# Harnessing Generative AI for Overcoming Labeled Data Challenges in Social Media NLP

**Chandreen Ravihari Liyanage**

Computer Science Department

Lakehead University, Thunder Bay, Ontario

A thesis submitted to Lakehead University in partial fulfillment of the requirements for the Master of Science degree in Computer Science in the Faculty of Science and Environmental Studies

Fall 2023

# Examining Committee Members

This thesis was reviewed and approved by the following examining committee:

| | |
|---|---|
| Supervisor: | Dr. Thiago E. Alves de Oliveira<br>Assistant Professor, Lakehead University |
| Co-supervisor: | Dr. Vijay Mago<br>Associate Professor, York University |
| Internal Committee Member: | Dr. Garima Bajwa<br>Assistant Professor, Lakehead University |
| External Committee Member: | Dr. Rajesh Sharma<br>Associate Professor, University of Tartu |

# Declaration

I declare that the work presented in this thesis is entirely original and has been conducted by myself, under the overall guidance of my supervisor(s). The content herein has not been submitted to any other institute for the purpose of obtaining a degree or diploma. In instances where external materials, including concepts, ideas, text, expressions, data, graphs, diagrams, theoretical analysis, results, etc., have been utilized, I have appropriately acknowledged and credited them in the text of the thesis, providing detailed references. Any verbatim sentences derived from published works have been clearly identified and quoted. I affirm that, to the best of my knowledge and understanding, no part of this thesis constitutes plagiarism, and I willingly accept full responsibility in the event of any complaint arising.

This thesis comprises two original papers previously published or submitted to journals or conferences for publication, listed as follows:

| Thesis chapter | Publication/ Citation | Status |
|---|---|---|
| Chapter 3 | C. Liyanage, M. Garg, V. Mago, and S. Sohn, 'Augmenting Reddit Posts to Determine Wellness Dimensions impacting Mental Health', in The 22nd Workshop on Biomedical Natural Language Processing and BioNLP Shared Tasks, 2023, pp. 306–312. | Published |
| Chapter 4 | C. Liyanage, R. Gokani and V. Mago, "GPT-4 as a Twitter Data Annotator: Unraveling Its Performance on a Stance Classification Task". TechRxiv, 15-Sep-2023, doi: 10.36227/techrxiv.24143706.v1. | Under Review |

Adhering to the Lakehead University Policy on Authorship, I acknowledge the contributions of other researchers to my dissertation and have obtained explicit permission from each co-author for the inclusion of their materials. With the aforementioned clarification, I certify that this dissertation, along with the associated research, reflects my individual work.

# Acknowledgements

I extend my sincere gratitude to Dr. Vijay Mago, my thesis supervisor, whose guidance, encouragement, and financial support were invaluable throughout my research journey. I am also thankful to Dr. Ravi Gokani and Dr. Muskan Garg for their ongoing support and guidance, contributing to the refinement of my ideas and the enhancement of the quality of my work.

Special appreciation goes to my colleagues at DaTALab for their encouragement, support, and insightful discussions, which served as a constant source of inspiration and motivation. Their collaborative spirit significantly enriched my research experience. I owe a debt of gratitude to my family for their unwavering love, support, and encouragement. Their steadfast belief in me provided the strength to persevere and surmount the challenges encountered.

This thesis represents the culmination of two years of dedicated effort, and I am thankful for the support of numerous individuals who made this achievement possible. Finally, I am immensely thankful to Lakehead University for granting access to the resources at the CASES building and providing financial support crucial for the successful execution of this work.

# Dedication

*I dedicate this thesis to my beloved husband, son, parents, brother, and sister whose support, encouragement, love, and sacrifices have been my greatest source of strength throughout this journey. You are my inspiration and the reason I strive for excellence.*

# Abstract

With the introduction of Transformers and Large Language Models, the field of NLP has significantly evolved. Generative AI, a prominent transformer-based technology for crafting human-like content, has proven powerful skills across numerous NLP tasks. Simultaneously, social media emerges as a rich source for NLP explorations, offering vast and diverse datasets that capture real-time language usage, making it a valuable resource for understanding and advancing NLP techniques. Given that supervised learning is the most popular Machine Learning training method, numerous NLP studies necessitate labor-intensive annotation of social media text. However, despite the large amount of data available, the social media data annotation process is usually difficult for human experts due to unique characteristics of text, such as shortness, lack of context, embedded socio-cultural perspectives, and varied writing styles. The challenges in constructing labeled social media datasets often result in a scarcity of labeled data and the generation of low-quality labels. Moreover, these datasets frequently face class imbalance due to the limitations of labeled samples. Hence, ensuring a balanced, high-quality dataset in sufficient quantities is crucial for the robust and accurate development of NLP models. To address these challenges, this study has identified the usage of generative AI for social media labeled text generation. Specifically, this study focuses on two key objectives: augmenting existing labeled text samples and annotating unlabeled text samples using generative AI. As the generative AI technology, the Generative Pre-trained Transformer model, a prevalent choice for AI-based content generation is employed in different versions throughout the study and evaluated its performance against traditional text augmentation and annotation methods. While both studies centered around multi-class classification problems, the text augmentation approach delves into augmenting human wellness dimensions using Reddit posts, and text annotation tackles stance detection on abortion legalization using Twitter posts. By employing various classifiers, the subsequent investigations aim to enhance classification performance in social media NLP, emphasizing the common goal of expanding labeled datasets, while enhancing the quality of labels.

# Table of Contents

# List of Figures

# List of Tables

# List of Acronyms

| | |
|---|---|
| AI | Artificial Intelligence |
| BERT | Bidirectional Encoder Representations from Transformers |
| CoT | Chain-of-Thoughts |
| EDA | Easy Data Augmentation |
| FN | False Negative |
| FP | False Positive |
| GPT | Generative Pre-trained Transformer |
| IVA | Intellectual and Vocational Aspect |
| LLM | Large language model |
| MCC | Matthew's Correlation Coefficient |
| ML | Machine Learning |
| MLP | Multi-Layer Perceptron |
| NER | Named-entity recognition |
| NLP | Natural Language Processing |
| PA | Physical Aspect |
| POS | Part-of-speech |
| RLHF | Reinforcement Learning from Human Feedback |
| ROC_AUC | Area under the Receiver Operating Characteristic curve |
| SA | Social Aspect |
| SDOH | Social Determinants Of Health |
| SEA | Spiritual and Emotional Aspect |
| SVM | Support Vector Machine |
| TP | True Positive |
| TRob | A Roberta-based model pre-trained on a general Twitter dataset |
| TRobSen | A Roberta-based model pre-trained on a Twitter sentiment dataset |
| TRobStan | A Roberta-based model pre-trained on a Twitter stance dataset |
| WD | Wellness Dimension |
| XGBoost | Extreme Gradient Boosting |

# Chapter 1

# Introduction

## 1.1 Problem Description and Motivation

In recent years, the field of Natural Language Processing (NLP) has witnessed remarkable advancements, driven by the introduction of Transformers and the rapid growth in the generation of Large Language Models (LLMs) [3]. Social media is a valuable data source for NLP research in various domains, as it provides diverse, real-time, large-scale, user-generated, and multimodal text with embedded images, videos, and audio data [4]. Researchers can access APIs, such as Twitter API[1], Reddit API[2], Facebook Graph API[3], and YouTube Data API[4] to collect these data for their analyses. Social media serves as an excellent resource for investigating a wide range of societal problems, offering insights into issues like public opinion, social trends, and even emergency responses during crises.

However, despite the numerous advantages of leveraging social media text data for NLP research, there are significant challenges that researchers must contend with [5, 6]. Firstly, the data can be noisy and unstructured, containing slang, abbreviations, misspellings, and grammatical errors, making it challenging to understand [7, 8]. Secondly, social media data often lacks context, as posts are short and may not provide the full story, necessitating context reconstruction for comprehension [7, 8]. Moreover, these texts are often embedded with diverse social and cultural beliefs and perspectives, as the users come from various backgrounds and communities, leading to a wide range of writing styles, and contextual meanings [6, 8]. These unique characteristics and diversity introduce complexity to accurately interpret and understand the textual data on social media.

On the other hand, most popular Machine Learning (ML) and NLP training method is supervised learning, which necessitates the labeling of social media text for training and evaluation [9]. Labeling the text data is a time-consuming, expensive, and resource-intensive task, often requiring human annotators to go through vast volumes of content and training sessions [8, 10]. Moreover, the characteristics mentioned earlier, including brevity, lack of context, and diverse writing styles, further complicate the labeling process. They make it difficult for annotators to capture the full

[1] https://developer.twitter.com/en/docs/twitter-api

[2] https://www.reddit.com/dev/api/

[3] https://developers.facebook.com/docs/graph-api/

[4] https://developers.google.com/youtube/v3

context and nuances of the content, leading to potential labeling errors and ambiguity. Further, the presence of a wide array of cultural and regional references in social media texts can confuse annotators and introduce cultural bias into labeled datasets. As a result, using human experts to reliably label large amounts of social media text has posed a significant challenge for NLP researchers.

Besides, ensuring the availability of a class-wise balanced dataset for ML modeling is also crucial, as it enables the development of more accurate and unbiased NLP models [11, 12]. However, acquiring a balanced dataset is difficult due to the limitations in generating labeled data. These challenges hinder the development of robust and unbiased NLP models from this rich and varied data source. Hence, it is necessary to experiment with the scenarios for obtaining quality labeled data, in balanced and sufficient quantities.

To tackle these issues, researchers are increasingly investigating Artificial Intelligence (AI) based methods for generating content. In this context, this study has identified two potential solutions for expanding the labeled datasets using generative AI as follows:

1. Augmentation of existing labeled text samples.
2. Automatic annotation of unlabeled text samples.

These strategies are vital in utilizing most of the available data and improving NLP models' performance on social media text. While both approaches can effectively increase the size of the labeled dataset, concerns arise regarding label reliability and the quality of the added data. Hence, this thesis explores how the above two approaches, both of which share common goal despite being distinct in their approaches, could address this problem. In summary, this study is motivated by the necessity to address data acquisition issues by harnessing the potential of Generative AI techniques, particularly in response to the shortage of labeled data in the context of social media content.

## 1.2 Thesis Contribution

The main aim of this study is to investigate how generative AI techniques can respond to text data augmentation and annotation; two techniques for addressing the shortage of labeled data in the context of social media content and ultimately for the enhancement of the classification performance. To achieve this primary goal, the current study addressed two main problems as follows. The breakdown of some general objectives for these two studies is also listed here.

1. **Study 1**: Data augmentation using generative AI.
   - Objectives:
     - Use textual data augmentation techniques to create new samples.
     - Assess the syntactic and semantic similarity of original and augmented text samples.
     - Compare and contrast the traditional and generative AI-based augmentation techniques in terms of classification performances.
2. **Study 2**: Data annotation using generative AI.
   - Objectives:
     - Construct a reliable human-annotated dataset and use generative AI models to relabel the dataset.
     - Compare and contrast the performance of models trained on human-labeled datasets with those trained on generative AI-based labels.
     - Compare and contrast the performance of models trained on labels generated using different AI-based prompting techniques.

Both studies center around a multi-class classification problem. The first focuses on classifying dimensions of human wellness using Reddit posts, while the second handles a stance classification problem using Twitter posts. We selected Twitter and Reddit as the social media platforms, given their prominence in NLP research, highlighting their extensive user interactions and diverse

**Table 1.1:** Comparison of the studies

| Aspect | Study 1 | Study 2 |
|---|---|---|
| ML problems | Data Augmentation and multi-class classification | Data Annotation and multi-class classification |
| Data source | Reddit text posts | Twitter text posts |
| Problem domain | Human wellness-dimension identification | Stance detection on abortion legalization |
| Generative AI-models | GPT-3, GPT-3.5 | GPT-4 |
| Classifiers | BERT | 26 classifiers, including LLMs and traditional ML models |

content. These studies employed Generative Pre-trained Transformer (GPT) models to leverage generative AI technology, due to their widespread popularity, broad availability, proven expertise, and effective applicability across a variety of applications.

Moreover, each study evaluates its performance against traditional methods, such as Back-translation and Easy Data Augmentation (EDA) for data augmentation, and human expert labels for data annotation. Despite differences in their selected classification problems, data sources, and methodological approaches, both studies share the common goal of expanding labeled datasets using generative AI. Table 1.1 provides a concise comparison of the two studies, offering a comprehensive overview of the upcoming chapters in this thesis.

This overall work underscores the promise of GPT models as valuable tools for data augmentation and annotation, ultimately contributing to the broader field of ML and NLP. All the work conducted for this study is open-sourced and publicly available in GitHub repositories[5].

## 1.3 Thesis Organization

This thesis is structured as follows: In Chapter 2, background information on the ML and NLP technologies employed in this study is presented. Chapter 3 presents the first study, introducing a novel generative AI approach for augmenting social media text. Chapter 4, outlines the specifics of the second study, designed to assess and establish a benchmark for generative AI-based data

[5] https://github.com/Ravihari123

annotation in social media text classification. Finally, Chapter 5 concludes the research conducted for this thesis, summarizing the key findings and contributions.

# Chapter 2

# Background

## 2.1 Machine Learning Background

This study used a mix of basic and advanced ML techniques to implement and compare various models and methods related to text generation and classification problems. The goal of this section is to provide the fundamental details and background necessary to understand the ML methods used in this study.

Machine learning, a subset of artificial intelligence, empowers systems to obtain insights from data, recognize patterns, and make decisions with minimal human intervention [13, 14]. ML scenarios can be categorized into distinct types based on the nature of the training data, the order and method of data reception, and the test data used to assess the learning algorithm [15]. These primary categories include supervised learning, unsupervised learning, semi-supervised learning, transductive inference, online learning, reinforcement learning, and active learning. Machine learning techniques have found successful applications in computer vision [16], NLP [17], speech recognition [18], and many more across a wide spectrum of fields, including but not limited to finance [19], entertainment [20], education [21], agriculture [22], and medical [23] domains. Despite the wide applications, ML encompasses issues, such as the quality and quantity of input data, model interpretability, ethical considerations, and the need for large computational resources [13, 15, 24]. Addressing these challenges is essential for realizing the full potential of ML in various applications. The subsequent sections will provide a concise overview of the ML models and techniques used in this study.

### 2.1.1 Traditional Machine Learning Models

Classification is a fundamental task in ML and data analysis. Traditional ML models have been widely employed for this purpose, by extracting patterns in data using various feature engineering techniques [25]. These models are also adept at learning patterns in text data and making predictions or classifications based on those patterns. Some of the common feature engineering techniques used to extract patterns in text are word embeddings, named entity recognition (NER), TF-IDF (Term Frequency-Inverse Document Frequency), and N-grams [25, 26]. This study employed several common traditional ML algorithms, namely Logistic Regression, Random Forest, Support Vector Machine (SVM), Multi-Layer Perceptron (MLP), Gradient Boosting trees, and Extreme Gradient Boosting (XGBoost). Each of these models operates based on distinct principles and mathematical foundations.

### Logistic Regression

Logistic regression is a fundamental statistical and ML technique used for binary classification and, with slight modifications, for multi-class classification. Unlike Linear Regression (LR), which is used for continuous target variables, logistic regression is designed to predict the probability of an instance belonging to one of two classes [27]. The output of a logistic regression model is a logistic curve (S-shaped curve) that maps input features to a probability score between 0 and 1 [28]. In logistic regression, the model makes use of a logistic function (also called the sigmoid function) to transform a linear combination of input features into a probability score [27]. The logistic function ensures that the output is bounded within the [0, 1] range, making it suitable for binary classification tasks [28]. The model is trained using a technique called maximum likelihood estimation, which aims to find the parameters that maximize the likelihood of the observed data given the model.

Logistic regression is not only widely used for classification but also provides insights into the relationship between input features and the likelihood of an outcome. It is a linear model, which means it assumes a linear relationship between the input features and the log-odds of the target variable [29]. Logistic regression is interpretable, computationally efficient, and performs well in a wide range of scenarios, especially when the classes are well-separated and the data is relatively simple [27].

### Random Forest

A Random Forest is a powerful ensemble learning method in ML, primarily used for classification and regression tasks [30]. It belongs to the family of decision tree-based algorithms and is known for its robustness, versatility, and high predictive accuracy [31]. The ”forest” in Random Forest is composed of multiple decision trees, hence the term ”ensemble” is used [30, 31]. These decision trees are constructed independently during training, and they collectively make predictions by either voting (in classification) or averaging (in regression) their individual outputs.

What sets Random Forest apart is its ability to mitigate the overfitting issues commonly associated with single decision trees [32]. Each tree is trained on a bootstrap sample of the original dataset, and at each node in the tree, only a random subset of the features is considered for splitting. This randomness introduces diversity among the individual trees, reducing their correlation and enhancing the model's generalization capabilities [31]. Random Forests are also well-equipped to handle high-dimensional data, are robust to outliers, and can effectively capture complex, nonlinear relationships within the data. Additionally, they provide a feature importance score, allowing users to assess the relevance of each feature in making predictions [33].

### Support Vector Machines

SVM is an ML algorithm used for both classification and regression tasks. SVM finds an optimal hyperplane that best separates data into different classes while maximizing the margin, which is the distance between the hyperplane and the nearest data points (support vectors) [34, 35]. SVM aims to achieve the best balance between maximizing this margin and minimizing classification errors [35]. This technique is particularly effective in scenarios where the data is not linearly separable, as it can employ kernel functions to map the data into higher-dimensional feature spaces where separation becomes possible [34, 36]. This property allows SVM to capture complex, nonlinear relationships between features. SVMs are also robust against overfitting, making them well-suited for high-dimensional data and scenarios with limited training samples [35,37]. Additionally, SVMs offer a unique advantage in classification by providing a clear decision boundary and the ability to handle multi-class classification problems through various strategies. They also provide insight into feature importance, allowing users to identify the most influential features in their models [36].

### Multi-Layer Perceptron

MLP is a foundational neural network architecture in ML. It belongs to the class of feedforward artificial neural networks which consists of multiple layers of interconnected neurons, or artificial nodes [38]. The MLP typically consists of an input layer, one or more hidden layers, and an output layer. Each neuron in the network performs a weighted sum of its inputs, followed by the application of an activation function, and then passes its output to subsequent layers [38].

MLPs are capable of modeling complex, nonlinear relationships in data, making them suitable for a wide range of tasks, including classification, and regression [39]. They are known for their ability to approximate arbitrary functions and are effective at feature learning, allowing them to automatically extract relevant features from the input data. Training an MLP involves optimizing the weights and biases of the network to minimize a chosen loss function, typically through gradient-based optimization techniques like backpropagation [38, 40]. The choice of activation functions, network architecture, and hyperparameters plays a critical role in determining the performance of an MLP.

Despite their effectiveness, MLPs have some limitations, such as the potential for overfitting, and their performance can heavily depend on the quality and quantity of training data [40]. Nevertheless, they remain a fundamental building block in deep learning and have been instrumental in the success of various AI applications, including image and speech recognition, natural language processing, and many others [39].

### Gradient Boosting Trees

Gradient Boosting Trees is an ML technique that can be used for both regression and classification problems. It belongs to the ensemble learning family, and its primary goal is to create a strong predictive model by combining the predictions of multiple decision trees [41, 42]. Unlike other ensemble methods like Random Forest, Gradient Boosting Trees builds trees sequentially, with each new tree aiming to correct the errors made by the ensemble of the previously built trees [42].

Gradient Boosting Trees are known for their exceptional predictive accuracy and ability to capture complex relationships within the data. They work by optimizing a loss function, typically mean squared error for regression or cross-entropy for classification, by iteratively training weak decision trees that focus on the instances where the current model performs poorly [41]. The idea is to find the optimal combination of these trees by assigning weights to each one based on their performance.

XGBoost, LightGBM, and CatBoost are popular variations of Gradient Boosting Trees, each with its own optimization techniques and hyperparameters [41]. The models produced by Gradient Boosting Trees are interpretable, and they also provide feature importance scores, enabling users to understand the relevance of different features in the predictive process. Despite their exceptional performance, Gradient Boosting Trees can be computationally intensive and require careful tuning to prevent overfitting [43].

### Xtream Gradient Boosting

XGBoost is a popular ML algorithm, which is an optimized and scalable implementation of the gradient boosting framework [41]. XGBoost has gained wide attention across a range of tasks, including classification, regression, ranking, and anomaly detection [44]. XGBoost improves upon traditional gradient boosting methods by employing several key innovations, such as a regularized objective function to control overfitting, a specialized algorithm for handling missing values, and the ability to parallelize and distribute the training process, making it faster and more scalable [41,45]. It also incorporates a flexible framework for handling user-defined custom loss functions and evaluation criteria [45]. The algorithm is known for its capability to handle high-dimensional data with a large number of features and can automatically learn feature importance for model interpretability [46].

## 2.2 Natural Language Processing

NLP is a multidisciplinary field that combines linguistics, computer science, mathematics, and AI to bridge the gap between human communication and machine understanding [47]. NLP algorithms and models are trained on vast amounts of text data to learn the nuances of human language,

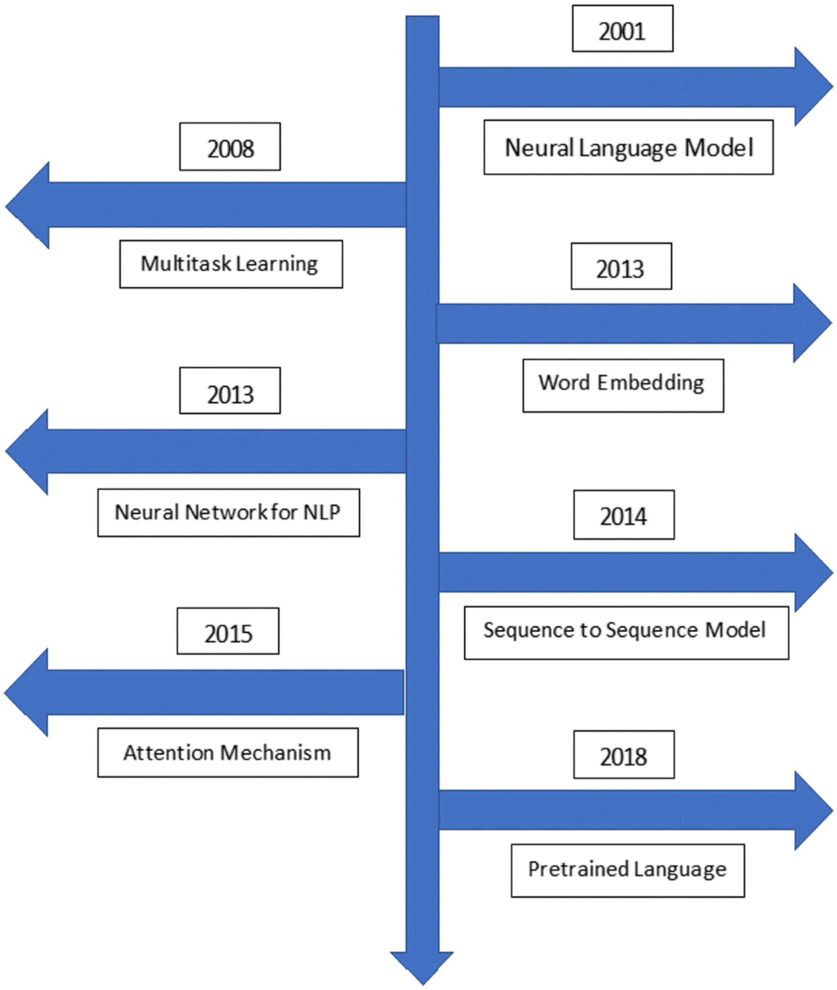


**Figure 2.1:** A brief overview of the recent developments in NLP (sourced from [1]).

enabling them to perform a wide range of tasks [48]. Recent breakthroughs in NLP have been driven by the Transformer technology [49] and the large pre-trained language models built upon this Transformer architecture [3]. These models, with millions or even billions of parameters, have achieved unprecedented levels of performance in tasks, such as text generation, text summarization, classification, question answering, and language translation [47]. Figure2.1 depicts significant advancements in NLP tools and methods, sourced from [1].

NLP has a profound impact on diverse industries, including but not limited to healthcare, finance, customer support, and content creation. In healthcare, NLP is used for clinical documentation [50], disease diagnosis [51], and medical record analysis [52]. In finance, it aids in fraud detection [53], sentiment analysis for trading [54], and risk assessment [55]. Customer support chatbots employ NLP to provide instant assistance [56], while content creators use NLP to automate content generation and recommendations [57].

The following sections will discuss the concepts of NLP techniques, which are related to this study, including language models, their learning approaches, generative AI technology, and models and prompt-based learning techniques related to generative AI in detail.

### 2.2.1 Large Langauge Models

Large language models have revolutionized the field of NLP by expanding the boundaries of what machines can do with human language [58, 59]. The base of the LLMs is Transformers [49], a deep neural network architecture that captures long-range dependencies by identifying contextual relationships between the tokens in the input text (sequential data) using self-attention mechanisms [59]. These LLMs, such as GPT by OpenAI [60,61], Bidirectional Encoder Representations from Transformers (BERT) by Google [62], Large Language Model Meta AI (LLaMa) by Meta [1], and Cross-lingual Language Model (XLM) by Facebook AI Research [63] are characterized by their massive scale, usually containing tens or hundreds of billions of parameters. There are three classes of transformer-based LLM architectures; decoder-only(GPT), encoder-only (BERT, XLM-R), or encoderdecoder(BART, T5) [64]. They excel at a wide range of NLP tasks, from language translation and text summarization to sentiment analysis and question-answering [59, 65]. By pre-training on vast corpora of text data and fine-tuning on specific tasks, these models can generalize and adapt to various linguistic challenges, making them versatile tools for applications in chatbots, content generation, language understanding, and many more. Presently, there is extensive research being conducted on LLMs, with a primary focus on areas, such as fine-tuning these models for specific tasks, optimizing prompts, and assessing their performance across various problem-solving scenarios [58, 59].

This section discusses the details of the LLMs used in this study, namely BERT, ALBERT, DeBERTa, MPNet, and RoBERTa. The specific versions of the models and the details of the datasets that they have been pre-trained on will be further discussed in chapter 3 and 4.

[1] https://ai.meta.com/blog/large-language-model-llama-meta-ai/

### BERT: Bidirectional Encoder Representations from Transformers

BERT is a multi-layer bidirectional transformer encoder-based language model built for different downstream NLP tasks, including text classification, question and answering and language translation [62]. By using the transformer architecture as the feature extractor, BERT has introduced mask language modeling and prediction of the next sentence from both directions in a sentence, allowing it to understand the meaning of words in the context of the entire sentence [62, 66].

Through pre-training on a massive corpus of unlabeled text, BERT learned to represent words and sentences in a continuous vector space, creating contextual embeddings that reflect the relationships between words and their surrounding context. These pre-trained embeddings can then be fine-tuned for specific NLP tasks, such as text classification, by only adding a simple classification layer to the pre-trained model [67]. The original work performed these downstream tasks using GLUE benchmark datasets, such as Multi-Genre Natural Language Inference (MNLI), Quora Question Pairs(QQP), Question Natural Language Inference (QNLI), Stanford Sentiment Treebank (SST-2), and Corpus of Linguistic Acceptability (CoLA). The best hyperparameter configuration recommended for any downstream task is; batch size = 16/ 32, Learning rate (Adam): 5e-5/ 3e-5/ 2e-5, and number of epochs: 2/ 3/ 4 [62].

BERT's contributions to NLP are various, including state-of-the-art performance on a wide range of tasks, elimination of the need for task-specific feature engineering, and improved model generalization.

### ALBERT: A Lite BERT

ALBERT is a highly influential model in the field of NLP [68]. ALBERT builds on the architecture of BERT, however, introduces several innovations to reduce its size and computational requirements while maintaining or even improving its performance. ALBERT leverages two key techniques: parameter sharing and cross-layer parameter sharing. The former reduces the number of model parameters by sharing them across layers, and the latter shares parameters across the transformer layers, making the model significantly more efficient [68, 69].

This reduction in model size and the total number of parameters enables ALBERT to be trained on larger datasets and fine-tuned more effectively for various NLP tasks. Despite its lighter architecture, ALBERT often outperforms or matches the performance of BERT on tasks, such as text classification, question and answering, and language understanding [69,70]. ALBERT's efficiency and effectiveness have made it a good choice in NLP, particularly for resource-constrained environments where computational power and memory are limited [71].

### DeBERTa: Decoding-enhanced BERT with Disentangled Attention

DeBERTa is developed as an enhancement of the BERT model, which incorporates several innovative features that significantly improve its performance and capabilities [72]. One of its main contributions is the introduction of disentangled attention mechanisms, which allow the model to separate different types of information during processing, making it more efficient and effective in understanding context and relationships in text. DeBERTa also introduces the concept of ”masked sentence prediction,” which aims to predict missing sentences within a paragraph, encouraging the model to understand more profound contextual relationships. Moreover, DeBERTa takes into account both a word's absolute position and its relative position to capture structured information more effectively, and it incorporates techniques to reduce the training and inference time, making it more efficient than some previous models like BERT which simply sums word position and content [73]. These enhancements lead to better contextual embeddings, which are invaluable for various NLP tasks.

### MPNet

MPNet, which stands for ”Masked and Permuted Pre-training for Language Understanding” is designed to address the limitations of earlier pre-training models, such as BERT and XLNet [66]. While BERT excels in masked language modeling during pre-training, it overlooks dependencies among predicted tokens. XLNet, on the other hand, introduces permuted language modeling (PLM) to address this issue, however, it does not fully consider the position information of tokens in a sentence, resulting in position discrepancies between pre-training and fine-tuning [74].

In response to these challenges, MPNet is introduced as a novel pre-training approach that combines the strengths of both BERT and XLNet while mitigating their shortcomings. MPNet employs permuted language modeling, similar to XLNet, to capture token dependencies, thereby enhancing its contextual understanding of text. Additionally, it leverages auxiliary position information to ensure that the model processes the full sentence, thus reducing the position discrepancy observed in XLNet [66, 75]. MPNet's pre-training phase is conducted on a vast dataset consisting of over 160GB of text corpora, followed by fine-tuning on a diverse set of downstream NLP tasks, including GLUE and SQUAD.

### RoBERTa: Robustly Optimized BERT Pre-training Approach

The RoBERTa is built upon the same pre-training framework as BERT, however, introduces several innovative optimizations to improve its performance and robustness [76]. One of its key enhancements is the use of larger datasets for pre-training, encompassing more web text, books, and Wikipedia content, which allows RoBERTa to learn more nuanced language representations. It also incorporates dynamic masking during training, which replaces BERT's static masking strategy, further improving the model's contextual understanding. RoBERTa optimizes the hyperparameters and training schedule, and it removes BERT's next sentence prediction (NSP) task. This streamlining of pre-training objectives enables RoBERTa to outperform BERT on a wide range of NLP benchmarks [77]. Its training techniques also include sentence-level and document-level training, helping the model capture document-level semantics more effectively.

### Models trained on Twitter Data: TweetEval and BERTweet

Twitter is known for its informal language, non-standard abbreviations, hashtags, and a variety of cultural references, which often pose challenges for traditional NLP models. TweetEval and BERTweet are two models fine-tuned on two base LLMs, which obtained the ability to handle the intricacies of Twitter language and the informal nature of tweets. This study uses different versions of these two models for classifying Twitter text.

TweetEval is a framework trained for seven different Twitter-specific classification tasks, which are emoji, hate, offensive, irony, sentiment, emotion, and stance classification [78]. TweetEval is a retrained version of the RoBERTa-base model [76] which used a nearly 60M tweets dataset and the same hyper-parameter settings to re-train the models. The choice of RoBERTa was motivated by two key reasons: it is a high-performing model in the GLUE benchmark, and it is better suited for tasks involving single sentences like tweets, as it does not employ the Next-Sentence-Prediction loss. However, to provide a better context on social media text, they experimented with three training strategies using three different RoBERTa variants, 1) use original pre-trained language models (RoB-Bs: pre-trained RoBERTa-base), 2) train original architecture from scratch using Twitter data (RoB-Tw) and 3) use pre-trained models and continue training on more Twitter data (RoB-RT: RoB-Bs re-trained on Twitter). They used classification finetuning steps similar to the original RoBERTa, however, added one dense layer at the end and trained all parameters during fine-tuning.

BERTweet is a specialized pre-trained language model tailored for processing and understanding English language text found in tweets [79]. Developed by the research community, BERTweet is built upon the BERT architecture, fine-tuned using the RoBERTa pretraining technique specifically for the unique characteristics of tweets. BERTweet, however, is trained on a vast corpus of Twitter data, allowing it to effectively work on three downstream NLP tasks, Part-of-speech (POS) tagging, NER and text classification. This makes it particularly well-suited for tasks such as sentiment analysis, topic classification, and information extraction in the context of social media data.

### 2.2.2 Learning Approaches in Large Language Models

#### Transfer Learning

One of the huge challenges faced by NLP is the difficulty of acquiring large amounts of data for training big language models. Especially, in practice obtaining labeled data is not an easy task [80]. Moreover, deep learning models present an additional challenge due to their substantial need for extensive computational capabilities. These challenges can be effectively addressed through the application of Transfer Learning, a technique that involves transferring parameters or knowledge from a pre-trained model to a different one [81]. These pre-trained language models capture contextual information and sophisticated language features, modeling both syntax and semantics, thereby delivering state-of-the-art performance on a diverse set of tasks [82]. Depending on the availability of labeled datasets and the type of task, transfer learning can be categorized into two main types: transductive and inductive transfer learning. In transductive transfer learning, both source and target domains/tasks are same and we have no or very few labeled data in the target task. In the context of inductive transfer learning, we utilize insights acquired from one task or domain to enhance performance in another different, but related task/domain, and we have labeled data for the downstream task [81]. By utilizing the transfer learning approach, the power of advanced LLMs can be harnessed on a broader range of downstream tasks to improve their performances. Three of the widely used techniques for employing pre-trained LLMs in downstream tasks are fine-tuning, few-shot learning, and zero-shot learning [83].

### Fine-tuning

Fine-tuning is one of the most common transfer learning techniques in NLP [84]. It is a type of inductive transfer learning technique, which involves further training of the pre-trained model to improve its performance on a different related task [62]. This takes advantage of the knowledge encoded in the pre-trained model's parameters and adapts it to the new task, often with fewer training examples than would be required for training a model from scratch. This process allows the model to transfer its learned features and representations from the source task/domain to the target task/domain. This fine-tuning process usually involves updating the weights of the pre-trained model's layers while keeping the lower layers, which capture more general features relatively fixed, and updating the higher layers to specialize in the target task. Hence, fine-tuning is considered as more parameter-efficient approach as the lower layers of a network are shared between source and target tasks [84]. This technique is useful when training a new model from scratch would be too costly, or time-consuming [85–87]. It can also help to make the new models more robust and improve the performance by using the previous knowledge from the pre-trained model and new knowledge by training on a small new dataset which can be much smaller in scale compared to the vast dataset the LLM was initially trained on [78].

### Few-shot Learning

Few-shot learning is a specialized subfield of ML that focuses on training models to make accurate predictions with only a very limited amount of labeled data. Unlike fine-tuning, which requires a substantial volume of new data for training, the few-shot learning technique typically comes in a few variations, the most common being one-shot learning and few-shot learning [88]. One-shot learning involves training a model to recognize new classes with only one example per class, while few-shot learning generally refers to tasks with a small number of examples, often ranging from a few to several examples per class [89]. This flexibility is made possible by the transfer learning ability of LLMs to generalize from their pre-trained knowledge on extensive text data to the specific task with limited data. Techniques for few-shot learning often involve meta-learning, which focuses on training models to adapt quickly to new tasks or classes, as well as methods like siamese networks, matching networks, and prototypical networks that learn to understand the similarities and differences between classes, even when data is scarce [90, 91]. Few-shot learning has applications in image recognition, natural language processing, and other domains where obtaining abundant labeled data is challenging, making it a valuable approach for real-world ML problems [88].

### Zero-shot Learning

Zero-shot learning is a ML approach where a model is trained to recognize and classify objects or concepts it has never seen or encountered during training [92, 93]. In traditional supervised learning, a model is trained on a labeled dataset with examples of all the classes it will need to classify. In contrast, zero-shot learning enables a model to generalize its knowledge to previously unseen classes by learning to understand the relationships and attributes that describe these classes [93, 94].

Zero-shot learning consists of two distinct stages: the training phase, which involves capturing attribute knowledge, and the subsequent inference phase, where this acquired knowledge is applied to classify instances within a novel class set [95]. The key to zero-shot learning is the use of semantic embeddings or attributes that provide a description of each class or concept in a continuous feature space [92]. These attributes capture the characteristics, properties, and relationships of different classes. During training, the model learns to map the visual or textual features of the input data to these attributes [96]. Next, when presented with a new, unseen class, the model can make predictions based on the similarity between the attributes of the known and unknown classes, even if it has never seen specific examples of the new class. Zero-shot learning is particularly useful in situations where it is impractical or costly to provide labeled data for every possible class, making it applicable in various fields such as computer vision, natural language processing, and more, where novel or rare concepts can emerge [97].

### 2.2.3 Generative Artificial Intelligence

Generative AI is a subfield of artificial intelligence, which can be defined as a technology that uses deep learning models to create human-like content (such as images and text) in response to diverse and complex prompts, including various languages, instructions, and questions [98]. This approach is different from other AI approaches in its capacity to produce novel, human-like output that goes beyond simple pattern recognition or classification [2]. This technology has seen remarkable advancements in recent years, with models like OpenAI's GPT [61] and Google's PaLM [99] showcasing the potential of generative AI. Generative AI caused impacts in various domains, including economic, medical, education, law, and even scientific research [100, 101].

Generative AI, with its capacity to create human-like content across various media, brings substantial advantages, including automation of content generation, improved translation and localization, and personalized recommendations [101, 102]. However, it also poses notable shortcomings, such as challenges in ensuring the quality and accuracy of generated content, ethical concerns regarding the potential for misuse, limitations in true creativity, resource-intensive requirements, and the perpetuation of data biases [100, 102, 103]. Balancing these advantages and shortcomings

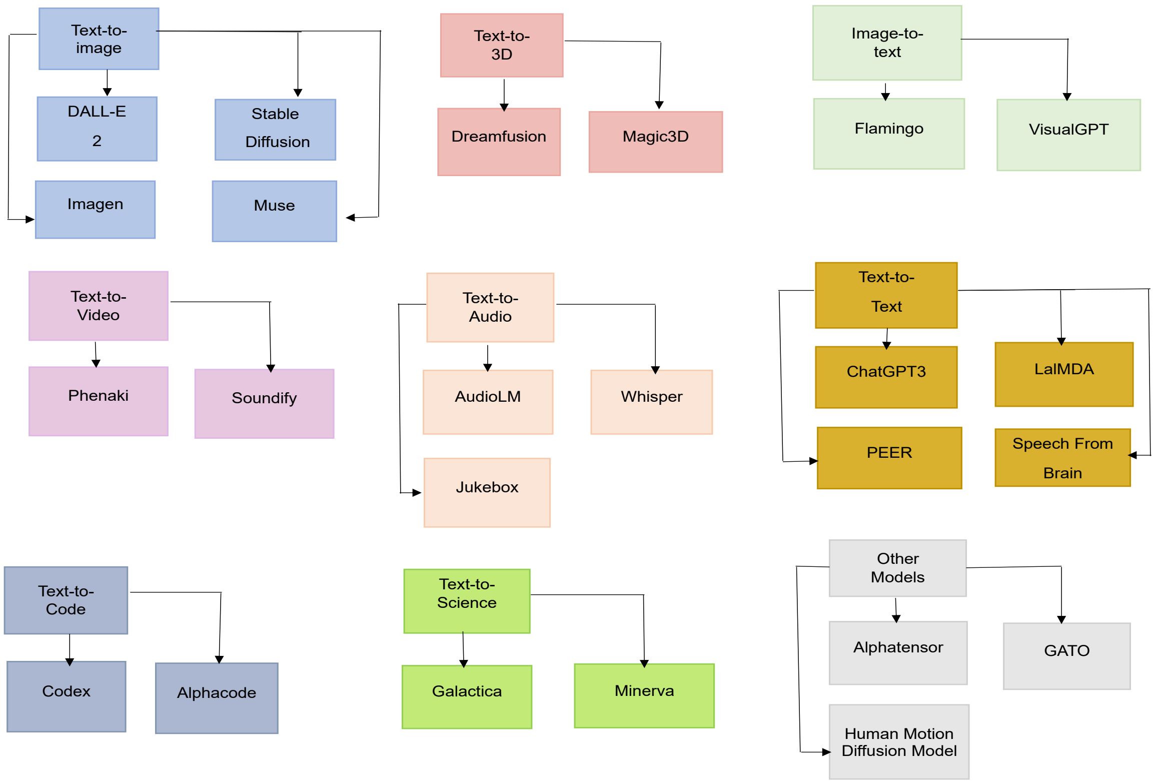


**Figure 2.2:** A taxonomy of the recent most popular generative AI models classified according to input and output formats (sourced from [2]).

necessitates responsible use, the development of ethical guidelines, and ongoing efforts to refine the technology to maximize its benefits while minimizing its risks.

A study has presented a taxonomy of current generative artificial models that outlines the primary relationships between different types of multimedia inputs and outputs [2]. The outcome is depicted in Figure2.2. They identified a total of 9 categories, however notably, only six organizations are responsible for developing these models, which include Google Research, Meta AI, OpenAI, DeepMind, NVIDIA, and Runway.

### Generative Pre-trained Transformers

OpenAI's GPT models which are based on decoder-only Transformer architecture, have been developed to comprehend both human language and computer code [2]. These GPTs generate textual responses based on the input they receive. The utility of GPTs spans a wide spectrum of functions, encompassing tasks such as generating content or code, text classification, summarizing information, engaging in conversations, creative writing, and more [2].

OpenAI introduced the first GPT model (GPT-1) with 117 million model parameters in 2018 trained on a large corpus of books (4.5 GB of text, from 7000 unpublished books of various genres). In the subsequent year, they launched GPT-2, an enlarged model with 1.5 billion parameters, proficient in generating coherent text. By 2020, the introduction of GPT-3 marked a significant leap, presenting a model with 100 times as many parameters (175 billion) as GPT-2 [104]. This GPT-3 has been tested on new NLP tasks to improve the rapid adaptation to different tasks in new datasets. It introduced fine-tuning, a transfer learning technique to train language models in downstream tasks by proving many examples, generally, a couple of hundred as recommended[3]. Recently in 2022, OpenAI has introduced a chatbot called “ChatGPT”, which is built upon the GPT-3.5 model series and fine-tuned through Reinforcement Learning from Human Feedback (RLHF) technique[4].This supervised training technique has employed human AI trainers for conducting conversations in both sides as the user and an AI assistant. Despite its powerful language understanding ability, the researchers have listed several limitations, such as writing incorrect or nonsensical answers, asking questions when user queries are unclear, and giving wordy answers by overusing some phrases.

In March 2023, GPT-4 was introduced as the latest addition, featuring a novel capability to process both text and images, generating text outputs [105]. GPT-4 introduced a rule-based reward model (RBRM) approach in addition to RLHF to ensure correct behavior and prevent harmful content generation. While GPT-4 retains the transformer-based architecture of its predecessors, OpenAI has not released detailed technical reports, including information about the architecture (including model size), hardware utilization, dataset construction, and training method as they did with previous models, citing competitive and safety considerations [105, 106].

[2] https://platform.openai.com/docs/introduction

[3] https://platform.openai.com/docs/guides/fine-tuning

[4] https://openai.com/blog/chatgpt

The primary objective of these models is to enhance their understanding and generation of natural language text, especially in complex scenarios [107]. GPT-4 was tested on various human-designed exams and consistently performed well, surpassing the majority of human test takers. For instance, on a simulated bar exam, it ranked in the top 10%, unlike GPT-3.5, which scored in the bottom 10%. GPT-4 also outperforms previous language models and state-of-the-art systems on traditional NLP benchmarks and the MMLU benchmark [105, 106]. When compared to its predecessor, GPT-4 is reported to be capable of handling approximately eight times more words, demonstrating greater resilience to deception, exhibiting image comprehension, and showing a reduced likelihood of responding to requests that are not permitted [107].

Despite its capabilities, GPT-4 shares limitations similar to the earlier GPT models, including occasional reliability issues, a limited context window, and the inability to learn from experience. Caution is advised when using GPT-4's outputs, especially in contexts where reliability is crucial.

### Prompt-based Learning

A prompt is a directive or set of instructions given to a language model with the intent of customizing its behavior, enhancing its performance, or refining its capabilities to generate desired outputs [108, 109]. Prompt-based learning represents an innovative approach in the realm of NLP, offering a more efficient and cost-effective method for leveraging LLMs. As fine-tuning pre-trained models can be a resource-intensive approach, involving a significant amount of annotated data and computational power, prompt-based learning introduces a way to empower language models to engage in few or zero-shot learning, adapting to novel scenarios with minimal labeled data [110,111].

The process of prompt-based learning can be divided into five essential steps [110, 112]. First, the selection of an appropriate pre-training model is crucial. Next, in the prompt engineering phase, prompts are designed for specific tasks. The third step involves designing responses aligned with the task's objectives, ensuring the model generates the desired output. Expanding this paradigm further to enhance results or adaptability comes as the fourth step. Lastly, designing effective training strategies is essential for the model to learn efficiently and effectively.

Prompt engineering assumes a central role in this approach, guiding pre-trained language models for downstream tasks [112]. Research has highlighted that the effectiveness of prompt design significantly influences the model's performance in these tasks [113]. Notably, language models can produce significantly distinct outputs given different prompts, even if the prompts seem semantically similar [113]. Therefore, prompt engineering is key to aligning the model effectively with the downstream task. For this, identifying the components of a prompt to design them properly and optimizing prompt parameters is crucial.

The components of a prompt encompass the following aspects [109]:

1. **Instruction**: This explains a specific task or directive that serves as a guide for the model's behavior, steering it toward the intended output.

2. **Context**: External information or supplementary context is provided to furnish the model with background knowledge.

3. **Input Data**: At the core of the prompt lies the input data or query, which the model is expected to process and respond to.

4. **Output Indicator**: This defines the desired output's type or format, whether it's a concise answer, an extensive paragraph, or any other specific format.

Similarly, when using GPT models, crafting a suitable prompt by providing the provision of guidelines or illustrative instances is important in completing a task effectively[5]. For this, OpenAI has instructed to adjust the model parameters, such as **model**: defines the type of the model, **temperature**: a measure that indicates the randomness of the output, and **max_tokens**: a hard cutoff limit for the tocken generation. Additionally, they provided a set of instructions to design reliable prompts as listed below[6].

- Use the latest model.

[5] https://platform.openai.com/docs/introduction

[6] https://help.openai.com/en/articles/6654000-best-practices-for-prompt-engineering-with-openai-api

- Put instructions at the beginning of the prompt and use ### or ”””” to separate the instruction and context.

- Be specific, descriptive and as detailed as possible about the desired context, outcome, length, format, style, etc.

- Articulate the desired output format through examples (example 1, example 2).

- Start with zero-shot, then few-shot (example), neither of them worked, then fine-tune.

- Reduce ”fluffy” and imprecise descriptions.

- Instead of just saying what not to do, say what to do instead.

- Code Generation Specific - Use “leading words” to nudge the model toward a particular pattern.

### 2.2.4 Text Embedding and Sentence Similarity

#### Text Embedding

Text embedding is a crucial component in NLP that involves converting textual data into numerical representations, facilitating the analysis and processing of language by ML models. One widely used technique for text embedding is Word Embedding, which represents words as dense vectors in a continuous vector space [114]. Models like Word2Vec [115], GloVe (Global Vectors for Word Representation) [116], and FastText [117] utilize different mechanisms to generate these embeddings. Word2Vec employs a shallow neural network to predict context words based on a target word, while GloVe combines global statistics of the corpus to generate embeddings. FastText, an extension of Word2Vec, considers sub-word information, enhancing its ability to capture morphological nuances [118].

Beyond Word Embedding, sentence and document embeddings capture the semantic meaning of larger text segments [119]. Models like Universal Sentence Encoder (USE), BERT, and sentenceTransformer [7] generate embeddings for entire sentences. Recently, OpenAI has introduced embedding-based API endpoints for generating text embeddings [8]. These models capture contextual information and relationships between words in a way that traditional word embeddings often struggle to achieve. The embeddings generated by GPT-based models not only consider the immediate context of a word but also the broader context of the entire text. This results in richer, more contextually aware representations of language.

### Syntactic and Semantic Similarity

In NLP, syntactic and semantic similarity are crucial aspects that contribute to understanding the structure and meaning of language. The integration of both syntactic and semantic similarity in NLP is essential for tasks that require a nuanced understanding of language [120]. Syntactic information ensures grammatical coherence, while semantic insights contribute to a better comprehension of meaning.

Syntactic similarity involves assessing the structural likeness between sentences by focusing on the arrangement and order of words [1, 120]. Techniques for generating syntactic similarity often include parsing sentences to extract grammatical structures, such as part-of-speech tags, dependency trees, or syntactic constituents [121]. Methods, such as tree edit distance quantify the similarity between syntactic parse trees, offering a measure of how closely related the structures are [121]. Applications of syntactic similarity are diverse, ranging from grammar checking and sentence paraphrasing to machine translation [122], where maintaining syntactic coherence is crucial.

Semantic similarity, on the other hand, focuses on understanding the meaning conveyed by sentences. This involves identifying keywords crucial for comprehending the interactions between words or various concepts within the sentence. At the semantic level, words are inspected for their dictionary definition, or their interpretation is derived from the contextual cues provided by the sentence [1]. Various techniques contribute to measuring semantic similarity, with traditional methods and advanced neural network-based approaches being prominent [123]. Vector space models,

[7] https://www.sbert.net/examples/applications/computing-embeddings/README.html

[8] https://platform.openai.com/docs/guides/embeddings/what-are-embeddings

a traditional approach, represent words or sentences as vectors in a high-dimensional space, calculating similarity using metrics, such as Cosine, Jaccard, and Manhattan similarity [123]. Advanced techniques involve the use of deep learning models, such as Siamese Networks [124] or Transformer-based models like BERT, to generate embeddings that capture semantic relationships between sentences [125]. Applications of semantic similarity span information retrieval, question and answering systems, and sentiment analysis, where discerning the underlying meaning and context is critical [123].

## 2.3 Data Imbalance and Augmentation

Class imbalance in ML occurs when the distribution of classes within a dataset is significantly skewed, with one or more classes having notably fewer instances than others [126]. This imbalance poses challenges as models tend to be biased toward the majority class, leading to poor generalization of minority classes. Addressing class imbalance is crucial as in real-world scenarios, certain classes may be rare but still of significant interest [127]. In NLP, the class imbalance can affect tasks, such as sentiment analysis, medical diagnostics, fraud detection, or rare event prediction [126, 127]. Techniques to mitigate class imbalance include resampling methods (oversampling minority or undersampling majority classes), using different evaluation metrics (precision, recall, F1-score), and employing ensemble methods that handle imbalanced datasets more effectively [128, 129]. Furthermore, the incorporation of advanced algorithms, such as cost-sensitive learning and synthetic data generation, contributes to improving model performance on minority classes in NLP tasks [128, 129].

Data augmentation in NLP involves creating variations of existing training data to enhance the robustness and generalization ability of ML models. The primary goal is to increase the diversity of the dataset, improving the model's ability to handle various input scenarios [130]. Data augmentation serves multiple purposes in NLP, including addressing the scarcity of labeled data, preventing overfitting/ mitigating bias, and handling class imbalance [131, 132]. Data augmentation methods in NLP can be classified into three main types: paraphrasing, noising, and sampling [130]. In paraphrasing-based methods, augmented data is generated by creating paraphrases of the original data, which introduces relatively fewer modifications compared to the original data. Noising-based methods introduce additional continuous or discrete noises, such as word insertion, swapping, deletion, or synonym replacement to the original data, resulting in more substantial changes. Sampling-based methods focus on understanding the distribution of the original data and use this knowledge to generate new data as augmented data. By introducing variations, data augmentation exposes the model to a broader range of linguistic patterns, making it more adept at handling diverse inputs and improving overall performance.

### 2.3.1 Easy Data Augmentation

EDA is a noising data augmentation technique tailored commonly for NLP tasks [133]. EDA involves applying operations, such as synonym replacement, random insertion, random deletion, and random swapping to augment the training data. Synonym substitution involves the random selection of non-stop words from sentences, which are then replaced with synonyms chosen at random. Random insertion, on the other hand, entails identifying a non-stop word, selecting a random synonym, and inserting it at a random position within the sentence. Random swap includes the random selection of two words in a sentence, with their positions exchanged. Lastly, random deletion involves the probabilistic removal of each word in a sentence with a probability P [134]. An example of these four operations are demonstrated in Table 2.1. The input text of ”The playful kitten chased a colorful ball” has changed into ”small and curious kitty pursued a playful, vibrant”, after the four types of EDA operations. These simple yet effective operations introduce variations in the data, making the model more robust and reducing the risk of overfitting. However, EDA comes with certain weaknesses as well. These include the potential introduction of semantic inconsistencies, inaccuracies in representing the intended context or sentiment, and the alteration of sentence structures, thereby affecting syntactic patterns. These limitations suggest that EDA might not yield significant performance improvements, particularly when applied to pre-trained large language models used as classifiers. [133].

**Table 2.1:** Example of performing four EDA operations on the original text ”The playful kitten chased a colorful ball”.

| **Operation** | **Text after the operation** |
|---|---|
| Synonym replacement | The playful kitty pursued a vibrant ball |
| Random Insertion | The playful and curious kitty pursued a small, vibrant ball |
| Random Swapping | The small and curious kitty pursued a playful, vibrant ball |
| Random Deletion | small and curious kitty pursued a playful, vibrant |

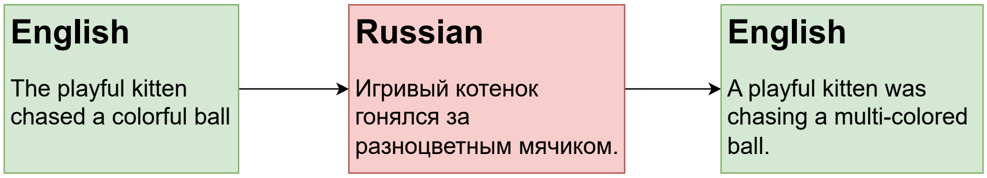

English
The playful kitten chased a colorful ball
Russian
Игривый котенок гонялся за разноцветным мячиком.
English
A playful kitten was chasing a multi-colored ball.

**Figure 2.3:** Backtranslation augments text using language translation approach.

### 2.3.2 Backtranslation

Backtranslation is a popular data augmentation technique in NLP that involves translating sentences from the target language to a foreign language and then back to the original language [130, 135]. This can be categorized as a paraphrasing data augmentation technique. For example, in Fig2.3, the original English text will be converted into Russian, and again that Russian text will be converted into English. This process introduces variations in the input data while preserving the semantic meaning. However, the quality of the augmented text depends on the machine translation task, where most of the translated data are not accurate [134].

## 2.4 Data Annotation

Data annotation is a critical process in ML and NLP that involves labeling or tagging data with specific information to train models effectively. The primary purpose of data annotation is to create a labeled dataset for supervised learning so that algorithms can use it to learn patterns, relationships, and associations between different elements in the data. In the context of NLP, data annotation can involve labeling entities in text (named entity recognition), identifying sentiment in sentences, or marking syntactic structures [136].

Traditional approaches to data annotation often involve manual labeling by human annotators who are experts in the domain. This can be a time-consuming and resource-intensive process, however, it often ensures high-quality annotations [137]. Guidelines and annotation manuals are provided to maintain consistency among annotators [88]. Novel approaches in data annotation leverage advances in technology, including crowdsourcing platforms, to distribute annotation tasks among a large number of annotators, making the process more scalable and cost-effective [138]. Additionally, active learning techniques are employed, where ML models are used to identify the most uncertain or challenging instances for human annotators to focus on, optimizing the annotation process [138].

In NLP, specifically, with the increasing complexity and diversity of tasks, advanced annotation techniques like distant supervision, where weakly labeled data is utilized along with a small set of strongly labeled data, have gained prominence [139]. Transfer learning through pre-trained language models is also a novel approach that leverages existing knowledge to enhance the efficiency of the annotation process [137, 140]. These approaches contribute to the creation of large, high-quality annotated datasets, essential for training and improving the performance of sophisticated NLP models.

# Chapter 3

# Augmenting Reddit Posts to Determine Wellness Dimensions Impacting Mental Health

# Chapter Abstract

Amid ongoing health crisis, there is a growing necessity to discern possible signs of Wellness Dimensions (WD)[1] manifested in self-narrated text. As the distribution of WD on social media data is intrinsically imbalanced, we experiment the generative NLP models for data augmentation to enable further improvement in the pre-screening task of classifying WD. To this end, we propose a simple yet effective data augmentation approach through prompt-based generative NLP models, and evaluate the ROUGE scores and syntactic/semantic similarity among *existing interpretations* and *augmented data*. Our approach with ChatGPT model surpasses all the other methods and achieves improvement over baselines such as Easy-Data Augmentation and Backtranslation. Introducing data augmentation to generate more training samples and balanced dataset, results in the improved F-score and the Matthew's Correlation Coefficient for upto 13.11% and 15.95%, respectively.

[1] The concept of Wellness Dimensions is often used in holistic approaches to health, recognizing that well-being encompasses multiple areas of life.

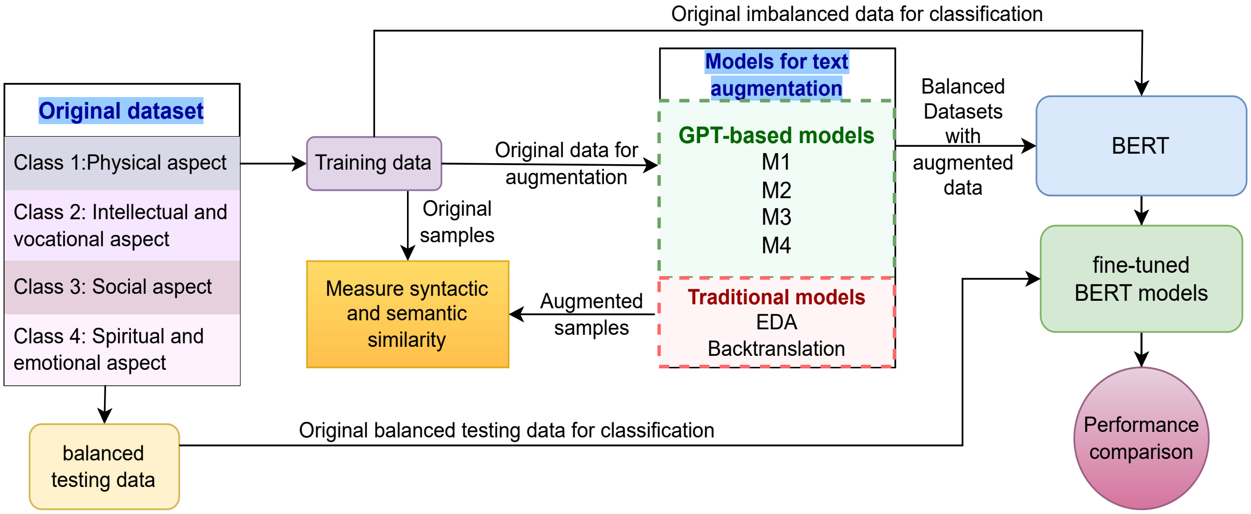

Original imbalanced data for classification
Models for text augmentation
Original dataset
Class 1:Physical aspect
Class 2: Intellectual and vocational aspect
Class 3: Social aspect
Class 4: Spiritual and emotional aspect
Training data
Original data for augmentation
GPT-based models
M1
M2
M3
M4
Balanced Datasets with augmented data
BERT
Original samples
Measure syntactic and semantic similarity
Augmented samples
Traditional models
EDA
Backtranslation
fine-tuned BERT models
Performance comparison
balanced testing data
Original balanced testing data for classification

**Figure 3.1:** Overview of the task. Generating balanced dataset through data augmentation to facilitate the development of classifiers screening Reddit posts through a lens of Wellness Dimensions.

## 3.1 Introduction

The social determinants of health (SDOH) refer to various factors present in the surroundings where individuals are born, reside, acquire knowledge, work, engage in leisure activities, practice religion, grow older, impacting a broad range of health-related outcomes, risks and quality-of-life indicators.[2][3] A rapid expansion of research in SDOH 2030 encourages the social NLP research community to design and develop computational intelligence models for enhancement of an individual's well-being [141]. In this work, we choose to pre-screen human-writings for biomedical therapy by investigating latent indicators of *wellness dimensions* in Reddit posts (see illustration in Figure 3.1).

Wellness dimensions (WD) refer to different aspects of an individual's overall well-being that contribute to their physical, spiritual, social, emotional, intellectual, occupational, environmental, and financial well-being. The disturbed WD, if remains unaddressed, have adverse impact on mental health of an individual. As social media becomes integral part of our daily lives [142],

[2] https://health.gov/healthypeople/priority-areas/social-determinants-health

[3] Social Determinants of Health (SDOH) are the social and economic factors that influence an individual's health outcomes.

studies in the past suggest that individuals tend to express their thoughts and emotions impacted by one or more wellness dimensions more easily on social media platforms as compared to during in-person sessions with clinical psychologists and mental healthcare [143, 144]. We construct, annotate and observe the original (natural) composition of WD dataset as an imbalanced dataset. In this work, we augment a multi-class dataset on WD to facilitate design and development of NLP models for classifying WD impacting mental health in Reddit posts during mental health screening. Pre-screening filters are helpful in biomedical therapy by facilitating early detection of WD impacting mental health, which if left untreated may cause severe mental disorders. Dunn highlights the holistic nature of wellness in 1961 as a *high-level wellness*, denoting a superior level of healthy living [145].

We reduce multiple WD to four key dimensions of well-being based on the frequency and recognition in human writings: Physical Aspect (PA), Intellectual and Vocational Aspect (IVA), Social Aspect (SA), Spiritual and Emotional Aspect (SEA) [146,147]. Our major contributions (as illustrated in Fig. 3.1) include (i) the applicability of generative NLP models for domain-specific data augmentation, (ii) examining the diversity among generated and original instances through semantic and syntactic similarity measure, (iii) test and validate the efficacy of data augmentation by investigating classifiers' performance.

## 3.2 Background

According to Weiss (1975), sociologists put forth a theory that outlines six social needs to prevent loneliness: attachment, social integration, nurturance, reassurance of worth, sense of reliable alliance, and guidance in stressful situations [148]. The Self-Determination Theory (SDT)[4] highlights the importance of balancing relatedness, competency, and autonomy for intrinsic motivation and genuine self-esteem, which contribute to overall well-being. Neglecting mental disturbance can escalate sub-clinical depression to clinical depression by activating interpersonal risks. This research seeks to examine the origins and outcomes of mental disturbance to mitigate these risks.

[4] https://en.wikipedia.org/wiki/Self-determination_theory

**Corpus Construction:** We present a new dataset with 3,092 instances and 72,813 words to identify wellness dimensions impacting mental disturbance: PA, IVA, SA, and SEA. A senior clinical psychologist, a rehabilitation councilor, and a social NLP researcher framed annotation schemes and perplexity guidelines for text annotation through pre-defined wellness dimensions. Our experts trained three postgraduate students to annotate the data based on predefined dimensions. The annotations were validated using Fleiss' Kappa inter-observer agreement, resulting in a kappa score of 74.39%. Final annotations were determined through majority voting and expert verification. The experts achieved a kappa score of 87.32% for the selection of explanatory text spans. Despite slight confusion between PA and SEA, there was a higher agreement for the selection of explanations. To facilitate future research and developments, we publicly release our dataset at Github.[5]

**Problem Formulation:** We collect and annotate Reddit data from subreddits `r/depression` and `r/suicidewatch` for the task of identifying WD and found imbalanced dataset in its natural composition, suggesting the need of data augmentation. To evaluate the effectiveness of generative NLP models for data augmentation, we frame the task of *augmenting Reddit posts* as a *text generation* problem. We compare and contrast the performance of model trained on data augmented with (i) GPT models [149], and (ii) conventional data augmentation approach for NLP such as EDA [133] and Backtranslation [135].

## 3.3 Materials and methods

We first generate the data using two-fold measures: (i) traditional data augmentation methods for NLP - EDA and Backtranslation, and (ii) prompt-based Generative Pre-trained Transformer models [150]. We further investigate the diversity of the generated samples in comparison to the original samples and fine-tune BERT language model to observe improvements in WD classification, if any. Hinged on the classification results and similarity measures, we select the best model for augmenting WD dataset.

[5] https://github.com/drmuskangarg/WellnessDimensions

| WD | $\alpha$ | Red | RC | AS | Tot. |
|---|---|---|---|---|---|
| PA | 740 | 6.0 | 695 | 399 | 1094 |
| IVA | 592 | 7.6 | 547 | 547 | 1094 |
| SA | 1139 | 4.0 | 1094 | 0 | 1094 |
| SEA | 621 | 7.2 | 576 | 518 | 1094 |

**Table 3.1:** The statistics of original composition ($\alpha$), the reduction percentage (Red), reduced composition (RC), the number of augmented samples (AS) and total number of samples (Tot.)

### 3.3.1 Methods: Data Augmentation

We use pre-trained generative models[6] [151] for this task: (i) ChatGPT models: *gpt-3.5-turbo and gpt-3.5-turbo-0301*, and (ii) other GPT-3 models: *text-curie-001 and text-davinci-003*. The original dataset consists of 3092 samples, with 740, 592, 1139 and 621 records from classes PA, IVA, SA, and SEA respectively. We first split the dataset such that we maximize the number of training samples required for each WD. After augmentation, the training set comprises a total of 4376 records, with an equal distribution of 1094 records per class.

**Training and Testing Split:** Consider the data containing $D$ documents representing a collection of Reddit posts $\{D = d_1, d_2, ..., d_n\}$ where $n = 3092$. For each document $d_i$, there exist a tuple representing $< E_i, C_i >$ where $E_i$ is text-span/ explanation and $C_i$ is the aspect class for $i^{th}$ instance. Thus, the original WD dataset consists of three columns for 3092 samples: $< D_i, E_i, C_i >$. The aspect class $C_i \in \alpha$ where $\alpha =$[PA, IVA, SA, SEA] and the composition of original dataset contains imbalanced distribution of aspect classes (see Table 3.1). The number of samples for every class $\alpha[j]$ where $1 \leq j \leq 4$ suggests the need of data augmentation to facilitate development of NLP models over balanced dataset. To this end, we propose the Algorithm 1- *Required augmentation count* to decide the number of samples that needs to generated for each WD. Given an input $\alpha$ as a list of the number of text samples for different WD (PA, IVA, SA, SEA) where PA, IVA, SA, SEA defines the count of instances for each class.

[6] https://platform.openai.com/docs/models

As such, our goal is to achieve a balanced dataset by obtaining $1094$ samples for each WD, resulting in $1094 * 4 = 4376$ data samples. We observe that all the samples for IVA class must be augmented while no augmentation is required for SA class.

**Algorithm 1:** Required Augmentation Count

```
Result: Return $AS$=[]
// AS: augmented sample
Input: $\alpha$ : [PA, IVA, SA, SEA]
Set: $\beta$= [], R, Red=[], RC=[]
Set: $min_{value} := min(\alpha)$ // Get the record count of minority class
for j in count($\alpha$) do
    $\beta[j] := max(\alpha) - \alpha[j]$
    /* For each class, get the record count difference from
       majority class                                                   */
end
// Estimate the size of the test set
$R = min_{value} - max(\beta[j])$
for j in count($\alpha$) do
    /* For each class, calculate the Reduction Percentage
        (percentage of reduction after separating the testing
       set)                                                             */
    $Red[j] = \frac{R}{\alpha[j]} * 100$
    /* For each class, calculate the Reduced Composition
        (number of training records before augmentation)                */
    $RC[j] = \alpha[j] - R$
end
/* Get the maximum number of records per class for
   augmentation                                                         */
$max_{RC} = max(RC)$
for j in count($\alpha$) do
    /* Augment each class up to the maximum record count                */
    $AS[j] = max_{RC} - RC[j]$
end
// return the augmented dataset
return $AS$
```

**Prompt Design and Parameter Setup:** As shown in Figure 3.2, we design following prompts to produce, a) text similar to the original text (topic and text), and b) an explanation of newly generated text (text and explanation).

```
Prompt Designs for GPT models

Considering the given topic, generate similar text to the given text.
Topic: ≪class label as a string≫
Text: ≪original text sentence≫

Similar text:
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
Consider the examples and generate a very short explanation of the given text.

text: ≪example1-text≫
explanation: ≪example1-explanation≫
...
text: ≪example5-text≫
explanation: ≪example5-explanation≫
text: ≪original text sentence≫
explanation: ≪original explanation≫

text: ≪augmented text sentence≫
explanation:
```

**Figure 3.2:** The prompt designs for generating Text and Explanation.

While designing prompts according to Open-AI prompt design instructions[7], we begin with explanation through instructions and examples or both. During the text creation, we only provide instructions to the model. As every text belongs to one of the four pre-defined WD, we provide class name as an input, for example, ”Physical Aspect”, hypothesizing its contribution towards contextual consciousness required for enhancing similar text generation. Furthermore, the explanation generation is developed as a few-shot learning approach [85], where we provide *five* text-explanation pairs as examples. The selective examples ensure the representation of all four classes and are made static for every call. We keep temperature as 0.7 to preserve the creativity/ randomness of generated text.

[7] https://platform.openai.com/docs/guides/completion/prompt-design

### 3.3.2 Method: Similarity Measures

First we calculate ROUGE scores $<$ROUGE-1, ROUGE-2 and ROUGE-L$>$ to examine similarity[8]. Next, for *semantic similarity,* we calculate the embedding for each sentence through eleven pre-trained language models (See Table 3.2), including nine SentenceTransformers and two OpenAI models [152, 153]. SentenceTransformers are a set of state-of-the-art language models implemented in Python for generating text embeddings. The two different GPT-3 models used for this task accessed the embeddings API endpoint. The resulting sentence embeddings[9] of each original and augmented data instance were then compared using *cosine similarity*. Lastly, for *syntactic similarity*, we first parsed given sentences into syntactic trees and then mapped them into vector representations using the ”en_core_web_md” English pipeline in the spaCy library[10]. Next, these vector representations are used to compute the similarity score between sentences. Furthermore, we compute the set overlap between the POS tag sequences of the original and augmented sentences to determine their similarity[11].

**Table 3.2:** Language models used to evaluate generate sentence embedding.

| **Base Model** | **Version** |
|---|---|
| BERT | all-MiniLM-L6-v2 |
| BERT | all-MiniLM-L12-v2 |
| MPNet | all-mpnet-base-v2 |
| MPNet | paraphrase-mpnet-base-v2 |
| Albert | paraphrase-albert-small-v2 |
| DistilBERT | quora-distilbert-base |
| DistilRoberta | all-distilroberta-v1 |
| DistilRoberta | paraphrase-distilroberta-base-v1 |
| Roberta | msmarco-roberta-base-v3 |
| GPT-3 | text-embedding-ada-002 |
| GPT-3 | text-similarity-davinci-001 |

[8] https://pypi.org/project/rouge/

[9] https://platform.openai.com/docs/guides/embeddings/what-are-embeddings

[10] https://spacy.io/models/en

[11] https://www.nltk.org/api/nltk.tag.pos_tag.html

### 3.3.3 Classification with BERT

As the final evaluation, we build BERT [62], a baseline classifier, with 6 augmented datasets and compare its performance with the BERT classifier built over original data. We used the training data in WD dataset for finetuning for 10 epochs with a batch size of 32 and a learning rate of 3e-5. To preserve the lengths of texts, we set the max_length to 256 during tokenization. We use the validation set (20% of the training set) and testing set (180 samples) to examine the efficiency and effectiveness of a classifier through F-score and Matthew's Correlation Coefficient (MCC), respectively.

## 3.4 Results and discussion

**Similarity Analysis:** We report three types of ROUGE scores: ROUGE-1, ROUGE-2 and ROUGE-L between the original and augmented text. The ChatGPT models show the lowest ROUGE scores and *Backtranslation* versions surpass all other augmentation methods (see Figure 3.3). We further examine semantic and syntactic similarities through average of all 13 models in Figure 3.4(top) and 3.4(bottom). We observe high diversity and low similarity with GPT based models where ChatGPT based models illustrate the least similarity. However, compared to the other GPT models, text-curie-001 shows a notably higher similarities through all the similarity models.

**Classification Performances:** We obtain the validation accuracy (Val-A), and testing results with precision (T-P), recall (T-R), F-score (T-F), accuracy (T-A) and MCC value (T-MCC) with experimental results for evaluation (see Table 3.3). Even though we keep the testing dataset to be a small chunk of 180 samples, we observe significant difference in the results in training on the *original imbalanced dataset* and *augmented dataset*. The *gpt-3.5-turbo* model over testing dataset outperforms all the baseline models, specifically the original dataset by 13.11% F1-score and 9.52% Accuracy followed by the second best model: *gpt-3.5-turbo-0301*. Moreover, compared to the best traditional augmentation method (Backtranslation), the top ChatGPT model shows 7.81% im-

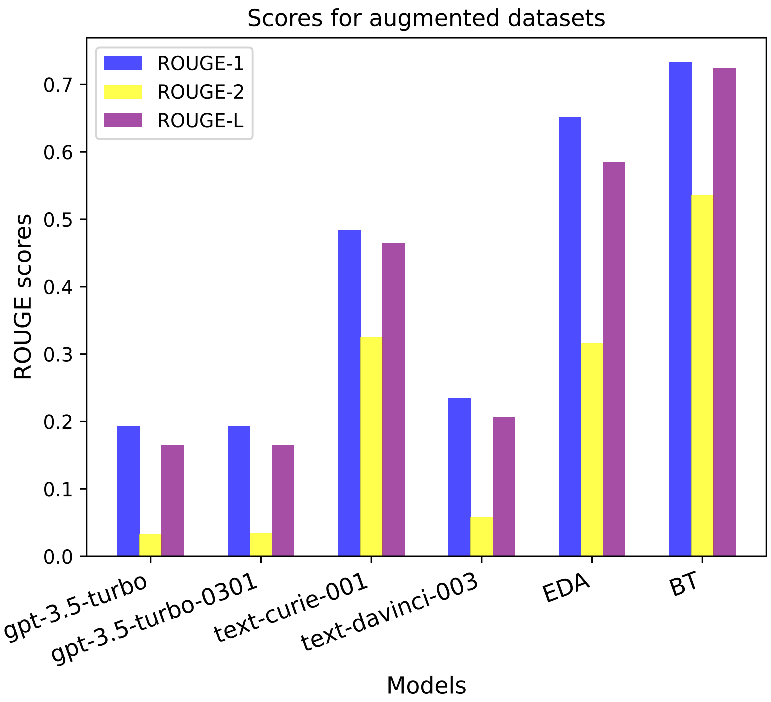


**Figure 3.3:** The ROUGE scores for six different augmentation mechanisms leveraging the augmented samples in comparison to the original text, averaged over all the texts generated.

**Table 3.3:** Improvement in classifiers. M1: gpt-3.5-turbo, M2: gpt-3.5-turbo-0301, M3: text-curie-001, M4: text-davinci-003, BT: Backtranslation

| Type | Val-A | T-P | T-R | T-F | T-A | T-MCC |
|---|---|---|---|---|---|---|
| Original | 0.427 | 0.65 | 0.63 | 0.61 | 0.63 | 0.514 |
| M1 | 0.504 | **0.70** | **0.69** | **0.69** | **0.69** | **0.596** |
| M2 | 0.499 | **0.69** | **0.68** | **0.67** | **0.68** | **0.581** |
| M3 | 0.498 | 0.63 | 0.63 | 0.62 | 0.63 | 0.519 |
| M4 | 0.502 | 0.66 | 0.67 | 0.66 | 0.67 | 0.559 |
| EDA | 0.498 | 0.63 | 0.63 | 0.62 | 0.63 | 0.518 |
| BT | 0.504 | 0.65 | 0.64 | 0.63 | 0.64 | 0.527 |

provement in testing accuracy. Notably, the datasets from text-curie-001 and EDA which gained higher similarity values have shown lowest performance on all classification measurements.

We further examine the MCC values to determine the effectiveness of the classifier [154]. MCC values vary between -1 and 1 such that values closer to 0 and 1 suggest increased randomness and perfect prediction towards decision making correspondingly. Compared to the original, we found 15.95% improvement in MCC score when model is trained on augmented training samples

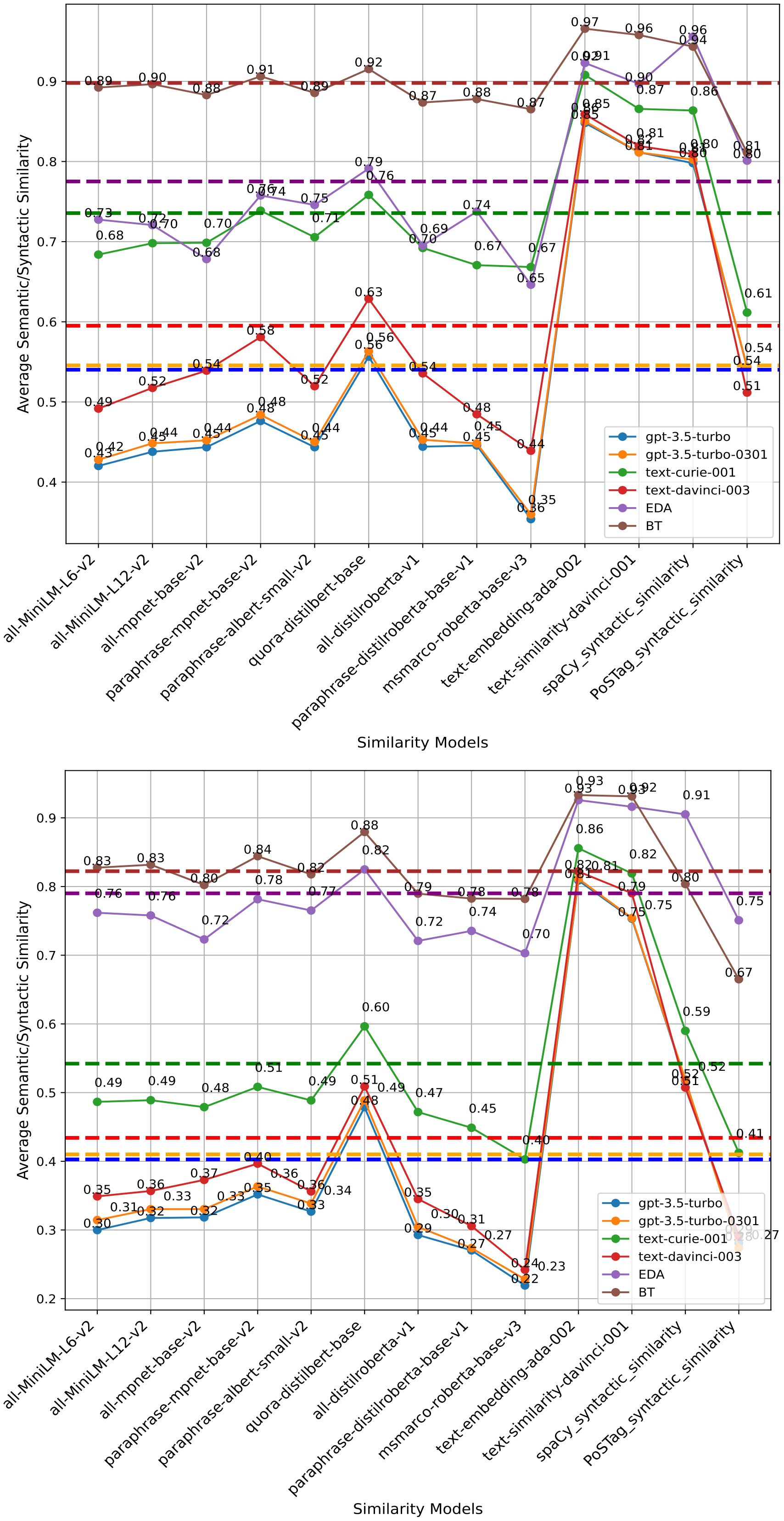


**Figure 3.4:** (top): Average Textual Similarity among Original and Augmented Text. (bottom): Average Textual Similarity among Original and Augmented Explanations.

**Table 3.4:** Class-vise classification performance. NS: Number of correctly classified samples, INS: Improvement in NS (in %), OD: Original dataset, AD: Dataset augmented by M1 method.

| Class | Type | T-P | T-R | T-F | NS | INS |
|---|---|---|---|---|---|---|
| PA | OD | 0.78 | 0.71 | 0.74 | 32 | 4.44 |
| | AD | 0.76 | 0.76 | 0.76 | 34 | |
| IVA | OD | 0.67 | 0.31 | 0.42 | 14 | 17.78 |
| | AD | 0.69 | 0.49 | 0.57 | 22 | |
| SA | OD | 0.61 | 0.76 | 0.67 | 34 | 2.22 |
| | AD | 0.69 | 0.78 | 0.73 | 35 | |
| SEA | OD | 0.53 | 0.73 | 0.62 | 33 | 2.22 |
| | AD | 0.65 | 0.76 | 0.70 | 34 | |

with *gpt-3.5-turbo* model. Overall, the augmented text with lowest ROUGE scores, syntactic and semantic similarities showed the highest classification performance on BERT.

Moreover, the following Table 3.4 compares the class-vise classification performance between the original and M1 (best performed dataset) datasets. We notice a significant improvement in all the measurements of all the classes after augmenting data. Additionally, compared to other classes, the IVA- class with the least number of original samples shows a significantly higher improvement in the number of correctly classified samples.

## 3.5 Conclusion and Future Scope

In this work, we augment the Reddit posts for a four-class classification problem of determining Wellness Dimensions impacting mental health. The GPT models are outperforming in terms of generating diverse text by preserving the context of the corresponding original text. In future, we plan to experiment with different parameter settings and prompts for generating datasets and develop improved classifiers to determine WD in a well balanced dataset. Furthermore, we will evaluate the classification performance of short explanation text we generated in this dataset.

# Ethics and Broader Impact

The data used in this study is obtained from Reddit, a platform designed for anonymous posting, and the user IDs have been anonymized. Furthermore, all sample posts displayed in this study have been obfuscated, paraphrased, and anonymized to protect user privacy and prevent any misuse. As annotation is subjective in nature, we acknowledge that there may be some biases present in our gold-labeled data and the distribution of labels in Wellness Dimensions dataset. We urge researchers to be mindful of the potential risks associated with WD dataset based on personal textual information. To prevent this, human intervention by a moderator is necessary. We acknowledge that we do not release user's metadata and the augmented samples further increase the privacy. The dataset and the source code required to replicate the baseline results can be accessed at Github.[12]

# Acknowledgement

We express our gratitude to Veena Krishnan, a senior clinical psychologist, and Ruchi Joshi, a rehabilitation counselor, for their unwavering support throughout the project. This project was partially supported by NIH R01 AG068007. This project is funded by NSERC Discovery Grant (RGPIN-2017-05377), held by Vijay Mago, Department of Computer Science, Lakehead University, Canada.

[12] https://github.com/Ravihari123/Data-Augmentation

# Chapter 4

# GPT-4 as a Twitter Data Annotator: Unraveling Its Performance on a Stance Classification Task

# Chapter Abstract

Data annotation in NLP is a costly and time-consuming task, traditionally handled by human experts who require extensive training to enhance the task-related background knowledge. Besides, labeling social media texts is particularly challenging due to their brevity, informality, creativity, and varying human perceptions regarding the sociocultural context of the world. With the emergence of GPT models and their proficiency in various NLP tasks, this study aims to establish a performance baseline for GPT-4 as a social media text annotator. To achieve this, we employ our own dataset of tweets, expertly labeled for stance detection with full inter-rater agreement among three annotators. We experiment with three techniques: Zero-shot, Few-shot, and Zero-shot with Chain-of-Thoughts to create prompts for the labeling task. We utilize four training sets constructed with different label sets, including human labels, to fine-tune transformer-based large language models and various combinations of traditional machine learning models with embeddings for stance classification. Finally, all fine-tuned models undergo evaluation using a common testing set with human-generated labels. We use the results from models trained on human labels as the benchmark to assess GPT-4's potential as an annotator across the three prompting techniques. Based on the experimental findings, GPT-4 achieves comparable results through the Few-shot and Zero-shot Chain-of-Thoughts prompting methods. However, none of these labeling techniques surpass the top three models fine-tuned on human labels. Moreover, we introduce the Zero-shot Chain-of-Thoughts as an effective strategy for aspect-based social media text labeling, which performs better than the standard Zero-shot and yields results similar to the high-performing yet expensive Few-shot approach.

## 4.1 Introduction

Among the LLMs, the GPT series has emerged as a pioneer, showcasing powerful skills on numerous tasks in NLP, such as content generation, completion, translations, summarizations, classifications, and many more[1]. However, the ability of GPT models to comprehend and generate human-like text has not only redefined the landscape of NLP applications but also highlights significant capabilities related to handling many human jobs, such as data analysts [155], data evaluators [156, 157], software developers [158, 159], and teaching assistants [160]. Besides, GPT has proven applications in diverse domains, including finance [161], health [162, 163], social science [164] and law [165]. Among the potentialities for replacing diverse human tasks, GPT has demonstrated itself as a remarkably effective tool for data annotation across various domains [164, 166–171]. Its ability to understand context, generate coherent content, and follow specific guidelines has made it a versatile data annotator, in labeling a wide range of content from generic to domain-specific text.

Data annotation is the primary step of many NLP tasks. Nevertheless, the process of labeling by skilled human experts proves to be expensive and time-consuming due to the costs associated with labor, tools, and the time needed for training and manual annotation [80, 167, 170]. Furthermore, maintaining a high standard training process through setting perplexity benchmarks and enough foundation of background knowledge is crucial for high-quality labeling outcomes [172]. Due to these requirements, the consideration of substituting human annotators with AI tools has become justifiable.

From another perspective, given the emergence of social media as a significant data source for various NLP studies, addressing the challenges posed by the inherent traits of brevity, informality, creativity, and poor grammar in tweets is essential during annotation [172–174]. Additionally, considering that these texts are embedded within the cultural and social context of human ideas, values, and perceptions of the world, comprehending them necessitates a thorough understanding of context and the ability to empathize by adopting different perspectives [171]. Consequently, the examination and annotation of social media texts, especially those pertaining to social debates,

[1] https://platform.openai.com/docs/quickstart

will demand specialized annotation capabilities. This prompts the investigation into the potential of GPT-based models to replace human annotation tasks.

While existing studies have demonstrated GPT's effectiveness in data annotation, limited attention has been paid to its application in social media stance labeling. The challenges encountered by humans in social media text labeling and stance identification present an opportunity to investigate the potentiality of AI tools in this context. Hence, this research aims to evaluate the capacity of the most recent and powerful GPT-4 model [60] in labeling social media text on stance detection. By comparing GPT-4's performance against human annotators, and potentially incorporating innovative prompting techniques, this study seeks to contribute to the field of NLP and social text analysis as follows.

1. Create and release a labeled Twitter corpus on stance detection.

2. Benchmark the performance of GPT-4 as a data annotator for labeling social media text on stance detection tasks compared to human experts.

3. Investigate the applicability of integrating the Chain-of-Thoughts concept into the prompt design for labeling the stance of social media texts.

4. Conduct a performance comparison among three distinct prompt-designing strategies in the context of annotating the stance of social media texts.

## 4.2 Background Motivation

In the literature, many studies have explored the role of GPT as a textual data annotator. A recent investigation assessed the performance of GPT-4 in annotating domain-specific multi-label legal text, a task usually requiring individuals well-versed in legal matters for accurate annotation [168]. Utilizing a dataset comprising 256 records with Krippendorff's inter-annotator agreement of 0.79, this study demonstrated GPT-4's capacity to achieve results comparable to human annotators when provided with almost the same copy of instructions. Further, they explained the cost-effectiveness of this approach during batch predictions without a major reduction in performance compared to manual labeling. Nevertheless, slight adjustments to the prompts led to decreased model robustness, significantly impacting outcomes. Moreover, the authors engaged in a failed attempt to improve the performance with the Chain-of-Thoughts (CoT) prompting technique. Another approach has developed to label the political affiliation of tweets collected from the USA politicians [171]. The researcher has used 500 records and executed the GPT-4 model 5 times each with different temperature values; 0.2 and 1.0 to gain both the creativeness and robustness during label prediction. This work achieved better results for accuracy, reliability and bias of GPT-4 compared to human coders for a Zero-shot learning classification task.

The authors of another study have explored three methods to employ GPT-3 for data annotation [167]. The initial approach employed a Few-shot prompt to generate labels for unlabeled data, while the second method designed a prompt to guide the GPT-3 model in self-generating label data. In the third approach, a dictionary was used as an external source of knowledge to assist GPT-3 in creating domain-specific labeled data. They conducted experiments using text-davinci-003 and ChatGPT as GPT-3 models, along with Bert-base as the classifier for evaluation. Findings indicated that the first approach yielded subpar results compared to humans in both accuracy and cost, while the third approach achieved higher performance for GPT-3, surpassing both humans and ChatGPT. Furthermore, the authors highlighted the AI models' capability to generate training data from scratch without relying on unlabeled data. Another study has investigated the application of GPT-3.5 and GPT-4 in automated psychological text analysis, assessing their performance as data annotators [164]. This evaluated GPT's capability to label psychological aspects like sentiment, emotions, and offensiveness across 15 datasets encompassing multiple languages. The results revealed GPT's remarkable performance compared to dictionary-based analysis and comparable performance to fine-tuned machine learning (ML) models, suggesting its potential as a versatile tool for automated text labeling with simple prompts and less programming experience.

Besides the inherited complexities of annotating tweets, some labeling tasks, such as sentiment labeling are relatively straightforward as they focus on identifying sentiments that are often expressed explicitly in the text. Whereas stance classification is a more challenging task for humans as it involves determining the author's position or perspective toward a particular topic or issue as in favor of, against to, or neutral, which is not always explicitly stated in the text [174–176]. In the existing literature, there are limited studies that have engaged in stance labeling by humans and common target topics of their studies are Atheism, Climate change, Feminism, Elections, and the Legalization of abortion [78,172,175].

The earliest dataset of English tweets annotated for stance detection became available to the research community quite recently, in 2016 [172]. This dataset consisted of 4870 tweets, and the annotation process was conducted through crowdsourcing using the CrowdFlower platform. They aimed for high-quality labels by offering clear and simple labeling instructions, assigning each tweet to 8 annotators, and discarding poorly annotated records based on an analysis of annotator responses. Moreover, they shared the finalized dataset, comprising records where over 60% of the annotators had agreed on the majority label. Many recent studies have utilized this dataset in their stance classification tasks [78,175,176]. Another study has annotated a corpus of French tweets for detecting stances for a fake news recognition problem [173]. They have implemented a novel annotation approach by presenting the tweets to the annotators as a bundle, comprising a root tweet and all thread tweets as children. They argue the advantage of this approach as annotators gain context from whole threads, improving topic consistency and reducing topic-switching during annotation. However, they stated a few limitations of this approach, as cases like unrelated responses or incomprehensible tweets were not covered by their stance categories, and certain classes lacked distinctness, potentially creating uncertainty for annotators.

While those studies have only provided the text of tweets for the annotators, a different study explored utilizing associated metadata to enrich the labeling process [174]. In the context of political stance detection on Twitter, this study has experimented with a novel labeling approach by providing 6 pieces of additional information related to the authors of tweets other than the tweets' texts. Initially, these details were given to human raters (via Amazon Mechanical Turk) during annotation and revealed that providing insufficient context related to tweets can lead to ambiguous and noisy annotations, while an excessively strong context might overpower other signals. Consequently, the researchers designed a classifier that employed both individual human annotations and author-related information to determine the final tweet label. This classifier outperformed the common practice of using majority voting to decide the label.

The latest development in LLMs involves utilizing prompts to train these models with very little or no prior training data. These techniques are known as Few-shot and Zero-shot learning, and the GPT series of models have proven to excel in these learning scenarios [85]. However, research has demonstrated that GPT models are significantly influenced by their prompts, often producing diverse outcomes [168]. The concept of ”Chain-of-Thoughts” was introduced through a Few-shot method that involves presenting a series of intermediate steps to explain a given example answer [177]. They conducted experiments using various versions of prompt-based large language models, including GPT-3, LaMDA, PaLM, UL2 20B, and Codex. Remarkably, the PaLM 540B model achieved outstanding accuracy on the GSM8K benchmark for math word problems with only eight CoT exemplars and this performance was even better than a fine-tuned GPT-3 model. Subsequently, another study has incorporated this mechanism in Zero-shot prompting [178]. In contrast to the original approach, they omitted to provide examples and instead utilized a two-prompt method, adding the instruction ”Let’s think step by step” before each answer in the first prompt. Comparing this Zero-shot approach to the original mechanism, they observed improvements in various reasoning tasks, including arithmetic, symbolic, and logical reasoning. They highlight the advantage of exploring Zero-shot knowledge prior to employing manually crafted Few-shot examples.

## 4.3 Methodology

Initially, we constructed a labeled corpus of Twitter posts related to the stance classification problem towards abortion legalization. Subsequently, we employed 3 distinct prompting methods to reassign labels to the training tweets using GPT-4. Utilizing these variedly generated labels, along with human annotations, we constructed 4 training datasets containing the same tweets for multi-class classification fine-tuning. Next, the fine-tuned models underwent testing on a shared testing set equipped with human-annotated labels. Finally, we compared the outcomes from the 4 sets of

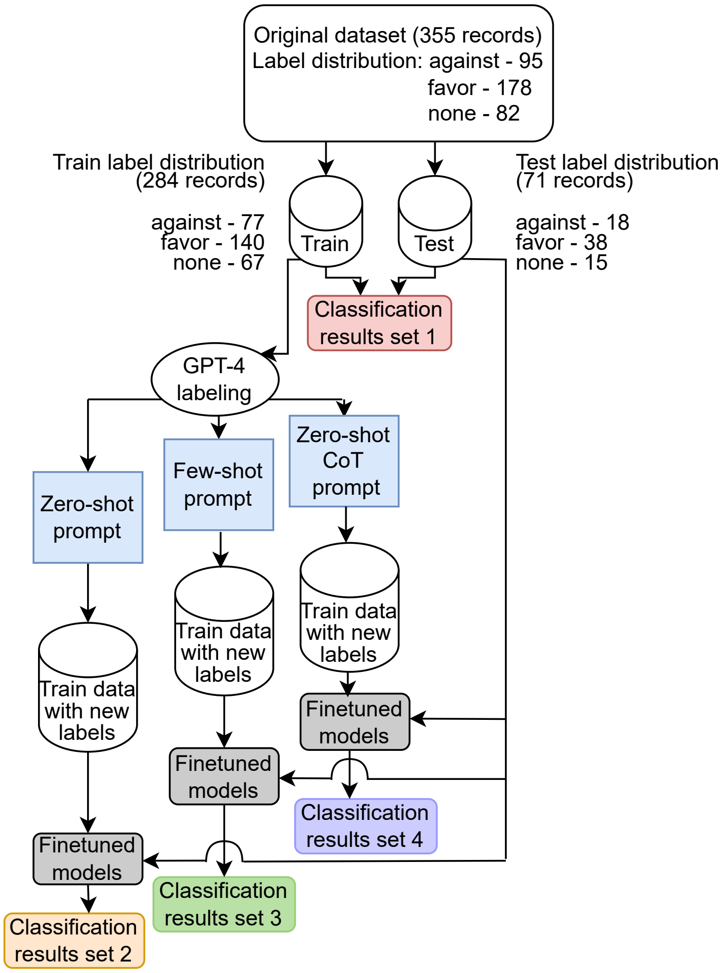


**Figure 4.1:** Overall methodology of the study.

test results to generate comprehensive findings. The complete research methodology is depicted in Fig. 4.1.

### 4.3.1 Dataset Collection

Motivated by the limited datasets for stance detection, we constructed a dataset by downloading texts related to the topic of abortion legalization from Twitter through Twitter academic API[2]. Focusing on the recent Supreme Court decision to ban abortion in the USA[3], we extracted tweets originating from the USA at three distinct time stamps (TS): i) TS1 - before the court decision was

[2] https://developer.twitter.com/en/use-cases/do-research/academic-research

[3] https://www.plannedparenthoodaction.org/issues/abortion/roe-v-wade

leaked (106 days from 16th January 2022 to 1st May 2022), ii) TS2 - following the leak (53 days from 2nd May 2022 to 23 June 2022), and iii) TS3 - after the court decision (53 days from 24th June 2022 to 15th August 2022), by yielding 250 records from each time stamp. We determined these dates by calculating the number of days between May 2nd (the date of the leak) and June 24th (the date of the court decision). For TS1, we extended the period to twice the duration, as the volume of tweets related to the topic of abortion legalization can be relatively lower. Our research adhered to ethical guidelines by solely utilizing publicly available tweets without any interest in or disclosure of author identities, thereby eliminating the need for any ethical considerations related to human subjects.

### 4.3.2 Human Data Annotation

Under the guidance of a senior academician in Social Science, three postgraduate students underwent specialized training using annotation and perplexity guidelines. Through a series of trial sessions by annotating a few samples, they familiarized themselves with the requirements for achieving a shared understanding. Subsequently, each coder annotated all 750 data points in the corpus for the multi-class stance classification task, regarding the author's stance on the legalization of abortion as a favor, against, or none. Additionally, the label ”uncertain” was provided as an option to indicate instances where annotators are unsure about the suitable label. In our annotation task, we only provided the texts of tweets, omitting their associated metadata. To ensure the reliability of the annotations, we evaluated the results using both Fliess' Kappa[4] and Krippendorf's alpha[5] inter-observer agreements [179]. After removing records with at least one uncertain label among annotators, the calculated kappa and alpha were found to be 64.54% and 61.26% respectively. Finally, we employed the majority voting mechanism to finalize the label for each record. We are releasing this dataset of 533 tweets to the public for research purposes[6].

[4] https://www.statsmodels.org/dev/generated/statsmodels.stats.inter_rater.fleiss_kappa.html

[5] https://github.com/surge-ai/krippendorffs-alpha/blob/main/kalpha.py

[6] https://github.com/Ravihari123/Twitter-Stance-Labeling/tree/main

### 4.3.3 GPT-4 Label Generation

As one of the main objectives of our study is to compare GPT-4's capabilities as an annotator with respect to humans, we needed to utilize reliable baseline labels. As the original dataset shows only substantial agreement among 3 annotators [180], we opted to work with a subset of our corpus, comprising 355 records that achieved 100% inter-reliability agreement among all raters.

We explored three different prompting strategies: 1) Zero-shot, 2) Few-shot, and 3) Zero-shot with CoT to generate labels for the tweets in our dataset using GPT-4. We set the temperature[7] as 0.5 which is a lower temperature value as it makes the model more confident in its predictions and leads to more deterministic and focused outputs. However, we did not set the temperature to 0.0, as we needed the model to have some randomness and creativity in predicting our labels [171]. Even though this can help in generating more conservative and precise responses, this will also lead to different answers during different runs. Due to this nature, each prompt type was run 3 times to generate labels for each tweet in the training set and then majority voting was used to finalize the final labels.

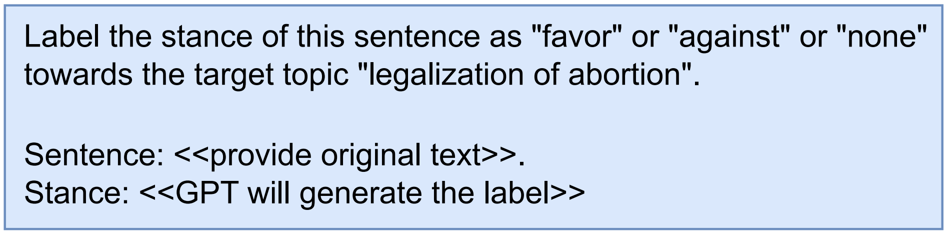
#### Zero-shot

The first approach is to design a prompt with only instructions (no examples) about the task and provide the tweets without the human-annotated labels in the training set to GPT-4 API call [181]. Within the prompt, we requested the model to produce an appropriate label for the provided text. The prompt design employed for generating labels through the Zero-shot mechanism is illustrated in Fig. 4.2.

#### Few-shot

The second method uses a Few-shot learning approach that teaches the GPT-4 model to perform the labeling task utilizing a combination of user instructions and a limited number of examples [181]. To introduce all three classes equally, we provided two fresh examples of tweets and their corresponding human-annotated labels for each class which are mutually exclusive from the training

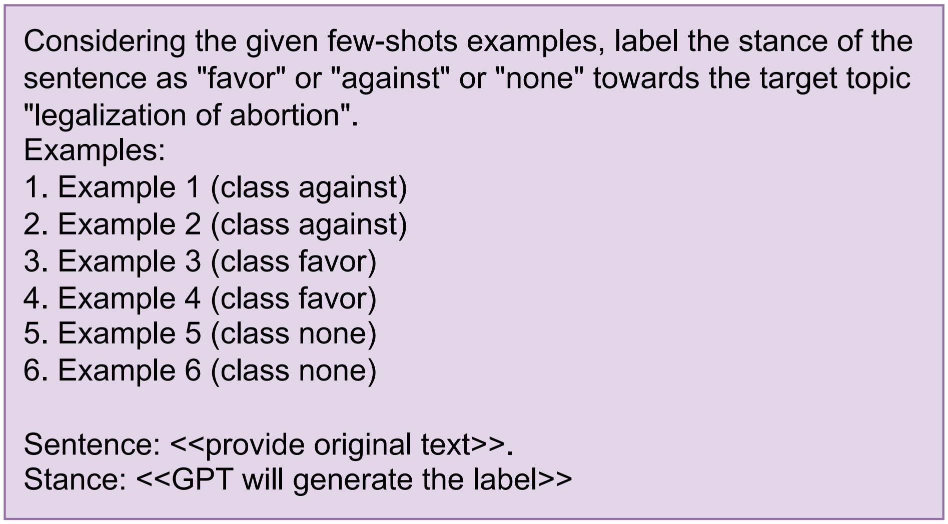
[7] https://platform.openai.com/docs/models/overview



Label the stance of this sentence as "favor" or "against" or "none" towards the target topic "legalization of abortion".

Sentence: <<provide original text>>.
Stance: <<GPT will generate the label>>

**Figure 4.2:** Zero-shot prompt for generating labels.



Considering the given few-shots examples, label the stance of the sentence as "favor" or "against" or "none" towards the target topic "legalization of abortion".
Examples:
1. Example 1 (class against)
2. Example 2 (class against)
3. Example 3 (class favor)
4. Example 4 (class favor)
5. Example 5 (class none)
6. Example 6 (class none)

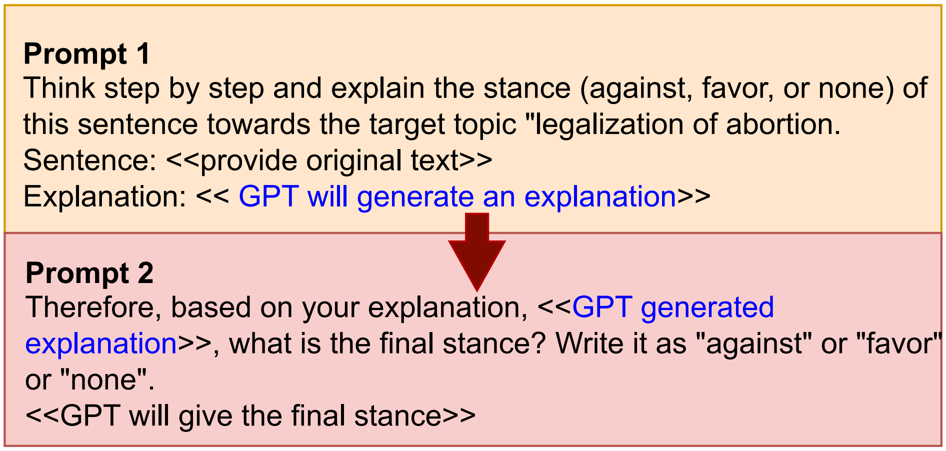
Sentence: <<provide original text>>.
Stance: <<GPT will generate the label>>

**Figure 4.3:** Few-shot prompt for generating labels.

and testing sets (See Fig. 4.3). The Few-shot approach tends to be more expensive compared to the Zero-shot method due to the larger number of tokens in each prompt and the requirement of few samples for the prompt will reduce data from the original dataset.

### Zero-shot Chain-of-Thought

This is an extension of Zero-shot prompting where we only provide instructions to the GPT-4 without any examples. The difference between this and the Zero-shot mechanism is that Zero-shot uses only a single prompt and the model will generate the final output at the end. However, as shown in Fig. 4.4, for the concept of Zero-shot CoT, we implemented two prompts, 1) to get a step-by-step explanation of how it decides the author's stance toward the target topic, and 2) to generate the final stance based on its own explanation. Similar to the original study [178], we instructed



**Prompt 1**
Think step by step and explain the stance (against, favor, or none) of this sentence towards the target topic "legalization of abortion.
Sentence: <<provide original text>>
Explanation: << GPT will generate an explanation>>

**Prompt 2**
Therefore, based on your explanation, <<GPT generated explanation>>, what is the final stance? Write it as "against" or "favor" or "none".
<<GPT will give the final stance>>

**Figure 4.4:** Zero-shot Chain-of-Thoughts prompt for label generation.

the model to think step by step and explain the answer before determining the final stance of the text. Through this two-prompt mechanism, we provide an opportunity for the model to reassess its answer. The advantages of this concept will be further discussed with examples in section 4.5.

### 4.3.4 Stance Classification

Stance detection is a multi-class classification problem, often with three stance labels. Our initial dataset with tweets and corresponding human labels was partitioned into an 80:20 ratio as the training and testing sets. Additionally, as mentioned earlier, we generated 3 more training sets featuring the same tweets but with new labels obtained through 3 distinct prompting techniques utilizing GPT-4. Subsequently, we fine-tuned eight transformer-based LLMs, namely Bert [62], Albert [68], Deberta [72], BerTweet [182], MPNet [66], and three Roberta-based models pre-trained on i) a general Twitter dataset (TRob) [79], ii) a Twitter sentiment dataset (TRobSen) [78], and iii) a Twitter stance dataset (TRobStan) [78]. These models were separately fine-tuned using our four training datasets. The list of model versions employed in the study, along with the datasets they were pre-trained on is provided in Table4.1.

In addition, 18 multiple combinations of classifiers composed of 6 traditional ML models and 3 embedding techniques, namely OpenAI ADA embedding (ADA), Sentence Transformers embedding (SenTr), and Glove embeddings were individually fine-tuned on our 4 training sets. The

**Table 4.1:** Large language models, their pre-trained versions, and pre-trained datasets.

| Model | Version | Pre-trained dataset |
|---|---|---|
| Bert | bert-base-uncased | BooksCorpus (800M words) and English Wikipedia (2,500M words) |
| Albert | albert-base-v2 | Same dataset of Bert |
| Deberta | microsoft/deberta-base-mnli | English Wikipedia (12GB), BookCorpus (6GB),<br>OpenWebText (public Reddit content of 38GB), and<br>STORIES (a subset of CommonCrawl of 31GB).<br>The size of the total data set after deduplication is about 78G. |
| BerTweet | vinai/bertweet-base | 850M English Tweets containing 845M Tweets streamed from 01/2012 to 08/2019 and 5M Tweets related to the COVID-19 pandemic. |
| MPNet | microsoft/mpnet-base | 160GB data from Wikipedia,<br>BooksCorpus, OpenWebText, CC-News and Stories. |
| Roberta | cardiffnlp/twitter-roberta-base-2022-154m | 154M tweets of general conversations between 2018-01 and 2022-12. |
| Roberta | cardiffnlp/twitter-roberta-base-sentiment-latest | 60M tweets were obtained by extracting a large corpus of English tweets (using the automatic labeling provided by Twitter). |
| Roberta | cardiffnlp/twitter-roberta-base-stance-abortion | |

embedding techniques were used to convert the tweets of the training set to their numerical vectors before feeding into the models [183]. Finally, all 104 types of fine-tuned models (32 LLMs and 72 traditional classifiers+embeddings) were tested individually on the common testing set to compare the classification performance of models trained on 4 different label sets.

### 4.3.5 Selection of Performance Metrics

We reported the testing performance in terms of precision, recall, f1-score, MCC[8] and area under the receiver operating characteristic curve (ROC_AUC). Accuracy was not reported due to its inability to account for class distributions, which makes it unsuitable for evaluating an imbalanced dataset [184, 185].

We used the macro averaging over micro and weighted for calculating precision, recall, f1-score and ROC_AUC as it calculates these metrics for each class independently and then takes the average across all classes. This approach gives equal consideration to all classes, irrespective of their frequency in the dataset. Hence, there is no difference between majority and minority classes, making the evaluations fair for an imbalanced dataset [185]. It is particularly useful in our study as we lack prior knowledge of the real-world class distribution and need to prevent evaluation bias towards dominant classes in different training datasets.

[8] https://scikit-learn.org/stable/modules/generated/sklearn.metrics.matthews_corrcoef.html

Equations (4.1) and (4.2) show the calculation of precision and recall, where True Positive ($TP$) is the correctly classified samples for the class $k$, whereas False Positive ($FP$) and False Negative ($FN$) are the incorrectly classified samples on the predicted and actual classifications of the class $k$ [185]. Equations (4.3), (4.4), and (4.5) represent the macro average precision, recall, and f1-score respectively, where $N$ is the total number of classes in the dataset [185]. The harmonic mean of macro precision and macro recall represents the multi-class macro F1-score.

$$Precision_k = \frac{TP_k}{TP_k + FP_k} \tag{4.1}$$

$$Recall_k = \frac{TP_k}{TP_k + FN_k} \tag{4.2}$$

$$MacroAveragePrecision(MP) = \frac{\sum_{k=1}^{N} Precision_k}{N} \tag{4.3}$$

$$MacroAverageRecall(MR) = \frac{\sum_{k=1}^{N} Recall_k}{N} \tag{4.4}$$

$$MacroF1 - Score = 2 * \frac{MP * MR}{MP^{\ 1} + MR^{\ 1}} \tag{4.5}$$

MCC is a metric ranging between -1 and 1, where a value close to 1 indicates excellent prediction, signifying a robust positive correlation between predicted and actual labels. Conversely, an MCC of 0 signifies no correlation, indicating that the classifier assigns samples to classes randomly, unrelated to their true values. Furthermore, MCC produces negative values, representing an inverse relationship between the predicted and actual classes [184, 185]. For multi-class classification, the MCC can be expressed using (4.6), based on the number of classes $N$, and confusion matrix $C$ with actual results on rows ($i$) and predicted results on columns($j$) [185].

$$MCC = \frac{c * s - \sum_{k}^{N} P_k * t_k}{\sqrt{(s^2 - \sum_{k}^{N} P_k^2)(s^2 - \sum_{k}^{N} t_k^2)}} \tag{4.6}$$

Where,

- $c = \sum_{k}^{N} C_{kk}$ the total number of elements correctly predicted

- $s = \sum_i^N \sum_j^N C_{ij}$ the total number of elements

- $P_k = \sum_i^N C_{ki}$ the number of times that class k was predicted (column total)

- $t_k = \sum_i^N C_{ik}$ the number of times that class k truly occurred (row total)

ROC_AUC is one of the best metrics to measure the performance of imbalanced datasets and it is regarded as a reliable metric, even when dealing with heavily skewed class distributions [186, 187]. For calculating ROC_AUC[9] in multi-class classification, the $TP$ rate or $FP$ rate is established only after transforming the output into binary form. For this we used the One-vs-Rest (OvR) method to compare each class to all others, treating the others as a single class.

### 4.3.6 Hyperparameter tuning

The LLMs underwent fine-tuning using identical hyperparameter configurations: a learning rate of 3e-5, batch size of 16, maximum epochs set at 10 with early stopping based on validation loss, and a patience of 2. Conversely, a grid search[10] was conducted to determine the optimal hyperparameter combinations for traditional ML models. However, for boosting algorithms, we utilized the default setup due to the expected computational complexity associated with hyperparameter evaluation. The traditional models and their corresponding hyperparameter settings are detailed in Table 4.2. Additionally, a 5-fold cross-validation[11] strategy was employed during model training to mitigate potential overfitting and yield more precise outcomes. Where possible, we employed the ”balanced” class weight option to ensure equal significance across all classes to handle class imbalance. All experiments were conducted using a constant random seed value.

### 4.3.7 Wilcoxon signed-rank test

The Wilcoxon signed-rank test is a fundamental non-parametric statistical test used to compare the central tendencies of paired data or matched samples [188]. This test assesses whether there

[9] https://scikit-learn.org/stable/modules/generated/sklearn.metrics.roc_auc_score.html

[10] https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html

[11] https://scikit-learn.org/stable/modules/cross_validation.html

**Table 4.2:** Hyperparameter settings utilized for traditional machine learning models during hyperparameter tuning.

| **ML model** | **Hyperparameter settings** |
|---|---|
| Logistic regression (LR) | ‘class_weight’: [None, ”balanced”]<br>‘penalty’: [None, ’l2’]<br>‘solver’: [’lbfgs’, ’newton-cg’] |
| Random Forest (RF) | ’n_estimators’: [50, 100, 200]<br>’max_depth’: [None, 5, 10]<br>’class_weight’: [”balanced”, ”balanced_subsample”, None] |
| Support Vector Classifier(SVC) | ’C’: [1.0, 2.0]<br>’class_weight’: [’balanced’, None] |
| Multi-Layer Perceptron (MLP) | ’activation’: [’logistic’, ’relu’]<br>’solver’: [’sgd’, ’adam’]<br>’hidden_layer_sizes’: [(100,), (200,), (50,)] |
| Gradient Boosting Tree (GB) | Default settings |
| Extreme Gradient Boosting (XGBoost) | Default settings |

is a statistically significant difference between two related groups, often before-and-after measurements or two treatments applied to the same subjects. It accomplishes this by analyzing the distribution of the signed differences between the pairs, effectively testing whether the median of these differences is zero [189, 190]. For our study, we used the Wilcoxon signed-rank test[12] to assess and summarize the similarity between performance metrics of various combinations of prompting outcomes.

We utilized the conventional value of 0.05 as the threshold for accepting or rejecting the null hypothesis, which assumes there is no significant difference between the corresponding performance metrics (either, precision, recall, f1-score, or ROC_AUC) of any two labeling sets. Here, in addition to the null hypothesis, we used an alternative hypothesis called ‘greater’ which suggests that the median of the paired differences is greater than zero. This test produces two main outputs, 1) test-statistics - the sum of ranks of positive differences, which measures the extent to which the positive differences between paired observations are greater than the negative differences, and 2) P-value - which determines whether this difference holds statistical significance. Consequently, higher test-statistics (larger positive difference between the two groups) indicate that the first group

[12] https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.wilcoxon.html

tends to have higher values than the second group, and the P-values below the selected significance level of 0.05 present there are statistically significant evidence to prove this difference. Equation (4.7) and (4.8) represents the calculation of the test-statistic and P-value of the Wilcoxon signed-rank test with the 'greater' alternative hypothesis [191].

- The test-statistic ($W+$):

$$W+ = \sum_{i=1}^{n} sign(d_i).R_i^+, \tag{4.7}$$

where, $n$ is the sample size, $di$ represents the paired differences, $sign(di)$ is the sign of the difference (+1 if $di$ is positive, -1 if $di$ is negative), and $Ri+$ is the rank of the positive differences among all the positive differences.

- The P-value ($P\_val$):

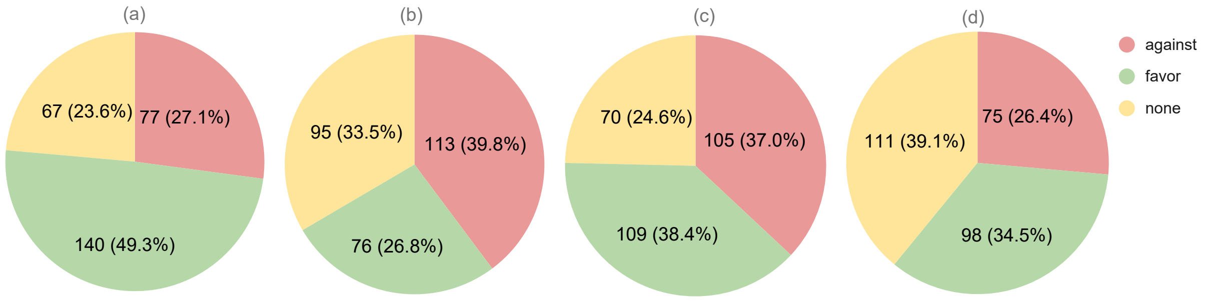
$$P{-}val = P(W+ \geq W_{observed}) \tag{4.8}$$

Where, $W+$ is the test-statistic calculated from our data, $W_{observed}$ is the test-statistic from the Wilcoxon signed-rank table[13] (based on the chosen significance level of 0.05 and sample size of 26), and $P$ is the probability of observing a $W+$ value greater than or equal to $W_{observed}$ under the null hypothesis.

## 4.4 Experimental Results and Initial Discussion

First, we analyze the outcomes of the relabeling process by examining the distribution of class labels in both the original and new label sets. Following this, we present the classification results of various ML models which were fine-tuned using the four distinct training sets.

[13] https://users.stat.ufl.edu/~winner/tables/wilcox_signrank.pdf



**(a)** Human labels **(b)** Zero-shot labels **(c)** Few-shot labels **(d)** Zero-shot CoT labels

**Figure 4.5:** The distribution of class labels in the four different label sets.

### 4.4.1 Results of Label Generation

Fig. 4.5 illustrates the distribution of class labels within the four training sets, created using different labeling techniques. Notably, datasets labeled by humans and the Few-shot approach exhibit a similarity, showcasing almost equal ratios in their 'none' class and gaining the 'favor' as the majority class. However, a significant change has occurred due to the 'against' class incrementing to 37% in the Few-shot labeled dataset, resulting in an almost 1:1 ratio with the 'favor' class. This contrast stands against the nearly 2:1 'favor: against' ratio seen in the human-labeled dataset. On the other hand, compared to human labels, the Zero-shot and Zero-shot CoT datasets have undergone a shift, with their majority classes changing to 'against' and 'none', respectively. Furthermore, the 'favor' and 'against' classes in the Zero-shot and Zero-shot CoT datasets have become the minority respectively, departing from the 'none' which served as the minority class in the human-labeled datasets. Nevertheless, the sizes of the 'against' class in both the Zero-shot CoT and human-labeled datasets are nearly similar.

Fig. 4.6 displays the percentage of changes observed with new label sets compared to the human labels. This demonstrates that the highest number of changes in the whole dataset appeared as 25.35% during the Zero-shot approach, whereas a minimum of 13.73% is recorded at the Few-shot. Analyzing class-wise percentages[14], the 'favor' class experienced the highest variations,

[14] The percentage of changes in a given class $k$ is calculated using (the number of changes in new labels compared to the human labels in class $k$ / total number of records belonging to class $k$ *100). Example: If there are 77 records of against class in the human-annotated dataset and 6 of the labels have changed to a different label during Zero-shot labeling, then the percentage of change in against class during Zero-shot is (6/77*100 = 7.79)

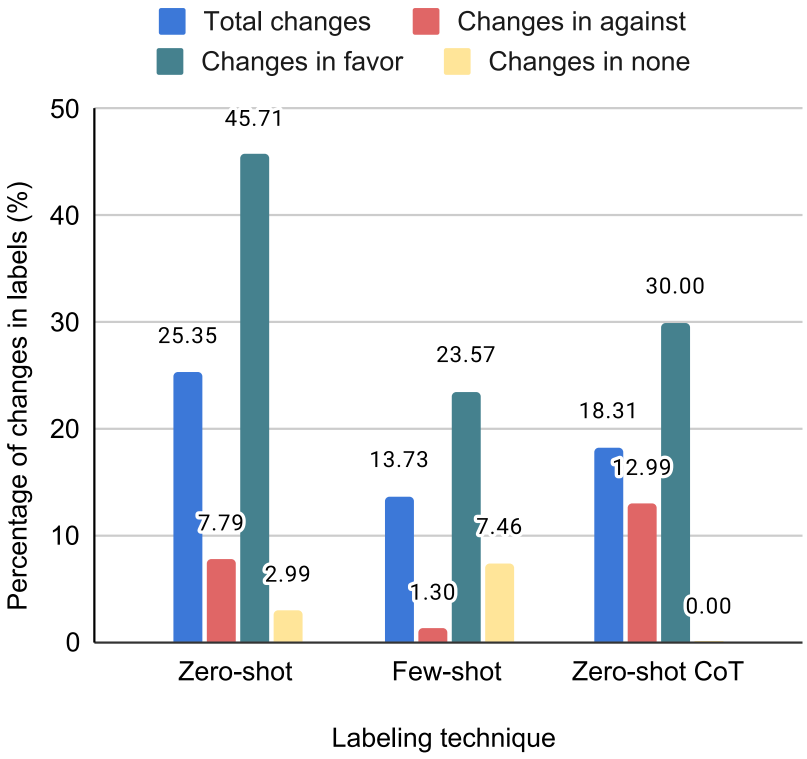


**Figure 4.6:** The percentages of changes in the three types of new label sets; Zero-shot, Few-shot, and Zero-shot CoT compared to human labels.

reaching 45.71%, 23.57%, and 30.0% in the Zero-shot, Few-shot, and Zero-shot CoT methods, respectively. Moreover, the minimum change percentage of the 'against' class is recorded as 1.30% in the Few-shot technique, whereas a minimum of 0.0% in the 'none' class is reported in the Zero-shot CoT approach.

By considering both the label distribution and the percentage of changes, we observe that, in comparison to the labels generated by the Zero-shot method, both Few-shot and Zero-shot CoT approaches produce labels that are more similar to those generated by humans.

### 4.4.2 Classification Results

The classification results obtained for five evaluation metrics are shown in Table 4.3. The rows represent all combinations of classification models, including transformer-based LLMs and combinations of embeddings and traditional ML models. Whereas the main columns represent the four training sets with different labels used to fine-tune these models. By setting the results of models fine-tuned on human labels as the ground truth, we highlighted (in green) the instances of the other three labeling sets that surpassed the corresponding baseline value. Overall, the Few-shot and Zero-shot CoT have obtained better results for many models. According to LLMs' results, BerTweet; a model pre-trained on 850M English Tweets (See Appendix) has outperformed the ground truth when fine-tuned on Few-shot and Zero-shot CoT labels. Similarly, this model has gained better or equal precision, recall, and MCC when fine-tuned on Zero-shot labels. Besides, MPNet and TRobStan on Few-shot labels, and Bert on Zero-shot CoT labels, have shown remarkable results on various metrics.

Noticeably, the traditional ML models have gained surpassing results, when the embedding techniques are Sentence Transformers or Glove. Besides, many of the embedding and traditional ML model combinations, such as Random Forest and Gradient Boosting Tree with Sentence Transformers and Gradient Boosting Tree and XGBoost classifier with Golve have exceeded the baseline margins when they are trained on Zero-shot CoT labels. However, for Few-shot learning, only SVM with GLove embedding has fully overpassed the human-label performance. On average, we noticed that the recalls of all the models when trained on Few-shot and Zero-shot CoT labels have reached or improved upon the baseline performance.

### 4.4.3 Results of Wilcoxon signed-rank test

Next, to summarize and compare the classification results mentioned above, we conducted a Wilcoxon signed-rank test by analyzing the performance metrics of different pairs of labeling sets. The results for each of the six possible pairs of labeling sets are presented in Table 4.4, showing the corresponding test-statistic and P-values. Here, we calculated the difference between the two groups as (Training label set 'a' - Training label set 'b'). The test-statistic values, which are larger and fall within the range of 250 to 350, along with significantly smaller P-values ranging from E-08 to E-02 for precision, f1-score, MCC, and ROC_AUC, indicate that the classification results for H-Z, H-F, and H-ZC are notably better when the models are trained using human labels com-

**Table 4.3:** Testing results of models fine-tuned on four training sets with different labels.

| Model | Classification results set 1 : Human labels | | | | | Classification results set 2 : Zero-shot labels | | | | | Classification results set 3 : Few-shot labels | | | | | Classification results set 4 : Zero-shot CoT labels | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | pre | rec | f1 | mcc | roc | pre | rec | f1 | mcc | roc | pre | rec | f1 | mcc | roc | pre | rec | f1 | mcc | roc |
| Bert | 0.70 | 0.69 | 0.69 | 0.53 | 0.84 | 0.64 | 0.67 | 0.59 | 0.44 | 0.81 | 0.68 | 0.61 | 0.60 | 0.40 | 0.83 | 0.68 | 0.74 | 0.69 | 0.56 | 0.84 |
| Albert | 0.57 | 0.59 | 0.54 | 0.33 | 0.70 | 0.44 | 0.48 | 0.42 | 0.15 | 0.66 | 0.46 | 0.41 | 0.41 | 0.10 | 0.65 | 0.48 | 0.53 | 0.46 | 0.21 | 0.73 |
| Debert | 0.73 | 0.64 | 0.67 | 0.52 | 0.82 | 0.63 | 0.67 | 0.57 | 0.44 | 0.76 | 0.64 | 0.63 | 0.61 | 0.41 | 0.81 | 0.60 | 0.64 | 0.59 | 0.42 | 0.80 |
| BerTweet | 0.72 | 0.70 | 0.71 | 0.55 | 0.89 | 0.72 | 0.76 | 0.69 | 0.57 | 0.86 | 0.75 | 0.76 | 0.72 | 0.59 | 0.91 | 0.76 | 0.74 | 0.75 | 0.62 | 0.88 |
| MPNet | 0.81 | 0.76 | 0.77 | 0.67 | 0.91 | 0.69 | 0.71 | 0.65 | 0.49 | 0.81 | 0.82 | 0.79 | 0.79 | 0.66 | 0.95 | 0.75 | 0.73 | 0.74 | 0.63 | 0.85 |
| TRob | 0.83 | 0.77 | 0.79 | 0.67 | 0.94 | 0.78 | 0.76 | 0.72 | 0.58 | 0.88 | 0.76 | 0.75 | 0.74 | 0.59 | 0.92 | 0.70 | 0.73 | 0.71 | 0.59 | 0.92 |
| TRobSen | 0.82 | 0.73 | 0.75 | 0.62 | 0.92 | 0.69 | 0.68 | 0.62 | 0.48 | 0.85 | 0.75 | 0.66 | 0.64 | 0.48 | 0.88 | 0.72 | 0.77 | 0.73 | 0.61 | 0.89 |
| TRobStan | 0.82 | 0.74 | 0.77 | 0.64 | 0.89 | 0.73 | 0.71 | 0.69 | 0.51 | 0.87 | 0.82 | 0.79 | 0.77 | 0.66 | 0.92 | 0.72 | 0.75 | 0.72 | 0.59 | 0.90 |
| LR-ADA | 0.78 | 0.77 | 0.77 | 0.65 | 0.92 | 0.65 | 0.70 | 0.63 | 0.49 | 0.87 | 0.76 | 0.80 | 0.77 | 0.66 | 0.91 | 0.70 | 0.75 | 0.70 | 0.58 | 0.90 |
| RF-ADA | 0.86 | 0.65 | 0.70 | 0.58 | 0.86 | 0.63 | 0.67 | 0.58 | 0.44 | 0.86 | 0.73 | 0.73 | 0.71 | 0.56 | 0.89 | 0.73 | 0.70 | 0.66 | 0.53 | 0.88 |
| SVM-ADA | 0.81 | 0.83 | 0.81 | 0.71 | 0.93 | 0.69 | 0.73 | 0.68 | 0.54 | 0.88 | 0.75 | 0.77 | 0.74 | 0.62 | 0.92 | 0.71 | 0.76 | 0.72 | 0.60 | 0.90 |
| MLP-ADA | 0.89 | 0.87 | 0.88 | 0.81 | 0.94 | 0.62 | 0.66 | 0.58 | 0.43 | 0.87 | 0.78 | 0.79 | 0.77 | 0.64 | 0.92 | 0.72 | 0.75 | 0.71 | 0.59 | 0.89 |
| GB-ADA | 0.77 | 0.64 | 0.67 | 0.55 | 0.91 | 0.68 | 0.71 | 0.60 | 0.50 | 0.82 | 0.66 | 0.65 | 0.64 | 0.47 | 0.86 | 0.64 | 0.66 | 0.61 | 0.46 | 0.86 |
| XGBoost-ADA | 0.79 | 0.71 | 0.74 | 0.62 | 0.91 | 0.64 | 0.64 | 0.54 | 0.42 | 0.87 | 0.68 | 0.72 | 0.68 | 0.52 | 0.87 | 0.68 | 0.69 | 0.61 | 0.48 | 0.88 |
| LR-SenTr | 0.74 | 0.78 | 0.75 | 0.63 | 0.90 | 0.68 | 0.68 | 0.56 | 0.47 | 0.85 | 0.71 | 0.76 | 0.71 | 0.58 | 0.88 | 0.71 | 0.77 | 0.72 | 0.60 | 0.88 |
| RF-SenTr | 0.71 | 0.64 | 0.66 | 0.49 | 0.85 | 0.64 | 0.62 | 0.49 | 0.39 | 0.80 | 0.66 | 0.67 | 0.64 | 0.48 | 0.86 | 0.71 | 0.69 | 0.66 | 0.52 | 0.85 |
| SVM-SenTr | 0.71 | 0.69 | 0.69 | 0.53 | 0.89 | 0.64 | 0.66 | 0.60 | 0.43 | 0.84 | 0.70 | 0.70 | 0.67 | 0.53 | 0.87 | 0.70 | 0.75 | 0.71 | 0.58 | 0.86 |
| MLP-SenTr | 0.75 | 0.69 | 0.71 | 0.57 | 0.91 | 0.68 | 0.70 | 0.63 | 0.49 | 0.87 | 0.77 | 0.80 | 0.77 | 0.65 | 0.88 | 0.69 | 0.72 | 0.68 | 0.55 | 0.87 |
| GB-SenTr | 0.63 | 0.58 | 0.59 | 0.39 | 0.84 | 0.62 | 0.63 | 0.56 | 0.40 | 0.80 | 0.69 | 0.62 | 0.60 | 0.43 | 0.81 | 0.68 | 0.70 | 0.67 | 0.54 | 0.86 |
| XGBoost-SenTr | 0.72 | 0.68 | 0.69 | 0.52 | 0.88 | 0.63 | 0.65 | 0.56 | 0.42 | 0.79 | 0.69 | 0.69 | 0.65 | 0.49 | 0.82 | 0.61 | 0.65 | 0.60 | 0.43 | 0.81 |
| LR-Glove | 0.63 | 0.60 | 0.61 | 0.40 | 0.80 | 0.52 | 0.55 | 0.52 | 0.29 | 0.75 | 0.59 | 0.56 | 0.54 | 0.32 | 0.78 | 0.57 | 0.62 | 0.56 | 0.37 | 0.77 |
| RF-Glove | 0.62 | 0.54 | 0.56 | 0.36 | 0.80 | 0.61 | 0.63 | 0.54 | 0.40 | 0.78 | 0.59 | 0.61 | 0.58 | 0.43 | 0.79 | 0.53 | 0.56 | 0.52 | 0.32 | 0.75 |
| SVM-Glove | 0.58 | 0.51 | 0.53 | 0.28 | 0.78 | 0.54 | 0.53 | 0.51 | 0.29 | 0.74 | 0.60 | 0.58 | 0.56 | 0.37 | 0.80 | 0.53 | 0.56 | 0.52 | 0.30 | 0.73 |
| MLP-Glove | 0.63 | 0.56 | 0.58 | 0.34 | 0.78 | 0.51 | 0.52 | 0.46 | 0.25 | 0.75 | 0.59 | 0.56 | 0.55 | 0.31 | 0.78 | 0.59 | 0.62 | 0.58 | 0.40 | 0.78 |
| GB-Glove | 0.52 | 0.50 | 0.51 | 0.25 | 0.76 | 0.48 | 0.51 | 0.43 | 0.24 | 0.70 | 0.52 | 0.52 | 0.51 | 0.29 | 0.71 | 0.55 | 0.55 | 0.53 | 0.32 | 0.78 |
| XGBoost-Glove | 0.59 | 0.55 | 0.56 | 0.36 | 0.77 | 0.49 | 0.52 | 0.45 | 0.24 | 0.73 | 0.58 | 0.53 | 0.54 | 0.29 | 0.77 | 0.59 | 0.62 | 0.58 | 0.40 | 0.79 |
| AVERAGE | 0.72 | 0.67 | 0.68 | 0.52 | 0.86 | 0.63 | 0.64 | 0.57 | 0.42 | 0.81 | 0.68 | 0.67 | 0.65 | 0.48 | 0.84 | 0.66 | 0.68 | 0.64 | 0.49 | 0.84 |

pared to the corresponding three label types. On the contrary, relatively larger P-values (6.91E-01, 9.25E-01) and smaller test-statistic values (144.0, 119.5) for recall in the H-F and H-ZC comparisons illustrate that the classification results of Few-shot and Zero-shot CoT label types are closer to that of human labels.

When comparing Zero-shot to both Few-shot and Zero-shot CoT performances, it is evident that the test-statistic values are consistently smaller, falling within the range of 2.0 to 78.0. This observation suggests that Zero-shot generally results in smaller values compared to the other two. Furthermore, the larger P-values, which range from E-01 to E+00, indicate that there is no statistically significant evidence to support the claim that Zero-shot tends to yield larger values. This indicates that these two techniques outperform the basic Zero-shot method significantly across all metrics. Based on the larger P-values obtained for the comparison of Few-shot and Zero-shot CoT, we describe that the recall, f1-score, MCC, and ROC_AUC of these two labeling techniques are not significantly different. However, due to the smaller P-value, it is clear that the precision of the Few-shot is significantly larger than that of the Zero-shot ZoT. Besides, the higher test-statistic values across all these 5 metrics indicate that the Few-shot has performed better than the Zero-shot CoT.

## 4.5 Further Discussion

In the subsequent section, we further analyze our primary results to extract more insightful observations.

### 4.5.1 Performance of GPT labeling on best classifiers of human labels

Referring to Table 4.3, it is evident that the baseline experiment showcased the highest performance from models, namely MLP-ADA, SVM-ADA, and TRob (Twitter Roberta) across a majority of metrics. In Fig. 4.7, we visualize the percentage improvements in performance[15] achieved by GPT-based labeling techniques across the top 12 models that achieved the best f1-scores (f1 ≥

[15] improvement percentage = (GPT result - human result) * 100

**Table 4.4:** Results of Wilcoxon signed-rank test performed to compare the evaluation metrics of each of two sets of labels generated by different approaches. The ‘W’ refers to the test-statistic and p-val refers to the P-value.

| Training label | Training label | Precision | | Recall | | F1-score | | MCC | | ROC_AUC | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| set ’a’ | set ’b’ | W | p-val | W | p-val | W | p-val | W | p-val | W | p-val |
| Human (H) | Zero-shot (Z) | 351.0 | 1.49E-08 | 250.0 | 2.97E-02 | 351.0 | 1.49E-08 | 340.0 | 8.20E-07 | 350.0 | 2.98E-08 |
| Human (H) | Few-shot (F) | 282.0 | 6.50E-04 | 144.0 | 6.91E-01 | 281.5 | 3.07E-03 | 262.0 | 1.36E-02 | 275.0 | 5.09E-03 |
| Human (H) | Zero-shot CoT (ZC) | 306.0 | 5.64E-05 | 119.5 | 9.25E-01 | 295.0 | 8.02E-04 | 247.0 | 3.55E-02 | 276.0 | 1.12E-03 |
| Zero-shot (Z) | Few-shot (F) | 11.5 | 1.00E+00 | 78.0 | 9.89E-01 | 2.0 | 1.00E+00 | 33.5 | 1.00E+00 | 6.0 | 1.00E+00 |
| Zero-shot (Z) | Zero-shot CoT (ZC) | 66.5 | 9.98E-01 | 43.0 | 9.99E-01 | 6.5 | 1.00E+00 | 22.0 | 1.00E+00 | 19.5 | 1.00E+00 |
| Few-shot (F) | Zero-shot CoT (ZC) | 275.5 | 5.09E-03 | 153.0 | 7.17E-01 | 213.5 | 1.77E-01 | 155.0 | 7.00E-01 | 187.0 | 1.45E-01 |

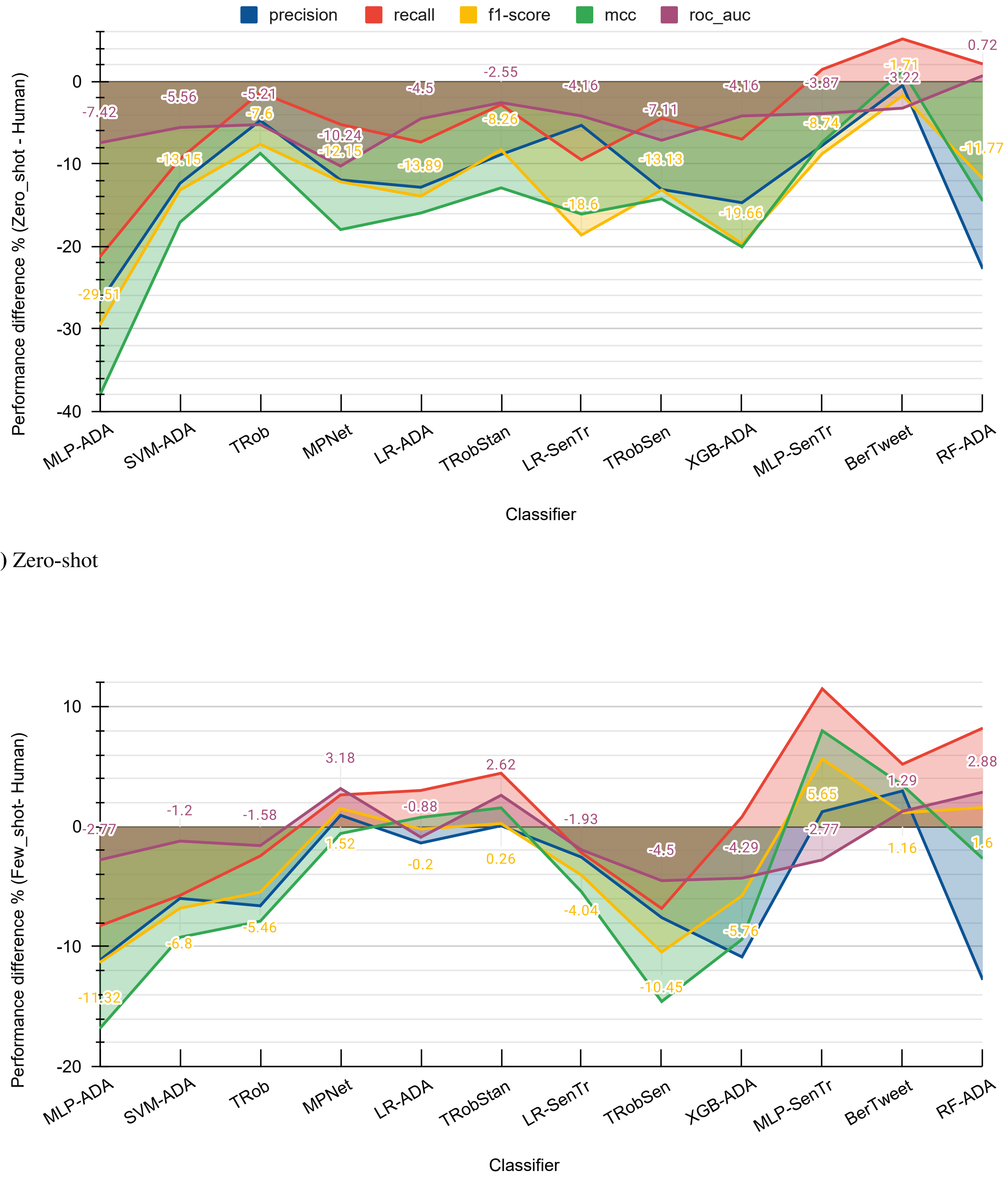
0.70) with human labels. Additionally, on the graphs, we numerically labeled the differences in performance for f1-score and ROC_AUC, two crucial metrics for evaluating an imbalanced multi-class classification task [185, 186]. In these graphs, the positive regions signify enhanced performance, while the negative regions reflect performance that failed to achieve the standards set by human labeling.

When comparing with Few-shot and Zero-shot CoT, the majority of the area in the Zero-shot category lies in the negative region, with a more substantial negative difference, reaching as low as -40.00%. Notably, BerTweet and TRobStan stand out as the top-performing models in the Zero-shot category, closely aligning with human labels across all metrics. In contrast, the performance of Few-shot occupies a larger positive area for many ML models. TRobStan and BerTweet emerge as the leading models, surpassing human labels through all the metrics, while MPNet, LR-ADA, and MLP-SenTr are a few other models performing at par with human labels. Among these models, BerTweet is highlighted as the best model for Zero-shot CoT labels, with only a minor decrease in ROC_AUC compared to human labels. Additionally, LR-SenTr and MPNet are two of the models with considerable performance.

However, it is essential to note that none of the GPT-4 techniques were able to match or surpass the human benchmark set by the top-performing three models, MLP-ADA, SVM-ADA, and TRob. Apart from that, out of all the labeling techniques, it is noteworthy that the percentages in the gap of recall and ROC_AUC between GPT and human labels are relatively lower compared to the other metrics. Moreover, similar to the literature that suggests MLP as one of the robust traditional classifiers on imbalanced datasets [187], we found MLP with ADA or Sentence Transformers produced better results when fine-tuned on human labels.

### 4.5.2 The Best Classifiers of GPT-based Labels

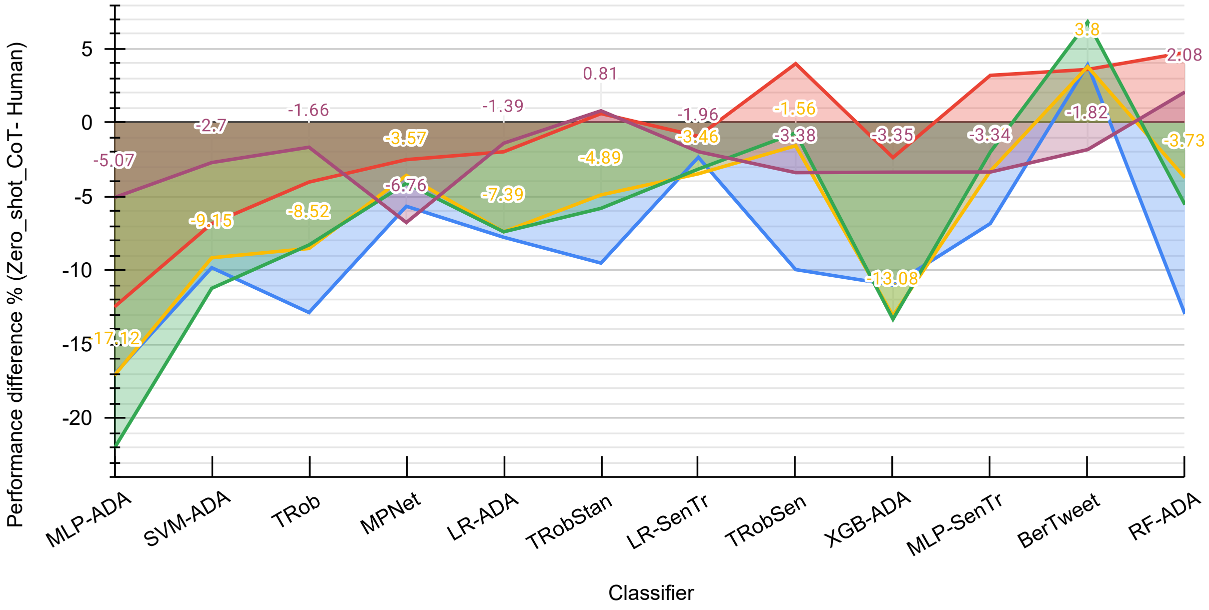
Table 4.5 lists the best-performed classifiers trained on GPT-based training labels, ordered by f1-score. Noticeably, the LLMs, such as BerTweet, TRob, TRobSen, and TRobStan which were pre-trained on Twitter datasets were among the top ten of all the three prompting techniques. MPNet, SVM-ADA, and LR-ADA embedding are the other classifiers commonly performed when trained



**(a)** Zero-shot

**(b)** Few-shot

**Figure 4.7:** The percentage increase in performance compared to human-labeled data, observed across the top-performing classifiers of human labeling.



**(a)** Zero-shot CoT

**Figure 4.7:** The percentage increase in performance compared to human-labeled data, observed across the top-performing classifiers of human labeling (Continued).

**Table 4.5:** Top classifiers trained on different GPT-based labeling sets based on f1-score.

| **Rank** | **Zero-shot** | **Few-shot** | **Zero-shot CoT** |
|---|---|---|---|
| 1 | TRob | MPNet | BerTweet |
| 2 | BerTweet | TRobStance | MPNet |
| 3 | TRobStance | LR-ADA | TRobSentiment |
| 4 | SVM-ADA | MLP-SenTrans | SVM-ADA |
| 5 | MPNet | MLP-ADA | TRobStance |
| 6 | LR-ADA | SVM-ADA | LR-SenTrans |
| 7 | MLP-SenTrans | TRob | TRob |
| 8 | TRobSentiment | BerTweet | SVM-SenTrans |
| 9 | GB-ADA | RF-ADA | MLP-ADA |
| 10 | SVM-SenTrans | LR-SenTrans | LR-ADA |

on any GPT-based labeling set. Additionally, no traditional classifiers with Glove embeddings are within the best performances and all six combinations of them are listed within the ten worst-performed classifiers of all three GPT-based labeling methods. Moreover, we noticed Albert as the model gained the least performance over all the five metrics in all the three labeling approaches.

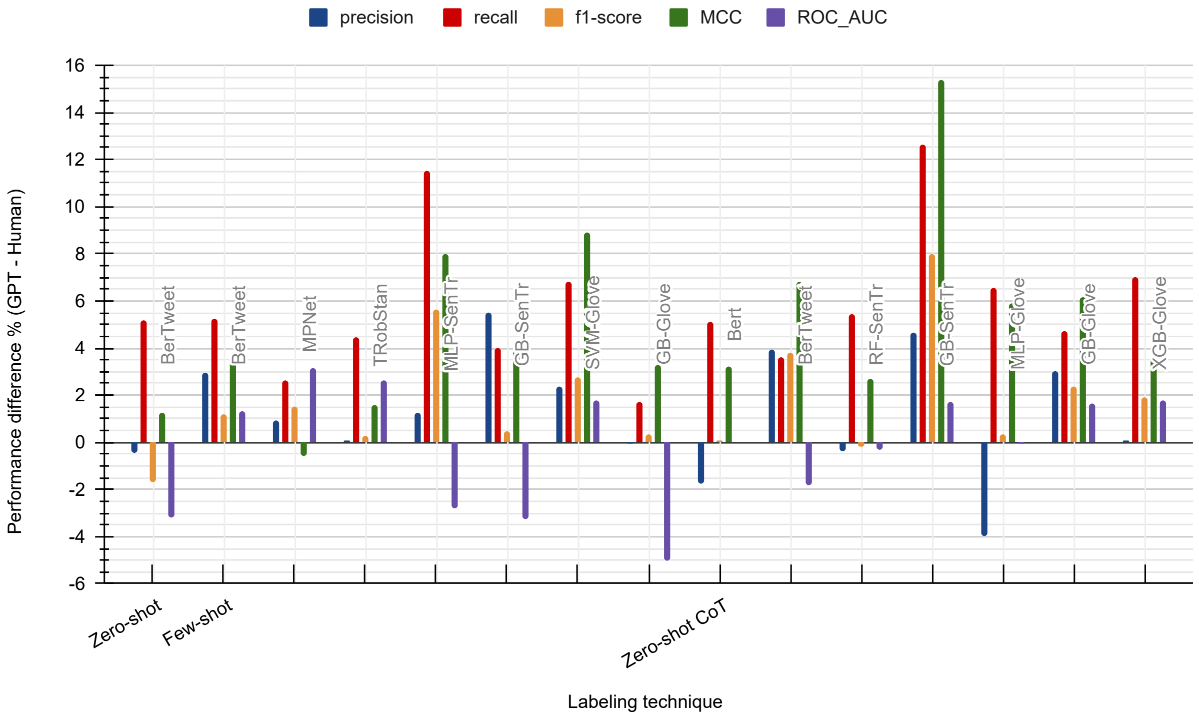


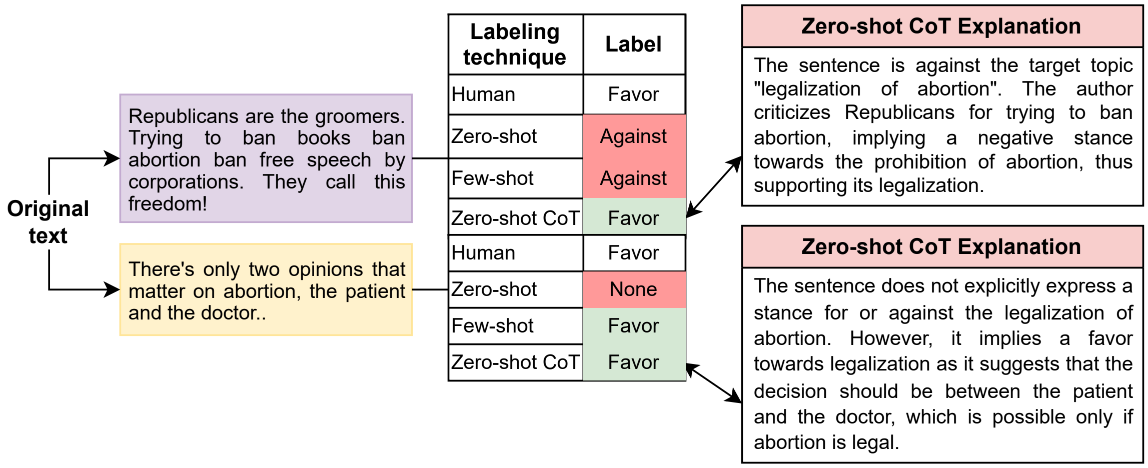
**Figure 4.8:** Performance analysis of classifiers trained on GPT-4's labeled datasets, which outperformed ground truth labels.

### 4.5.3 GPT Performance above the Benchmark

In this section, we focus on highlighting the classifiers trained using GPT-4's labeled datasets that have exceeded the performance of ground truth labels. Based on the cells highlighted in Table 4.3, we selected the models that excelled in at least four out of five metrics compared to the baseline. However, with Zero-shot labeling, we observed improved performance in a maximum of three out of five key metrics[16]. The percentages of performance gaps between GPT-4 techniques and human labels of these models are presented in Fig. 4.8.

In Zero-shot method, only BerTweet satisfies this criterion. On the other hand, Few-shot labeling has exhibited enhanced performance across seven models, with three of them being LLMs. Out of the seven classifiers that outperformed during Zero-shot CoT, the one using Gradient Boosting Tree with Sentence Transformer embedding emerged as the best, surpassing human label per-

[16] Please note that Table II displays the values rounded up to two decimals. Hence, a highlighted cell with equal performance in Table 4.3 can be displayed as a negative difference percentage of less than 0.5 in Fig. 4.8.



Republicans are the groomers. Trying to ban books ban abortion ban free speech by corporations. They call this freedom!

There's only two opinions that matter on abortion, the patient and the doctor..

| Labeling technique | Label |
|---|---|
| Human | Favor |
| Zero-shot | Against |
| Few-shot | Against |
| Zero-shot CoT | Favor |
| Human | Favor |
| Zero-shot | None |
| Few-shot | Favor |
| Zero-shot CoT | Favor |

**Zero-shot CoT Explanation**

The sentence is against the target topic "legalization of abortion". The author criticizes Republicans for trying to ban abortion, implying a negative stance towards the prohibition of abortion, thus supporting its legalization.

**Zero-shot CoT Explanation**

The sentence does not explicitly express a stance for or against the legalization of abortion. However, it implies a favor towards legalization as it suggests that the decision should be between the patient and the doctor, which is possible only if abortion is legal.

**Figure 4.9:** Two examples explaining the advantage of Zero-shot CoT over the basic Zero-shot prompting mechanism.

formance. It is worth noting that there were no classifier-embedding combinations using ADA embedding, despite its presence among the top-performing classifiers based on human labels. Additionally, BerTweet consistently delivered impressive results across all three GPT-4 labeling techniques.

Finally, it is noteworthy to compare the models presented in this section and the best classifiers based on human labels in Fig. 4.7 to understand how GPT-4 labeling techniques have achieved or exceeded the high standards set by humans. While Zero-shot labeling failed to meet this threshold, four models in the Few-shot category; BerTweet, MPNet, TRobStan, and MLP-SenTr along with Bertweet in Zero-shot CoT, surpassed the best ground truth performances across various metrics.

### 4.5.4 Improvements with Zero-shot CoT Mechanism

This approach has been implemented in generating answers to arithmetic, symbolic, and logical reasoning problems [178]. In this paper, we applied the Chain-of-Thoughts concept to comprehend and label social media texts, which exhibit their own unique characteristics. As mentioned, this prompting approach has the benefit of allowing the model to reassess its answer before determining the final label. Fig. 4.9 shows a few examples of how GPT-4 has changed its final answer based on this re-thinking strategy.

In both examples, Zero-shot assigns an incorrect label. In contrast, in Zero-shot CoT, it reads its own explanation and corrects the label. Both explanations clarify how GPT-4 initially generates incorrect answers for Zero-shot prompts. For instance, in the second explanation, it first states that the sentence does not explicitly express a stance on the legalization of abortion, leading to a 'none' label. However, it later expands its explanation, understanding an alternative viewpoint, and correctly labels it as 'favor'.

### 4.5.5 Limitations and Future Work

It is worth acknowledging that there is room for improvement in the quality of data annotated by GPT-4 when compared to human-annotated data. This study has some limitations, including a smaller dataset size and the use of a single dataset for stance detection, which may not fully capture the complexities of labeling social media text in stance classification, requiring domain-specific expertise. Furthermore, GPT models are highly sensitive to prompts and continually evolving, hence reproducibility of results must be considered. Our future work will involve expanding to multiple datasets and investigating the impact of the number of examples in Few-shot learning. Additionally, a comprehensive examination of GPT model robustness will be valuable, given that our approach employed fixed prompts and was resource-intensive due to the repeated execution of prompts to balance robustness and creativity in label generation.

## 4.6 Conclusion

Annotating social media text is a challenging task for humans due to the brevity, informality, and embedded socio-cultural opinions and perceptions in these texts where insufficient context understanding can result in low-quality annotations. To address this challenge, this study explores the potential of the GPT-4 model as an effective tool for labeling social media text, selecting stance labeling as the problem due to its relative complexity among other NLP tasks. We compare its performance across three prompting techniques, Zero-shot, Few-shot, and Zero-shot Chain-of-Thoughts (CoT) with human-labeled data. By observing the label distribution and the extent of alterations made to the original labels, it became evident that the Few-shot approach, followed by the Zero-shot CoT method, exhibits a higher degree of similarity to human experts in the assignment of labels to tweets. The overall results gained through 26 classifiers highlight the superiority of human labels, achieving higher performance across numerous metrics. However, several machine learning models fine-tuned on both Few-shot and Zero-shot CoT labels demonstrate enhanced or competitive individual performance, showcasing their ability to match human annotators in this task. Remarkably, we noticed that BerTweet has exhibited outstanding performance across all three labeling techniques. The Large Language Models, pre-trained on Twitter data, such as BerTweet, Twitter Roberta (TRob), Twitter Roberta Stance (TRobStan), and Twitter Roberta Sentiment (TRobSen), generally yield better results when fine-tuned on GPT-4-based labels or human labels. Furthermore, Zero-shot CoT demonstrated its strength compared to basic Zero-shot methods in labeling social media text for stance classification. Moreover, it competes effectively with the resource-intensive Few-shot approach, highlighting its capacity to produce reliable results without relying on labeled data samples. We anticipate that our findings will shed light on the utility of the GPT-4 model, for automating data annotation in social media text and inspire future research aimed at improving the quality and dependability of generated data.

# Chapter 5

# Conclusion

In summary, this work aims to harness the potential of generative AI for overcoming labeled data challenges in social media NLP. The work explores novel approaches to augmenting and annotating data, addressing the limitations of traditional methods.

Identification of Wellness Dimensions (WD) in self-narrated text is crucial, especially amid ongoing health crises. By leveraging generative NLP models, specifically ChatGPT, this novel data augmentation approach enhances the pre-screening task of classifying WD of humans. The results of this study showcase significant improvements, with up to a 13.11% increase in F-score and a 15.95% boost in Matthew's Correlation Coefficient. The gpt-3.5-turbo model outperforms baselines, displaying a 7.81% improvement in testing accuracy over the best traditional augmentation method, which is Backtranslation. This study not only enhances the robustness of NLP models but also contributes to mental health analysis through improved classification.

Data annotation in NLP is a time-consuming task, particularly challenging for social media texts due to their brevity and varying human perceptions. The second study of this thesis explores GPT-4's potential as a social media text annotator for a stance classification task. While GPT-4 demonstrates promising results through Few-shot and Zero-shot Chain-of-Thoughts prompting methods, it falls short of surpassing models fine-tuned on human labels. However, the Zero-shot Chain-of-Thoughts emerges as an effective and resource-efficient strategy for aspect-based social media text labeling. While addressing the challenges of data annotation, this study also contributes to the stance classification problem by constructing a novel dataset tailored to the nuances of stance identification. Furthermore, this work delves into prompt engineering, experimenting with various prompt-based learning techniques to enhance the effectiveness of the data labeling and classification process.

These findings underscore the potential of generative AI to enhance the efficiency and robustness of NLP models, opening new avenues for social media text data augmentation and annotation. In future studies, there is an opportunity to enhance the methodology and outcomes of both investigations. To improve robustness and reproducibility, future studies could expand on multiple social media datasets for data augmentation and annotation. Considering variations in refining prompt engineering strategies, including prompt designs, and parameters is also a possible pathway to experiment for enhancing performance. Furthermore, investigating other related techniques, such as leveraging ensemble methods, and active learning could offer alternative pathways for improving the overall quality and reliability of generative AI-based data annotation. The consideration of these aspects in future studies will contribute to the continuous refinement and applicability of the proposed approaches.

# Bibliography

[1] D. Khurana, A. Koli, K. Khatter, and S. Singh, “Natural language processing: State of the art, current trends and challenges,” *Multimedia tools and applications*, vol. 82, no. 3, pp. 3713–3744, 2023.

[2] R. Gozalo-Brizuela and E. C. Garrido-Merchan, “Chatgpt is not all you need. a state of the art review of large generative ai models,” *arXiv preprint arXiv:2301.04655*, 2023.

[3] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, *et al.*, “Transformers: State-of-the-art natural language processing,” in *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*, pp. 38–45, 2020.

[4] A. Farzindar, D. Inkpen, and G. Hirst, *Natural language processing for social media*. Springer, 2015.

[5] D. U. Patton, W. R. Frey, K. A. McGregor, F.-T. Lee, K. McKeown, and E. Moss, “Contextual analysis of social media: The promise and challenge of eliciting context in social media posts with natural language processing,” in *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, pp. 337–342, 2020.

[6] D. Maynard, K. Bontcheva, and D. Rout, “Challenges in developing opinion mining tools for social media,” *Proceedings of@ NLP can u tag# usergeneratedcontent*, pp. 15–22, 2012.

[7] S. K. Singh and K. Manoj, “Importance and challenges of social media text,” *International Journal of Advanced Research in Computer Science*, vol. 8, no. 3, pp. 831–834, 2017.

[8] C. Liyanage, R. Gokani, and V. Mago, “GPT-4 as a Twitter Data Annotator: Unraveling Its Performance on a Stance Classification Task,” 9 2023.

[9] P. Cunningham, M. Cord, and S. J. Delany, “Supervised learning,” in *Machine learning techniques for multimedia: case studies on organization and retrieval*, pp. 21–49, Springer, 2008.

[10] Y. Ouali, C. Hudelot, and M. Tami, “An overview of deep semi-supervised learning,” *arXiv preprint arXiv:2006.05278*, 2020.

[11] D. Tamming, “Data augmentation for text classification tasks,” Master’s thesis, University of Waterloo, 2020.

[12] Y. Shi, T. ValizadehAslani, J. Wang, P. Ren, Y. Zhang, M. Hu, L. Zhao, and H. Liang, “Improving imbalanced learning by pre-finetuning with data augmentation,” in *Fourth International Workshop on Learning with Imbalanced Domains: Theory and Applications*, pp. 68–82, PMLR, 2022.

[13] M. I. Jordan and T. M. Mitchell, “Machine learning: Trends, perspectives, and prospects,” *Science*, vol. 349, no. 6245, pp. 255–260, 2015.

[14] I. El Naqa and M. J. Murphy, *What is machine learning?* Springer, 2015.

[15] M. Mohri, A. Rostamizadeh, and A. Talwalkar, *Foundations of machine learning*. MIT press, 2018.

[16] A. R. Islam, “Machine learning in computer vision,” in *Applications of Machine Learning and Artificial Intelligence in Education*, pp. 48–72, IGI Global, 2022.

[17] T. P. Nagarhalli, V. Vaze, and N. Rana, “Impact of machine learning in natural language processing: A review,” in *2021 third international conference on intelligent communication technologies and virtual mobile networks (ICICV)*, pp. 1529–1534, IEEE, 2021.

[18] V. Vashisht, A. K. Pandey, and S. P. Yadav, "Speech recognition using machine learning," *IEIE Transactions on Smart Processing & Computing*, vol. 10, no. 3, pp. 233–239, 2021.

[19] M. F. Dixon, I. Halperin, and P. Bilokon, *Machine learning in finance*, vol. 1170. Springer, 2020.

[20] S. Suresh, N. Sinha, S. Prusty, *et al.*, "Latent approach in entertainment industry using machine learning," *International Research Journal on Advanced Science Hub*, vol. 2, no. Special Issue ICARD 2020, pp. 304–307, 2020.

[21] H. Luan and C.-C. Tsai, "A review of using machine learning approaches for precision education," *Educational Technology & Society*, vol. 24, no. 1, pp. 250–266, 2021.

[22] S. T. Jagtap, K. Phasinam, T. Kassanuk, S. S. Jha, T. Ghosh, and C. M. Thakar, "Towards application of various machine learning techniques in agriculture," *Materials Today: Proceedings*, vol. 51, pp. 793–797, 2022.

[23] A. Rajkomar, J. Dean, and I. Kohane, "Machine learning in medicine," *New England Journal of Medicine*, vol. 380, no. 14, pp. 1347–1358, 2019.

[24] A. L'heureux, K. Grolinger, H. F. Elyamany, and M. A. Capretz, "Machine learning with big data: Challenges and approaches," *Ieee Access*, vol. 5, pp. 7776–7797, 2017.

[25] G. Dong and H. Liu, *Feature engineering for machine learning and data analytics*. CRC press, 2018.

[26] W. Zhou, H. Wang, H. Sun, and T. Sun, "A method of short text representation based on the feature probability embedded vector," *Sensors*, vol. 19, no. 17, p. 3728, 2019.

[27] E. Bisong and E. Bisong, "Logistic regression," *Building Machine Learning and Deep Learning Models on Google Cloud Platform: A Comprehensive Guide for Beginners*, pp. 243–250, 2019.

[28] E. Y. Boateng and D. A. Abaye, “A review of the logistic regression model with emphasis on medical research,” *Journal of data analysis and information processing*, vol. 7, no. 4, pp. 190–207, 2019.

[29] Q. Wang, S. Yu, X. Qi, Y. Hu, W. Zheng, J. Shi, and H. Yao, “Overview of logistic regression model analysis and application,” *Zhonghua yu fang yi xue za zhi [Chinese journal of preventive medicine]*, vol. 53, no. 9, pp. 955–960, 2019.

[30] V. Y. Kulkarni and P. K. Sinha, “Random forest classifiers: a survey and future research directions,” *Int. J. Adv. Comput*, vol. 36, no. 1, pp. 1144–1153, 2013.

[31] K. Fawagreh, M. M. Gaber, and E. Elyan, “Random forests: from early developments to recent advancements,” *Systems Science & Control Engineering: An Open Access Journal*, vol. 2, no. 1, pp. 602–609, 2014.

[32] M. Schonlau and R. Y. Zou, “The random forest algorithm for statistical learning,” *The Stata Journal*, vol. 20, no. 1, pp. 3–29, 2020.

[33] J. L. Speiser, M. E. Miller, J. Tooze, and E. Ip, “A comparison of random forest variable selection methods for classification prediction modeling,” *Expert systems with applications*, vol. 134, pp. 93–101, 2019.

[34] C. Campbell and Y. Ying, *Learning with support vector machines*. Springer Nature, 2022.

[35] D. A. Pisner and D. M. Schnyer, “Support vector machine,” in *Machine learning*, pp. 101–121, Elsevier, 2020.

[36] J. Cervantes, F. Garcia-Lamont, L. Rodríguez-Mazahua, and A. Lopez, “A comprehensive survey on support vector machine classification: Applications, challenges and trends,” *Neurocomputing*, vol. 408, pp. 189–215, 2020.

[37] M. Tanveer, T. Rajani, R. Rastogi, Y.-H. Shao, and M. Ganaie, “Comprehensive review on twin support vector machines,” *Annals of Operations Research*, pp. 1–46, 2022.

[38] L. B. Almeida, “Multilayer perceptrons,” in *Handbook of Neural Computation*, pp. C1–2, CRC Press, 2020.

[39] A. A. Alnuaim, M. Zakariah, P. K. Shukla, A. Alhadlaq, W. A. Hatamleh, H. Tarazi, R. Sureshbabu, R. Ratna, *et al.*, “Human-computer interaction for recognizing speech emotions using multilayer perceptron classifier,” *Journal of Healthcare Engineering*, vol. 2022, 2022.

[40] S. Mirjalili, H. Faris, and I. Aljarah, “Evolutionary machine learning techniques,” *Cham, Switzerland: Springer*, 2019.

[41] C. Bentéjac, A. Csörgő, and G. Martínez-Muñoz, “A comparative analysis of gradient boosting algorithms,” *Artificial Intelligence Review*, vol. 54, pp. 1937–1967, 2021.

[42] Z. Zhang and C. Jung, “Gbdt-mo: Gradient-boosted decision trees for multiple outputs,” *IEEE transactions on neural networks and learning systems*, vol. 32, no. 7, pp. 3156–3167, 2020.

[43] Q. Li, Z. Wen, and B. He, “Practical federated gradient boosting decision trees,” in *Proceedings of the AAAI conference on artificial intelligence*, vol. 34, pp. 4642–4649, 2020.

[44] T. Chen, T. He, M. Benesty, V. Khotilovich, Y. Tang, H. Cho, K. Chen, R. Mitchell, I. Cano, T. Zhou, *et al.*, “Xgboost: extreme gradient boosting,” *R package version 0.4-2*, vol. 1, no. 4, pp. 1–4, 2015.

[45] Y.-C. Chang, K.-H. Chang, and G.-J. Wu, “Application of extreme gradient boosting trees in the construction of credit risk assessment models for financial institutions,” *Applied Soft Computing*, vol. 73, pp. 914–920, 2018.

[46] J. G. Ghatkar, R. K. Singh, and P. Shanmugam, “Classification of algal bloom species from remote sensing data using an extreme gradient boosted decision tree model,” *International Journal of Remote Sensing*, vol. 40, no. 24, pp. 9412–9438, 2019.

[47] K. Chowdhary and K. Chowdhary, "Natural language processing," *Fundamentals of artificial intelligence*, pp. 603–649, 2020.

[48] A. Torfi, R. A. Shirvani, Y. Keneshloo, N. Tavaf, and E. A. Fox, "Natural language processing advancements by deep learning: A survey," *arXiv preprint arXiv:2003.01200*, 2020.

[49] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[50] D. Scharp, M. Hobensack, A. Davoudi, and M. Topaz, "Natural language processing applied to clinical documentation in post-acute care settings: A scoping review," *Journal of the American Medical Directors Association*, 2023.

[51] K. Roitero, B. Portelli, M. H. Popescu, and V. Della Mea, "Dilbert: Cheap embeddings for disease related medical nlp," *IEEE Access*, vol. 9, pp. 159714–159723, 2021.

[52] G. K. Savova, I. Danciu, F. Alamudun, T. Miller, C. Lin, D. S. Bitterman, G. Tourassi, and J. L. Warner, "Use of natural language processing to extract clinical cancer phenotypes from electronic medical records," *Cancer research*, vol. 79, no. 21, pp. 5463–5470, 2019.

[53] J.-W. Chang, N. Yen, and J. C. Hung, "Design of a nlp-empowered finance fraud awareness model: the anti-fraud chatbot for fraud detection and fraud classification as an instance," *Journal of Ambient Intelligence and Humanized Computing*, vol. 13, no. 10, pp. 4663–4679, 2022.

[54] K. Mishev, A. Gjorgjevikj, I. Vodenska, L. T. Chitkushev, and D. Trajanov, "Evaluation of sentiment analysis in finance: from lexicons to transformers," *IEEE access*, vol. 8, pp. 131662–131682, 2020.

[55] Y. Jallan and B. Ashuri, "Text mining of the securities and exchange commission financial filings of publicly traded construction firms using deep learning to identify and assess risk," *Journal of Construction Engineering and Management*, vol. 146, no. 12, p. 04020137, 2020.

[56] N. Patel and S. Trivedi, “Leveraging predictive modeling, machine learning personalization, nlp customer support, and ai chatbots to increase customer loyalty,” *Empirical Quests for Management Essences*, vol. 3, no. 3, pp. 1–24, 2020.

[57] A. A. A. Ahmed and A. Ganapathy, “Creation of automated content with embedded artificial intelligence: a study on learning management system for educational entrepreneurship,” *Academy of Entrepreneurship Journal*, vol. 27, no. 3, pp. 1–10, 2021.

[58] Y. Chang, X. Wang, J. Wang, Y. Wu, K. Zhu, H. Chen, L. Yang, X. Yi, C. Wang, Y. Wang, *et al.*, “A survey on evaluation of large language models,” *arXiv preprint arXiv:2307.03109*, 2023.

[59] L. Fan, L. Li, Z. Ma, S. Lee, H. Yu, and L. Hemphill, “A bibliometric review of large language models research from 2017 to 2023,” *arXiv preprint arXiv:2304.02020*, 2023.

[60] B. Peng, C. Li, P. He, M. Galley, and J. Gao, “Instruction tuning with gpt-4,” *arXiv preprint arXiv:2304.03277*, 2023.

[61] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever, *et al.*, “Improving language understanding by generative pre-training,” 2018.

[62] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “Bert: Pre-training of deep bidirectional transformers for language understanding,” *arXiv preprint arXiv:1810.04805*, 2018.

[63] A. Conneau and G. Lample, “Cross-lingual language model pretraining,” *Advances in neural information processing systems*, vol. 32, 2019.

[64] B. Min, H. Ross, E. Sulem, A. P. B. Veyseh, T. H. Nguyen, O. Sainz, E. Agirre, I. Heintz, and D. Roth, “Recent advances in natural language processing via large pre-trained language models: A survey,” *ACM Computing Surveys*, vol. 56, no. 2, pp. 1–40, 2023.

[65] Y. Liu, T. Han, S. Ma, J. Zhang, Y. Yang, J. Tian, H. He, A. Li, M. He, Z. Liu, *et al.*, “Summary of chatgpt-related research and perspective towards the future of large language models,” *Meta-Radiology*, p. 100017, 2023.

[66] K. Song, X. Tan, T. Qin, J. Lu, and T.-Y. Liu, “Mpnet: Masked and permuted pre-training for language understanding,” *Advances in Neural Information Processing Systems*, vol. 33, pp. 16857–16867, 2020.

[67] Y. Hao, L. Dong, F. Wei, and K. Xu, “Visualizing and understanding the effectiveness of bert,” *arXiv preprint arXiv:1908.05620*, 2019.

[68] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut, “Albert: A lite bert for self-supervised learning of language representations,” *arXiv preprint arXiv:1909.11942*, 2019.

[69] H. Wang, X. Hu, and H. Zhang, “Sentiment analysis of commodity reviews based on albert-lstm,” in *Journal of Physics: Conference Series*, vol. 1651, p. 012022, IOP Publishing, 2020.

[70] B. Choi, Y. Lee, Y. Kyung, and E. Kim, “Albert with knowledge graph encoder utilizing semantic similarity for commonsense question answering,” *arXiv preprint arXiv:2211.07065*, 2022.

[71] P.-H. Chi, P.-H. Chung, T.-H. Wu, C.-C. Hsieh, Y.-H. Chen, S.-W. Li, and H.-y. Lee, “Audio albert: A lite bert for self-supervised learning of audio representation,” in *2021 IEEE Spoken Language Technology Workshop (SLT)*, pp. 344–350, IEEE, 2021.

[72] P. He, X. Liu, J. Gao, and W. Chen, “Deberta: Decoding-enhanced bert with disentangled attention,” *arXiv preprint arXiv:2006.03654*, 2020.

[73] J. Ganesh and A. Bansal, “Transformer-based automatic mapping of clinical notes to specific clinical concepts,” in *2023 IEEE 47th Annual Computers, Software, and Applications Conference (COMPSAC)*, pp. 558–563, IEEE, 2023.

[74] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. R. Salakhutdinov, and Q. V. Le, “Xlnet: Generalized autoregressive pretraining for language understanding,” *Advances in neural information processing systems*, vol. 32, 2019.

[75] S. Qianmin, P. Wei, C. Xiaoqiong, L. Hongxing, and H. Jihan, “Covid-19 clinical medical relationship extraction based on mpnet,” *IET Cyber-Physical Systems: Theory & Applications*, 2023.

[76] 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,” *arXiv preprint arXiv:1907.11692*, 2019.

[77] P. Delobelle, T. Winters, and B. Berendt, “Robbert: a dutch roberta-based language model,” *arXiv preprint arXiv:2001.06286*, 2020.

[78] F. Barbieri, J. Camacho-Collados, L. Neves, and L. Espinosa-Anke, “Tweeteval: Unified benchmark and comparative evaluation for tweet classification,” *arXiv preprint arXiv:2010.12421*, 2020.

[79] D. Loureiro, F. Barbieri, L. Neves, L. E. Anke, and J. Camacho-Collados, “Timelms: Diachronic language models from twitter,” *arXiv preprint arXiv:2202.03829*, 2022.

[80] P. Bansal and A. Sharma, “Large language models as annotators: Enhancing generalization of nlp models at minimal cost,” *arXiv preprint arXiv:2306.15766*, 2023.

[81] Z. Alyafeai, M. S. AlShaibani, and I. Ahmad, “A survey on transfer learning in natural language processing,” *arXiv preprint arXiv:2007.04239*, 2020.

[82] A. Chronopoulou, C. Baziotis, and A. Potamianos, “An embarrassingly simple approach for transfer learning from pretrained language models,” *arXiv preprint arXiv:1902.10547*, 2019.

[83] J. Wei, M. Bosma, V. Y. Zhao, K. Guu, A. W. Yu, B. Lester, N. Du, A. M. Dai, and Q. V. Le, “Finetuned language models are zero-shot learners,” *arXiv preprint arXiv:2109.01652*, 2021.

[84] N. Houlsby, A. Giurgiu, S. Jastrzebski, B. Morrone, Q. De Laroussilhe, A. Gesmundo, M. Attariyan, and S. Gelly, "Parameter-efficient transfer learning for nlp," in *International Conference on Machine Learning*, pp. 2790–2799, PMLR, 2019.

[85] 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.

[86] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter," *arXiv preprint arXiv:1910.01108*, 2019.

[87] A. Malte and P. Ratadiya, "Evolution of transfer learning in natural language processing," *arXiv preprint arXiv:1910.07370*, 2019.

[88] Y. Ge, Y. Guo, Y.-C. Yang, M. A. Al-Garadi, and A. Sarker, "Few-shot learning for medical text: A systematic review," *arXiv preprint arXiv:2204.14081*, 2022.

[89] M. Yang, "A survey on few-shot learning in natural language processing," in *2021 International Conference on Artificial Intelligence and Electromechanical Automation (AIEA)*, pp. 294–297, IEEE, 2021.

[90] W. Yin, "Meta-learning for few-shot natural language processing: A survey," *arXiv preprint arXiv:2007.09604*, 2020.

[91] T. Bansal, R. Jha, and A. McCallum, "Learning to few-shot learn across diverse natural language classification tasks," *arXiv preprint arXiv:1911.03863*, 2019.

[92] X. Sun, J. Gu, and H. Sun, "Research progress of zero-shot learning," *Applied Intelligence*, vol. 51, pp. 3600–3614, 2021.

[93] Y. Xian, B. Schiele, and Z. Akata, "Zero-shot learning-the good, the bad and the ugly," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4582–4591, 2017.

[94] 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.

[95] B. Romera-Paredes and P. Torr, "An embarrassingly simple approach to zero-shot learning," in *International conference on machine learning*, pp. 2152–2161, PMLR, 2015.

[96] F. Pourpanah, M. Abdar, Y. Luo, X. Zhou, R. Wang, C. P. Lim, X.-Z. Wang, and Q. J. Wu, "A review of generalized zero-shot learning methods," *IEEE transactions on pattern analysis and machine intelligence*, 2022.

[97] W. Wang, V. W. Zheng, H. Yu, and C. Miao, "A survey of zero-shot learning: Settings, methods, and applications," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 10, no. 2, pp. 1–37, 2019.

[98] W. M. Lim, A. Gunasekara, J. L. Pallant, J. I. Pallant, and E. Pechenkina, "Generative ai and the future of education: Ragnarök or reformation? a paradoxical perspective from management educators," *The International Journal of Management Education*, vol. 21, no. 2, p. 100790, 2023.

[99] A. Chowdhery, S. Narang, J. Devlin, M. Bosma, G. Mishra, A. Roberts, P. Barham, H. W. Chung, C. Sutton, S. Gehrmann, *et al.*, "Palm: Scaling language modeling with pathways," *arXiv preprint arXiv:2204.02311*, 2022.

[100] Z. Epstein, A. Hertzmann, I. of Human Creativity, M. Akten, H. Farid, J. Fjeld, M. R. Frank, M. Groh, L. Herman, N. Leach, *et al.*, "Art and the science of generative ai," *Science*, vol. 380, no. 6650, pp. 1110–1111, 2023.

[101] E. Brynjolfsson, D. Li, and L. R. Raymond, "Generative ai at work," tech. rep., National Bureau of Economic Research, 2023.

[102] A. Jo, "The promise and peril of generative ai," *Nature*, vol. 614, no. 1, pp. 214–216, 2023.

[103] I. Solaiman, "The gradient of generative ai release: Methods and considerations," in *Proceedings of the 2023 ACM Conference on Fairness, Accountability, and Transparency*, pp. 111–122, 2023.

[104] S. Coyne, K. Sakaguchi, D. Galvan-Sosa, M. Zock, and K. Inui, "Analyzing the performance of gpt-3.5 and gpt-4 in grammatical error correction," *arXiv preprint arXiv:2303.14342*, 2023.

[105] OpenAI, "Gpt-4 technical report," *ArXiv*, vol. abs/2303.08774, 2023.

[106] A. Koubaa, "Gpt-4 vs. gpt-3.5: A concise showdown," 2023.

[107] E. Waisberg, J. Ong, M. Masalkhi, S. A. Kamran, N. Zaman, P. Sarker, A. G. Lee, and A. Tavakkoli, "Gpt-4: a new era of artificial intelligence in medicine," *Irish Journal of Medical Science (1971-)*, pp. 1–4, 2023.

[108] J. White, Q. Fu, S. Hays, M. Sandborn, C. Olea, H. Gilbert, A. Elnashar, J. Spencer-Smith, and D. C. Schmidt, "A prompt pattern catalog to enhance prompt engineering with chatgpt," *arXiv preprint arXiv:2302.11382*, 2023.

[109] L. Giray, "Prompt engineering with chatgpt: A guide for academic writers," *Annals of Biomedical Engineering*, pp. 1–5, 2023.

[110] J. Wang, E. Shi, S. Yu, Z. Wu, C. Ma, H. Dai, Q. Yang, Y. Kang, J. Wu, H. Hu, *et al.*, "Prompt engineering for healthcare: Methodologies and applications," *arXiv preprint arXiv:2304.14670*, 2023.

[111] L. Reynolds and K. McDonell, "Prompt programming for large language models: Beyond the few-shot paradigm," in *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems*, pp. 1–7, 2021.

[112] P. Liu, W. Yuan, J. Fu, Z. Jiang, H. Hayashi, and G. Neubig, "Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing," *ACM Computing Surveys*, vol. 55, no. 9, pp. 1–35, 2023.

[113] T. Sorensen, J. Robinson, C. M. Rytting, A. G. Shaw, K. J. Rogers, A. P. Delorey, M. Khalil, N. Fulda, and D. Wingate, "An information-theoretic approach to prompt engineering without ground truth labels," *arXiv preprint arXiv:2203.11364*, 2022.

[114] R. Patil, S. Boit, V. Gudivada, and J. Nandigam, "A survey of text representation and embedding techniques in nlp," *IEEE Access*, 2023.

[115] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," *Advances in neural information processing systems*, vol. 26, 2013.

[116] J. Pennington, R. Socher, and C. Manning, "GloVe: Global vectors for word representation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (A. Moschitti, B. Pang, and W. Daelemans, eds.), (Doha, Qatar), pp. 1532–1543, Association for Computational Linguistics, Oct. 2014.

[117] T. Mikolov, E. Grave, P. Bojanowski, C. Puhrsch, and A. Joulin, "Advances in pre-training distributed word representations," in *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*, 2018.

[118] I. B. Drexel, "Feature engineering and word embedding impacts for automatic personality detection on instant message," in *2019 International Conference on Information Management and Technology (ICIMTech)*, vol. 1, pp. 155–159, IEEE, 2019.

[119] N. Muennighoff, N. Tazi, L. Magne, and N. Reimers, "Mteb: Massive text embedding benchmark," *arXiv preprint arXiv:2210.07316*, 2022.

[120] J. Yang, Y. Li, C. Gao, and Y. Zhang, "Measuring the short text similarity based on semantic and syntactic information," *Future Generation Computer Systems*, vol. 114, pp. 169–180, 2021.

[121] Y. Bao, H. Zhou, S. Huang, L. Li, L. Mou, O. Vechtomova, X. Dai, and J. Chen, "Generating sentences from disentangled syntactic and semantic spaces," *arXiv preprint arXiv:1907.05789*, 2019.

[122] H. Chen, S. Huang, D. Chiang, and J. Chen, "Improved neural machine translation with a syntax-aware encoder and decoder," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (R. Barzilay and M.-Y. Kan, eds.), (Vancouver, Canada), pp. 1936–1945, Association for Computational Linguistics, July 2017.

[123] D. Chandrasekaran and V. Mago, "Evolution of semantic similarity—a survey," *ACM Computing Surveys (CSUR)*, vol. 54, no. 2, pp. 1–37, 2021.

[124] P. Neculoiu, M. Versteegh, and M. Rotaru, "Learning text similarity with siamese recurrent networks," in *Proceedings of the 1st Workshop on Representation Learning for NLP*, pp. 148–157, 2016.

[125] N. Peinelt, D. Nguyen, and M. Liakata, "tbert: Topic models and bert joining forces for semantic similarity detection," in *Proceedings of the 58th annual meeting of the association for computational linguistics*, pp. 7047–7055, 2020.

[126] H. Ali, M. M. Salleh, R. Saedudin, K. Hussain, and M. F. Mushtaq, "Imbalance class problems in data mining: A review," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 14, no. 3, pp. 1560–1571, 2019.

[127] F. Thabtah, S. Hammoud, F. Kamalov, and A. Gonsalves, "Data imbalance in classification: Experimental evaluation," *Information Sciences*, vol. 513, pp. 429–441, 2020.

[128] C. Padurariu and M. E. Breaban, "Dealing with data imbalance in text classification," *Procedia Computer Science*, vol. 159, pp. 736–745, 2019.

[129] A. Kulkarni, D. Chong, and F. A. Batarseh, "Foundations of data imbalance and solutions for a data democracy," in *Data democracy*, pp. 83–106, Elsevier, 2020.

[130] B. Li, Y. Hou, and W. Che, “Data augmentation approaches in natural language processing: A survey,” *Ai Open*, vol. 3, pp. 71–90, 2022.

[131] S. Y. Feng, V. Gangal, J. Wei, S. Chandar, S. Vosoughi, T. Mitamura, and E. Hovy, “A survey of data augmentation approaches for nlp,” *arXiv preprint arXiv:2105.03075*, 2021.

[132] A. Mumuni and F. Mumuni, “Data augmentation: A comprehensive survey of modern approaches,” *Array*, p. 100258, 2022.

[133] J. Wei and K. Zou, “Eda: Easy data augmentation techniques for boosting performance on text classification tasks,” *arXiv preprint arXiv:1901.11196*, 2019.

[134] P. Liu, X. Wang, C. Xiang, and W. Meng, “A survey of text data augmentation,” in *2020 International Conference on Computer Communication and Network Security (CCNS)*, pp. 191–195, 2020.

[135] R. Sennrich, B. Haddow, and A. Birch, “Improving neural machine translation models with monolingual data,” in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (K. Erk and N. A. Smith, eds.), (Berlin, Germany), pp. 86–96, Association for Computational Linguistics, Aug. 2016.

[136] U. Hahn, E. Buyko, K. Tomanek, S. S. Piao, J. McNaught, Y. Tsuruoka, and S. Ananiadou, “An annotation type system for a data-driven nlp pipeline,” in *Proceedings of the Linguistic Annotation Workshop*, pp. 33–40, 2007.

[137] P. Stenetorp, S. Pyysalo, G. Topić, T. Ohta, S. Ananiadou, and J. Tsujii, “Brat: a web-based tool for nlp-assisted text annotation,” in *Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 102–107, 2012.

[138] Z. Zhang, E. Strubell, and E. Hovy, “A survey of active learning for natural language processing,” *arXiv preprint arXiv:2210.10109*, 2022.

[139] M. A. Hedderich, L. Lange, and D. Klakow, “Anea: distant supervision for low-resource named entity recognition,” *arXiv preprint arXiv:2102.13129*, 2021.

[140] C. Tan, F. Sun, T. Kong, W. Zhang, C. Yang, and C. Liu, “A survey on deep transfer learning,” in *Artificial Neural Networks and Machine Learning–ICANN 2018: 27th International Conference on Artificial Neural Networks, Rhodes, Greece, October 4-7, 2018, Proceedings, Part III 27*, pp. 270–279, Springer, 2018.

[141] A. Bompelli, Y. Wang, R. Wan, E. Singh, Y. Zhou, L. Xu, D. Oniani, B. S. A. Kshatriya, J. J. E. Balls-Berry, and R. Zhang, “Social and behavioral determinants of health in the era of artificial intelligence with electronic health records: a scoping review,” *Health Data Science*, vol. 2021, 2021.

[142] S. Wang, M. Schraagen, E. T. K. Sang, and M. Dastani, “Public sentiment on governmental covid-19 measures in dutch social media,” in *Proceedings of the 1st Workshop on NLP for COVID-19 (Part 2) at EMNLP 2020*, 2020.

[143] T. Zhang, K. Yang, S. Ji, and S. Ananiadou, “Emotion fusion for mental illness detection from social media: A survey,” *Information Fusion*, vol. 92, pp. 231–246, 2023.

[144] M. Garg, “Mental health analysis in social media posts: A survey,” *Archives of Computational Methods in Engineering*, pp. 1–24, 2023.

[145] E. Printz-Markó and Z. Ivancsóné Horváth, “Applicability of american wellness research methods in case of central-european countries,” in *DIEM: Dubrovnik International Economic Meeting*, vol. 3, pp. 825–842, Sveučilište u Dubrovniku, 2017.

[146] P. C. Wickramarathne, J. C. Phuoc, and A. R. S. Albattat, “A review of wellness dimension models: For the advancement of the society,” *European Journal of Social Sciences Studies*, 2020.

[147] A. K. Dillette, A. C. Douglas, and C. Andrzejewski, "Dimensions of holistic wellness as a result of international wellness tourism experiences," *Current Issues in Tourism*, vol. 24, no. 6, pp. 794–810, 2021.

[148] R. Weiss, *Loneliness: The experience of emotional and social isolation.* MIT press, 1975.

[149] K. Yang, S. Ji, T. Zhang, Q. Xie, and S. Ananiadou, "On the evaluations of chatgpt and emotion-enhanced prompting for mental health analysis," *arXiv preprint arXiv:2304.03347*, 2023.

[150] N. Ding, Y. Qin, G. Yang, F. Wei, Z. Yang, Y. Su, S. Hu, Y. Chen, C.-M. Chan, W. Chen, *et al.*, "Parameter-efficient fine-tuning of large-scale pre-trained language models," *Nature Machine Intelligence*, pp. 1–16, 2023.

[151] Y. Meng, J. Huang, Y. Zhang, and J. Han, "Generating training data with language models: Towards zero-shot language understanding," in *Advances in Neural Information Processing Systems*, 2021.

[152] M. Ormerod, J. Martínez del Rincón, and B. Devereux, "Predicting semantic similarity between clinical sentence pairs using transformer models: Evaluation and representational analysis," *JMIR Medical Informatics*, vol. 9, no. 5, p. e23099, 2021.

[153] N. Reimers, I. Gurevych, N. Reimers, I. Gurevych, N. Thakur, N. Reimers, J. Daxenberger, I. Gurevych, N. Reimers, I. Gurevych, *et al.*, "Sentence-bert: Sentence embeddings using siamese bert-networks," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, pp. 671–688, Association for Computational Linguistics, 2019.

[154] S. Boughorbel, F. Jarray, and M. El-Anbari, "Optimal classifier for imbalanced data using matthews correlation coefficient metric," *PloS one*, vol. 12, no. 6, p. e0177678, 2017.

[155] L. Cheng, X. Li, and L. Bing, "Is gpt-4 a good data analyst?," *arXiv preprint arXiv:2305.15038*, 2023.

[156] C.-H. Chiang and H.-y. Lee, “Can large language models be an alternative to human evaluations?,” *arXiv preprint arXiv:2305.01937*, 2023.

[157] J. Wang, Y. Liang, F. Meng, H. Shi, Z. Li, J. Xu, J. Qu, and J. Zhou, “Is chatgpt a good nlg evaluator? a preliminary study,” *arXiv preprint arXiv:2303.04048*, 2023.

[158] Y. Feng, S. Vanam, M. Cherukupally, W. Zheng, M. Qiu, and H. Chen, “Investigating code generation performance of chat-gpt with crowdsourcing social data,” in *Proceedings of the 47th IEEE Computer Software and Applications Conference*, pp. 1–10, 2023.

[159] R. A. Poldrack, T. Lu, and G. Beguš, “Ai-assisted coding: Experiments with gpt-4,” *arXiv preprint arXiv:2304.13187*, 2023.

[160] S. MacNeil, A. Tran, D. Mogil, S. Bernstein, E. Ross, and Z. Huang, “Generating diverse code explanations using the gpt-3 large language model,” in *Proceedings of the 2022 ACM Conference on International Computing Education Research-Volume 2*, pp. 37–39, 2022.

[161] R. S. de Padua, I. Qureshi, and M. U. Karakaplan, “Gpt-3 models are few-shot financial reasoners,” *arXiv preprint arXiv:2307.13617*, 2023.

[162] Z. Liu, X. Yu, L. Zhang, Z. Wu, C. Cao, H. Dai, L. Zhao, W. Liu, D. Shen, Q. Li, *et al.*, “Deid-gpt: Zero-shot medical text de-identification by gpt-4,” *arXiv preprint arXiv:2303.11032*, 2023.

[163] E. T. R. Schneider, J. V. A. de Souza, Y. B. Gumiel, C. Moro, and E. C. Paraiso, “A gpt-2 language model for biomedical texts in portuguese,” in *2021 IEEE 34th international symposium on computer-based medical systems (CBMS)*, pp. 474–479, IEEE, 2021.

[164] S. Rathje, D.-M. Mirea, I. Sucholutsky, R. Marjieh, C. Robertson, and J. J. Van Bavel, “Gpt is an effective tool for multilingual psychological text analysis,” 2023.

[165] D. M. Katz, M. J. Bommarito, S. Gao, and P. Arredondo, “Gpt-4 passes the bar exam,” *Available at SSRN 4389233*, 2023.

[166] Z. Xiao, X. Yuan, Q. V. Liao, R. Abdelghani, and P.-Y. Oudeyer, “Supporting qualitative analysis with large language models: Combining codebook with gpt-3 for deductive coding,” in *Companion Proceedings of the 28th International Conference on Intelligent User Interfaces*, pp. 75–78, 2023.

[167] B. Ding, C. Qin, L. Liu, L. Bing, S. Joty, and B. Li, “Is gpt-3 a good data annotator?,” *arXiv preprint arXiv:2212.10450*, 2022.

[168] J. Savelka, K. D. Ashley, M. A. Gray, H. Westermann, and H. Xu, “Can gpt-4 support analysis of textual data in tasks requiring highly specialized domain expertise?,” *arXiv preprint arXiv:2306.13906*, 2023.

[169] J. Savelka, “Unlocking practical applications in legal domain: Evaluation of gpt for zero-shot semantic annotation of legal texts,” *arXiv preprint arXiv:2305.04417*, 2023.

[170] S. Wang, Y. Liu, Y. Xu, C. Zhu, and M. Zeng, “Want to reduce labeling cost? gpt-3 can help,” *arXiv preprint arXiv:2108.13487*, 2021.

[171] P. Törnberg, “Chatgpt-4 outperforms experts and crowd workers in annotating political twitter messages with zero-shot learning,” *arXiv preprint arXiv:2304.06588*, 2023.

[172] S. Mohammad, S. Kiritchenko, P. Sobhani, X. Zhu, and C. Cherry, “A dataset for detecting stance in tweets,” in *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC’16)*, pp. 3945–3952, 2016.

[173] M. Evrard, R. Uro, N. Hervé, and B. Mazoyer, “French tweet corpus for automatic stance detection,” in *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pp. 6317–6322, 2020.

[174] K. Joseph, L. Friedland, W. Hobbs, O. Tsur, and D. Lazer, “Constance: Modeling annotation contexts to improve stance classification,” *arXiv preprint arXiv:1708.06309*, 2017.

[175] S. M. Mohammad, P. Sobhani, and S. Kiritchenko, “Stance and sentiment in tweets,” *ACM Transactions on Internet Technology (TOIT)*, vol. 17, no. 3, pp. 1–23, 2017.

[176] S. Mohammad, S. Kiritchenko, P. Sobhani, X. Zhu, and C. Cherry, "Semeval-2016 task 6: Detecting stance in tweets," in *Proceedings of the 10th international workshop on semantic evaluation (SemEval-2016)*, pp. 31–41, 2016.

[177] J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. V. Le, D. Zhou, *et al.*, "Chain-of-thought prompting elicits reasoning in large language models," *Advances in Neural Information Processing Systems*, vol. 35, pp. 24824–24837, 2022.

[178] T. Kojima, S. S. Gu, M. Reid, Y. Matsuo, and Y. Iwasawa, "Large language models are zero-shot reasoners," *Advances in neural information processing systems*, vol. 35, pp. 22199–22213, 2022.

[179] A. Zapf, S. Castell, L. Morawietz, and A. Karch, "Measuring inter-rater reliability for nominal data–which coefficients and confidence intervals are appropriate?," *BMC medical research methodology*, vol. 16, pp. 1–10, 2016.

[180] M. L. McHugh, "Interrater reliability: the kappa statistic," *Biochemia medica*, vol. 22, no. 3, pp. 276–282, 2012.

[181] P. Liu, W. Yuan, J. Fu, Z. Jiang, H. Hayashi, and G. Neubig, "Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing," *ACM Computing Surveys*, vol. 55, no. 9, pp. 1–35, 2023.

[182] D. Q. Nguyen, T. Vu, and A. T. Nguyen, "Bertweet: A pre-trained language model for english tweets," *arXiv preprint arXiv:2005.10200*, 2020.

[183] A. Shahbandegan, V. Mago, A. Alaref, C. B. van der Pol, and D. W. Savage, "Developing a machine learning model to predict patient need for computed tomography imaging in the emergency department," *Plos One*, vol. 17, no. 12, p. e0278229, 2022.

[184] D. Chicco and G. Jurman, "The advantages of the matthews correlation coefficient (mcc) over f1 score and accuracy in binary classification evaluation," *BMC genomics*, vol. 21, no. 1, pp. 1–13, 2020.

[185] M. Grandini, E. Bagli, and G. Visani, “Metrics for multi-class classification: an overview,” *arXiv preprint arXiv:2008.05756*, 2020.

[186] C. Halimu, A. Kasem, and S. S. Newaz, “Empirical comparison of area under roc curve (auc) and mathew correlation coefficient (mcc) for evaluating machine learning algorithms on imbalanced datasets for binary classification,” in *Proceedings of the 3rd international conference on machine learning and soft computing*, pp. 1–6, 2019.

[187] C. Lemnaru and R. Potolea, “Imbalanced classification problems: systematic study, issues and best practices,” in *Enterprise Information Systems: 13th International Conference, ICEIS 2011, Beijing, China, June 8-11, 2011, Revised Selected Papers 13*, pp. 35–50, Springer, 2012.

[188] S. W. Scheff, *Fundamental statistical principles for the neurobiologist: A survival guide*. Academic Press, 2016.

[189] S. Taheri and G. Hesamian, “A generalization of the wilcoxon signed-rank test and its applications,” *Statistical Papers*, vol. 54, pp. 457–470, 2013.

[190] J. H. McDonald, *Handbook of biolological statistics*. New York•, 2014.

[191] A. Benavoli, G. Corani, F. Mangili, M. Zaffalon, and F. Ruggeri, “A bayesian wilcoxon signed-rank test based on the dirichlet process,” in *International conference on machine learning*, pp. 1026–1034, PMLR, 2014.