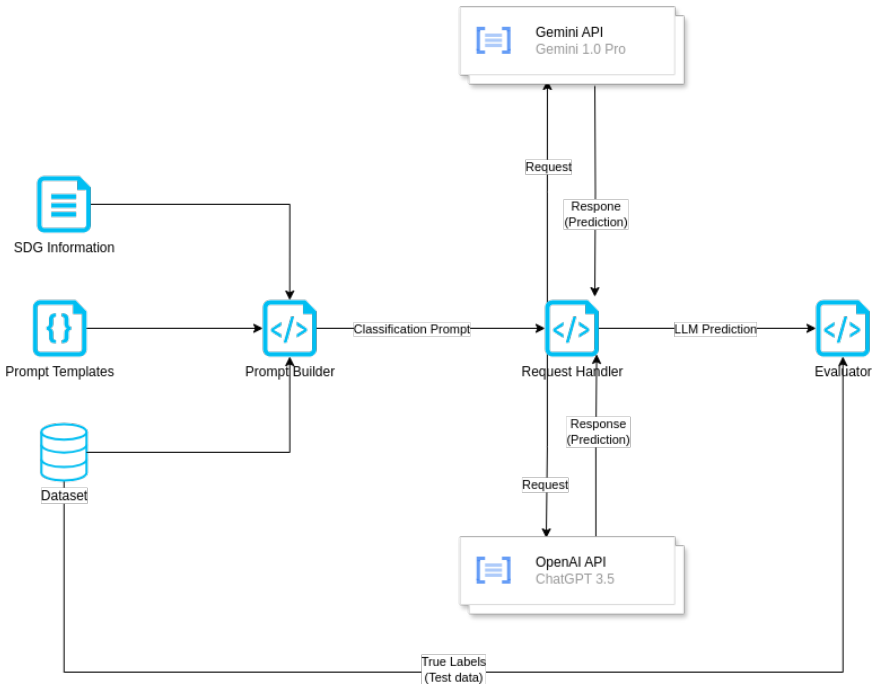


Figure 4.1: Prompt-Based Binary Text Classification Pipeline for SDG Relevancy

### 4.2.1 Zero-Shot Prompts

Zero-Shot Classification Prompts employed a parametric structure where there are three contextual levels for each given SDG. Prompts with different context levels incorporated varying contextual data about SDGs, including a descriptive paragraph and detailed information about their targets and indicators (see Chapter 3). In all contextual levels, the identifier numbers and titles of the SDGs are used. Table 4.3 provides a breakdown of contextual data included across three context levels:

Table 4.3: Breakdown of Information Components Across Context Levels in Zero-Shot Prompting

| Context Level | SDG Number | SDG Title | SDG Description | SDG Targets & Indicators |
|---|---|---|---|---|
| 1 | ✓ | ✓ | ✗ | ✗ |
| 2 | ✓ | ✓ | ✓ | ✗ |
| 3 | ✓ | ✓ | ✓ | ✓ |

Contextual data for each SDG was stored locally within the classification system.

Builder scripts were implemented to craft classification Prompts by compiling the parametric Prompts based on the context level and the respective SDG during classification.

**Example of SDG Descriptions Used in Zero-Shot Prompts. Description of SDG 14: Life Below Water**

Goal 14 focuses on conserving and sustainably using the oceans, seas, and marine resources for sustainable development by preventing marine pollution, protecting marine and coastal ecosystems, and promoting sustainable fisheries management.

**Zero-Shot Context Level 1 Prompt Template**

Task: Perform binary text classification to determine the relevance of a text excerpt to a specified United Nations' Sustainable Development Goal (SDG).

Information:
- SDG Number: "'{sdg_number}"'
- SDG Title: "'{sdg_title}"'
- Text Excerpt: "'{text_excerpt}"'

Instructions:
1. Analyze the given text excerpt and classify it as relevant (1) or irrelevant (0) to the specified SDG.
2. Respond only with the integer classification result.

**Zero-Shot Context Level 2 Prompt Template**

Task: Perform binary text classification to determine the relevance of a text excerpt to a specified United Nations' Sustainable Development Goal (SDG).

Information:
- SDG Number: “‘{sdg_number}“‘
- SDG Title: “‘{sdg_title}“‘
- SDG Description: “‘{sdg_description}“‘
- Text Excerpt: “‘{text_excerpt}“‘

Instructions:
1. Analyze the given text excerpt and classify it as relevant (1) or irrelevant (0) to the specified SDG.
2. Utilize the provided SDG description to enhance classification accuracy.
3. Respond only with the integer classification result.

**Zero-Shot Context Level 3 Prompt Template**

Task: Perform binary text classification to determine the relevance of a text excerpt to a specified United Nations' Sustainable Development Goal (SDG).

Information:
- SDG Number: “‘{sdg_number}“‘
- SDG Title: “‘{sdg_title}“‘
- SDG Description: “‘{sdg_description}“‘
- Targets and Indicators: “‘{targets_and_indicators}“‘
- Text Excerpt: “‘{text_excerpt}“‘

Instructions:
1. Analyze the given text excerpt and classify it as relevant (1) or irrelevant (0) to the specified SDG.
2. Utilize the provided SDG description, along with its targets and indicators, to enhance classification accuracy.
3. Respond only with the integer classification result.

### 4.2.2 Few-Shot Prompts

Few-Shot classification Prompts were also structured parametrically to include the respective number of data examples for 1-shot, 2-shot, and 3-shot classification scenarios. The selection of data examples aimed to mitigate randomness. Therefore, when building the Prompts with builder scripts, examples with the highest values of "labels_positive" and "agreement" were chosen under the assumption that these text excerpts were more clearly associated with the given SDG, as they were predominantly labeled as relevant by volunteers with a consensus. The data examples were selected from the training datasets.

**Few-Shot Prompt Template**

Task: Perform binary text classification to determine the relevance of a text excerpt to a specified United Nations' Sustainable Development Goal (SDG).

Information:
- SDG Number: “‘{sdg_number}“‘
- SDG Title: “‘{sdg_title}“‘
- Text Excerpt: “‘{text_excerpt}“‘
- Training Data: {training_data}

Instructions:
1. Analyze the given text excerpt and classify it as relevant (1) or irrelevant (0) to the specified SDG.
2. Fine-tune the classification using text excerpt examples labeled as relevant to the specified SDG in the training data.
3. Respond only with the integer classification result.

## 4.3 Binary Text Classification of SDG Relevance Using ChatGPT and Gemini with Prompting

After crafting the Prompt templates for Zero-Shot and Few-Shot cases, data examples from test datasets are used to evalute the binary text classification performance of ChatGPT and Gemini models. To facilitate a clearer comparison between ChatGPT

and Gemini models, identical Prompts were used with both LLMs during classification.

During classification, **ChatGPT 3.5** and **Gemini 1.0 Pro** models were used, as they are widely available and represent state-of-the-art technology developed by industry leaders, OpenAI and Google, respectively. The interaction with the models were done via API requests. For ChatGPT, OpenAI API and for Gemini, Gemini API were used with their respective Python packages. A temperature parameter of 0 was set for both model inputs to ensure deterministic responses. To prevent unintended blocks by the Gemini model, the safety settings parameter was uniformly configured with a threshold of "BLOCK_NONE" across all categories.

Afterwards, the outputs from both models were stored in DataFrames[2]. Responses that did not conform to the expected output format ("1" or "0") for each binary classification scenario were identified and excluded from the evaluation for all SDGs, separately addressing Zero-Shot and Few-Shot results.

### 4.4 Fine-Tuning DistilBERT and BERT for Binary Text Classification of SDG Relevance

For Fine-Tuning, DistilBERT [38] and BERT Base [14] models were employed. BERT model was employed as it laid the foundation for adopting LLMs for Fine-Tuning. DistilBERT model was used to compare model sizes and capabilities relative to the dataset, demonstrating a comparison of performance under limited computational resources.

Hyperparameter optimization with Grid Search approach was conducted during Fine-Tuning using the search space recommended by Devlin et al. [14]:

- **Epochs:** 2, 3, 4
- **Batch size:** 16, 32
- **Learning rate:** 5e-5, 3e-5, 2e-5

[2] Two-dimensional, size-mutable, potentially heterogeneous tabular data. Part of pandas Python package: https://pandas.pydata.org.

The model downloads and Fine-Tuning were conducted using the HuggingFace [47] Python package. Both models utilized the default model configurations, including a dropout probability of 0.1, weight decay value of 0.01 and AdamW optimizer from PyTorch [48], which is the implementation of fixing Adam algorithm with weight decay [49]. During hyperparameter optimization, validation loss was used as the metric for identifying best model. Binary Cross-Entropy Loss was employed as the loss function. To ensure reproducible results and models, a random seed was employed with the HuggingFace library.

Since the task was binary classification, separate DistilBERT and BERT Base models were Fine-Tuned for each SDG, including hyperparameter optimization. Finally, the best models were evaluated using the test datasets of each SDG.

# CHAPTER 5

# RESULTS

## 5.1 Evaluation of Prompting for Binary Text Classification of SDGs

### 5.1.1 Zero-Shot Prompting Results

Table 5.1: Evaluation of Zero-Shot Prompting Combined Across All SDGs and Context Levels

| Model | Precision | Recall | Accuracy | F1 |
|---|---|---|---|---|
| ChatGPT | 0.73 | 0.68 | 0.68 | 0.66 |
| Gemini | 0.78 | 0.78 | 0.78 | 0.78 |

The results presented in Table 5.1 highlight the performance of ChatGPT and Gemini in the Zero-Shot Prompting scenario across all SDGs and context levels.

Gemini consistently outperformed ChatGPT in all evaluation metrics. These findings demonstrate that Gemini is more effective and reliable for Zero-Shot binary text classification withing SDG relevancy domain, maintaining higher performance across all evaluated metrics.

The results in Table 5.2 reveal the impact of context level on the classification performance of both models in the Zero-Shot scenario.

At context level 1, Gemini significantly outperformed ChatGPT, demonstrating better handling of the least context-rich scenarios. As the context level increased to level 2, both models showed improved performance, but the improvement was more no-

Table 5.2: Evaluation of Zero-Shot Prompting Across Context Levels

| Context Level | Model | Precision | Recall | Accuracy | F1 |
|---|---|---|---|---|---|
| 1 | ChatGPT | 0.68 | 0.56 | 0.56 | 0.46 |
| | Gemini | 0.75 | 0.74 | 0.74 | 0.74 |
| 2 | ChatGPT | 0.75 | 0.66 | 0.66 | 0.63 |
| | Gemini | 0.79 | 0.79 | 0.79 | 0.79 |
| 3 | ChatGPT | 0.82 | 0.82 | 0.82 | 0.82 |
| | Gemini | 0.80 | 0.80 | 0.80 | 0.80 |

ticeable for ChatGPT. By context level 3, ChatGPT achieved its highest performance, surpassing Gemini.

The context level had a more substantial effect on ChatGPT's performance, with clear improvements across all metrics as context increased. This indicates that ChatGPT heavily relies on and effectively leverages contextual information for more accurate classification. In contrast, Gemini's performance remained consistently high across lower context levels, showing less variation and a more robust ability to handle varying amounts of context. Nevertheless, Gemini also showed improved performance as context level increased.

Table 5.3 provides insights into the Zero-Shot binary classification performance of ChatGPT and Gemini across various SDGs, combining their results for all context levels.

Gemini consistently demonstrated better performance compared to ChatGPT across all SDGs. Notably, Gemini achieved its highest performance in SDGs 4, 7, and 14. These results underscore Gemini's robust performance in accurately classifying content related to sustainable infrastructure (SDG 9), affordable and clean energy (SDG 7), and life below water (SDG 14). Gemini's lowest performance was observed in SDGs 8, 9, and 10. Despite these challenges, Gemini's performance remains noteworthy and significantly outperforms ChatGPT's lowest performance in the same SDGs. This indicates that both models encountered difficulties in accurately classifying content related to goals such as decent work and economic growth, industry innovation

Table 5.3: Evaluation of Zero-Shot Prompting Across SDGs

| SDG | Model | Precision | Recall | Accuracy | F1 |
|---|---|---|---|---|---|
| 1 | ChatGPT | 0.71 | 0.64 | 0.63 | 0.60 |
| | Gemini | 0.76 | 0.72 | 0.73 | 0.71 |
| 2 | ChatGPT | 0.70 | 0.69 | 0.69 | 0.69 |
| | Gemini | 0.77 | 0.75 | 0.75 | 0.74 |
| 3 | ChatGPT | 0.77 | 0.74 | 0.74 | 0.73 |
| | Gemini | 0.81 | 0.81 | 0.81 | 0.81 |
| 4 | ChatGPT | 0.74 | 0.70 | 0.70 | 0.69 |
| | Gemini | 0.85 | 0.84 | 0.84 | 0.84 |
| 5 | ChatGPT | 0.76 | 0.74 | 0.74 | 0.73 |
| | Gemini | 0.86 | 0.83 | 0.83 | 0.83 |
| 6 | ChatGPT | 0.75 | 0.73 | 0.74 | 0.73 |
| | Gemini | 0.83 | 0.81 | 0.81 | 0.81 |
| 7 | ChatGPT | 0.78 | 0.71 | 0.71 | 0.69 |
| | Gemini | 0.85 | 0.84 | 0.84 | 0.84 |
| 8 | ChatGPT | 0.76 | 0.59 | 0.59 | 0.51 |
| | Gemini | 0.76 | 0.72 | 0.72 | 0.70 |
| 9 | ChatGPT | 0.74 | 0.62 | 0.62 | 0.56 |
| | Gemini | 0.74 | 0.68 | 0.68 | 0.66 |
| 10 | ChatGPT | 0.70 | 0.57 | 0.57 | 0.48 |
| | Gemini | 0.67 | 0.67 | 0.67 | 0.67 |
| 11 | ChatGPT | 0.72 | 0.65 | 0.65 | 0.62 |
| | Gemini | 0.83 | 0.80 | 0.80 | 0.80 |
| 12 | ChatGPT | 0.76 | 0.63 | 0.63 | 0.58 |
| | Gemini | 0.78 | 0.73 | 0.74 | 0.72 |
| 13 | ChatGPT | 0.74 | 0.68 | 0.68 | 0.66 |
| | Gemini | 0.80 | 0.79 | 0.79 | 0.78 |
| 14 | ChatGPT | 0.80 | 0.79 | 0.79 | 0.79 |
| | Gemini | 0.85 | 0.84 | 0.84 | 0.84 |
| 15 | ChatGPT | 0.79 | 0.74 | 0.74 | 0.72 |
| | Gemini | 0.83 | 0.82 | 0.82 | 0.82 |
| 16 | ChatGPT | 0.74 | 0.64 | 0.64 | 0.60 |
| | Gemini | 0.78 | 0.78 | 0.78 | 0.78 |

and infrastructure, and reduced inequalities. However, Gemini still maintained relatively strong performance compared to ChatGPT in these areas.

On the other hand, ChatGPT exhibited weaker overall performance across all SDGs. It showed competitive performance in specific SDGs such as 3, 5, and 14, where ChatGPT achieved its highest scores, still being outperformed by Gemini.

### 5.1.2 Few-Shot Prompting Results

Table 5.4: Evaluation of Few-Shot Prompting Combined Across All SDGs and N-shots

| Model | Precision | Recall | Accuracy | F1 |
|---|---|---|---|---|
| ChatGPT | 0.74 | 0.51 | 0.51 | 0.35 |
| Gemini | 0.77 | 0.75 | 0.76 | 0.75 |

The results summarized in Table 5.4 provide the evaluation of ChatGPT and Gemini in the Few-Shot Prompting scenario across all SDGs and N-shots.

Notably, ChatGPT exhibited a significant performance fall in the Few-Shot setting, making it obsolete for the given scenario. In contrast, Gemini demonstrated consistently good performance across all metrics. By achieving balanced metrics across the board, Gemini shows a more thorough and accurate approach in identifying both positive and negative instances, making it better suited for Few-Shot binary text classification of SDG relevance compared to ChatGPT.

The effects of N-shot on model performance, are presented in Table 5.5 for both ChatGPT and Gemini models. The overall performance metrics of both models remained consistent across different N-shot settings, indicating that varying the number of examples (N-shot) did not have any noticeable impact on the models' overall performance.

While both models exhibit worse performance in the Few-Shot setting compared to Zero-Shot, the extent of performance downgrade varies significantly between the two models. Gemini demonstrates relatively minor changes in performance. In contrast,

Table 5.5: Evaluation of Few-Shot Prompting Across N-shot

| N-shot | Model | Precision | Recall | Accuracy | F1 |
|---|---|---|---|---|---|
| 1 | ChatGPT | 0.75 | 0.50 | 0.51 | 0.34 |
| | Gemini | 0.77 | 0.75 | 0.75 | 0.74 |
| 2 | ChatGPT | 0.75 | 0.50 | 0.51 | 0.34 |
| | Gemini | 0.77 | 0.75 | 0.75 | 0.74 |
| 3 | ChatGPT | 0.75 | 0.50 | 0.51 | 0.34 |
| | Gemini | 0.77 | 0.75 | 0.75 | 0.74 |

ChatGPT shows significant performance degradation in the Few-Shot setting, making it an unsuitable choice for binary text classification of SDGs.

Moreover, it is evident that Zero-Shot Prompting with high-context settings yields the best performance for both models, outperforming their performance in the Few-Shot setting. This indicates that the introduction of additional training examples in the Few-Shot setting does not consistently enhance model performance, contrary to expectations.

These observations can be contextualized through Prompt Engineering. In the Zero-Shot setting, models only rely on their pre-existing knowledge and Prompt understanding to make predictions, which may align better with the intended task requirements without the introduction of potentially conflicting or insufficient training examples. On the other hand, the Few-Shot setting introduces a limited number of task-specific examples, which may not effectively guide the models towards optimal performance due to constraints in data diversity, model adaptation, or Prompt formulation.

Table 5.3 presents the Few-Shot performance across Sustainable Development Goals (SDGs) for ChatGPT and Gemini models. Observing the results reveals notable variations in performance metrics across different SDGs and between the two models.

Similar to the previous evaluations, comparison between Zero-Shot and Few-Shot settings highlights that both models generally perform better across all SDGs in the

Table 5.6: Evaluation of Few-Shot Prompting Across SDGs

| SDG | Model | Precision | Recall | Accuracy | F1 |
|---|---|---|---|---|---|
| 1 | ChatGPT | 0.25 | 0.50 | 0.80 | 0.33 |
| | Gemini | 0.71 | 0.69 | 0.69 | 0.68 |
| 2 | ChatGPT | 0.76 | 0.52 | 0.52 | 0.38 |
| | Gemini | 0.76 | 0.75 | 0.75 | 0.75 |
| 3 | ChatGPT | 0.75 | 0.50 | 0.50 | 0.34 |
| | Gemini | 0.74 | 0.73 | 0.73 | 0.72 |
| 4 | ChatGPT | 0.75 | 0.50 | 0.51 | 0.34 |
| | Gemini | 0.84 | 0.81 | 0.81 | 0.81 |
| 5 | ChatGPT | 0.25 | 0.50 | 0.50 | 0.33 |
| | Gemini | 0.88 | 0.88 | 0.88 | 0.88 |
| 6 | ChatGPT | 0.75 | 0.51 | 0.52 | 0.36 |
| | Gemini | 0.83 | 0.82 | 0.82 | 0.81 |
| 7 | ChatGPT | 0.59 | 0.50 | 0.51 | 0.34 |
| | Gemini | 0.81 | 0.81 | 0.81 | 0.81 |
| 8 | ChatGPT | 0.25 | 0.50 | 0.50 | 0.33 |
| | Gemini | 0.77 | 0.73 | 0.73 | 0.72 |
| 9 | ChatGPT | 0.25 | 0.50 | 0.50 | 0.33 |
| | Gemini | 0.75 | 0.66 | 0.66 | 0.63 |
| 10 | ChatGPT | 0.25 | 0.50 | 0.50 | 0.33 |
| | Gemini | 0.64 | 0.63 | 0.63 | 0.62 |
| 11 | ChatGPT | 0.75 | 0.51 | 0.52 | 0.36 |
| | Gemini | 0.83 | 0.81 | 0.81 | 0.80 |
| 12 | ChatGPT | 0.25 | 0.50 | 0.50 | 0.33 |
| | Gemini | 0.77 | 0.70 | 0.70 | 0.68 |
| 13 | ChatGPT | 0.75 | 0.51 | 0.51 | 0.35 |
| | Gemini | 0.81 | 0.78 | 0.78 | 0.77 |
| 14 | ChatGPT | 0.77 | 0.56 | 0.56 | 0.45 |
| | Gemini | 0.80 | 0.80 | 0.80 | 0.80 |
| 15 | ChatGPT | 0.25 | 0.50 | 0.50 | 0.33 |
| | Gemini | 0.76 | 0.72 | 0.72 | 0.71 |
| 16 | ChatGPT | 0.75 | 0.51 | 0.50 | 0.35 |
| | Gemini | 0.77 | 0.77 | 0.77 | 0.77 |

Zero-Shot setting, indicating that the introduction of a small number of task-specific examples does not consistently improve performance. This highlights the importance of Prompting in achieving optimal results. A well-designed Zero-Shot Prompt can significantly outperform a Prompt based on groundtruth data in the binary text classification of SDGs.

Additionally, these results underscore Gemini's reliability and adaptability in both Zero-Shot and Few-Shot Prompting scenarios across diverse SDGs, while highlighting areas for improvement in ChatGPT's Few-Shot Prompting performance.

## 5.2 Evaluation of Fine-Tuning for Binary Text Classification of SDGs

Table 5.7: Optimal Hyperparameters of DistilBERT Models

| SDG | Epoch | Batch Size | Learning Rate | Validation Loss | F1 |
|---|---|---|---|---|---|
| 1 | 2 | 16 | 5e-05 | 0.223 | 0.93 |
| 2 | 3 | 16 | 5e-05 | 0.182 | 0.97 |
| 3 | 2 | 16 | 5e-05 | 0.184 | 0.97 |
| 4 | 2 | 16 | 5e-05 | 0.223 | 0.93 |
| 5 | 2 | 16 | 5e-05 | 0.409 | 0.89 |
| 6 | 2 | 16 | 5e-05 | 0.329 | 0.90 |
| 7 | 2 | 16 | 5e-05 | 0.279 | 0.91 |
| 8 | 2 | 16 | 5e-05 | 0.348 | 0.85 |
| 9 | 2 | 16 | 5e-05 | 0.491 | 0.88 |
| 10 | 2 | 16 | 5e-05 | 1.018 | 0.81 |
| 11 | 2 | 16 | 5e-05 | 0.811 | 0.84 |
| 12 | 2 | 16 | 5e-05 | 0.205 | 0.95 |
| 13 | 3 | 32 | 5e-05 | 1.168 | 0.92 |
| 14 | 2 | 16 | 5e-05 | 0.328 | 0.93 |
| 15 | 2 | 16 | 5e-05 | 0.945 | 0.85 |
| 16 | 4 | 16 | 5e-05 | 0.596 | 0.94 |

Table 5.8: Optimal Hyperparameters of BERT Base Models

| SDG | Epoch | Batch Size | Learning Rate | Validation Loss | F1 |
|---|---|---|---|---|---|
| 1 | 2 | 16 | 5e-05 | 0.217 | 0.94 |
| 2 | 3 | 16 | 5e-05 | 0.237 | 0.95 |
| 3 | 2 | 16 | 5e-05 | 0.188 | 0.97 |
| 4 | 2 | 16 | 5e-05 | 0.288 | 0.89 |
| 5 | 2 | 16 | 5e-05 | 0.431 | 0.91 |
| 6 | 2 | 16 | 5e-05 | 0.525 | 0.88 |
| 7 | 2 | 16 | 5e-05 | 0.792 | 0.88 |
| 8 | 2 | 16 | 5e-05 | 0.788 | 0.86 |
| 9 | 2 | 16 | 5e-05 | 0.231 | 0.91 |
| 10 | 2 | 16 | 5e-05 | 0.357 | 0.87 |
| 11 | 2 | 16 | 5e-05 | 0.325 | 0.92 |
| 12 | 2 | 16 | 5e-05 | 0.376 | 0.89 |
| 13 | 3 | 32 | 5e-05 | 0.860 | 0.85 |
| 14 | 2 | 16 | 5e-05 | 0.459 | 0.87 |
| 15 | 4 | 16 | 5e-05 | 1.148 | 0.87 |
| 16 | 3 | 16 | 5e-05 | 0.789 | 0.92 |

Tables 5.7 and 5.8 summarize the hyperparameter configurations and corresponding to best performance (lowest validation loss) of DistilBERT and BERT models respectively.

Upon reviewing the tables, it is clear that the majority of SDG models consistently achieved optimal validation results using the same hyperparameters. Specifically, configurations featuring 2 epochs, a batch size of 16, and a learning rate of 5e-05 were effective across most SDGs, with the exceptions of SDGs 2, 13, and 16 for DistilBERT models, and additionally SDG 15 for BERT models.

The consistent optimal performance of configurations using 2 epochs, a batch size of 16, and a learning rate of 5e-05 across the majority of SDG models can be attributed to several key factors. Firstly, employing 2 epochs strikes a balance between model

convergence and overfitting, allowing the model to sufficiently learn from the data without excessively memorizing noise or outliers. This duration appears to be adequate for the DistilBERT and BERT models to converge to a stable state of learning, optimizing their performance on the validation set.

Secondly, the batch size of 16 facilitates efficient computation during training, ensuring that the model receives diverse and informative data samples while maintaining computational efficiency. This batch size likely contributes to smoother convergence and better generalization capabilities.

Thirdly, the learning rate of 5e-05 proves effective in controlling the step size of gradient descent updates, enabling the models to make steady progress toward minimizing the loss function without encountering issues such as divergent behavior or slow convergence.

These hyperparameters collectively support robust training dynamics that enhance the models' ability to generalize well to unseen data, thereby consistently achieving better validation results across a range of SDG configurations.

Table 5.9: Average Performance of DistilBERT and BERT Models on Binary Text Classification of SDGs

| Model | Precision | Recall | Accuracy | F1 |
|---|---|---|---|---|
| DistilBERT | 0.93 | 0.93 | 0.93 | 0.93 |
| BERT | 0.93 | 0.93 | 0.93 | 0.93 |

Tables 5.10 and 5.11 present an evaluation of DistilBERT and BERT models respectively providing metrics such as precision, recall, accuracy, and F1 score across different SDG models. Table 5.9 presents average performances of the models. Both models demonstrate strong performance across various configurations.

Since both models exhibit similar average performances, DistilBERT emerges as more advantageous due to its computational efficiency stemming from its smaller model size and simplified architecture.

Upon reviewing the overall results (see Table 5.12), it is evident that Fine-Tuned

Table 5.10: Evaluation of DistilBERT Models

| SDG | Precision | Recall | Accuracy | F1 |
|---|---|---|---|---|
| 1 | 0.95 | 0.95 | 0.95 | 0.95 |
| 2 | 0.94 | 0.94 | 0.94 | 0.94 |
| 3 | 0.99 | 0.99 | 0.99 | 0.99 |
| 4 | 0.95 | 0.95 | 0.95 | 0.95 |
| 5 | 0.94 | 0.94 | 0.94 | 0.94 |
| 6 | 0.94 | 0.94 | 0.94 | 0.94 |
| 7 | 0.95 | 0.95 | 0.95 | 0.95 |
| 8 | 0.90 | 0.89 | 0.89 | 0.89 |
| 9 | 0.88 | 0.87 | 0.87 | 0.87 |
| 10 | 0.90 | 0.90 | 0.90 | 0.90 |
| 11 | 0.94 | 0.94 | 0.94 | 0.94 |
| 12 | 0.92 | 0.91 | 0.91 | 0.91 |
| 13 | 0.91 | 0.90 | 0.90 | 0.90 |
| 14 | 0.94 | 0.94 | 0.94 | 0.94 |
| 15 | 0.94 | 0.94 | 0.94 | 0.94 |
| 16 | 0.96 | 0.96 | 0.96 | 0.96 |

DistilBERT and BERT models outperform Prompt-based binary text classification of SDGs using Gemini and ChatGPT. This can be attributed to the following reasons: Firstly, Fine-Tuning enables models to adjust their parameters to the specific nuances and complexities of the dataset, thereby optimizing their performance for the particular classification task. This process involves refining pre-trained weights through additional training on task-specific data, which enhances the model's ability to extract relevant features and make accurate predictions. On the other hand, Prompting approaches depend on manually crafted Prompts to guide the model's decision-making process. These prompts may struggle to capture subtle patterns or domain-specific nuances present in the dataset. This limitation is particularly evident in Zero-Shot or Few-Shot Prompting scenarios, where little to no training data is available for adapting the Prompts. Moreover, Fine-Tuning allows for iterative refinement based

Table 5.11: Evaluation of BERT Base Models

| SDG | Precision | Recall | Accuracy | F1 |
|---|---|---|---|---|
| 1 | 0.93 | 0.93 | 0.93 | 0.93 |
| 2 | 0.96 | 0.96 | 0.96 | 0.96 |
| 3 | 0.99 | 0.99 | 0.99 | 0.99 |
| 4 | 0.95 | 0.95 | 0.95 | 0.95 |
| 5 | 0.93 | 0.93 | 0.93 | 0.93 |
| 6 | 0.94 | 0.94 | 0.94 | 0.94 |
| 7 | 0.93 | 0.93 | 0.93 | 0.93 |
| 8 | 0.91 | 0.91 | 0.91 | 0.91 |
| 9 | 0.88 | 0.88 | 0.88 | 0.88 |
| 10 | 0.92 | 0.91 | 0.91 | 0.91 |
| 11 | 0.94 | 0.94 | 0.94 | 0.94 |
| 12 | 0.89 | 0.88 | 0.88 | 0.88 |
| 13 | 0.88 | 0.88 | 0.88 | 0.88 |
| 14 | 0.91 | 0.91 | 0.91 | 0.91 |
| 15 | 0.95 | 0.95 | 0.95 | 0.95 |
| 16 | 0.97 | 0.97 | 0.97 | 0.97 |

Table 5.12: Overall Performance Comparison of Prompting and Fine-Tuning Approaches

| Model | Methodology | Precision | Recall | Accuracy | F1 |
|---|---|---|---|---|---|
| ChatGPT | Zero-Shot Prompting | 0.73 | 0.68 | 0.68 | 0.66 |
| ChatGPT | Few-Shot Prompting | 0.74 | 0.51 | 0.51 | 0.35 |
| Gemini | Zero-Shot Prompting | 0.78 | 0.78 | 0.78 | 0.78 |
| Gemini | Few-Shot Prompting | 0.77 | 0.75 | 0.76 | 0.75 |
| DistilBERT | Fine-Tuning | 0.93 | 0.93 | 0.93 | 0.93 |
| BERT | Fine-Tuning | 0.93 | 0.93 | 0.93 | 0.93 |

on validation performance, enabling the model to continuously improve its performance.

# CHAPTER 6

# DISCUSSION

In the Results section, it is noted that Fine-Tuning with DistilBERT and BERT models outperform Prompting with ChatGPT and Gemini on binary text classification. This advantage can be attributed to Fine-Tuning, which allows models to adjust their parameters to the dataset's specific nuances, optimizing performance for the classification task. In contrast, Prompting relies on manually crafted prompts that may not fully capture subtle patterns or domain-specific nuances, especially in zero-shot or few-shot scenarios. Additionally, fine-tuning enables iterative refinement based on validation performance.

Another contributing factor to these results could be the architectural differences between the models used. BERT models employ an encoder-only architecture, while ChatGPT 3.5 and Gemini are based on a decoder-only architecture. As a result, encoder-only models tend to excel in sentence classification tasks, which aligns with findings in the literature (Benayas et al. 2024 [50]).

Finally, the findings of this research indicate that fine-tuning outperforms prompting in binary text classification for SDG relevance. However, prompting-based approaches have the potential for improvement with the development of newer, more capable and more optimized models. With advanced models and more effective prompting techniques in the future, it is reasonable to expect an increase in the performance of prompting-based models for sentence classification, making them a valid option that requires no training costs.

## 6.1 Future Work

Building on the presented findings, several opportunities for future research can be identified. To expand the coverage of Prompting comparison, the performance of few-shot prompts that provide only examples and labels, without any prompt text, could be evaluated. This approach would reveal how prompting-based models perform in a few-shot scenario without explicit instructions.

Additionally, the overall comparison of models in the binary text classification of SDG relevance could be broadened by exploring traditional machine learning models (e.g., SVMs, logistic regression, random forests, etc.).

Finally, the multilabel classification of SDG relevance could be examined for both Fine-Tuning and Prompting approaches, allowing for an enhanced comparison between these two methods.

# CHAPTER 7

# CONCLUSIONS

The traditional methodology of Fine-Tuning Transformer-based models significantly outperforms Prompting with LLMs in the task of binary text classification for SDG relevancy, consistent with findings from comparative research in the literature on different domains. However, Prompting via LLMs has the potential to demonstrate improved text classification performance with the next iteration of more capable models and introduction of effective Prompting techniques.

For both ChatGPT and Gemini, Zero-Shot Prompting outperforms Few-Shot Prompting in the binary text classification of SDGs. Furthermore, Gemini consistently outperforms ChatGPT in SDG-related binary text classification across both Prompting scenarios, showing its reliability.

In the Zero-Shot Prompting scenario, both models demonstrate improved binary text classification performance as the context level of the Prompt increases. The context level notably influenced ChatGPT's performance, enabling it to outperform Gemini at the highest context level. This underscores ChatGPT's sensitivity to and reliance on contextual information specific to the Prompt's task description for achieving optimal results. In contrast, Gemini exhibited less variation but consistently high and robust performance.

In the Few-Shot Prompting scenario, both models exhibited poorer performance compared to the Zero-Shot Prompting setting. ChatGPT experienced a significant decline in its performance, rendering it ineffective as a text classifier in the Few-Shot prompting setting. Increasing the number of examples in the Few-Shot prompts had no effect on performance for either model.

For Fine-Tuned models, DistilBERT and BERT show similar average performance in the binary text classification of SDGs. In this scenario, DistilBERT proves more advantageous as it achieves comparable performance to the BERT model despite its smaller size and simpler architecture, thereby offering improved computational efficiency.

# REFERENCES


[1] United Nations, "Sustainable development goals," n.d. Accessed: 2024-08-11.

[2] OpenAI, "Introducing ChatGPT." https://openai.com/index/chatgpt/, 2024. Accessed: 2024-06-30.

[3] G. Team, R. Anil, S. Borgeaud, Y. Wu, J.-B. Alayrac, J. Yu, R. Soricut, J. Schalkwyk, A. M. Dai, A. Hauth, *et al.*, "Gemini: a family of highly capable multimodal models," *arXiv preprint arXiv:2312.11805*, 2023.

[4] OpenAI, "Chatgpt." https://www.openai.com/chatgpt, 2024. Accessed: 2024-06-30.

[5] Google, "Google gemini." https://gemini.google.com/?hl=en, 2024. Accessed: 2024-07-12.

[6] N. Bautista-Puig, G. Statulevičiūtė, V. Stančiauskas, G. Dikmener, and D. Akylbekova, "Osdg community dataset: Unraveling the societal perceptions around the sustainable development goals."

[7] OSDG and others, "OSDG Community Dataset (OSDG-CD)," January 2024. Accessed: 2024-06-17.

[8] Databricks, "Generative ai - databricks documentation." https://docs.databricks.com/en/generative-ai/generative-ai.html, Accessed: 2024-07-06.

[9] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, *et al.*, "Language models are unsupervised multitask learners," *OpenAI blog*, vol. 1, no. 8, p. 9, 2019.

[10] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, *et al.*, "Language models are few-shot learners," *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020.

[11] D. Dai, Y. Sun, L. Dong, Y. Hao, S. Ma, Z. Sui, and F. Wei, "Why can gpt learn in-context? language models implicitly perform gradient descent as meta-optimizers," *arXiv preprint arXiv:2212.10559*, 2022.

[12] J. Howard and S. Ruder, "Universal language model fine-tuning for text classification," *arXiv preprint arXiv:1801.06146*, 2018.

[13] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," *CoRR*, vol. abs/1706.03762, 2017.

[14] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," *CoRR*, vol. abs/1810.04805, 2018.

[15] A. Salinas and F. Morstatter, "The butterfly effect of altering prompts: How small changes and jailbreaks affect large language model performance," *arXiv preprint arXiv:2401.03729*, 2024.

[16] A. H. Parizi, Y. Liu, P. Nokku, S. Gholamian, and D. Emerson, "A comparative study of prompting strategies for legal text classification," in *Proceedings of the Natural Legal Language Processing Workshop 2023*, pp. 258–265, 2023.

[17] B. Fatemi, F. Rabbi, and A. L. Opdahl, "Evaluating the effectiveness of gpt large language model for news classification in the iptc news ontology," *IEEE Access*, 2023.

[18] L. Loukas, I. Stogiannidis, O. Diamantopoulos, P. Malakasiotis, and S. Vassos, "Making llms worth every penny: Resource-limited text classification in banking," in *Proceedings of the Fourth ACM International Conference on AI in Finance*, pp. 392–400, 2023.

[19] E. Gunes and C. K. Florczak, "Multiclass classification of policy documents with large language models," *arXiv preprint arXiv:2310.08167*, 2023.

[20] X.-Q. Dao, "Performance comparison of large language models on vnhsge english dataset: Openai chatgpt, microsoft bing chat, and google bard," *arXiv preprint arXiv:2307.02288*, 2023.

[21] Y. Chae and T. Davidson, "Large language models for text classification: From zero-shot learning to fine-tuning," *Open Science Foundation*, 2023.

[22] D. Trajanov, G. Lazarev, L. Chitkushev, and I. Vodenska, "Comparing the performance of chatgpt and state-of-the-art climate nlp models on climate-related text classification tasks," in *Proceedings of the 4th International Conference on Environmental Design (ICED2023)*, 2023.

[23] B. Clavié, A. Ciceu, F. Naylor, G. Soulié, and T. Brightwell, "Large language models in the workplace: A case study on prompt engineering for job type classification," in *International Conference on Applications of Natural Language to Information Systems*, pp. 3–17, Springer, 2023.

[24] M. Bosley, M. Jacobs-Harukawa, H. Licht, and A. Hoyle, "Do we still need bert in the age of gpt? comparing the benefits of domain-adaptation and in-context-learning approaches to using llms for political science research," 2023.

[25] H. Yu, Z. Yang, K. Pelrine, J. F. Godbout, and R. Rabbany, "Open, closed, or small language models for text classification?," *arXiv preprint arXiv:2308.10092*, 2023.

[26] X. Wang, Y. Wang, C. Xu, X. Geng, B. Zhang, C. Tao, F. Rudzicz, R. E. Mercer, and D. Jiang, "Investigating the learning behaviour of in-context learning: a comparison with supervised learning," in *ECAI 2023*, pp. 2543–2551, IOS Press, 2023.

[27] R. Eisenstadt, A. Asi, and R. Ronen, "Comparing fine-tuned transformers and large language models for sales call classification: A case study," in *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pp. 5240–5241, 2023.

[28] Y. Qiu and Y. Jin, "Chatgpt and finetuned bert: A comparative study for developing intelligent design support systems," *Intelligent Systems with Applications*, vol. 21, p. 200308, 2024.

[29] M. A. K. Raiaan, M. S. H. Mukta, K. Fatema, N. M. Fahad, S. Sakib, M. M. J. Mim, J. Ahmad, M. E. Ali, and S. Azam, "A review on large language mod-

els: Architectures, applications, taxonomies, open issues and challenges," *IEEE Access*, 2024.

[30] M. U. Hadi, R. Qureshi, A. Shah, M. Irfan, A. Zafar, M. B. Shaikh, N. Akhtar, J. Wu, S. Mirjalili, *et al.*, "A survey on large language models: Applications, challenges, limitations, and practical usage," *Authorea Preprints*, 2023.

[31] K. S. Kalyan, "A survey of gpt-3 family large language models including chatgpt and gpt-4," *Natural Language Processing Journal*, p. 100048, 2023.

[32] S. Pahune and M. Chandrasekharan, "Several categories of large language models (llms): A short survey," *arXiv preprint arXiv:2307.10188*, 2023.

[33] L. Pukelis, N. Bautista-Puig, G. Statulevičiūtė, V. Stančiauskas, G. Dikmener, and D. Akylbekova, "Osdg 2.0: a multilingual tool for classifying text data by un sustainable development goals (sdgs)," *arXiv preprint arXiv:2211.11252*, 2022.

[34] M. Angin, B. Taşdemir, C. A. Yılmaz, G. Demiralp, M. Atay, P. Angin, and G. Dikmener, "A roberta approach for automated processing of sustainability reports," *Sustainability*, vol. 14, no. 23, p. 16139, 2022.

[35] S. Rodríguez Medina, "Multi-label text classification with transfer learning for policy documents: the case of the sustainable development goals," 2019.

[36] A. Hajikhani and A. Suominen, "The interrelation of sustainable development goals in publications and patents: A machine learning approach," *AI+ Informetrics*, pp. 183–193, 2021.

[37] J. E. Guisiano, R. Chiky, and J. De Mello, "Sdg-meter: A deep learning based tool for automatic text classification of the sustainable development goals," in *Asian conference on intelligent information and database systems*, pp. 259–271, Springer, 2022.

[38] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "Distilbert, a distilled version of BERT: smaller, faster, cheaper and lighter," *CoRR*, vol. abs/1910.01108, 2019.

[39] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized BERT pretraining approach," *CoRR*, vol. abs/1907.11692, 2019.

[40] P. P. Ray, "Chatgpt: A comprehensive review on background, applications, key challenges, bias, ethics, limitations and future scope," *Internet of Things and Cyber-Physical Systems*, vol. 3, pp. 121–154, 2023.

[41] T. Kudo and J. Richardson, "Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing," *CoRR*, vol. abs/1808.06226, 2018.

[42] P. Sahoo, A. K. Singh, S. Saha, V. Jain, S. Mondal, and A. Chadha, "A systematic survey of prompt engineering in large language models: Techniques and applications," *arXiv preprint arXiv:2402.07927*, 2024.

[43] OpenAI, "Prompt Engineering." https://platform.openai.com/docs/guides/prompt-engineering. Accessed: 2024-06-17.

[44] Google Cloud, "Design Text Prompts." https://cloud.google.com/vertex-ai/generative-ai/docs/text/text-prompts. Accessed: 2024-06-17.

[45] I. Kondrashchenko and O. Kostromin, "Scikit-llm: Scikit-learn meets large language models," 2023.

[46] A. Pal, "Promptify: Structured output from llms." https://github.com/promptslab/Promptify, 2022. Prompt-Engineering components for NLP tasks in Python.

[47] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, and J. Brew, "Huggingface's transformers: State-of-the-art natural language processing," *CoRR*, vol. abs/1910.03771, 2019.

[48] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Z. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," *CoRR*, vol. abs/1912.01703, 2019.

[49] I. Loshchilov and F. Hutter, "Fixing weight decay regularization in adam," *CoRR*, vol. abs/1711.05101, 2017.

[50] A. Benayas, M. Sicilia, and M. Mora-Cantallops, “A comparative analysis of encoder only and decoder only models in intent classification and sentiment analysis: Navigating the trade-offs in model size and performance,” 01 2024.

Table A1: ChatGPT Zero-shot Results Across SDGs and Context Levels

| SDG | Context Level | Precision | Recall | Accuracy | F1 |
|---|---|---|---|---|---|
| 1 | 1 | 0.56 | 0.51 | 0.51 | 0.39 |
| | 2 | 0.71 | 0.61 | 0.60 | 0.55 |
| | 3 | 0.80 | 0.79 | 0.79 | 0.79 |
| 2 | 1 | 0.65 | 0.59 | 0.59 | 0.54 |
| | 2 | 0.73 | 0.72 | 0.72 | 0.71 |
| | 3 | 0.78 | 0.77 | 0.77 | 0.76 |
| 3 | 1 | 0.77 | 0.58 | 0.57 | 0.48 |
| | 2 | 0.83 | 0.78 | 0.78 | 0.77 |
| | 3 | 0.88 | 0.86 | 0.86 | 0.86 |
| 4 | 1 | 0.66 | 0.54 | 0.55 | 0.44 |
| | 2 | 0.78 | 0.71 | 0.71 | 0.69 |
| | 3 | 0.87 | 0.86 | 0.86 | 0.86 |
| 5 | 1 | 0.67 | 0.56 | 0.56 | 0.48 |
| | 2 | 0.79 | 0.79 | 0.79 | 0.79 |
| | 3 | 0.87 | 0.86 | 0.87 | 0.86 |
| 6 | 1 | 0.68 | 0.59 | 0.60 | 0.54 |
| | 2 | 0.78 | 0.77 | 0.77 | 0.76 |
| | 3 | 0.87 | 0.84 | 0.84 | 0.84 |
| 7 | 1 | 0.70 | 0.57 | 0.57 | 0.49 |
| | 2 | 0.82 | 0.71 | 0.71 | 0.68 |
| | 3 | 0.84 | 0.84 | 0.84 | 0.84 |
| 8 | 1 | 0.75 | 0.51 | 0.50 | 0.35 |
| | 2 | 0.76 | 0.53 | 0.52 | 0.39 |
| | 3 | 0.81 | 0.74 | 0.74 | 0.72 |
| 9 | 1 | 0.75 | 0.51 | 0.51 | 0.35 |
| | 2 | 0.76 | 0.55 | 0.55 | 0.44 |
| | 3 | 0.82 | 0.79 | 0.79 | 0.79 |
| 10 | 1 | 0.50 | 0.50 | 0.50 | 0.35 |
| | 2 | 0.75 | 0.51 | 0.51 | 0.35 |
| | 3 | 0.76 | 0.69 | 0.70 | 0.67 |
| 11 | 1 | 0.76 | 0.54 | 0.55 | 0.42 |
| | 2 | 0.77 | 0.57 | 0.57 | 0.48 |
| | 3 | 0.84 | 0.84 | 0.84 | 0.84 |
| 12 | 1 | 0.76 | 0.51 | 0.52 | 0.37 |
| | 2 | 0.77 | 0.56 | 0.56 | 0.45 |
| | 3 | 0.84 | 0.81 | 0.82 | 0.81 |
| 13 | 1 | 0.72 | 0.56 | 0.57 | 0.47 |
| | 2 | 0.76 | 0.67 | 0.67 | 0.64 |
| | 3 | 0.80 | 0.80 | 0.80 | 0.80 |
| 14 | 1 | 0.72 | 0.68 | 0.68 | 0.66 |
| | 2 | 0.83 | 0.83 | 0.83 | 0.83 |
| | 3 | 0.89 | 0.87 | 0.87 | 0.87 |
| 15 | 1 | 0.74 | 0.60 | 0.60 | 0.53 |
| | 2 | 0.79 | 0.72 | 0.72 | 0.70 |
| | 3 | 0.89 | 0.89 | 0.89 | 0.89 |
| 16 | 1 | 0.75 | 0.52 | 0.52 | 0.38 |
| | 2 | 0.77 | 0.57 | 0.57 | 0.47 |
| | 3 | 0.84 | 0.84 | 0.84 | 0.84 |

Table A2: Gemini Zero-shot Results Across SDGs and Context Levels

| SDG | Context Level | Precision | Recall | Accuracy | F1 |
|---|---|---|---|---|---|
| 1 | 1 | 0.74 | 0.69 | 0.69 | 0.67 |
| | 2 | 0.78 | 0.73 | 0.73 | 0.72 |
| | 3 | 0.78 | 0.75 | 0.75 | 0.74 |
| 2 | 1 | 0.75 | 0.74 | 0.74 | 0.73 |
| | 2 | 0.78 | 0.72 | 0.72 | 0.71 |
| | 3 | 0.80 | 0.78 | 0.78 | 0.78 |
| 3 | 1 | 0.78 | 0.77 | 0.77 | 0.76 |
| | 2 | 0.82 | 0.82 | 0.82 | 0.82 |
| | 3 | 0.86 | 0.85 | 0.85 | 0.85 |
| 4 | 1 | 0.86 | 0.84 | 0.84 | 0.84 |
| | 2 | 0.87 | 0.87 | 0.87 | 0.87 |
| | 3 | 0.82 | 0.80 | 0.80 | 0.80 |
| 5 | 1 | 0.86 | 0.85 | 0.85 | 0.85 |
| | 2 | 0.87 | 0.83 | 0.84 | 0.83 |
| | 3 | 0.85 | 0.81 | 0.81 | 0.80 |
| 6 | 1 | 0.81 | 0.80 | 0.79 | 0.79 |
| | 2 | 0.84 | 0.80 | 0.79 | 0.79 |
| | 3 | 0.86 | 0.84 | 0.84 | 0.84 |
| 7 | 1 | 0.81 | 0.81 | 0.81 | 0.81 |
| | 2 | 0.86 | 0.85 | 0.85 | 0.85 |
| | 3 | 0.88 | 0.86 | 0.86 | 0.86 |
| 8 | 1 | 0.73 | 0.68 | 0.67 | 0.65 |
| | 2 | 0.78 | 0.74 | 0.74 | 0.73 |
| | 3 | 0.78 | 0.74 | 0.74 | 0.73 |
| 9 | 1 | 0.73 | 0.64 | 0.64 | 0.60 |
| | 2 | 0.75 | 0.71 | 0.71 | 0.70 |
| | 3 | 0.74 | 0.70 | 0.70 | 0.69 |
| 10 | 1 | 0.66 | 0.65 | 0.65 | 0.65 |
| | 2 | 0.69 | 0.69 | 0.69 | 0.69 |
| | 3 | 0.68 | 0.68 | 0.68 | 0.68 |
| 11 | 1 | 0.83 | 0.75 | 0.75 | 0.73 |
| | 2 | 0.83 | 0.81 | 0.82 | 0.81 |
| | 3 | 0.85 | 0.84 | 0.84 | 0.83 |
| 12 | 1 | 0.76 | 0.68 | 0.68 | 0.65 |
| | 2 | 0.79 | 0.76 | 0.76 | 0.75 |
| | 3 | 0.80 | 0.76 | 0.77 | 0.76 |
| 13 | 1 | 0.79 | 0.76 | 0.77 | 0.76 |
| | 2 | 0.83 | 0.82 | 0.82 | 0.82 |
| | 3 | 0.80 | 0.77 | 0.77 | 0.77 |
| 14 | 1 | 0.77 | 0.77 | 0.77 | 0.77 |
| | 2 | 0.87 | 0.87 | 0.87 | 0.86 |
| | 3 | 0.90 | 0.89 | 0.89 | 0.89 |
| 15 | 1 | 0.76 | 0.72 | 0.72 | 0.71 |
| | 2 | 0.84 | 0.84 | 0.84 | 0.84 |
| | 3 | 0.90 | 0.90 | 0.90 | 0.90 |
| 16 | 1 | 0.74 | 0.72 | 0.72 | 0.72 |
| | 2 | 0.84 | 0.84 | 0.84 | 0.84 |
| | 3 | 0.83 | 0.79 | 0.79 | 0.78 |

Table A3: ChatGPT Few-shot Results Across SDGs and Context Levels

| SDG | N-shot | Precision | Recall | Accuracy | F1 |
|---|---|---|---|---|---|
| 1 | 1 | 0.25 | 0.50 | 0.50 | 0.33 |
| | 2 | 0.25 | 0.50 | 0.50 | 0.33 |
| | 3 | 0.25 | 0.50 | 0.50 | 0.33 |
| 2 | 1 | 0.76 | 0.53 | 0.53 | 0.40 |
| | 2 | 0.75 | 0.51 | 0.51 | 0.35 |
| | 3 | 0.76 | 0.52 | 0.52 | 0.38 |
| 3 | 1 | 0.75 | 0.51 | 0.50 | 0.35 |
| | 2 | 0.25 | 0.50 | 0.50 | 0.33 |
| | 3 | 0.75 | 0.51 | 0.50 | 0.35 |
| 4 | 1 | 0.75 | 0.51 | 0.51 | 0.35 |
| | 2 | 0.25 | 0.50 | 0.50 | 0.33 |
| | 3 | 0.25 | 0.50 | 0.50 | 0.33 |
| 5 | 1 | 0.25 | 0.50 | 0.50 | 0.33 |
| | 2 | 0.25 | 0.50 | 0.50 | 0.33 |
| | 3 | 0.25 | 0.50 | 0.50 | 0.33 |
| 6 | 1 | 0.76 | 0.51 | 0.52 | 0.37 |
| | 2 | 0.75 | 0.51 | 0.51 | 0.35 |
| | 3 | 0.76 | 0.51 | 0.52 | 0.37 |
| 7 | 1 | 0.59 | 0.51 | 0.51 | 0.36 |
| | 2 | 0.25 | 0.50 | 0.50 | 0.33 |
| | 3 | 0.25 | 0.50 | 0.50 | 0.33 |
| 8 | 1 | 0.25 | 0.50 | 0.50 | 0.33 |
| | 2 | 0.25 | 0.50 | 0.50 | 0.33 |
| | 3 | 0.25 | 0.50 | 0.50 | 0.33 |
| 9 | 1 | 0.25 | 0.50 | 0.50 | 0.33 |
| | 2 | 0.25 | 0.50 | 0.50 | 0.33 |
| | 3 | 0.25 | 0.50 | 0.50 | 0.33 |
| 10 | 1 | 0.25 | 0.50 | 0.50 | 0.33 |
| | 2 | 0.25 | 0.50 | 0.50 | 0.33 |
| | 3 | 0.25 | 0.50 | 0.50 | 0.33 |
| 11 | 1 | 0.76 | 0.51 | 0.52 | 0.37 |
| | 2 | 0.76 | 0.52 | 0.52 | 0.38 |
| | 3 | 0.25 | 0.50 | 0.50 | 0.33 |
| 12 | 1 | 0.25 | 0.50 | 0.50 | 0.33 |
| | 2 | 0.25 | 0.50 | 0.50 | 0.33 |
| | 3 | 0.25 | 0.50 | 0.50 | 0.33 |
| 13 | 1 | 0.75 | 0.51 | 0.51 | 0.35 |
| | 2 | 0.75 | 0.51 | 0.51 | 0.35 |
| | 3 | 0.75 | 0.51 | 0.51 | 0.35 |
| 14 | 1 | 0.76 | 0.54 | 0.55 | 0.42 |
| | 2 | 0.78 | 0.61 | 0.62 | 0.55 |
| | 3 | 0.76 | 0.52 | 0.52 | 0.35 |
| 15 | 1 | 0.25 | 0.50 | 0.50 | 0.33 |
| | 2 | 0.25 | 0.50 | 0.50 | 0.33 |
| | 3 | 0.25 | 0.50 | 0.50 | 0.33 |
| 16 | 1 | 0.75 | 0.51 | 0.50 | 0.35 |
| | 2 | 0.75 | 0.51 | 0.50 | 0.35 |
| | 3 | 0.75 | 0.51 | 0.50 | 0.35 |

Table A4: Gemini Few-shot Results Across SDGs and Context Levels

| SDG | N-shot | Precision | Recall | Accuracy | F1 |
|---|---|---|---|---|---|
| 1 | 1 | 0.75 | 0.75 | 0.75 | 0.75 |
| | 2 | 0.71 | 0.68 | 0.67 | 0.66 |
| | 3 | 0.68 | 0.65 | 0.65 | 0.63 |
| 2 | 1 | 0.76 | 0.75 | 0.74 | 0.74 |
| | 2 | 0.76 | 0.76 | 0.76 | 0.76 |
| | 3 | 0.76 | 0.76 | 0.76 | 0.76 |
| 3 | 1 | 0.74 | 0.74 | 0.74 | 0.74 |
| | 2 | 0.73 | 0.72 | 0.72 | 0.71 |
| | 3 | 0.75 | 0.73 | 0.73 | 0.73 |
| 4 | 1 | 0.84 | 0.82 | 0.82 | 0.82 |
| | 2 | 0.85 | 0.83 | 0.83 | 0.83 |
| | 3 | 0.83 | 0.79 | 0.79 | 0.78 |
| 5 | 1 | 0.87 | 0.87 | 0.87 | 0.87 |
| | 2 | 0.87 | 0.87 | 0.87 | 0.87 |
| | 3 | 0.89 | 0.89 | 0.89 | 0.89 |
| 6 | 1 | 0.81 | 0.78 | 0.78 | 0.77 |
| | 2 | 0.85 | 0.84 | 0.84 | 0.84 |
| | 3 | 0.84 | 0.83 | 0.83 | 0.83 |
| 7 | 1 | 0.80 | 0.80 | 0.80 | 0.80 |
| | 2 | 0.83 | 0.83 | 0.83 | 0.83 |
| | 3 | 0.80 | 0.79 | 0.79 | 0.79 |
| 8 | 1 | 0.79 | 0.77 | 0.77 | 0.76 |
| | 2 | 0.77 | 0.72 | 0.72 | 0.70 |
| | 3 | 0.76 | 0.70 | 0.70 | 0.68 |
| 9 | 1 | 0.76 | 0.69 | 0.70 | 0.67 |
| | 2 | 0.74 | 0.64 | 0.65 | 0.61 |
| | 3 | 0.75 | 0.64 | 0.65 | 0.60 |
| 10 | 1 | 0.62 | 0.62 | 0.62 | 0.61 |
| | 2 | 0.66 | 0.64 | 0.64 | 0.63 |
| | 3 | 0.65 | 0.64 | 0.64 | 0.63 |
| 11 | 1 | 0.82 | 0.81 | 0.81 | 0.81 |
| | 2 | 0.84 | 0.82 | 0.82 | 0.82 |
| | 3 | 0.83 | 0.79 | 0.79 | 0.78 |
| 12 | 1 | 0.78 | 0.74 | 0.74 | 0.73 |
| | 2 | 0.77 | 0.69 | 0.69 | 0.66 |
| | 3 | 0.77 | 0.67 | 0.67 | 0.64 |
| 13 | 1 | 0.81 | 0.79 | 0.79 | 0.79 |
| | 2 | 0.81 | 0.78 | 0.78 | 0.77 |
| | 3 | 0.82 | 0.76 | 0.76 | 0.75 |
| 14 | 1 | 0.77 | 0.77 | 0.77 | 0.77 |
| | 2 | 0.81 | 0.81 | 0.81 | 0.81 |
| | 3 | 0.82 | 0.82 | 0.82 | 0.82 |
| 15 | 1 | 0.76 | 0.72 | 0.72 | 0.72 |
| | 2 | 0.76 | 0.72 | 0.72 | 0.70 |
| | 3 | 0.78 | 0.72 | 0.72 | 0.71 |
| 16 | 1 | 0.78 | 0.78 | 0.78 | 0.78 |
| | 2 | 0.77 | 0.76 | 0.76 | 0.76 |
| | 3 | 0.77 | 0.77 | 0.77 | 0.76 |