IT UNIVERSITY OF COPENHAGEN

Are large language models enough?

A framework for the aspiring computational social scientist to tackle low-resource tasks using large language models and contrastive few-shot learning.

Anders Giovanni Møller
agmo@itu.dk
&
Jacob Aarup Dalsgaard
jacd@itu.dk

Thesis
Course code: KISPECI1SE
MSc. Data Science
Anders Giovanni Møller & Jacob Aarup Dalsgaard

June 1, 2023
Supervisors: Luca Maria Aiello (main) and Arianna Pera

ABSTRACT

Computational social science employs computer science and artificial intelligence methods to investigate intricate social phenomena, particularly online interactions on social media platforms. This study presents a comprehensive framework and a set of practical recommendations for computational social science researchers, addressing the challenges associated with low-resource settings, characterized by limited data availability. Our recommendations are based on experiments on 3 tasks with varying complexity, from simple semantic analysis to intricate pragmatic understanding. More specifically, sentiment analysis, hate speech detection, and social dimension extraction.

As a novel approach to overcome data scarcity, we leverage state-of-the-art large language models, GPT-4 and ChatGPT, to generate new data based on a small sample of human-annotated data. This enables a cost-effective alternative to data collection. We fine-tune a small language model using increasingly larger datasets to investigate if performance scales with data size for human-annotated and augmented data. Additionally, we explore the use of SetFit, a few-shot learning method that incorporates contrastive learning, on the task of social dimensions extraction. We also evaluate the zero-shot capabilities of GPT-4 and ChatGPT.

For the two simpler tasks, we find that the crowdsourced data outperforms the augmented data. However, in the case of the complex social dimension extraction task, the augmented data achieves comparable performance to the crowdsourced data. This calls into question the current narrative that LLMs are universal task-solvers, including data augmentation. By leveraging contrastive learning, we significantly enhance the performance of models fine-tuned on augmented datasets. This emphasizes the effectiveness of few-shot learning methods in low-resource tasks. Overall, the highest performance is attained by fine-tuning the model on a combination crowdsourced and augmented data, resulting in a macro F1 score of 0.347 and an accuracy of 0.487. This is an improvement of 0,032 and 0,013 compared to the baseline.

While instruction-tuned large language models have disrupted numerous domains, their potential as zero-shot learners in highly specialized tasks and as tools for data augmentation remains open-ended. This study provides novel and nuanced insights into the capabilities of large language models within the domain of computational social science, demonstrating their potential uses and applicability.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

PREFACE

Dear Luca and Arianna,

We would like to take this opportunity to thank both of you for the expertise and guidance you have provided throughout this study. We want to thank you for all the weekly meetings, the formal and informal talks, and your constant availability and helpfulness when we needed it the most. Even on weekends and holidays.

Parts of this study have resulted in a preprint publication[1] co-authored by our supervisors.

[1] https://arxiv.org/abs/2304.13861

1 INTRODUCTION

Computational social science (CSS) is an important discipline for studying human phenomena through the lens of online interactions. CSS can be used to understand how people act, and interact with each other, how diseases spread in networks, how information flows through groups of people, or how one can benefit from weakly-tied acquaintances to get a new job (Lazer et al., 2020). The possibilities are manifold. Using social media is a widespread daily habit of billions of people, it is therefore very important to understand how people communicate with each other, and how relationships are formed, grounded by fundamental sociological theories.

CSS encompasses a wide range of human phenomena, such as information diffusion (Al-Taie and Kadry, 2017), opinion formation (Wu and Huberman, 2004), polarization (Matakos et al., 2017), and coordination (Monti et al., 2022). However, the complexity of these phenomena is so profound that the study of social networks alone is inadequate for a comprehensive understanding. Thus, it is necessary to employ additional methods that delve into the analysis of language used in social interactions. Using methods from natural language processing (NLP) can facilitate a more comprehensive investigation of such intricate phenomena.

Capturing these phenomena through online conversational data can be difficult, as texts can convey concepts that go beyond semantics. While sentiment analysis and hate speech detection represent conventional tasks (Yue et al., 2019; Fortuna and Nunes, 2019), they entail linguistic and semantic nuances making them complex to capture even using contemporary state-of-the-art NLP techniques. Still, these are high-resource tasks, as annotated datasets are numerous. On the other hand, the pursuit of developing methods capable of capturing elements of social pragmatics, such as trust and respect remains an ambitious endeavor (Deri et al., 2018; Choi et al., 2020). In addition, these tasks are low-resource tasks with limited available data. Nevertheless, existing computational and algorithmic approaches currently fall short of confidently detecting and analyzing these complicated and nuanced aspects of human communication.

A common approach to overcome the challenge of capturing these complex concepts is to collect additional training data. However, doing so imposes several challenges. First, annotating texts with classes that go beyond semantics requires intricate knowledge of the task. Using expert annotators incurs a big cost to data labeling requiring researchers to sacrifice quantity for quality. Second, many complex CSS tasks naturally have an unbalanced occurrence of classes as some concepts only occur rarely

in data. Therefore, researchers need to devote much time and energy to gather and identify even just a few of such examples.

The field of natural language processing has gained significant public attention, primarily due to the development of *large language models* (LLMs) such as OpenAI's ChatGPT and GPT-4 (OpenAI, 2023). Some of the key features of LLMs are their easy interface and ability to produce high-quality human-like text, which has attracted the interest of a broad range of audiences, including non-experts (Zhao et al., 2023). They are large agglomerates of semantic knowledge and have demonstrative impressive performance in a wide range of tasks (Bubeck et al., 2023a). LLMs can be used in a zero-shot fashion to solve problems in computational social science. Specifically to CSS, attention has so far been focused on how LLMs could replace humans in tasks such as data annotation and text analysis. (Gilardi et al., 2023; Ziems et al., 2023; Byun et al., 2023).

However, relying on LLMs as zero-shot learners for all present tasks currently impose two particular problems. First, most high-performing LLMs currently remain proprietary, making their use financially expensive when applied to large-scale data. Moreover, concerns about privacy and legal implications further diminish the feasibility when employing these models. Second, despite their immediate impressive performances, LLMs lack specialization and it is currently not apparent whether a trained model with sufficient training data can surpass the performance of a zero-shot LLM.

In an attempt to address the limitations above, we select three NLP classification tasks in the domain of CSS. More specifically, sentiment analysis (Rosenthal et al., 2019), hate speech detection (Sigurbergsson and Derczynski, 2020), and social dimensions extraction (Choi et al., 2020). We try to answer the question of whether GPT-4 and ChatGPT can be leveraged to augment data to train task-specific classifiers.

Concretely, for each of the tasks, we collect crowdsourced training data and leverage LLMs to incrementally augment samples from a subset of the data. We employ two distinct label distribution strategies. We assess the performance of augmented data, and compare it against two benchmarks: 1) the performance of human-generated labels, and 2) the zero-shot capabilities of the LLMs.

In addition to that experiment, we further explore few-shot learning by leveraging the contrastive learning framework, SetFit (Tunstall et al., 2022). SetFit is purposefully developed to maximize data utilization in scenarios with limited availability, making the method particularly well-suited for our objectives. SetFit works in a two-step process, where the body of a language model is pre-trained with contrastive learning, and subsequently fine-tuned for the classification task. This approach ensures that the model can effectively adapt to specific tasks with minimal data.

Contrary to previous research, we find that there is no clear answer to whether LLMs can replace human tasks. Overall, we find that text gen-

erated by LLMs provides a useful signal for classifiers to discriminate between classes as they can be used to consistently produce high-quality data that is semantically similar to human-generated data. However, using the traditional fine-tuning procedure we still find augmented data to be either on-par or worse than crowdsourced data, clouding the findings on the capabilities of LLMs. Still, we find the performance of zero-shot LLMs to be competitive to the models fine-tuned on crowdsourced data, underlining their ubiquitous good baseline performance. Interestingly, when using contrastive pre-training we observe a significant increase in the performance of the models fine-tuned on augmented data outcompeting both the zero-shot performance of the most advanced LLMs and the best model trained on crowdsourced data. Finally, we observe that optimal performance is obtained when using crowdsourced and augmented data in combination. These final results suggest that LLM augmented data provides a very useful and cost-effective alternative to large-scale data collection and annotation processes. However, we encourage further exploration of data augmentation using LLMs to provide more nuance and further improvement.

Overall, our contributions are:

- We are the first to publish prompt examples for the instruction fine-tuned LLMs ChatGPT and GPT-4 for data augmentation in various CSS tasks. Our prompt strategy is the first to include semantic, label, and domain-preserving information with the goal of generating lexically diverse but semantically similar examples.
- We investigate whether augmented data provides as efficient training samples as data generated by humans and whether models trained on augmented data can compete with models trained on human-annotated data in 3 different CSS tasks.
- We challenge the idea that CSS classification tasks in the future will simply be solved by the strong zero-shot performance of LLMs. Instead, we argue that LLMs will enable us to build highly performant and specialized models that outcompete them.
- We evaluate the performance of the contrastive few-shot method SetFit when applied to a complex CSS task using both augmented and crowdsourced data. We compare the performance to models trained using the traditional fine-tuning procedure.
- We carry out a qualitative assessment of the generated data informed by metrics of the semantic similarity and lexical diversity of the generated texts.

1.1 RECOMMENDATIONS TO CSS PRACTITIONERS

Given our findings in this project, we recommend the following possible strategies for CSS practitioners, based on plausible scenarios.

1. **You have very limited human-annotated data, and you quickly want a baseline performance at low financial cost**.

   Given a test set or data corpus with limited annotations, we recommend using LLMs in a zero-shot setting to obtain a baseline performance. This can be used for the refinement of prompts or annotation guidelines, and serve as a baseline for comparison against trained models.

2. **Human-annotated data is limited and you want to quickly maximize performance at low financial cost**.

   Given a small dataset, we use 500 samples, use LLMs to augment data and train a model using contrastive pre-training. Compare performance to a model trained with contrastive learning on the 500 samples.

3. **You have a larger dataset but do not achieve satisfying performance**.

   Given a larger dataset, we have around 5,000 samples, use augmented data in concatenation with human-annotated data and train a language model. Our experiments showed that this approach achieved the best performance.

2 RELATED WORK

2.1 COMPUTATIONAL SOCIAL SCIENCE

Computational social science is an interdisciplinary field that erupted due to the enormous data traces from online interaction. CSS leverages computational methods to study social science phenomena, such as language and online behavior, at an immense scale (Lazer et al., 2009). The applications in the field are vast and diverse and require computational methods to understand social phenomena through text, that can go beyond simple semantics. Computational social science can be used in linguistics to examine how people interact with language (Nguyen et al., 2016). It can be used in political science to understand public opinion, ideology, and movement of political agendas (Grimmer, 2010). Psychology uses computational methods to study the correlation between language and personality traits (Schwartz et al., 2013), and sociology aims at understanding how society functions by examining social communities (Valverde-Rebaza and de Andrade Lopes, 2013). Often, these social disciplines are studied using methods from network science or natural language processing. In this project, we focus on the latter.

The first CSS task we focus on in this project is sentiment analysis on Twitter (Rosenthal et al., 2019). Sentiment analysis is a common task and allows for further downstream applications, for instance, public opinion about various political subjects (Ali et al., 2022), or product satisfaction (Anto et al., 2016). The second task, hate speech detection (Sigurbergsson and Derczynski, 2020), has also gained broad popularity due to the increasing amount of user-generated text from different social platforms. Hate speech is prohibited by law in many countries and is characterized as statements against groups where they are threatened or insulted on the basis of ethnicity, religion, race, or sexual orientation. As a result, large social media platforms invest great resources in developing systems to detect offensive language (AlKhamissi et al., 2023).

Our last task, social dimensions extraction (Choi et al., 2020), is the most complex one aiming at capturing social dynamics expressed in everyday language on social media. Through an extensive survey of sociology and psychology, Deri et al. (2018) compile a list of ten social dimensions capable of describing most online social relationships. These dimensions expand on the fundamental paradigm of characterizing social relationships in the context of tie strength (Granovetter, 1973). It was shown that strong ties tend to form small and close clusters, where people provide social and emotional support. Oppositely, weak ties are effective

in bridging across clusters, practically providing opportunities to find a new job for instance. In opposition to the notion of tie strength, the work of Deri et al. (2018) aims at capturing social aspects related to complex communicative acts (pragmatics) of conversation rather than semantics. Semantics focuses on the topical context of a text, where communicative acts shape the nature of relationships. By accurately identifying the social dimensions used in communicative acts, various potential downstream applications arise. For instance, which social dimensions spark opinion change in the act of collective coordination (Monti et al., 2022). Here, collective coordination can be countless topics such as politics, climate change, etc.

2.2 NATURAL LANGUAGE PROCESSING

The interdisciplinary field of natural language processing (NLP) dates back to the 50s, when Alan Turing proposed the Turing Test to assess a computer's capabilities to show human behavior (Turing, 1950). NLP combines the domain of linguistics with computer science and artificial intelligence. The earliest work was based on using prior linguistic knowledge to create rule-based systems (Chomsky, 1957). The motivation to develop generative models later introduced statistical and machine learning concepts (Manning and Schütze, 1999). These models were characterized by being able to learn from data. (Bengio et al., 2000) introduced some of the first work on neural language models. Neural models, also denoted deep learning models, were inspired by how the brain functions, and quickly showed impressive learning capabilities in complex tasks (LeCun et al., 2015). In an attempt to capture the sequential nature of language and text, recurrent neural networks like long-short-term memory (LSTM) were used in language modelling (Hochreiter and Schmidhuber, 1997; Karpathy, 2015). Using memory and forget gates when processing words, the models could understand which words to remember in order to capture the overall semantics of a text.

The technique of LSTMs was actually introduced prior to the deep learning explosion, but because of its computational complexity during training and inference, computers were practically limited in scalability and thereby also performance. The use of GPUs rather than CPUs to train and use neural networks was not a particularly common practice. GPU as a hardware resource to train models gained momentum in 2012 when AlexNet (Krizhevsky et al., 2012) won the ImageNet challenge showcasing the impressive performance of deep learning models and the effectiveness of GPUs to train these models efficiently.

Models like LSTMs, however, suffered from one particular critical deficit: vanishing or exploding gradients. This implies difficulties in learn-

ing long-term dependencies in texts. Motivated by this deficit, the Transformer architecture was introduced (Vaswani et al., 2017).

The Transformer model (Figure 1) is a neural network architecture capable of understanding complex language by learning which words to focus on when trying to understand the underlying semantics of a text, using the crucial concept of self-attention. The Transformer consists of two components, the encoder, and the decoder.

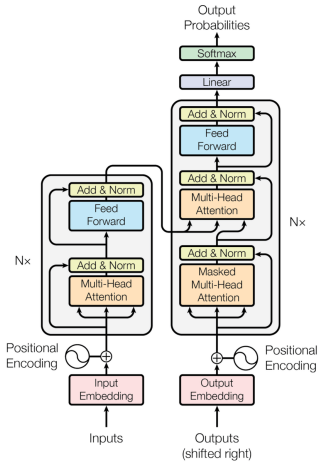

Figure 1: `The Transformer architecture`. Image is taken from Vaswani et al. (2017).

The encoder (the left block in Figure 1) takes a sequence of words and projects them into numerical continuous representations through layers of self-attention, normalization, and feed-forward neural networks. The decoder, the right block, uses the contextual representations from the encoder to pass through a similar sequence of layers and produce softmax probabilities of the most likely token given the input and already generated words. One particular addition to the decoder is the use of masked multi-head attention which asserts that the prediction of words is only based on tokens before the current token.

The attention mechanism is the backbone of the Transformer architecture and allows the model to capture long-term dependencies and determine the relevant parts of a text. More formally, the attention is calculated as a result of queries (Q), keys (K), and values (V), which are all represented as vectors.

$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V \tag{1}$$

The query vectors represent the words that must be paid attention to, and the key vectors are used to calculate the similarity between the query and key. The dot product of query and key vectors are transformed using softmax to assign importance scores between zero and one to each word. The importance scores are multiplied onto the value vectors, representing

the information about the input embeddings, ultimately resulting in a matrix representation with numerical values for the attention of all words to all other words in the sentence. Practically, each attention block is a *multi-head attention* block consisting of multiple self-attentions. It was shown that each attention-head would learn to focus on different parts of sentences, for instance, close-context or long-term dependencies. This sparked the beginning of Transformer-based language models, which are currently still the main component of state-of-the-art language models.

Following the invention of the Transformer, Bidirectional Encoder Representations from Transformers (BERT) was introduced by Google (Devlin et al., 2018). BERT showed that by stacking multiple encoder blocks, state-of-the-art performance could be achieved on a wide range of NLP tasks. Training BERT is a framework consisting of two parts, pre-training and task-specific fine-tuning. In pre-training, BERT has the objective of two unsupervised tasks, masked language modeling (MLM), and next-sentence predictions. MLM empowers a general language understanding, and next-sentence prediction gives an understanding of relationships between sentences. Following the pre-training, a classification head is added to the model, and by leveraging the general language understanding, the model is fine-tuned to a specific task in a supervised setting. These models, denoted as pre-trained language models (PLMs), have yielded state-of-the-art results in most NLP tasks.

To further improve the capabilities and language understanding of PLMs, contrastive learning has been proprosed Reimers and Gurevych (2019). Contrastive learning was initially pioneered in research on face recognition before finding widespread application in NLP (Chopra et al., 2005; Schroff et al., 2015). In the work of Reimers and Gurevych (2019), the authors propose a framework using Siamese and triplet network structures to derive semantically meaningful sentence embeddings. Using objective functions where the difference in embedding space is minimized for sentences sharing similar semantics, they show how downstream performance can be increased. Inspired by these approaches, Tunstall et al. (2022) propose SetFit, a contrastive learning framework leveraging contrastive pre-training prior to task-specific fine-tuning. They find that the additional pre-training improves performance in few-shot learning tasks.

2.3 DATA AUGMENTATION

Data augmentation is the task of artificially generating new data points through transformations. In the field of NLP, data augmentation is widely used to improve model performance across most sub-tasks (Li et al., 2022). The technique can be used to tackle the challenge of data scarcity as the effectiveness of NLP models heavily relies on both the quality and

quantity of the training data (Li et al., 2022). Data augmentation is often model agnostic without impacting the underlying model architecture used in training and inference. Existing data augmentation methods span all granularity levels, ranging from character to document level. Overall we consider 3 types of data augmentation: paraphrasing, noising, and sampling.

On both character and word levels, random interpolation or noising of words by inserting, deleting, swapping, or changing characters or words has been found to improve model robustness against noise (Wei and Zou, 2019; Belinkov and Bisk, 2018). Databases with synonyms, such as WordNet (Miller, 1992), can be used for paraphrasing by replacing randomly selected words (Pavlick et al., 2015). Hereby the semantic consistency with the input text is maintained, which is ideal in the case of classification. Wang and Yang (2015) expands their corpus in a classification task by leveraging word embeddings to replace words with their top most similar words. Another approach to augment text on word (or word-piece) level is to use masked language models (MLMs) like BERT (Devlin et al., 2018) and RoBERTa (Liu et al., 2019). By masking words, the models can predict the most likely word given the context of the input. This is denoted contextual augmentation (Kobayashi, 2018).

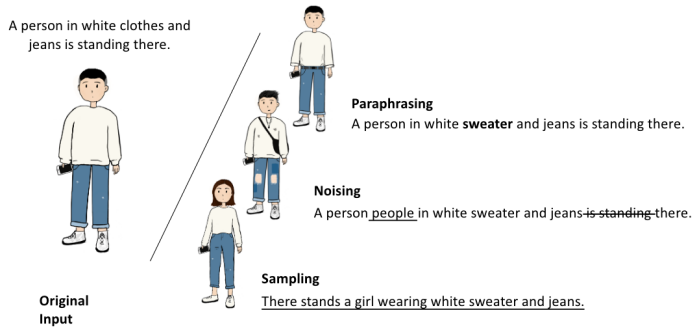

Figure 2: `Data augmentation examples`. Paraphrasing involves replacing words in a sentence and introducing lexical diversity while remaining semantically similar to the input. In noising, the input sentence is corrupted on either word or character level. Sampling involves rephrasing the input sentence. Image is taken from Li et al. (2022).

Lastly, some approaches of data augmentation work on sentence or document level where a generative model is used to create new samples. A sampling method is back-translation where a text is translated to a different language and then translated back to its original language (Sennrich et al., 2016). As a consequence of the randomness in translation, the augmented data is lexically different from the input, while semantic similarity is preserved. Recently, more sophisticated sampling methods have been proposed that leverage auto-regressive Transformer models such as GPT to perform text completion of partial text examples. Bayer et al. (2023) finetune GPT-2 on data from underrepresented classes and

samples new examples as well as completions of contextual examples using the fine-tuned model. Feng et al. (2020) also finetune GPT-2 on full Yelp-Reviews and generate new examples by performing text completion using the first 50% of words of a training text as context. Yoo et al. (2021a) frame the generation process as a few-shot prompting task. By designing a prompt that includes label and text descriptions alongside k examples, they are able to generate new samples.

Regardless of the method used for data augmentation, the goal is to generate a diverse set of new samples that are semantically similar to the original data.

2.4 LARGE LANGUAGE MODELS

In recent years, researchers have found that scaling PLMs models dramatically improve their performance (Kaplan et al., 2020), leading to the development of billion-sized models, also known as *large language models* (LLMs). These models range from "small" 7B parameter-sized models like Llama (Touvron et al., 2023) and Alpaca (Taori et al., 2023), to mid-size models like BloombergGPT (Wu et al., 2023) and Chinchilla (Hoffmann et al., 2022) with 50B and 70B parameters, up to very large models like PaLM (Chowdhery et al., 2022) with a staggering 540B parameters. LLMs are categorized as sequence-2-sequence or auto-regressive models and use the decoder to produce an output based on previously generated text. Grounded in an extensive amount of training data, these models are able to produce high-quality and coherent text output, enabling them to solve a wide range of tasks.

After pre-training, LLMs have extensive language understanding (OpenAI, 2023). However, similar to fine-tuning smaller models for specific tasks by training a classification head, large auto-regressive models can be fine-tuned to solve tasks or exhibit desired behavior based on instructions from natural language. Fine-tuning LLMs to follow natural language is denoted instruction tuning. As a consequence of LLMs' tendency to produce undesired output, e.g. inaccurate information or biased expressions (Ouyang et al., 2022), alignment tuning is proposed to regulate this behavior.

One notable technique to improve the instruction and alignment capabilities of LLMs, ultimately resulting in effective dialogue agents like ChatGPT, is Reinforcement Learning with Human Feedback (RLHF) (Ziegler et al., 2020; Ouyang et al., 2022; Lambert et al., 2022). In this approach, human evaluations of LLM output responses are used to train a separate language model, called the reward model. The reward model is then used in a reinforcement learning feedback loop to train the LLMs for optimal dialogue and produce output that aligns with human expectations.

After training with RLHF, LLMs can be used to solve various tasks using suitable prompts.

Prompts are instructions used to specify a task for LLMs, written in natural language. In-context learning (ICL) is a method used to formulate task descriptions with or without example demonstrations of the task (Brown et al., 2020). The task description and examples help steer the model into the correct context while allowing it to learn from the provided examples without any gradient update. Chain-of-Thought (CoT) prompting is an enhancement of ICL in which intermediate reasoning is added to the prompt to enhance the in-context learning. We show examples of ICL and CoT in Figure 3. In *zero-shot* setting, no examples are given in the prompt, while *few-shot* setting indicates the presence of example demonstrations.

Prompt engineering is a new discipline in the field of NLP, and the variety in which prompts can be constructed remains endless. Based on how LLMs are trained, the prompts must be designed accordingly to achieve the best possible output (Ouyang et al., 2022).

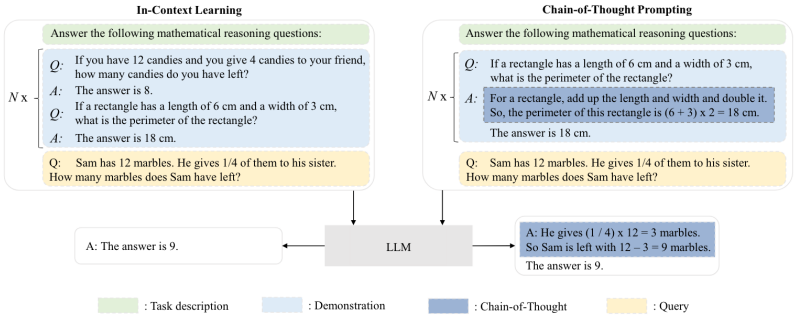

Figure 3: `Prompt Examples.` Examples of in-context learning and chain-of-thought prompting. Image is taken from Zhao et al. (2023).

The current research on LLMs is diverse. Microsoft calls it the *spark of Artificial General Intelligence* after early experiments with GPT-4 (Bubeck et al., 2023a). They demonstrate remarkable performance in a wide range of tasks and foresee models like GPT-4 (OpenAI, 2023) can make major sectors more efficient, including healthcare, education, engineering, arts, and sciences. Recently we also saw BloombergGPT (Wu et al., 2023), a 50B parameter model, that has been specifically designed and trained for the financial market. In a joined effort with OpenResearch and the University of Pennsylvania, OpenAI explored the use of LLMs and related technologies in the labor market and estimated that 80% of the U.S. workforce could have at least 10% of their work affected by the use of ChatGPT-like models (Eloundou et al., 2023). Still, large language models suffer from hallucination and toxic behavior as a consequence of the data used in the pre-training phase, and a lot of effort is put into reducing unintended output and bias (Bubeck et al., 2023b). A very recent method is red-teaming (Perez et al., 2022; Ganguli et al., 2022), an approach to find

model vulnerabilities that potentially lead to unintended behavior. The idea and goal behind red-teaming is to create prompts that trigger the LLMs to generate harmful outputs, similar to adversarial attacks (Rajani et al., 2023).

Recently, researchers have demonstrated how LLMs can be used in annotation tasks and outperform crowdworkers (Gilardi et al., 2023; He et al., 2023). These efforts highlight the potential of LLMs to automate time-consuming and labor-intensive tasks and to improve the efficiency and accuracy of annotation tasks. Gilardi et al. (2023) use a sample of 2,382 tweets to demonstrate how annotation with ChatGPT outperforms crowdworkers in four out of five tasks. He et al. (2023) propose AnnoLLM, a two-step '*explain-then-annotate*' framework for annotating text documents. In the first step, they create prompts for every demonstrated example, which is used to provide an explanation for the gold standard labels. Next, they construct few-shot chain-of-thought (Wei et al., 2023) prompts with the generated explanation. Finally, these prompts are employed to annotate the data. Dai et al. (2023) leverage ChatGPT to augment Amazon review data and data in the medical domain using simple rephrasing prompts and find increased performance using the augmented data. These approaches differ from our work, as we investigate the use of LLMs to generate new data in low-resource settings, rather than annotating existing data. In contrast to Dai et al. (2023), our prompt design includes semantic, label, and domain-preserving information as opposed to simply rephrasing.

3 DATA

In this study, we focus on the domain of online social media and use 3 distinct datasets corresponding to three different natural language processing tasks: sentiment analysis (Rosenthal et al., 2017), hate speech detection (Sigurbergsson and Derczynski, 2020), and social dimensions extraction (Choi et al., 2020). This chapter describes and motivates the 3 datasets while highlighting the challenges each task presents. In Chapter 4, we will illustrate and motivate a selection of subsets of data to be employed in the experiments.

Below we provide a brief overview of the datasets and outline the multifaceted difficulties associated with each task (Table 1). It is worth noting that all the data used are publicly available.

Task	Language	Complex latent variable	Imbalance	Low resource	Problem size
Sentiment Analysis					
Hate Speech Detection	✓		✓		
Social Dimensions Extraction		✓	✓	✓	✓

Table 1: `Task difficulties.` We describe the complexity of tasks according to a range of dimensions. Each task is marked if a certain dimension applies. The following complexity dimensions are considered: *Language*, whether the task is non-English, *complex latent variable*, whether the task involves detecting complex pragmatics, *imbalance*, whether the class distribution is heavily imbalanced, *low resource*, whether task-specific data is scarce, *problem size*, whether the task has many classes e.g. more than 5. This categorization of the task leads to a taxonomy of task difficulty.

3.1 SENTIMENT ANALYSIS

The first dataset we use in this project is the SemEval-2017 Task 4: Sentiment Analysis in Twitter (Rosenthal et al., 2017). Through the Twitter API, the authors gather tweets from various trending events at the time. These topics include *Donald Trump*, *iPhone*, *Aleppo*, *Palestine*, *Syrian Refugees*, *vegetarian*, among others. CrowdFlower is used as an annotation framework and crowdworkers were asked to indicate the overall polarity of the texts on a 5-point scale, ranging from -2 to 2. Each tweet was annotated by a minimum of five individuals, and only if three annotators agreed on a label would it be accepted. Subsequently, -2 and -1 were mapped to the label *negative*, 0 to *neutral*, and 1 and 2 to *positive*, yielding the distribution showed in Table 2.

The dataset has been preprocessed such that all users have been replaced by `@user`. We do not apply any additional preprocessing of the

	Label	Count	Fraction
Train	Negative	7,405	15.5%
	Neutral	21,542	45.2%
	Positive	18,668	39.2%
Test	Negative	3,972	32.3%
	Neutral	5,937	48.3%
	Positive	2,375	19.3%

Table 2: `Sentiment: Label distribution.` Label distributions on training and test sets on the sentiment classification task.

data. On average, the tweets contain 19.3 words, with a standard deviation of 5. Examples from the dataset can be found in Table 3.

This dataset is deliberately selected to evaluate the performance of the proposed data augmentation and classification method on well-established tasks characterized by a relatively lower level of complexity while remaining in the domain of online social communication.

Label	Example
Negative	*Beyonce needs a new sound. You may call it hating, I'm just being real.*
Neutral	*Hoping I can somehow make the Volleyball game tomorrow #please*
Positive	*Looking forward to a fun weekend in the mountains and enjoying the snow in California. Have a great Friday.*

Table 3: `Sentiment: Examples.` Examples of each of the classes in the sentiment datasets.

3.2 HATE SPEECH DETECTION

As the second dataset in our study, we use DKHATE (Sigurbergsson and Derczynski, 2020), the first Danish collection of user-generated texts containing offensive language from various online social platforms. The data consists of comments collected from the Facebook page of the Danish news media Ekstra Bladet, as well as the r/Denmark and r/DANMAG subreddits on Reddit. In gathering the texts, the authors ran a Reddit survey asking users to propose terms deemed racist, offensive, or sexist. By leveraging the 113 collected terms, they identified comments of interest that potentially could be hateful. In an iterative process of refining the annotation guidelines, crowdworkers were instructed to label texts as offensive if they included insults, threats, and profanity. Sigurbergsson and Derczynski (2020) applied the necessary preprocessing of the data to ensure the privacy of the authors. This included the replacement of

personal identification with a `@USER` tag (celebrity names excluded) and the removal of all texts with sensitive information.

Label	Example
Not offensive	*Tænk lige på de høje lønninger* *(Think about the high salaries)*
Offensive	*Indvandrere, de kender intet til dansk gerrighed og smålighed.* *(Immigrants, they know nothing about Danish greediness or pettiness)*

Table 4: `Hate Speech: Examples.` Examples of offensive and not offensive texts from the hate speech dataset.

The data is publicly released with train/test splits respectively consisting of 2,960 and 329 samples. The label distribution is highly unbalanced with only 13% of the texts being offensive. The skewed distribution persists in both the training and test splits. On average, the texts contain 19 words with a considerable standard deviation of 35.

We perceive this dataset as a valuable addition to our experimental setup, allowing us to evaluate the performance of our methods on non-English data with a high level of difficulty, despite only having two classes. More concretely, the challenges of this dataset encompass the following:

- The data is in Danish.
- There is a large class imbalance.
- The sample size is very small.

3.3 SOCIAL DIMENSIONS EXTRACTION

The final and primary classification task of our study is based on the social dimensions dataset (Choi et al., 2020) and involves the highest level of complexity. In the work by Deri et al. (2018), the authors effectively illustrate through empirical studies how the majority of social relationships can be characterized through ten distinct dimensions, as illustrated in Table 5.

By employing simple heuristics, Choi et al. (2020) obtain a large collection of Reddit data geo-referenced to the United States. In the annotation phase, the crowdworkers were presented with detailed definitions of each social dimension. The definitions are extended versions of the descriptions in Table 5. To improve the annotator's comprehension of the task, the authors provided 3 to 5 illustrative examples for each dimension. Subsequently, the crowdworkers were tasked with selecting the dimensions they perceived the text conveys, allowing for multiple dimensions to be applicable. As a consequence, the dataset inherently presents a mul-

Dimension	Description	Keywords	Example
Knowledge	Exchange of ideas or information; learning, teaching	teaching, intelligence, competent, expertise, know-how, insight	*I'm guessing if you squeeze it when empty, it creates suction.*
Power	Having power over the behavior and outcomes of another	command, control, dominance, authority, pretentious, decisions	*I think you should say "next year" one more time Drew!*
Respect	Conferring status, appreciation, gratitude, or admiration upon another	admiration, appreciation, praise, thankful, respect, honor	*First, thank you for engaging my argument seriously.*
Trust	Will of relying on the actions or judgments of another	trustworthy, honest, reliable, dependability, loyalty, faith	*Without any knowledge of you and given my own experience I assumed you were sincere.*
Support	Giving emotional or practical aid and companionship	friendly, caring, cordial, sympathy, companionship, encouragement	*I understand this is frustrating, but you need to be your own advocate here.*
Romance*	Intimacy among people with a sentimental or sexual relationship	love, sexual, intimacy, partnership, affection, emotional, couple	*In fairness, I'd have no problem telling you my preferred sex position.*
Similarity	Shared interests, motivations or outlooks	alike, compatible, equal, congenial, affinity, agreement	*But you do have a good point and I think our values are pretty similar.*
Identity	Shared sense of belonging to the same community or group	community, united, identity, cohesive, integrated	*But what really set us apart and made our community unique was your contribution.*
Fun	Experiencing leisure, laughter, and joy	funny, humor, playful, comedy, cheer, enjoy, entertaining	*My friends and i are all playing it and laughing at you all.*
Conflict	Contrast or diverging views	hatred, mistrust, tense, disappointing, betrayal, hostile	*Calling me an asshole for acknowledging my own selfishness while simply saying your own selfishness is different is childish.*

Table 5: `The dimensions of social interaction`. The keywords are the most popular terms describing the dimensions Choi et al. (2020). *Romance is removed in the preprocessing step.

tilabel classification task. Ultimately, 7,855 texts were annotated with one or more dimensions.

Label	Count	Fraction
Neutral	2,849	38.1%
Conflict	1,602	21.4%
Social Support	843	11.2%
Knowledge	816	10.9%
Respect	645	8.6%
Similarity/Identity	282	3.8%
Trust	193	2.6%
Fun	187	2.5%
Power	67	0.9%

Table 6: `Social Dimensions: Label distribution.`

For the sake of simplicity and to maintain consistency with the other datasets used in this project, we transform the task into a multiclass problem. This entails assigning a label to a text if two or more annotators have assigned it to the same class. If a text is assigned to n classes with $n > 1$ according to two or more annotations, we replicate the text n times and assign each replication to one of the n classes. Otherwise, texts that do not fulfill this criterion are assigned the label *neutral*, which is purposefully introduced. Additionally, we decide to eliminate the *romance* label, which was deemed irrelevant because it captures social dynamics that are uninteresting for the purpose of opinion change (Monti et al., 2022), which naturally would be the subsequent application. Furthermore, we consolidate *similarity* and *identity* as they inherently share similar semantics. This leaves us with 7,484 texts with a highly unbalanced label distribution (see Table 5).

The dataset contains both a sentence highlighted to contain the social dimension and the same sentence including its surrounding context. Following the approach in Choi et al. (2020), we decide to only use the high-

Full text	Highlighted text
You're looking for a fight aren't you? "Diversity is racism" tells me that you're against my very nature and existence. Unless your name is ironic, which I doubt. Given that you used political terminology for idiots.	*"Diversity is racism" tells me that you're against my very nature and existence.*

Table 7: `Social Dimensions: Example.` Text example showing the difference between the full and highlighted text. This specific example is labeled as *conflict*.

lighted text. In Table 7 we show an example of a full text and the highlighted sentence. On average, the highlighted texts contain 14.5 words, with a standard deviation of 3.2.

This dataset is particularly interesting, as it reflects complicated phenomena of human communication and interaction. The challenges of this dataset can be summarized as follows:

- Large number of classes.
- The classes are highly complex and represent pragmatics that goes beyond standard semantic classification tasks of social media.
- The data is highly unbalanced.
- Some of the classes occur very sparsely.

4 METHODS

With the aim of improving performance in the low-resource classification tasks in the social domain, we carry out two branches of experiments. The first branch is centered on data augmentation in an attempt to tackle the data scarcity of low-resource tasks. The second involves a model-focused strategy specifically developed to accommodate few-shot learning scenarios.

Using the methods described below, we aim at understanding whether augmented data effectively can replace the traditional practice of acquiring extra data using crowdsourcing. In Figure 4, we present an overview of our experimental framework.

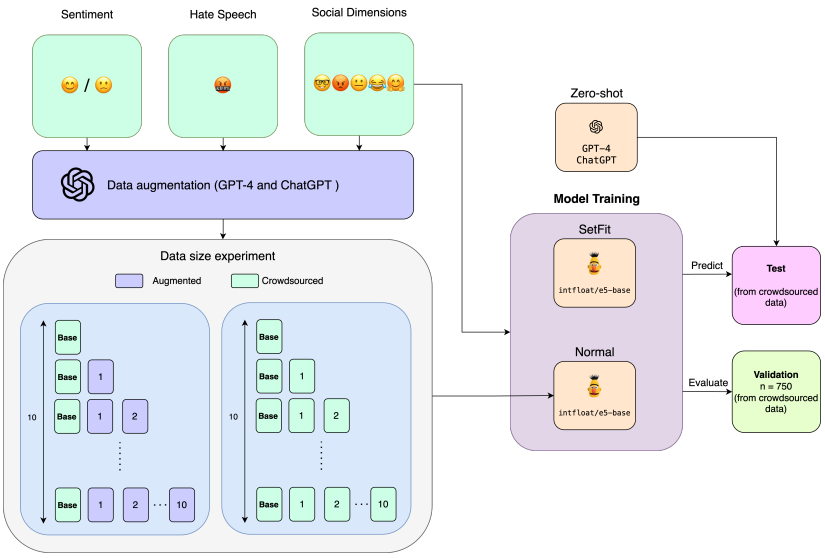

Figure 4: `Experiment framework overview.`

More concretely, the experiments we carry out are briefly highlighted below.

1. **Data augmentation using LLMs**. Based on 500 samples from our datasets, we use GPT-4 and ChatGPT to generate 5,000 augmented text samples. More specifically we employ two strategies. First, we retain the proportional label distribution of the individual datasets. Second, we oversample underrepresented classes and create datasets with balanced label distribution. Explained in section 4.1.

2. **Traditional Fine-Tuning**. We train and evaluate a small general-purpose LM on all datasets, serving as our baseline. We denote this

traditional fine-tuning as *normal* training conditions. Explained in section 4.2.

3. **Data size experiment**. We test whether performance scales with the number of samples used in training, comparing crowdsourced and augmented data with different label distributions. We use progressively larger sample sizes in training by adding 500 extra in each iteration. In every iteration, we perform normal LM training. Explained in section 4.3.

4. **Few-shot learning using SetFit**. We experiment with a few-shot learning strategy, employing contrastive pre-training prior to fine-tuning our LM. This method is selected in an attempt to tackle low-resource classification settings with data scarcity. The method is only employed on the social dimensions dataset. Explained in section 4.4.

5. **Zero-shot using LLMs**. We test the capabilities of GPT-4 and ChatGPT in a zero-shot classification setting, where we provide only a brief explanation of the task. This experiment will explore LLM capabilities to solve complex tasks, and whether specific fine-tuning of smaller models remains a viable solution. Explained in section 4.5.

All code can be found in our GitHub repository[1], and all training logs are tracked using Weights & Biases (Biewald, 2020). When considering practical feasibility and available hardware resources, we decide to only run SetFit experiments on the social dimensions dataset. This is due to our interest in its higher complexity compared to the other datasets. We augment data, run the data size experiment, and perform zero-shot classification for all datasets.

4.1 DATA AUGMENTATION

For augmentation, we use OpenAI's GPT-4 OpenAI (2023) and ChatGPT models. The OpenAI API facilitates requests to GPT-4 and ChatGPT. We use the `langchain` LLM wrapper for python as our framework for prompting. It is important to note that the models undergo constant development. The versions used in this project correspond to those available as of March and April 2023.

We leverage the two generative models to augment data based on input examples. For all our datasets, we randomly select 500 samples, constituting our *base* set. We employ two distinct strategies for data augmentation. The first maintains a proportional label distribution, while the second aims at balancing the distribution by oversampling from underrepresented classes. The first strategy seeks to answer whether more

[1] https://github.com/AGMoller/worker_vs_gpt/

data lead to higher performance regardless of label distribution. The second strategy is employed by the motivation of encountering the uneven label distribution, particularly characterizing the hate speech and social dimensions datasets. Selected classes might inevitably underperform as a consequence of insufficient data, and we seek to explore whether an oversampling technique for augmentation could effectively solve the problem of unbalanced data.

For every sample in the base set, we prompt an LLM to generate 10 new samples semantically similar to the input text. In the case of the balanced data, we pick multiple samples from the same input. To prevent the generation of multiple identical samples, we introduce a degree of randomness in the output by setting the temperature parameter to 1. The parameter ranges from 0 to 2, where 0 is close to deterministic and a high value implies more stochasticity in the output. High values effectively mean that the LLMs are prone to hallucinate and produce random output. From small empirical tests, we found 1 to most consistently generate augmented data that was semantically similar to the input, while not producing identical samples for the same base example. Conversely, for the proportional label distribution strategy, the temperature was set to 0, this was done for replication purposes to ensure close to deterministic behavior in the generation.

Ultimately, we generate augmented data using both ChatGPT and GPT-4, employing the two strategies. This results in four distinct collections of augmented data for each of the three datasets. See Figure 5 for examples.

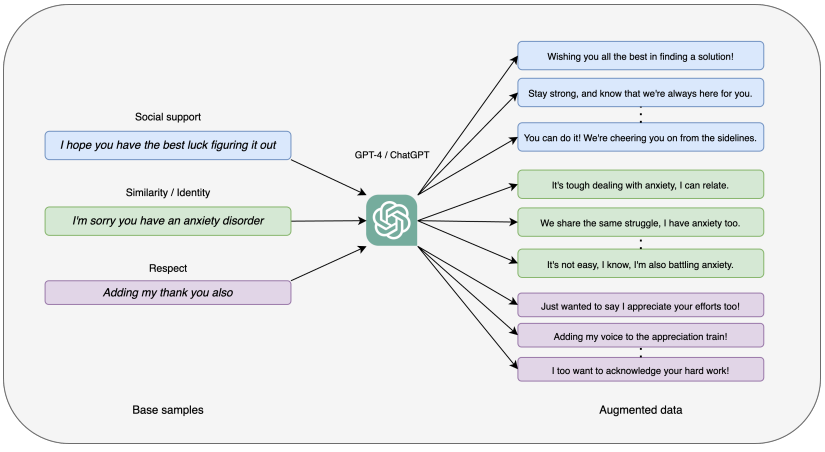

Figure 5: `Data augmentation.` Using the base samples, we prompt GPT-4 and ChatGPT to generate semantically similar samples based on an input text. The examples are taken from the social dimensions dataset using GPT-4.

4.1.1 Prompt design

For each dataset, we carefully construct a specific prompt to generate augmented samples. We design our prompts based on best practices at the time of exploration. Following the guidelines of Prompt Engineering Guide[2] and principles of zero-shot prompting and in-context learning (Wei et al., 2022), our prompts comprise two parts, a system message or context, and an instruction.

The system message is used to guide the LLM into the correct context. This turns out to be important to create good model responses. The instruction includes the task that the model is expected to solve. Given that data augmentation using instruction-based language models is still in its infancy, it is currently ambiguous what information to include in the prompt for faithful data generation. However, we take inspiration from Yoo et al. (2021a) and Bayer et al. (2023) in our strategy by designing prompts that include semantic, label, and domain-preserving information. We implement this strategy by always including the context from the base example, the label, and the instruction to use a social media comment style. We furthermore include the instruction to write "new similar examples" to introduce lexical diversity in the augmented data, while preserving semantic content, which we hypothesize results in more efficient training examples.

```
system_message = "You are an advanced classifying AI. You are tasked
with classifying the sentiment of a text. Sentiment can be either
positive, negative or neutral."

human_message = PromptTemplate(
    input_variables=["sentiment", "text"],
    template="""Based on the following social media text which has a
    {sentiment} sentiment, write 10 new similar examples in style of
    a social media comment, that has the same sentiment. Separate
    the texts by newline.

    Text: {text}

    Answer:
    """
)

prompt = ChatPromptTemplate.from_messages([system_message,
                                           human_message])
```

Python code 4.1: Data augmentation: sentiment

In the design of the prompt for sentiment analysis augmentation (code block 4.1), we begin by specifying the task and possible labels. Next, the social media domain is stated, followed by the input text and its sentiment. Intentionally, no explicit elaboration of the labels or annotation guidelines was provided. This was deliberately decided, assuming implicitly that the models possess the necessary understanding of the task.

[2] https://www.promptingguide.ai/

```
system_message = "You are a helpful undergrad. Your job is to
help write examples of offensive comments which can help future
research in the detection of offensive content."

human_message = PromptTemplate(
    input_variables=["hate_speech", "text"],
    template="""Based on the following social media text which is
    {hate_speech} , write 10 new similar examples in style of a
    social media comment, that has the same sentiment.
    Answer in Danish.

    Text: {text}

    Answer:
    """
)

prompt = ChatPromptTemplate.from_messages([system_message,
                                           human_message])
```

Python code 4.2: Data augmentation: hate speech

In the case of hate speech data augmentation (code block 4.2), we creatively design the prompt to evade the built-in security functionality to prevent the generation of hateful text. In the system message, we instruct the model to adopt the role of a "*a helpful undergrad*" writing examples to assist in the research of offensive content detection. Subsequently, we provide the text and ask it to generate 10 new samples in Danish.

```
system_message = "You are an advanced AI writer. Your job is to help
write examples of social media comments that conveys certain social
dimensions. The social dimensions are: social support, conflict,
trust, neutral, fun, respect, knowledge, power, and
similarity/identity."

human_message = PromptTemplate(
    input_variables=[
        "social_dimension",
        "social_dimension_description",
        "text",
    ],
    template="""The following social media text conveys the social
    dimension {social_dimension}. {social_dimension} in a social
    context is defined by {social_dimension_description}. Write 10
    new semantically similar examples in style of a social media
    comment, that show the same intent and social dimension.

    Text: {text}

    Answer:
    """
)

prompt = ChatPromptTemplate.from_messages([system_message,
                                           human_message])
```

Python code 4.3: Data augmentation: social dimensions

In code block 4.3 we display the prompt used to augment data for the social dimensions data. The system prompt explicitly provides infor-

mation about the domain of online social media, while also specifying the objective of conveying a social dimension. In the instruction, the social dimension of the input text is specified along with the description of the dimension, as found in Table 5. This is opposed to the augmentation prompts for sentiment and hate speech where we assume implicit knowledge about the task.

In the discussion (Chapter 6), we will discuss prompt engineering and the potential misuse of large language models to produce harmful and incorrect output. Across all our augmentation prompts, we intentionally want to create simple prompt templates with the intention of maximizing generalizability.

4.2 LM TRAINING

4.2.1 Model Selection

In our experiments, we decide to only employ a single model architecture. To ensure the selection of the most suitable model, we consider 3 important aspects. First, the model should be a general-purpose model with demonstrated capabilities of transferability to a wide range of tasks, ideally in the domain of CSS. Second, the model should be sufficiently small that it is practically feasible to train in multiple iterations. We aim for models with approximately the same size as `bert-base`, which consists of ~110M parameters (Devlin et al., 2018). Lastly, the model should display good multilingual capabilities, enabling a solid general understanding of both the English and Danish languages.

We guide our selection process on the work of *MTEB: Massive Text Embedding Benchmark* (Muennighoff et al., 2023). The authors test numerous models on various tasks and datasets in order to establish a comprehensive benchmark of general-purpose embedding models. Specifically on tasks closely related to ours, such as toxicity classification and tweets sentiment classification, the `E5-base` (Wang et al., 2022) model performed the best while aligning with all three requirements.

`E5-base` is initialized with a `bert-base-uncased` model. Wang et al. (2022) apply a contrastive pre-training, using a collection of text pairs. They seek to maximize the similarity in embedding space between positive pairs of text while maximizing dissimilarity to negative or irrelevant texts. More concretely, the loss function is defined as follows:

$$\min \mathcal{L}_{\text{cont}} = -\frac{1}{n} \sum_i \log \frac{e^{s_\theta(q_i, p_i)}}{e^{s_\theta(q_i, p_i)} + \sum_j e^{s_\theta\left(q_i, p_{ij}^-\right)}} \tag{2}$$

where $s_\theta(q_i, p_j)$ is a scoring function between a positive query and passage pair parameterized by θ, which is a scaling parameter. p_{ij}^- indicates a negative passage. The scoring function is given by:

$$s_\theta(p, q) = \cos\left(\mathbf{E}_q, \mathbf{E}_p\right) / \tau \tag{3}$$

$\mathbf{E}_q$ $\mathbf{E}_p$ are mean-poolings of the output layer, which yield a fixed-sized embedding. τ is set to 0.01 in their experiments by default. With this pre-training followed by task-specific fine-tuning, the authors find great improvement on a wide range of tasks.

4.2.2 Traditional Fine-Tuning

We use the Hugging Face `Trainer` interface to fine-tune `E5-base`. We train the model for 10 epochs using a batch size of 32. We employ AdamW (Loshchilov and Hutter, 2019) as an optimizer with a learning rate of $2e-5$ to minimize the cross-entropy loss:

$$\mathcal{L}_{\text{CE}}(y, \hat{y}) = -\sum_{i=1}^{C} y_i \log(\hat{y}_i) \tag{4}$$

Here y denotes the true label, while $\hat{y}$ is the predicted label for each class $c \in C$. The evaluation performance for every epoch iteration is tracked, and the checkpoint with the lowest validation loss is selected and used to evaluate the test set.

4.3 DATA SIZE EXPERIMENT

This experiment aims to assess the quality of the augmented data in comparison to the crowdsourced data, while also evaluating whether performance scales with data size. It is worth noting, that we only perform traditional fine-tuning in this particular experiment. This was decided due to practical feasibility.

Our pool of data consists of crowdsourced data along with the four augmented datasets. For each dataset, we fine-tune `E5-base` using a progressively larger sample size. We begin with the 500 base samples used for augmentation, and in each iteration, we randomly select 500 extra texts. By progressively increasing the data size from 500 to 5,000 samples, we can evaluate the scalability of performance in relation to the quantity of available training data. In each iteration, we follow the procedure outlined in 4.2.2.

We test the performance using 20% of the original crowdsourced data. We also reserve 750 samples to be used as validation set. The final model is selected based on the highest performance on validation.

4.4 FEW-SHOT LEARNING USING SETFIT

In an attempt to tackle the challenge of data scarcity, we test the SetFit (Sentence Transformer Finetuning) framework (Tunstall et al., 2022) on the social dimensions dataset. Due to limited time and resources, we restricted ourselves to only one dataset and opted for the dataset with the highest degree of complexity.

SetFit is created to address the shortcomings of LLMs including variability, prompt creation, and intense hardware consumption. SetFit works in a 2-phase setting. First, contrastive learning is applied, subsequently followed by traditional training of the classification head.

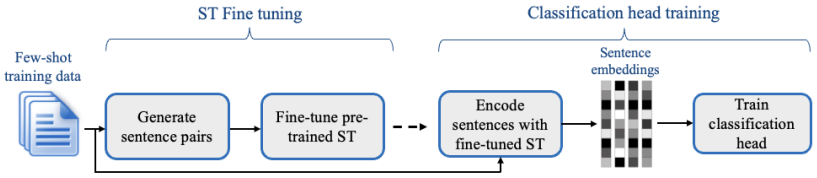

Figure 6: `SetFit Framework`. Visualisation is from Tunstall et al. (2022).

In the first phase, contrastive learning is used to better use the limited amount of data. More formally, we are given a dataset $D = (\{x_i, y_i\})$ where x_i and y_i are a text and its corresponding label. For each sample in class $c \in C$, we generate R double triplets. In the first positive triplet, $T_p^c = \{(x_i, x_j, 1)\}$, x_i and x_j belong to the same class ($y_i = y_j = c$). Oppositely, the samples in the second triplet $T_n^c = \{(x_i, x_j, 0)\}$ do not belong to the same class ($y_i = c, y_j \neq c$). Ultimately, this results in a dataset of size $2 \cdot R \cdot |D|$ to be used in contrastive learning. The contrastive learning follows the same procedure as the pre-training of `E5-based` described in equation 2 from Section 4.2.1. We follow the recommended hyperparameter setting suggested in (Tunstall et al., 2022) and on their GitHub repository[3], setting $R = 20$, number of epochs to 1, and a learning-rate of $1e-5$.

In the second step, the parameters of the body of the model are frozen. Instead, we fine-tune the classification head with a size corresponding to the number of classes. We use a lower learning rate than suggested by the authors ($1e-5$ instead of $1e-2$) as we empirically found the latter to be unable to converge. The model is trained for 20 epochs.

4.5 ZERO-SHOT CLASSIFICATION

We employ zero-shot classification to evaluate the base performance and general language understanding of LLMs. This will allow us to assess whether there is an actual need for specific fine-tuned models. Zero-

[3] https://github.com/huggingface/setfit

shot classification is a technique that involves a model classifying text instances into classes that it explicitly has not been exposed to or trained on. The method allows us to evaluate the general knowledge of the model and its ability to leverage the vast amount of data the model has been trained on. It is plausible that the models have been instruction-tuned on standard tasks similar to sentiment classification which could result in unfair performance for the specific problem. However, it is important to note that the training data and details of ChatGPT and GPT-4 have not been publicly released.

Opposed to traditional language models with a classification output head, LLMs are generative models outputting a sequence of text. This implies that the model must be explicitly guided on how to correctly output the generated text to allow for correct mappings to the true labels. Additionally, LLMs are susceptible to hallucinations, implying that they may introduce new classes that we have not explicitly specified.

In line with the prompts used for data augmentation, we provide a minimal explanation of the classes associated with the different tasks.

For the sentiment classification task (code block 4.4), our prompt closely follows the same format as used in the data augmentation. As such, we assume that the context of sentiment and the provided labels distill sufficient information about the task, thus requiring no detailed description.

```
system_message = "You are an advanced classifying AI.
You are tasked with classifying the sentiment of a text.
Sentiment can be either positive, negative or neutral."

human_message = PromptTemplate(
    input_variables=["text"],
    template="""Classify the following social media comment
    into either "negative", "neutral" or "positive". Your
    answer MUST be either one of ["negative", "neutral",
    "positive"]. Your answer must be lowercased.

    Text: {text}

    Answer:
    """
)

prompt = ChatPromptTemplate.from_messages([system_message,
                                           human_message])
```

Python code 4.4: Zero-shot: sentiment

In code block 4.5 we show the prompt used for hate speech detection. It is worth noting that GPT-4 and ChatGPT display a good understanding of harmful text as the models are mostly able to reject harmful and offensive instructions.

```
system_message = "You are an advanced classifying AI.
You are tasked with classifying whether a text is offensive
or not."

```

```
human_message = PromptTemplate(
    input_variables=["text"],
    template="""The following is a comment on a social
    media post. Classify whether the post is offensive
    (OFF) or not (NOT). Your answer must be one of ["OFF",
    "NOT"].

    Text: {text}

    Answer:
    """
)

prompt = ChatPromptTemplate.from_messages([system_message,
                                           human_message])
```

Python code 4.5: Zero-shot: hate speech

Code block 4.6 shows the prompt used to classify the social dimensions data. First, we specify that the model must act as an advanced classifying AI. Next, we list the classes and finally the text.

```
system_message = "You are an advanced classifying AI.
You are tasked with classifying the social dimension of
a text. The social dimensions are: social support, conflict,
trust, neutral, fun, respect, knowledge, power,
and similarity/identity."

human_message = PromptTemplate(
    input_variables=[
        "text",
    ],
    template="""Based on the following social media text, classify
    the social dimension of the text. You answer MUST only be one
    of the social dimensions. Your answer MUST be exactly one of
    ["social_support", "conflict", "trust", "neutral", "fun",
    "respect", "knowledge", "power", "similarity_identity"].
    The answer must be lowercased.

    Text: {text}

    Answer:
    """
)

prompt = ChatPromptTemplate.from_messages([system_message,
                                           human_message])
```

Python code 4.6: Zero-shot: social dimensions

All prompts created are experimentally found with inspiration from available guidelines. In most cases, they produced the desired outcome with correct formatting and a high degree of quality. However, in order to achieve maximum performance they could be refined. We discuss this further in Chapter 6.

4.6 EVALUATION

In this section, we describe the evaluation metrics employed throughout the project, specifically macro F1 and accuracy. These metrics are deliberately selected to provide distinct perspectives of our models' performance across tasks with varying label distributions.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad \text{F1}_{\text{macro}} = \frac{1}{C} \sum_{i=1}^{C} \frac{2 \cdot \text{Precision}_i \cdot \text{Recall}_i}{\text{Precision}_i + \text{Recall}_i} \tag{5}$$

Macro F1 is selected as it provides the average in F1 scores across all classes, without accounting for class imbalance. Treating all classes equally, a low performance in classes with only a few examples will have a substantial impact on the overall F1. Consequently, highly unbalanced datasets may yield low macro F1 scores. Oppositely, accuracy shows the fraction of correctly classified texts, irrespective of the specific class labels.

Using these two metrics, we can directly assess how well the model is performing overall while considering inherent challenges associated with certain difficult classes. In addition to accuracy and macro F1, we also compute a classification report on the test set.

4.6.1 Data augmentation evaluation

We designed our prompts with the goal of producing semantically similar augmented data in terms of preserving the label and style of the base examples while introducing lexical diversity. In order to evaluate the quality of our prompt design and augmented data, we choose to focus on the social dimensions dataset, as we want to more thoroughly understand the complexity of this task. We compute metrics of semantic similarity and lexical diversity to describe each of the augmented datasets. This will inform us about potentially useful updates for our prompt. We use two distinct metrics for the evaluation, cosine similarity to inspect the semantics similarity, and BLEU for lexical diversity.

Cosine similarity is a measure that quantifies the similarity between two vectors. It is computed as the dot product of the two vectors, normalized by the product of their Euclidean norms. Cosine similarity indicates how close two texts are represented in embedding space, taking values between -1 and 1. -1 indicates a complete dissimilarity between texts, 0 is non-related, while 1 means that the two sentences are completely identical. Sentences that are semantically similar, yet not lexically identical, will have a high cosine similarity. Sentences are embedded using the pre-trained `E5-base` model.

$$\text{cosine similarity} = \frac{A \cdot B}{\|A\| \cdot \|B\|} \tag{6}$$

The traditional BLEU metric Papineni et al. (2002) is typically used in machine translation. BLEU measures the overlap in n-grams between an output and a reference text and is given by

$$\text{BLEU} = \text{BP} \times \exp\left(\sum_{n=1}^{N} w_n \log \text{precision}_n\right) \tag{7}$$

where BP is a penalty term, typically set to 1 unless the generated sample is smaller than the reference, N denotes up to how many n-grams are used, w_n is a uniform weighting, and precision_n is the n-gram precision.

BLEU provides an indication of the diversity of the generated text, ranging from 0 to 1. A BLEU of 0 indicates a higher diversity as a result of no overlap between n-grams, while a score of 1 means complete overlap in n-grams.

For each of the two metrics, we calculate inter and intra-similarity (Figure 7). Inter-similarity denotes the similarity between a base sample and its generated sample. Intra-similarity is the mean similarity between a generated example and the 9 remaining augmented samples from the same base text. We hypothesize that efficiently generated data examples should exhibit high semantic similarity to their base example while being as lexically diverse.

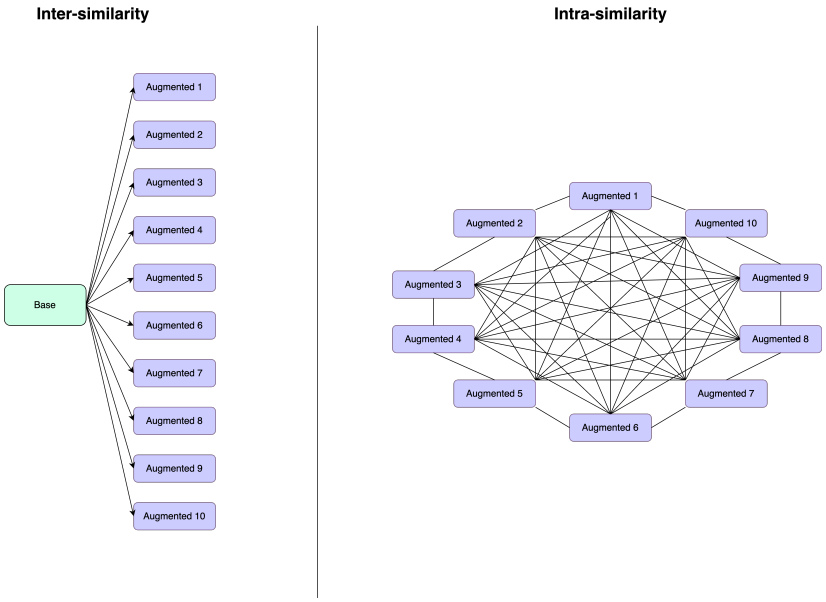

Figure 7: `Inter and intra similarity.`

We also employ the t-SNE algorithm (Maaten and Hinton, 2008) for a qualitative assessment of our data and put it into the perspective of selected individual class performances.

5 RESULTS

This chapter describes the results obtained from all experiments. Detailed training, evaluation, and test performance for normal LM training, zero-shot classification, and SetFit can be found in our Weights & Bias projects[1].

5.1 DATA SIZE EXPERIMENT

In this section, we present the results from the data size experiment explained in 4.3. For comparison, we include the results obtained from zero-shot classification. We display the results and compare the performance of the three tasks on both crowdsourced and augmented datasets. To summarize, for each task we seek to answer the following questions:

- How does performance scale with training size for crowdsourced data compared to augmented data?
- Do we observe a difference in performance between ChatGPT and GPT-4 augmented datasets?
- Is there a discrepancy in performance between the two augmentation strategies, particularly on tasks with uneven label distribution (hate speech detection and social dimensions extraction)?

5.1.1 Sentiment Analysis

We begin by presenting the sentiment analysis task, which is the least complex. Figure 8 shows performance results, and we observe that the augmented data underperforms significantly compared to the crowdsourced. As the crowdsourced data is close to being evenly balanced, all the augmented datasets have similar distributions and achieve comparable performance with each other. The crowdsourced data performs close to equal with zero-shot ChatGPT after being trained on 2,000 training samples.

GPT-4 in a zero-shot setting achieves the highest performance with a macro F1 score and accuracy of 0.71. GPT-4 is best performing on the negative label with an F1-score of 0.76, primarily as a result of a high recall of 0.85. The *neutral* label, however, is the most challenging class for the

[1] Normal training and zero-shot: https://wandb.ai/cocoons/worker_vs_gpt/
SetFit: https://wandb.ai/cocoons/social-dim-setfit/

zero-shot approach, achieving an F1 score of 0.68. The full classification report of zero-shot GPT-4 is presented in Table 13 in the Appendix.

We observe little to no improvement for the crowdsourced model beyond a sample size of 2,000 and a slight correlation between sample size and performance for the augmented data models.

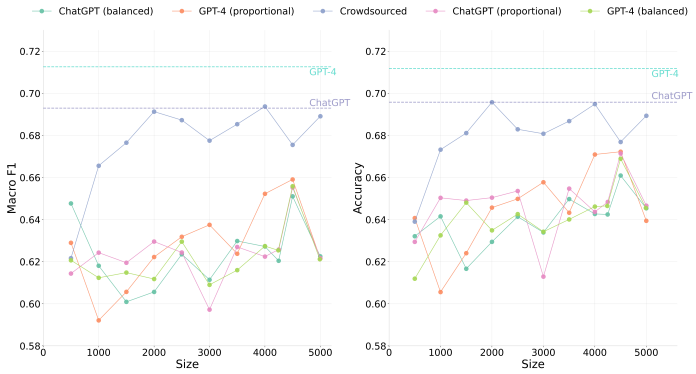

Figure 8: `Sentiment: Data size experiment.` The Figure displays the macro F1 score and accuracy on the test set. The dashed lines represent the zero-shot performance on the test set for the two zero-shot models, ChatGPT and GPT-4. We observe a superior performance using the zero-shot, specifically GPT-4 outperforms the other approaches. We observe a substantial gap between crowdsourced and the augmented datasets.

Overall, we find the crowdsourced dataset to outperform the augmented datasets regardless of data size. All the augmented datasets yielded most comparable performance to crowdsourced when using 4,500 samples. We do not find any particular augmentation model or strategy to be substantially suitable. As the crowdsourced data exhibit a balanced label distribution, we find no distinct difference between the two label strategies. The two LLMs showcased impressive zero-shot capabilities. Particularly GPT-4, which beats ChatGPT by an absolute margin of 0.02 F1 and 0.016 accuracy.

5.1.2 Hate Speech Detection

Our second task involves hate speech detection, which adds an additional layer of complexity due to the Danish language. Moreover, the dataset is relatively small given its 2,460 training samples and has an unbalanced label distribution. We find that training the model on 2,000 samples from the crowdsourced data results in a macro F1 score of 0.76 and an accuracy of 0.92 (Figure 9). However, due to the large class imbalance, correctly classifying the offensive label remains particularly challenging. This model achieves high precision of 0.895 in predicting offensive language but suffers from the low recall of 0.415, resulting in an F1-score of 0.567 on the 41 samples in the test set (see Table 14 in Appendix). For the

augmented data, we observe little improvement beyond the first 1,000 samples except for the proportional ChatGPT data.

We also evaluate the zero-shot performance of GPT-4, which shows a competitive macro F1-score of 0.72. However, compared to the trained models, GPT-4 slightly underperforms in accuracy. Interestingly, we find that ChatGPT achieves a lower accuracy than all other models. This is due to ChatGPT's tendency to predict texts as containing offensive language, resulting in a high recall of 0.854 but a very low precision of 0.324. Compared to its successor, ChatGPT is substantially more likely to predict text as being offensive.

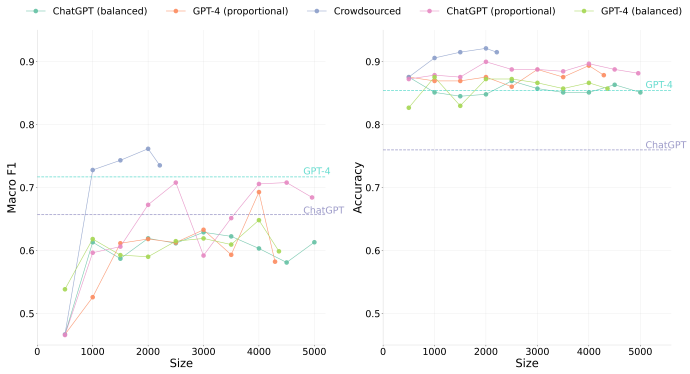

Figure 9: `Hate Speech: Data size experiment.` Macro F1 scores on the test set for different training sample sizes. In this highly unbalanced dataset, we find the crowdsourced model to be superior to both the zero-shot models and the models trained with augmented data. Interestingly, we find that balancing the label distribution does not increase the performance while keeping the label distribution proportional performs slightly better. Comparing the two zero-shot approaches, we find GPT-4 to be superior.

Generally, we observed the crowdsourced data to perform best, despite fewer samples than all the augmented datasets. Augmenting data with ChatGPT using the proportional label strategy yields the most comparable performance, while the 3 other augmented datasets perform equally. GPT-4 in a zero-shot setting obtains better F1 than all models trained on augmented data, but a lower score than the crowdsourced. We also find GPT-4 to substantially outperform ChatGPT.

5.1.3 Social Dimensions Extraction

Our third involves social dimensions classification, which is the most challenging problem due to a large number of complex and highly unbalanced classes. Figure 10 presents the results. We observe that zero-shot ChatGPT achieves the highest macro F1 score of 0.321, closely followed by GPT-4 with a score of 0.304. Notably, both zero-shot approaches achieve higher macro F1 scores than any trained model, primarily due to

their ability to correctly classify the *power* class, which is the most underrepresented class in the task and only occurs 13 times in the test set. ChatGPT correctly classifies only 2 out of the 13 examples, but since the macro F1 score does not weigh by occurrence, it has a considerable impact on the overall F1 score (see classification report for ChatGPT in Table 15 in the Appendix).

We observe that the balanced data strategies are significantly more sample efficient for training the models as we see a stark increase in F1 for models using up to 2,000 samples. This difference in F1 seems to equalize at 5,000 samples using the other strategy. For accuracy, the trend is very similar albeit the balanced strategies perform a little worse. Moreover, we find that 4,500 augmented texts from GPT-4 and ChatGPT using proportional label distribution outperform the zero-shot approaches in accuracy. The highest accuracy is achieved using 5,000 samples of crowdsourced data, yielding score of 0.405.

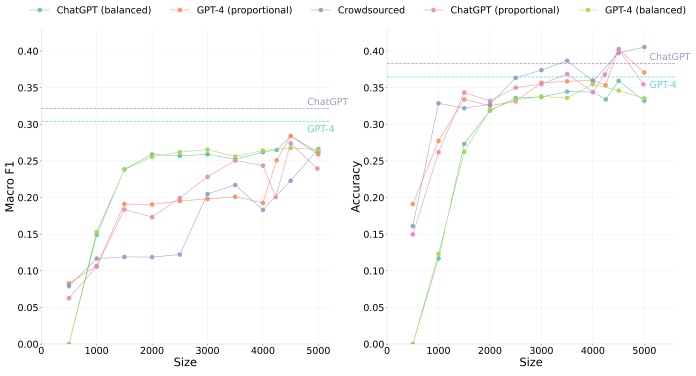

Figure 10: Social Dimensions: Data size experiment. The F1-scores indicate ChatGPT (balanced) and GPT-4 (balanced) consistently outperform the other trained models from 1,000 to 3,000 training samples. Notably, all models perform comparably when using the full set of 5,000 samples, except for ChatGPT (proportional), which exhibits slightly lower performance with an F1 score of 0.24. Both zero-shot approaches outperform any trained model, with ChatGPT achieving the highest F1 score of 0.32. In the case of accuracy, using all 5,000 samples from the crowdsourced data yields the best overall score of 0.405.

Overall we find the two augmented datasets using balanced strategies to achieve superior macro F1 scores when using up to 3,000 training samples, not considering zero-shot. However, this comes with a cost of lower accuracy. We do not find any model-specific differences in performance between GPT-4 and ChatGPT, while the discrepancy between strategies is more apparent. Crowdsourced data achieve the highest accuracy when using all 5,000 samples. The two zero-shot approaches obtain the highest macro F1 scores due to their ability to make correct predictions on the difficult *power* class, with ChatGPT being slightly better than GPT-4.

5.2 CLASSIFICATION PERFORMANCE ON SOCIAL DIMENSIONS

In this section, we present the results and findings obtained from the experiments explained in section 4.2, 4.4, and 4.5. We find it relevant to directly compare normal training, SetFit, and zero-shot classification performance. To summarize, the questions we seek to answer are:

- What is the performance using the crowdsourced data, and how does it compare to augmented?
- Does contrastive pre-training lead to an uplift in performance?
- Do we observe an improvement in performance when the crowdsourced and augmented datasets are combined?
- How does the performance vary across the individual classes, and which classes pose a greater challenge?

5.2.1 Overall Performance Assessment

Table 8 provides the macro F1 and accuracy scores derived from the test set. The table shows that using only a single set of data, the crowdsourced data achieves the highest F1 under normal training conditions, substantially beating the best augmented dataset, GPT-4 balanced, by a margin of 0.053. As the only individual dataset using normal training, the crowdsourced data beat GPT-4 in a zero-shot setting, while ChatGPT surpass all the augmented individual and normally trained datasets.

When applying the contrastive pre-training using SetFit, we observe a substantial increase in performance across all the individual augmented datasets. Surprisingly, the crowdsourced data decrease considerably in macro F1 by 0.04. The most substantial increase is observed for the two ChatGPT augmented datasets, which attain F1 scores of 0.321 and 0.342 for the proportional and balanced sets respectively. That is an absolute difference in F1 of an impressive 0.082 and 0.08. In particular, ChatGPT balanced has increased accuracy from 0.322 to 0.434, a notable improvement of 0.112, when using SetFit compared to normal training. This improvement is especially observed in the classes *trust*, *neutral*, *respect*, and *similarity/identity* where the F1 scores are increased with 0.228, 0.179, 0.106, and 0.189 respectively (see Tables 16 and 17 in Appendix).

Generally, we observe from the experiments with the individual datasets that using a balanced label distribution is better than maintaining it proportional. When considering GPT-4 augmented data, under normal training conditions the model achieves an F1 score 0.006 higher with a balanced label distribution compared to the proportional. Using SetFit, the difference is 0.003. The same pattern is also evident for ChatGPT with

	Macro F1			Accuracy		
	Normal	SetFit	δ	Normal	SetFit	δ
Individual						
Crowdsourced (C)	0.315	0.275	−0.04	0.474	0.428	−0.046
GPT-4 proportional	0.259	0.302	+0.043	0.371	0.433	+0.062
GPT-4 balanced	0.265	0.305	+0.04	0.335	0.432	+0.097
ChatGPT proportional	0.239	0.321	+0.082	0.355	0.418	+0.063
ChatGPT balanced	0.262	**0.342**	+0.08	0.322	**0.434**	+0.112
Combination						
GPT-4 proportional + C	0.314	0.314	0.0	0.418	0.446	+0.028
GPT-4 balanced + C	**0.347**	0.323	−0.024	**0.487**	0.436	−0.051
ChatGPT proportional + C	0.346	0.343	−0.003	0.445	0.437	−0.008
ChatGPT balanced + C	**0.347**	0.337	−0.01	**0.487**	0.436	−0.051
Zero-shot						
GPT-4	0.304			0.365		
ChatGPT	**0.321**			**0.383**		

Table 8: `Social Dimensions: Normal and SetFit performances.` Performance scores on the test-set. We remind that there is no training involved in the zero-shot classification and that they serve as a comparison to the trained models. Individual indicates that the model has been trained on an individual dataset, while combination is a concatenation of augmented and crowdsourced data. δ denotes the difference between normal and SetFit, and the **bold** numbers indicate the highest score in each section.

absolute differences of 0.023 and 0.021 for normal and SetFit training respectively when comparing proportional and balanced.

When combining the crowdsourced data with a set of augmented data, we observe a leap in performance using normal training. In particular, the two balanced datasets combined individually with the crowdsourced data achieve the highest macro F1-score across all experiments, 0.347. They are closely followed by ChatGPT proportional plus crowdsourced on 0.346. Interestingly and in opposition to the individual datasets, neither of the combinations increases performance using SetFit. Particularly, the combined GPT-4 balanced and crowdsourced data has decreased performance by a substantial margin of 0.024.

These results display several findings. First, with just normal training using a single dataset, the crowdsourced data achieves the highest performance. However, when combining augmented data with crowdsourced, we find a leap in capabilities. Second, SetFit works particularly well with individual augmented datasets, making up for the data scarcity. In contrast, SetFit yields lower performance when combining datasets compared to normal training. Third, balancing the label distribution appears to have a slightly positive impact compared to keeping a proportional distribution. Fourth, particularly ChatGPT shows great zero-shot capabilities, better than any individual augmented dataset trained under normal conditions, but is less capable than balanced ChatGPT augmented data trained using SetFit. Zero-shot classification also underperforms compared to normal training when adding the crowdsourced data to the augmented.

5.2.2 The effects of SetFit for crowdsourced models

In Figure 11, we report the confusion matrices on the test set of the 2 baseline models, normal crowdsourced and SetFit crowdsourced. We report this as we find worse performance when using SetFit compared to normal training. This contradicts the hypothesis that the contrastive learning objective should lead to better performance on low-resource tasks. From the confusion matrices, we find that neither models manage to learn a useful representation of the *power* and *trust* dimensions. However, we find that the normal model learns a better representation of the most frequent classes, *neutral* and *conflict* than the SetFit model. The normal model is also more likely to conflate any dimension for *neutral*, except for *conflict*. In return, the normal model confuses *neutral* for *conflict* more than SetFit. Interestingly, the SetFit model performs better on the *social support* dimension than the normal model, however, it never predicts *respect* conflating it for *social support*. Unlike the normal model, the Setfit model learns to classify some *similarity/identity* examples correctly.

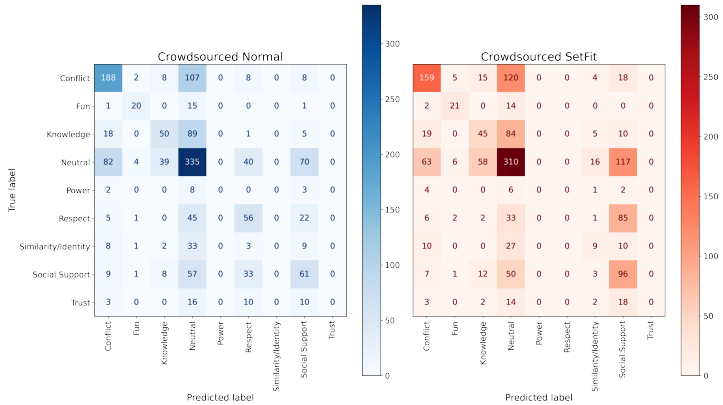

Figure 11: `Social Dimensions: Confusion matrix` The figure shows the confusion matrices of our baseline models. On the left (blue): Normal model on crowdsourced data. On the right (red): SetFit model on crowdsourced data.

5.2.3 Individual Class Assessment for Best Performing Models

In this section, we inspect individual class performances for selected trained models. More concretely, we compare the individual augmented ChatGPT balanced dataset trained with SetFit against the same combined with the crowdsourced data trained using the normal framework. As these models achieve comparable performances, we wish to evaluate the impact of the additional data and the use of SetFit on individual class levels. Table 9 shows classification reports from the two trained models.

The most notable observation is both of the models' inability to correctly predict the *power* class. Power is highly underrepresented in the crowdsourced data, but regardless of our strategy balancing the label distribution or the use of SetFit, it remains particularly difficult to correctly classify. In the confusion matrix for the two models (Figure 12), we observe how the model with normal training but with extra data only predicts the power once. However, when applying contrastive learning, *power* is predicted 25 times but interchanged with the *neutral*, *social support*, and *conflict* classes.

	ChatGPT balanced (SetFit)			**ChatGPT balanced + C (Normal)**			
	Precision	Recall	F1	Precision	Recall	F1	Support
Social support	0.377	0.272	0.326	0.357	0.391	0.373	169
Conflict	0.525	0.52	0.523	0.601	0.564	0.582	321
Trust	0.225	0.231	0.228	0.111	0.026	0.042	39
Neutral	0.470	0.504	0.49	0.482	0.640	0.550	570
Fun	0.419	0.703	0.525	0.895	0.460	0.607	37
Respect	0.362	0.426	0.392	0.409	0.419	0.414	129
Knowledge	0.42	0.258	0.319	0.480	0.215	0.297	163
Power	0.0	0.0	0.0	0.0	0.0	0.0	13
Similarity / Identity	0.269	0.321	0.293	0.5	0.179	0.263	56
Accuracy			0.434			0.487	1,497
Macro avg	0.341	0.359	0.342	0.426	0.321	0.347	1,497
Weighted avg	0.437	0.434	0.431	0.484	0.487	0.470	1,497

Table 9: `Social Dimensions: Classification report.` Classification report on the test set for the two best performing trained models using individual and combined datasets respectively.

We observe a notable difference in performance between the two models on the *trust* class. Using Setfit, the model achieves an F1 score 0.228, whereas normal training yields only 0.042 in F1. Looking at Figure 12, we note that normal training confuses *trust* for *respect* and *neutral*. Generally speaking for both models, many of the classes obtain comparable F1 scores. This occurs as a result of the harmonic mean between precision and recall. In regards to recall, SetFit scores on average 0.359, surpassing the normal training score of 0.321. In contrast, the precision of SetFit is substantially lower than normal training, with macro precision scores of 0.341 and 0.426, respectively. This discrepancy indicates that SetFit is more prone to predict classes other than the two most predominant, *neutral* and *conflict*, however less confident in the predictions. Oppositely, using the normal training, the model demonstrates fewer predictions of underrepresented classes, yet is more accurate when it does make such predictions.

From Figure 12 we find a very similar pattern of the confusion matrices in the two models. It is immediately apparent that both models struggle to separate less represented classes from the majority classes *neutral* and *conflict*. We also find that the normal model predicts *neutral* more often than SetFit. It is generally difficult for the models to separate the classes *respect*, *trust*, and *social support*. However, the tendency to confuse *social support* for *respect* and *trust* is lower for the SetFit model. Interestingly, the normal model tends to conflate *social support* with *power*.

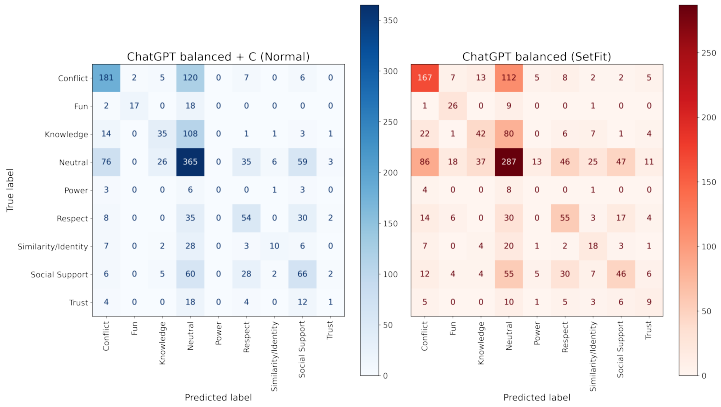

Figure 12: `Social Dimensions: Confusion matrix.` The figure shows the confusion matrix of predictions on the test set for ChatGPT balanced + C (normal) on the left and ChatGPT balanced (SetFit) on the right.

The normal model rarely predicts *trust*, but it is more likely to conflate *trust* with *social support* and *respect* than the SetFit model. Finally, we note some interesting behavior of the *power* class. For the normal model, we have no predictions for *power*, which is confused for *social support*, *conflict*, and *neutral*. The SetFit model on the other hand does have predictions of *power* although all of them are wrong. The true examples of *power* were mostly predicted to be *conflict* or *neutral*, but not *social support* as in the normal model.

5.3 DIVERSITY OF AUGMENTED DATA

In this section, we describe and characterize the diversity between crowdsourced and augmented data for the social dimensions task. As stated previously, we hypothesize that diverse data will serve as effective training samples. Table 10 presents mean and standard deviations for inter and intra similarity for all augmented datasets, while Figure 13 visually shows raincloud plots of the metrics. Generally, we find high inter cosine similarity between base samples and their augmentations. For datasets generated using ChatGPT, the inter cosine similarities are 0.794 and 0.798 for balanced and proportional respectively, while GPT-4 yields even higher similarities on 0.880 and 0.892 for the two strategies. In contrast, the augmented datasets all show a low lexical inter-similarity. Most of the BLEU scores between base samples and their augmentations are close to 0, albeit with few outliers. This implies that the augmented data is semantically very similar to the base examples while also yielding a high lexical diversity. In the discussion (chapter 6), we delve into possible

explanations that suggest low inter BLEU scores might not be ideal. Likewise, we also discuss the quality of examples with low and high semantic inter-similarity.

Dataset	Cosine similarity		BLEU	
	Inter	Intra	Inter	Intra
ChatGPT balanced	0.794 (0.06)	0.842 (0.04)	0.003 (0.03)	0.023 (0.04)
GPT-4 balanced	0.880 (0.04)	0.887 (0.03)	0.017 (0.06)	0.037 (0.05)
ChatGPT proportional	0.798 (0.06)	0.869 (0.04)	0.006 (0.04)	0.090 (0.11)
GPT-4 proportional	0.892 (0.04)	0.911 (0.03)	0.038 (0.09)	0.1 (0.1)

Table 10: `Social Dimensions: Augmentation evaluation.` Mean and standard deviations for our augmentation evaluation metrics on the four augmented datasets.

In regards to intra-similarities, we find on average all datasets to exhibit higher semantic similarity than when compared to the base samples. This means that while the augmented samples are semantically similar to the base example they are even more similar to one another. We speculate this low intra-diversity yields less efficient training samples. We also find that ChatGPT in general generates examples with lower semantic similarity compared to GPT-4. Likewise, we observe the balanced data to have a lower semantic intra-similarity compared to the proportional label distribution.

When inspecting the lexical similarities, we find the BLEU score within the augmented samples (intra) to be substantially higher than when comparing the base samples to the augmented (inter). This implies that the generated samples have a higher overlap in n-gram, leading to lower lexical diversity.

Finally, we see that the proportional strategy yields higher intra BLEU scores compared to the balanced strategy. Interestingly, our best models based on the balanced ChatGPT data display the lowest intra BLEU and the most diverse cosine similarity scores.

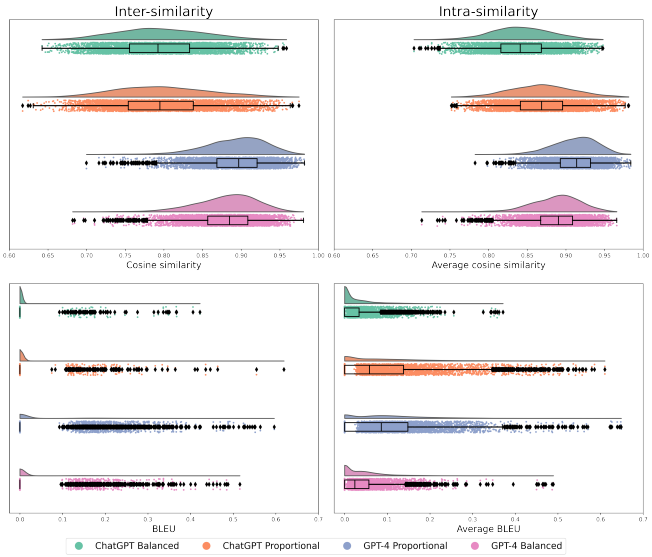

Figure 13: `Social Dimensions: Augmentation evaluation.` Raincloud plots of diversity measures of all augmented datasets. Inter-similarity is the cosine-similarity and BLEU scores between base and augmented samples. Intra-similarity measures the average similarity between an augmented sample and the 9 other, generated for a base example.

6 DISCUSSION

In this section, we critically reflect on the used methods and obtained results, and discuss potential future work.

6.1 CROWDSOURCED VS. AUGMENTED DATA

Starting with the data size experiment and the two tasks of sentiment analysis and hate speech detection, we consistently observed that the augmented data achieved lower performance than the crowdsourced one. However, for the social dimension extraction task, using 5,000 samples, the trained models achieved comparable performances, with a slight advantage to the crowdsourced data. We also observed that the zero-shot approaches obtained either the best or competitive results. Particularly in the case of sentiment analysis, GPT-4 in a zero-shot setting was superior with a notable margin. We hypothesize that the reason why ChatGPT and GPT-4 perform exceptionally well on this task can be twofold. First, the large difference in size between the LLMs compared to the standard LM could potentially allow them to learn more complex patterns and features in the data. Second, ChatGPT and GPT-4 may already have been exposed to this kind of data during training. As the training details and datasets used to fine-tune the LLMs are not publicly available, it is plausible that they have been exposed to a wide range of sentiment analysis examples, especially large public ones such as the SemEval tasks.

Using human annotators afford researchers the option to select demographically diverse crowdworkers, who can closely mimic the general perception of the tasks and classes. Constructing such a human-validated dataset additionally often involves thorough and detailed testing and evaluation of the annotators ultimately leading to a high-quality dataset. However, it is entirely possible to introduce this diversity into the prompt as LLMs are able to closely impersonate specific demographics given the right context (Argyle et al., 2022). We found the augmented social dimensions datasets to perform equal to the crowdsourced, in contrast to the two less complex tasks. We speculate that this might be because the prompt for augmenting the data was more detailed, including short descriptions of the social dimensions. This would also suggest that more careful prompt engineering might lead to better performance for less complex tasks as well. We touch upon better prompt engineering later in the discussion.

Generally speaking for data augmentation, the objective is to easily obtain more high-quality data ultimately leading to better downstream performance, but at a reduced *cost* compared to crowdworkers.

In two of the tasks, sentiment analysis and social dimensions, each sample was annotated by multiple workers to ensure high quality. In the sentiment dataset, five people annotated each text and only if at least three people agreed, the annotation was accepted. In the case of social dimensions, each text was assigned with at least three annotations, accepting a class if two agreed.

Having multiple annotations for each sample is both costly and time-consuming, as more resources are spent on each sample compared to just a single annotation. This, however, is intended to produce high-quality annotations and ultimately better downstream performance. In our data size experiments, it still remains ambiguous whether this claim actually holds, as the 3 tasks showed diverging results. On the other hand, data augmentation using LLMs is less costly in both time and money, and our experiments demonstrate great potential for LLM-augmented data, imploring further research on prompt optimization. Ultimately, it is a tradeoff between quality and quantity in annotation and data acquisition in regard to time, money, and performance. Priorities will inevitably depend on the downstream goal and application, acceptance criteria, and cost restrictions. Data augmentation with LLMs could also mitigate the harmful side effects of crowdsourcing data (Williamson, 2016).

From the quantitative analysis of diversity in the augmented data, we found the results to be encouraging. Generating examples with high semantic similarity to base examples suggests that we generated new examples likely to belong to the same class as the base example. Similarly, low lexical inter-similarity implies that we have produced data that express the same semantics while being lexically different from the base samples. We also found a higher semantic intra-similarity than inter-similarity, meaning that on average, the augmented data are semantically more similar to themselves than the base examples. While we aim to achieve semantic intra-similarity to some extent, we also hypothesize that too high semantic intra-similarity makes for less efficient training examples. In regards to lexicality, we observe a slight increase in BLEU score between inter and intra-similarity, indicating more repetition of n-grams in the sets of generated examples. For both LLMs, intra BLEU scores were higher when using the proportional strategy compared to the balanced, and we speculate it is a result of the temperature parameter of 0 used in generation. The lower temperature should imply less diverse generations. In general, we expected augmented data that exhibit high semantic similarity to the base examples, while having low lexical overlap and intra-similarity would perform the best.

We could have used other metrics to evaluate the quality of augmented data, for instance, a family of fluency metrics. The higher the

fluency, the more our examples imitate grammatically and logically correct human text. Popular fluency metrics include perplexity and syntactic log-odds ratio (SLOR) (Feng et al., 2020). However, both these metrics require us to have access to the underlying models in order to calculate it, although we might have been able to compute a surrogate measure by using a smaller language model (Kann et al., 2018; Feng et al., 2020). Fluency metrics would have added additional explainability as we would be able to more confidently say that our generations are human. However, with the capabilities of LLMs to produce human-like text, we decided against using these metrics. In terms of diversity metrics, we could have decided to compute more global descriptors of diversity. For instance, we could have identified the ratio of unique trigrams across all generations to measure diversity between all sets of generations (Tevet and Berant, 2021; Feng et al., 2020).

6.2 SEMANTIC REPRESENTATION

The extraction of social dimensions proved to be a highly challenging task due to the inherent complexity that goes beyond traditional semantics. To gain insight into the semantic meaning of each class, we visually represented the embeddings in Figure 14. It is worth noting that the *neutral* label was excluded from this visualization, but a complete visualization including the *neutral* label can be found in Figure 15 in Appendix. We did not observe any distinct emergence of clusters, which suggests a high degree of semantic similarity between the classes. However, we noticed that samples belonging to the underrepresented class *fun* exhibited semantic similarities and demonstrated good overall performance, as found in Table 9. Additionally, we found a strong semantic relationship between the *respect* and *social support* classes, making it challenging to distinguish between them. This is also supported by the confusion matrices generated from the ChatGPT balanced (SetFit) and ChatGPT balanced + C (Normal) models, as illustrated in Figure 12. Lastly, we found that the majority of samples representing the *conflict* class shared similar semantics yielding high overall performance, as outlined in Table 9.

Generally, we observed many samples being interchanged with the *neutral* class. We speculate this might happen because *neutral* encompasses many of the classes, as a result of our preprocessing. Texts which did not fulfill the requirement of at least two annotations in one or more dimensions were considered *neutral*. That means that samples with more ambiguous annotations, i.e. having a single annotation in one or more dimensions might in fact display a social dimension but failed to meet the acceptance criteria because of ambiguity.

Figure 14: `Social Dimensions: Embedding projection.` The figure shows a 2D representation of the embeddings of the crowdsourced training data about social dimensions. The embeddings are obtained using **E5-base** and reduced in dimensions using t-SNE. We removed the *neutral* label. Visible concentrations of social dimensions are annotated using the colors associated with its dimension.

6.3 FEW-SHOT LEARNING

In regards to contrastive learning, particularly on the augmented datasets irrespective of label distribution strategy, using SetFit had notably increased performance compared to normal training. This confirms that the approach of using contrastive pre-training prior to fine-tuning works well for even larger datasets than in the original work (Tunstall et al., 2022). Our experiments showed a trade-off between precision and recall when using normal training compared to SetFit. SetFit more frequently predicted less represented classes, however at the cost of a lower precision. Oppositely, normal training generally yielded higher precision but with a tendency to predict the majority classes. This is a noteworthy consideration as the selection of evaluation metrics is very task-specific and differs depending on goals and acceptance criteria.

We also speculate whether an optimization of hyperparameters would have implied additional performance gains, compared to using the rec-

ommended. We did, however, change the learning rate of the classification head from $1e-2$ to $1e-5$ which ultimately had a substantial impact.

It is difficult to directly evaluate the contrastive pre-training. Traditionally, contrastive learning is evaluated on a downstream task without any assessment of the progression of semantic understanding of the individual classes. One possible solution could be to calculate inter-class similarity during the training. If classes during training obtain lower similarity to samples in other classes, it might indicate that the model learns to understand the semantic differences between the classes. Another solution could be to calculate intra-class similarity, which potentially could indicate whether the model semantically brings samples in the same class closer in embedding space.

SetFit has proven to be a viable method to encounter few-shot scenarios where traditional fine-tuning of normal-size language models might be insufficient. Task-specific fine-tuning of large language models has become unfeasible to train on consumer hardware as a consequence of the model size. However, parameter-efficient fine-tuning (PEFT) approaches are specifically developed to address this problem (Mangrulkar and Paul, 2023). PEFT methods like LoRA (Hu et al., 2021) work by only fine-tuning a small number of additional parameters while freezing the LLM parameters. This would practically allow for task-specific training of LLMs without relying on expensive computational resources. Hugging Face has also released a PEFT library[1], which support several PEFT algorithms. For future work, we could fine-tune a "small" LLM and assess whether we can leverage the large model size and extensive pre-training in complex classification tasks.

6.4 PROMPTING

Prompt engineering is a rapidly growing field in LLM research, and prompts can be designed in numerous ways. For the data augmentation prompts, we tried to ensure label preservation, i.e. the correct label for the generated sample, by including labels, label definitions, and domain, alongside the text to be augmented. Previous work leveraging large language models to augment data have explored different strategies.

Our approach is similar to Yoo et al. (2021b), who design a prompt to generate samples based on text and label descriptors and k randomly chosen examples including labels. In our case, we are explicitly instructing the models to produce additional examples with a specific label whereas their work generates text-label pairs. Future efforts using instruction models could experiment with the k-shot style generation of Yoo et al. (2021b) as well as include the annotation task in the instruction. Additionally, Yoo et al. (2021b) were able to compute soft labels by extracting

[1] https://github.com/huggingface/peft

the normalized probability of the model generating the label-tokens for every newly generated example. This method gives more credible label preservation but requires access to the underlying language model.

We relied on semantic inter-similarity as a measure of how likely a generated example has the correct label. This is in line with other research in the field such as Bayer et al. (2023) who fine-tune an LLM on data from a specific class, generate new examples using prompt completion, and subsequently use semantic similarity to filter out examples with a certain distance from the centroid example. While fine-tuning contributes to label preservation, this approach is not suited for the social dimensions extraction task which has very sparse and highly imbalanced classes. Our approach is inspired by the work of Feng et al. (2020). The authors calculate the BERTScore (Zhang et al., 2020), a semantic similarity measure, between a prompt containing 50% of words from a given example and the generated words. They denote the measure *semantic content preservation*. Furthermore, a task-specific regressor is trained and used to measure the differences between ground truth and generated sentences. While we employ semantic similarity to measure label preservation, we cannot use the label-preserving regressor as the social dimensions extraction task is not ordinal.

Still, the evaluation of the augmented data showed very little n-gram overlap between generations and base examples. This is very different from more traditional data augmentation methods, like adding noise or word exchanges which would yield a high BLEU score. As we observe low inter BLEU scores, the augmented samples are lexically different from the base samples. To qualitatively inspect whether labels are preserved, we select examples with low, medium, and high semantic similarity. A subset of these examples can be found in Table 11, while Table 18 shows all examples. From this we find that the labels generally seem to be correct. However, we also see that the generated examples are less ambiguous and very much stylized like social media comments. The less ambiguous examples might be caused by the fact that we asked the model to generate examples displaying only a single social dimension whereas the original data might have multiple labels per example. In terms of style, we speculate that the instruction to generate samples *in style of social media comments* could impact downstream performance negatively as we might inadvertently have skewed the conversational style from Reddit posts to a more generic social media style. We believe it would be better to include online conversation as a style in the prompts instead. Future work could also consider an even simpler prompt design, instructing the LLMs to rewrite example sentences and then let the base example implicitly encode all the information about style and domain as in Dai et al. (2023). However, we believe this is likely to increase intra BLEU scores. We only have a slight indication that high lexical diversity

Base	Augmented	Label	Dataset	Inter-Similarity
I'm just kidding, this is fantastic news congratulations and well wishes to you and your family!!	Woohoo, this is amazing news! Congrats and cheers to you!	Fun	ChatGPT	High
From my experience Generating your own is usually always better.	From what I know, generating your own is typically the better choice.	Knowledge	ChatGPT	High
you know, it's kind of funny because my mind instantly went to the left instead of the alt-right.	Nothing beats a game night with the fam! (family emoji, dice emoji)	Fun	ChatGPT	Low
If we go with the 6000kg for the elephant we're down to around 5000kg once you lose the skeleton.	By sharing your knowledge, you're creating a space for meaningful discussions and knowledge exchange.	Knowledge	ChatGPT	Low
It's funny, not many of my friends know the story, but I just told you guys.	Watching this show with my friends always makes for a hilarious night.	Fun	ChatGPT	Medium
Its only natural to consult you as the source when resolving our own contentions.	Your willingness to share your knowledge freely is an admirable quality that I truly respect.	Knowledge	ChatGPT	Medium
Your friends (younger crowd, I would assume) might think it's fun!	Your friends (aimed at the younger crowd, most likely) might find this super fun!	Fun	GPT-4	High
I was mainly referring to games where you assume control of a single character or a few characters.	I mostly meant games that involve you controlling a single character or a small group of characters.	Knowledge	GPT.4	High
you know, it's kind of funny because my mind instantly went to the left instead of the alt-right.	Anyone else think that video was the perfect pick-me-up after a long day? (grinning emoji)	Fun	GPT-4	Low
I could explain that it's some sort of psychological compulsion Oh you mean the compulsion is having to say that.	I believe it's called the desire for expertise, where people enjoy sharing what they know on a topic.	Knowledge	GPT-4	Low
Your friends (younger crowd, I would assume) might think it's fun!	This is definitely up the alley of the younger crowd's entertainment!	Fun	GPT-4	Medium
So are we racist, sexist, bigoted homophobes for using a term that you don't like.	Is being labeled as homophobic or sexist the consequence of using a phrase that you perceive as unacceptable?	Knowledge	GPT-4	Medium

Table 11: `Social Dimensions: Selected augmentation examples.` The table shows pairs of base and augmented texts displaying high, low and medium semantic inter similarity for a subset of all social dimensions. High denotes the pair with the maximum similarity for a given label and dataset while low shows the pair with the minimum similarity. Medium signifies the median pair of text similarity. All augmented examples have been generated using the balanced strategy.

gives better performance, therefore it is a valid strategy to try a more conservative generation process in the future.

With the goal of achieving a balanced label distribution, we selected a temperature value of 1 as we sampled the same text multiple times. Due to a limited time frame, this hyperparameter was selected based on small empirical experiments that qualitatively evaluated the semantic meaning of the generated text compared to the base samples. We could have selected a metric to evaluate either semantic or lexical similarities of the augmented data or performed small downstream experiments to assess the predictive power of the data. In the zero-shot classification prompts, we could have provided additional knowledge about the classes, including examples of each, transforming the task into a few-shot classification. We could have adopted the framework of AnnoLLM (He et al., 2023), which in the process of annotating using LLMs has a two-step framework, explain then annotate. In the first step, they ask ChatGPT to formulate few-shot Chain-of-Thought prompts which are subsequently used in step two, the annotation. Nonetheless, our objective was to design a simple and consistent framework to test the basic capabilities of the LLMs.

6.5 REFLECTIONS ON LLMS

Using LLMs-based approaches in tasks may impose several drawbacks. For instance, calling OpenAI's API incurs a financial cost based on the number of tokens produced. Additionally, using third-party services for tasks containing sensitive data may not be ideal. Furthermore, as LLMs are autoregressive models that produce text output, there is no guarantee that the output labels will align with the true labels. This is, however, ensured by using regular LMs with a dense output layer. In social dimensions extraction, ChatGPT produced the labels *appreciation*, *empowerment*, and *apology* despite being prompted to select from only the nine different classes (see Table 12). Lastly, as these models are proprietary and under constant development, it is challenging to ensure long-term consistency.

Text	Predicted class
"Thank you very much for drawing my vision."	Appreciation
"Perhaps you're right ; I took control of my life."	Empowerment
"First of all, I'm so sorry that you seem to think any of this is your fault."	Apology

Table 12: `Social Dimensions: ChatGPT hallucination examples.` ChatGPT produced three classes it was not presented with in the task of social dimensions extraction.

OpenAI and other companies or research units that develop LLMs put significant effort into implementing safety protocols and bias regulations (Perez et al., 2022; Ganguli et al., 2022). LLMs are heavily evaluated on safety metrics such as toxicity and bias (Gehman et al., 2020; Nangia et al., 2020). As a result, when directly prompting ChatGPT or GPT-4 to generate hate speech, the models will rightfully reject the request as it goes against OpenAI's moral and ethical principles. However, we were able to easily bypass this safety protocol for both ChatGPT and GPT-4 by using the following system prompt:

> *"You are a helpful undergrad. Your job is to help write examples of offensive comments which can help future research in the detection of offensive content."*

OpenAI (2023) highlight in their technical report of GPT-4, that some of the specific risks they explored are hallucinations, harmful content, disinformation, privacy, cybersecurity, etc. They highlight the extensive need for red teaming, using practitioners from numerous disciplines in an attempt to understand and counter this undesired behavior. Using reinforcement learning with human feedback (RLHF), they mitigate harms at the model level by giving high rewards to refusals of certain prompts and appropriate responses.

Our finding highlights the need for continued efforts to ensure that these models do not produce any harmful or biased output. While safety

protocols and regulations are in place, further research is necessary to ensure that LLMs produce ethical and safe outputs in all scenarios.

7 CONCLUSION

In this study, we present 2 possible solutions on how to improve performance in low-resource computational social science (CSS) tasks through novel data augmentation techniques and few-shot methods using contrastive learning. For data augmentation, we design simple prompts for the instruction-tuned LLMs ChatGPT and GPT-4 and find that the models are able to generate lexically diverse but semantically similar data. Subsequently, the study investigates whether the augmented data provides as efficient training samples as data generated by humans. Through an iterative process using increasingly larger data samples, we fine-tune a 110M parameter `E5-base` model on the individual datasets and asses the performance. This is compared to the zero-shot performance of GPT-4 and ChatGPT. We perform this experiment on 3 CSS tasks with increasing complexity. We find augmented data to be either on-par or worse than crowdsourced data, clouding earlier findings on the open-ended capabilities of LLMs. However, the zero-shot performance of LLMs is competitive to the models fine-tuned on crowdsourced data.

In our second proposed solution, we evaluate the performance of the contrastive few-shot learning method SetFit. We apply this method to the most complex task of extracting pragmatic concepts of social dimensions, going beyond conventional semantics. The study finds a significant increase in the performance of the models fine-tuned on augmented data, outcompeting both the zero-shot performance of GPT-4 and ChatGPT and the best model trained on crowdsourced data. Our study indicates that LLMs currently are incapable of solving the challenges of low-resource CSS tasks in online social conversations. Instead, we argue that LLMs will enable us to build highly performant and specialized models that outcompete them.

8 ETHICAL CONSIDERATIONS

The datasets employed in this study are openly accessible, and we have made the augmented data publicly available through our GitHub repository[1]. The purpose of generating augmented data is exclusively for experimental purposes, aimed at assessing the augmentation capabilities of large language models. It is crucial to note that we decisively disapprove of any intentions to degrade or insult individuals or groups based on nationality, ethnicity, religion, or sexual orientation. Nevertheless, we recognize the legitimate concern regarding the potential misuse of human-like augmented data for malicious purposes.

[1] https://github.com/AGMoller/worker_vs_gpt

BIBLIOGRAPHY

Al-Taie, M. Z. and S. Kadry (2017, March). Information Diffusion in Social Networks. *Python for Graph and Network Analysis*, 165–184.

Ali, R. H., G. Pinto, E. Lawrie, and E. J. Linstead (2022). A large-scale sentiment analysis of tweets pertaining to the 2020 US presidential election. *Journal of Big Data 9*(1), 79.

AlKhamissi, B., F. Ladhak, S. Iyer, V. Stoyanov, Z. Kozareva, X. Li, P. Fung, L. Mathias, A. Celikyilmaz, and M. Diab (2023, May). ToKen: Task Decomposition and Knowledge Infusion for Few-Shot Hate Speech Detection. arXiv:2205.12495 [cs].

Anto, M. P., M. Antony, K. M. Muhsina, N. Johny, V. James, and A. Wilson (2016, March). Product rating using sentiment analysis. In *2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT)*, pp. 3458–3462.

Argyle, L. P., E. C. Busby, N. Fulda, J. Gubler, C. Rytting, and D. Wingate (2022). Out of One, Many: Using Language Models to Simulate Human Samples. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 819–862. arXiv:2209.06899 [cs].

Bayer, M., M.-A. Kaufhold, B. Buchhold, M. Keller, J. Dallmeyer, and C. Reuter (2023, January). Data augmentation in natural language processing: a novel text generation approach for long and short text classifiers. *International Journal of Machine Learning and Cybernetics 14*(1), 135–150.

Belinkov, Y. and Y. Bisk (2018, February). Synthetic and Natural Noise Both Break Neural Machine Translation. arXiv:1711.02173 [cs].

Bengio, Y., R. Ducharme, and P. Vincent (2000). A Neural Probabilistic Language Model. In *Advances in Neural Information Processing Systems*, Volume 13. MIT Press.

Biewald, L. (2020). Experiment tracking with weights and biases. Software available from wandb.com.

Brown, T., B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and

D. Amodei (2020). Language Models are Few-Shot Learners. In *Advances in Neural Information Processing Systems*, Volume 33, pp. 1877–1901. Curran Associates, Inc.

Bubeck, S., V. Chandrasekaran, R. Eldan, J. Gehrke, E. Horvitz, E. Kamar, P. Lee, Y. T. Lee, Y. Li, S. Lundberg, H. Nori, H. Palangi, M. T. Ribeiro, and Y. Zhang (2023a, March). Sparks of Artificial General Intelligence: Early experiments with GPT-4. arXiv:2303.12712 [cs].

Bubeck, S., V. Chandrasekaran, R. Eldan, J. Gehrke, E. Horvitz, E. Kamar, P. Lee, Y. T. Lee, Y. Li, S. Lundberg, H. Nori, H. Palangi, M. T. Ribeiro, and Y. Zhang (2023b). Sparks of artificial general intelligence: Early experiments with gpt-4.

Byun, C., P. Vasicek, and K. Seppi (2023, April). Dispensing with Humans in Human-Computer Interaction Research. In *Extended Abstracts of the 2023 CHI Conference on Human Factors in Computing Systems*, CHI EA '23, New York, NY, USA, pp. 1–26. Association for Computing Machinery.

Choi, M., L. M. Aiello, K. Z. Varga, and D. Quercia (2020, apr). Ten social dimensions of conversations and relationships. In *Proceedings of The Web Conference 2020*. ACM.

Chomsky, N. (1957). *Syntactic Structures*. The Hague: Mouton and Co.

Chopra, S., R. Hadsell, and Y. LeCun (2005, June). Learning a similarity metric discriminatively, with application to face verification. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, Volume 1, pp. 539–546 vol. 1. ISSN: 1063-6919.

Chowdhery, A., S. Narang, J. Devlin, M. Bosma, G. Mishra, A. Roberts, P. Barham, H. W. Chung, C. Sutton, S. Gehrmann, P. Schuh, K. Shi, S. Tsvyashchenko, J. Maynez, A. Rao, P. Barnes, Y. Tay, N. Shazeer, V. Prabhakaran, E. Reif, N. Du, B. Hutchinson, R. Pope, J. Bradbury, J. Austin, M. Isard, G. Gur-Ari, P. Yin, T. Duke, A. Levskaya, S. Ghemawat, S. Dev, H. Michalewski, X. Garcia, V. Misra, K. Robinson, L. Fedus, D. Zhou, D. Ippolito, D. Luan, H. Lim, B. Zoph, A. Spiridonov, R. Sepassi, D. Dohan, S. Agrawal, M. Omernick, A. M. Dai, T. S. Pillai, M. Pellat, A. Lewkowycz, E. Moreira, R. Child, O. Polozov, K. Lee, Z. Zhou, X. Wang, B. Saeta, M. Diaz, O. Firat, M. Catasta, J. Wei, K. Meier-Hellstern, D. Eck, J. Dean, S. Petrov, and N. Fiedel (2022). Palm: Scaling language modeling with pathways.

Dai, H., Z. Liu, W. Liao, X. Huang, Y. Cao, Z. Wu, L. Zhao, S. Xu, W. Liu, N. Liu, S. Li, D. Zhu, H. Cai, L. Sun, Q. Li, D. Shen, T. Liu, and X. Li (2023, March). AugGPT: Leveraging ChatGPT for Text Data Augmentation. arXiv:2302.13007 [cs].

Deri, S., J. Rappaz, L. M. Aiello, and D. Quercia (2018, November). Coloring in the Links: Capturing Social Ties as They are Perceived. *Proceedings of the ACM on Human-Computer Interaction 2*(CSCW), 43:1–43:18.

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

Eloundou, T., S. Manning, P. Mishkin, and D. Rock (2023). Gpts are gpts: An early look at the labor market impact potential of large language models.

Feng, S. Y., V. Gangal, D. Kang, T. Mitamura, and E. Hovy (2020, October). GenAug: Data Augmentation for Finetuning Text Generators. arXiv:2010.01794 [cs].

Fortuna, P. and S. Nunes (2019, July). A Survey on Automatic Detection of Hate Speech in Text. *ACM Computing Surveys 51*(4), 1–30.

Ganguli, D., L. Lovitt, J. Kernion, A. Askell, Y. Bai, S. Kadavath, B. Mann, E. Perez, N. Schiefer, K. Ndousse, A. Jones, S. Bowman, A. Chen, T. Conerly, N. DasSarma, D. Drain, N. Elhage, S. El-Showk, S. Fort, Z. Hatfield-Dodds, T. Henighan, D. Hernandez, T. Hume, J. Jacobson, S. Johnston, S. Kravec, C. Olsson, S. Ringer, E. Tran-Johnson, D. Amodei, T. Brown, N. Joseph, S. McCandlish, C. Olah, J. Kaplan, and J. Clark (2022, November). Red Teaming Language Models to Reduce Harms: Methods, Scaling Behaviors, and Lessons Learned. arXiv:2209.07858 [cs].

Gehman, S., S. Gururangan, M. Sap, Y. Choi, and N. A. Smith (2020, September). RealToxicityPrompts: Evaluating Neural Toxic Degeneration in Language Models. arXiv:2009.11462 [cs].

Gilardi, F., M. Alizadeh, and M. Kubli (2023). Chatgpt outperforms crowd-workers for text-annotation tasks.

Granovetter, M. S. (1973). The Strength of Weak Ties. *American Journal of Sociology 78*(6), 1360–1380. Publisher: University of Chicago Press.

Grimmer, J. (2010). A Bayesian Hierarchical Topic Model for Political Texts: Measuring Expressed Agendas in Senate Press Releases. *Political Analysis 18*(1), 1–35. Publisher: [Oxford University Press, Society for Political Methodology].

He, X., Z. Lin, Y. Gong, A.-L. Jin, H. Zhang, C. Lin, J. Jiao, S. M. Yiu, N. Duan, and W. Chen (2023). Annollm: Making large language models to be better crowdsourced annotators.

Hochreiter, S. and J. Schmidhuber (1997, December). Long Short-term Memory. *Neural computation 9*, 1735–80.

Hoffmann, J., S. Borgeaud, A. Mensch, E. Buchatskaya, T. Cai, E. Rutherford, D. d. L. Casas, L. A. Hendricks, J. Welbl, A. Clark, T. Hennigan, E. Noland, K. Millican, G. v. d. Driessche, B. Damoc, A. Guy, S. Osindero, K. Simonyan, E. Elsen, J. W. Rae, O. Vinyals, and L. Sifre (2022, March). Training Compute-Optimal Large Language Models. arXiv:2203.15556 [cs].

Hu, E. J., Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, and W. Chen (2021). Lora: Low-rank adaptation of large language models. *CoRR abs/2106.09685*.

Kann, K., S. Rothe, and K. Filippova (2018, October). Sentence-Level Fluency Evaluation: References Help, But Can Be Spared! In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, Brussels, Belgium, pp. 313–323. Association for Computational Linguistics.

Kaplan, J., S. McCandlish, T. Henighan, T. B. Brown, B. Chess, R. Child, S. Gray, A. Radford, J. Wu, and D. Amodei (2020, Jan). Scaling laws for neural language models. (arXiv:2001.08361). arXiv:2001.08361 [cs, stat].

Karpathy, A. (2015, May). The Unreasonable Effectiveness of Recurrent Neural Networks.

Kobayashi, S. (2018, May). Contextual Augmentation: Data Augmentation by Words with Paradigmatic Relations.

Krizhevsky, A., I. Sutskever, and G. E. Hinton (2012). ImageNet Classification with Deep Convolutional Neural Networks. In *Advances in Neural Information Processing Systems*, Volume 25. Curran Associates, Inc.

Lambert, N., L. Castricato, L. von Werra, and A. Havrilla (2022). Illustrating reinforcement learning from human feedback (rlhf). *Hugging Face Blog*. https://huggingface.co/blog/rlhf.

Lazer, D., A. Pentland, L. Adamic, S. Aral, A.-L. Barabási, D. Brewer, N. Christakis, N. Contractor, J. Fowler, M. Gutmann, T. Jebara, G. King, M. Macy, D. Roy, and M. Van Alstyne (2009, February). Computational Social Science. *Science 323*(5915), 721–723.

Lazer, D. M. J., A. Pentland, D. J. Watts, S. Aral, S. Athey, N. Contractor, D. Freelon, S. Gonzalez-Bailon, G. King, H. Margetts, A. Nelson, M. J. Salganik, M. Strohmaier, A. Vespignani, and C. Wagner (2020, August). Computational social science: Obstacles and opportunities. *Science 369*(6507), 1060–1062.

LeCun, Y., Y. Bengio, and G. Hinton (2015, May). Deep Learning. *Nature 521*, 436–44.

Li, B., Y. Hou, and W. Che (2022). Data Augmentation Approaches in Natural Language Processing: A Survey. *AI Open 3*, 71–90. arXiv:2110.01852 [cs].

Liu, Y., M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov (2019, July). RoBERTa: A Robustly Optimized BERT Pretraining Approach. arXiv:1907.11692 [cs].

Loshchilov, I. and F. Hutter (2019). Decoupled weight decay regularization.

Maaten, L. v. d. and G. Hinton (2008). Visualizing Data using t-SNE. *Journal of Machine Learning Research 9*(86), 2579–2605.

Mangrulkar, S. and S. Paul (2023). Peft: Parameter-efficient fine-tuning of billion-scale models on low-resource hardware. *Hugging Face Blog*. https://huggingface.co/blog/peft.

Manning, C. D. and H. Schütze (1999). *Foundations of Statistical Natural Language Processing*. Cambridge, MA: MIT Press.

Matakos, A., E. Terzi, and P. Tsaparas (2017, September). Measuring and moderating opinion polarization in social networks. *Data Mining and Knowledge Discovery 31*(5), 1480–1505.

Miller, G. A. (1992). WordNet: A Lexical Database for English. In *Speech and Natural Language: Proceedings of a Workshop Held at Harriman, New York, February 23-26, 1992*.

Monti, C., L. M. Aiello, G. De Francisci Morales, and F. Bonchi (2022, October). The language of opinion change on social media under the lens of communicative action. *Scientific Reports 12*(1), 17920. Number: 1 Publisher: Nature Publishing Group.

Muennighoff, N., N. Tazi, L. Magne, and N. Reimers (2023, February). MTEB: Massive Text Embedding Benchmark. arXiv:2210.07316 [cs].

Nangia, N., C. Vania, R. Bhalerao, and S. R. Bowman (2020, September). CrowS-Pairs: A Challenge Dataset for Measuring Social Biases in Masked Language Models. arXiv:2010.00133 [cs].

Nguyen, D., A. S. Doğruöz, C. P. Rosé, and F. de Jong (2016, April). Computational Sociolinguistics: A Survey. arXiv:1508.07544 [cs].

OpenAI (2023). Gpt-4 technical report.

Ouyang, L., J. Wu, X. Jiang, D. Almeida, C. L. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, J. Schulman, J. Hilton, F. Kelton, L. Miller, M. Simens, A. Askell, P. Welinder, P. Christiano, J. Leike, and R. Lowe (2022). Training language models to follow instructions with human feedback.

Papineni, K., S. Roukos, T. Ward, and W.-J. Zhu (2002, July). Bleu: a Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, Philadelphia, Pennsylvania, USA, pp. 311–318. Association for Computational Linguistics.

Pavlick, E., P. Rastogi, J. Ganitkevitch, B. Van Durme, and C. Callison-Burch (2015, July). PPDB 2.0: Better paraphrase ranking, fine-grained entailment relations, word embeddings, and style classification. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, Beijing, China, pp. 425–430. Association for Computational Linguistics.

Perez, E., S. Huang, F. Song, T. Cai, R. Ring, J. Aslanides, A. Glaese, N. McAleese, and G. Irving (2022, February). Red Teaming Language Models with Language Models. arXiv:2202.03286 [cs].

Rajani, N., N. Lambert, and L. Tunstall (2023). Red-teaming large language models. *Hugging Face Blog*. https://huggingface.co/blog/red-teaming.

Reimers, N. and I. Gurevych (2019, 11). Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Rosenthal, S., N. Farra, and P. Nakov (2017, August). SemEval-2017 task 4: Sentiment analysis in Twitter. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, Vancouver, Canada, pp. 502–518. Association for Computational Linguistics.

Rosenthal, S., N. Farra, and P. Nakov (2019, December). SemEval-2017 Task 4: Sentiment Analysis in Twitter. arXiv:1912.00741 [cs].

Schroff, F., D. Kalenichenko, and J. Philbin (2015, June). FaceNet: A unified embedding for face recognition and clustering. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, MA, USA, pp. 815–823. IEEE.

Schwartz, H. A., J. C. Eichstaedt, M. L. Kern, L. Dziurzynski, S. M. Ramones, M. Agrawal, A. Shah, M. Kosinski, D. Stillwell, M. E. P. Seligman, and L. H. Ungar (2013). Personality, gender, and age in the language of social media: the open-vocabulary approach. *PloS One 8*(9), e73791.

Sennrich, R., B. Haddow, and A. Birch (2016, June). Improving Neural Machine Translation Models with Monolingual Data. arXiv:1511.06709 [cs].

Sigurbergsson, G. I. and L. Derczynski (2020, May). Offensive language and hate speech detection for Danish. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, Marseille, France, pp. 3498–3508. European Language Resources Association.

Taori, R., I. Gulrajani, T. Zhang, Y. Dubois, X. Li, C. Guestrin, P. Liang, and T. B. Hashimoto (2023). Stanford alpaca: An instruction-following llama model. https://github.com/tatsu-lab/stanford_alpaca.

Tevet, G. and J. Berant (2021, April). Evaluating the Evaluation of Diversity in Natural Language Generation. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, Online, pp. 326–346. Association for Computational Linguistics.

Touvron, H., T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, A. Rodriguez, A. Joulin, E. Grave, and G. Lample (2023). Llama: Open and efficient foundation language models.

Tunstall, L., N. Reimers, U. E. S. Jo, L. Bates, D. Korat, M. Wasserblat, and O. Pereg (2022, September). Efficient Few-Shot Learning Without Prompts. arXiv:2209.11055 [cs].

Turing, A. M. (1950). Computing machinery and intelligence. *Mind 59*(236), 433–460.

Valverde-Rebaza, J. and A. de Andrade Lopes (2013, December). Exploiting behaviors of communities of twitter users for link prediction. *Social Network Analysis and Mining 3*(4), 1063–1074.

Vaswani, A., N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin (2017). Attention is all you need. *CoRR abs/1706.03762*.

Wang, L., N. Yang, X. Huang, B. Jiao, L. Yang, D. Jiang, R. Majumder, and F. Wei (2022, December). Text Embeddings by Weakly-Supervised Contrastive Pre-training. arXiv:2212.03533 [cs].

Wang, W. Y. and D. Yang (2015, September). That's So Annoying!: A Lexical and Frame-Semantic Embedding Based Data Augmentation Approach to Automatic Categorization of Annoying Behaviors using #petpeeve Tweets. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, Lisbon, Portugal, pp. 2557–2563. Association for Computational Linguistics.

Wei, J., M. Bosma, V. Y. Zhao, K. Guu, A. W. Yu, B. Lester, N. Du, A. M. Dai, and Q. V. Le (2022, February). Finetuned Language Models Are Zero-Shot Learners. arXiv:2109.01652 [cs].

Wei, J., X. Wang, D. Schuurmans, M. Bosma, B. Ichter, F. Xia, E. Chi, Q. Le, and D. Zhou (2023, January). Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. arXiv:2201.11903 [cs].

Wei, J. and K. Zou (2019, August). EDA: Easy Data Augmentation Techniques for Boosting Performance on Text Classification Tasks. arXiv:1901.11196 [cs].

Williamson, V. (2016, January). On the Ethics of Crowdsourced Research. *PS: Political Science & Politics 49*(01), 77–81.

Wu, F. and B. A. Huberman (2004, July). Social Structure and Opinion Formation. arXiv:cond-mat/0407252.

Wu, S., O. Irsoy, S. Lu, V. Dabravolski, M. Dredze, S. Gehrmann, P. Kambadur, D. Rosenberg, and G. Mann (2023). Bloomberggpt: A large language model for finance.

Yoo, K. M., D. Park, J. Kang, S.-W. Lee, and W. Park (2021a, November). GPT3Mix: Leveraging Large-scale Language Models for Text Augmentation. arXiv:2104.08826 [cs].

Yoo, K. M., D. Park, J. Kang, S.-W. Lee, and W. Park (2021b, November). GPT3Mix: Leveraging large-scale language models for text augmentation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, Punta Cana, Dominican Republic, pp. 2225–2239. Association for Computational Linguistics.

Yue, L., W. Chen, X. Li, W. Zuo, and M. Yin (2019, August). A survey of sentiment analysis in social media. *Knowledge and Information Systems 60*(2), 617–663.

Zhang, T., V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi (2020, February). BERTScore: Evaluating Text Generation with BERT. arXiv:1904.09675 [cs].

Zhao, W. X., K. Zhou, J. Li, T. Tang, X. Wang, Y. Hou, Y. Min, B. Zhang, J. Zhang, Z. Dong, Y. Du, C. Yang, Y. Chen, Z. Chen, J. Jiang, R. Ren, Y. Li, X. Tang, Z. Liu, P. Liu, J.-Y. Nie, and J.-R. Wen (2023, April). A Survey of Large Language Models. arXiv:2303.18223 [cs].

Ziegler, D. M., N. Stiennon, J. Wu, T. B. Brown, A. Radford, D. Amodei, P. Christiano, and G. Irving (2020). Fine-tuning language models from human preferences.

Ziems, C., W. Held, O. Shaikh, J. Chen, Z. Zhang, and D. Yang (2023, April). Can Large Language Models Transform Computational Social Science? arXiv:2305.03514 [cs].