electronics

MDPI

Article

Improving Training Dataset Balance with ChatGPT Prompt Engineering

Mateusz Kochanek, Igor Cichecki, Oliwier Kaszyca, Dominika Szydło, Michał Madej, Dawid Jędrzejewski, Przemysław Kazienko * and Jan Kocoń *

Department of Artificial Intelligence, Wrocław University of Science and Technology, Wyb. Wyspiańskiego 27, 50-370 Wrocław, Poland; mateusz.kochanek@pwr.edu.pl (M.K.); igor.cichecki@pwr.edu.pl (I.C.); oliwier.kaszyca@pwr.edu.pl (O.K.); dominika.szydlo@pwr.edu.pl (D.S.); michal.madej@pwr.edu.pl (M.M.); dawid.jedrzejewski@pwr.edu.pl (D.J.)

* Correspondence: kazienko@pwr.edu.pl (P.K.); jan.kocon@pwr.edu.pl (J.K.)

Abstract: The rapid evolution of large language models, in particular OpenAI's GPT-3.5-turbo and GPT-4, indicates a growing interest in advanced computational methodologies. This paper proposes a novel approach to synthetic data generation and knowledge distillation through prompt engineering. The potential of large language models (LLMs) is used to address the problem of unbalanced training datasets for other machine learning models. This is not only a common issue but also a crucial determinant of the final model quality and performance. Three prompting strategies have been considered: basic, composite, and similarity prompts. Although the initial results do not match the performance of comprehensive datasets, the similarity prompts method exhibits considerable promise, thus outperforming other methods. The investigation of our rebalancing methods opens pathways for future research on leveraging continuously developed LLMs for the enhanced generation of high-quality synthetic data. This could have an impact on many large-scale engineering applications.

Keywords: ChatGPT; natural language processing (NLP); sentiment analysis; unbalanced datasets; text classification; large language model (LLM); prompting; prompt engineering; rebalancing

Citation: Kochanek, M.; Cichecki, I.; Kaszyca, O.; Szydło, D.; Madej, M.; Jędrzejewski, D.; Kazienko, P.; Kocoń, J. Improving Training Dataset Balance with ChatGPT Prompt Engineering. *Electronics* **2024**, *13*, 2255. https://doi.org/10.3390/electronics13122255

Academic Editors: Hao Fei, Wei Ji and Fei Li

Received: 7 May 2024
Revised: 3 June 2024
Accepted: 6 June 2024
Published: 8 June 2024

1. Introduction

Recent advancements in artificial intelligence, particularly in natural language processing (NLP), have marked significant milestones. Generative language models like the Generative Pretrained Transformers (GPT) series have gained attention mostly for their human-like text generation capabilities. These models have also shown astounding effectiveness in numerous tasks, including translation, summarization, and sentiment analysis. However, the development of these models demands substantial computational resources and extensive data, thus presenting challenges in terms of environmental impact and resource allocation.

That is why it is essential to think about how these vast models can be used effectively via APIs or web applications, without the need to train them from the ground up. In addition, as promising as these models are, their application is not without challenges. A primary issue is controlling their output effectively and consistently. Ensuring that the generated text aligns with the desired goal is a nontrivial task, given their complexity and the vast solution spaces they operate in.

In response to this issue, the concept of using prompts to guide the model's output has emerged. Prompt engineering specifically serves to direct model output, thus enhancing alignment with user-defined goals—an essential factor in computational models. However, while intuitive and effective, the science of prompt engineering remains largely uncharted territory. How different prompts influence the model's behavior is under active research and development.

This article delves into the utilization of prompt-based data generation techniques with GPT-3.5-turbo, thus focusing on its application in addressing dataset imbalances, which is a fundamental issue in machine learning. We aim to demonstrate how refined prompt engineering can enhance data quality and contribute to the broader field of machine learning, thus potentially influencing model validation and integration across various applications.

Our main contributions include the following:

- The conceptualization and execution of three innovative prompt engineering strategies aimed at balancing machine learning datasets;
- The evaluation of these prompting methods for their performance in mitigating dataset imbalances, particularly in sentiment analysis tasks;
- The investigation the impact of prompt engineering in improving the quality of synthetic data generated;
- The creation and analysis of three synthetic negative review datasets to test the practical applications of our methods.

2. Materials and Methods

2.1. Related Work

Large language models have recently become a highly active area of research in AI mainly due to the release of the ChatGPT chatbot [1]. This development has had a profound impact not only on the research community but also on the general public. It has spurred the evolution of prompt engineering methods for such models and offered a new perspective on knowledge distillation. This section presents an overview of the potential resources for work involving large language models (LLMs) such as GPT by OpenAI [1], Bard by Google [2], and dozens of others.

2.1.1. Transformers

Developing autoregressive models based on the Transformer architecture introduced by Google [3] has been a fascinating journey marked by several significant milestones.

In the context of ChatGPT, the first significant step came in 2018 with the Generative Pretrained Transformer (GPT-1) model [4], which could generate coherent sentences and paragraphs. Next came GPT-2 [5] and GPT-3 [6], thereby offering more parameters and much better performance outcomes than their predecessors. The GPT models set new standards for the scale and performance of language models.

An interesting variant of these models, InstructGPT [7], was fine-tuned based on human feedback. Remarkably, InstructGPT (1.8B parameters) produced outputs that the labelers evaluated as better than GPT-3 (175B parameters). This method was further refined in ChatGPT [1], which is an interactive conversational model trained with even more human feedback. Finally, GPT-4 [8] represents the latest evolution in large language models from OpenAI. It is a large-scale multimodal model capable of processing text and image inputs. Furthermore, OpenAI has recently introduced ChatGPT plugins [9], such as connecting GPT-4 to the Internet. It makes the model highly responsive to recent events and new information.

As the complexity and capabilities of these models have grown, so have their potential applications. The ability to generate coherent and contextually relevant text opens up numerous possibilities. GPT models are already used for synthetic data generation, as seen in [10,11] or regarding data augmentation in [12–14]. Furthermore, researchers test the capabilities of these models to solve multiple different NLP tasks, as presented in [15–17].

Although these advances are promising, they also raise a number of practical and ethical considerations. Ensuring responsible and equitable use of these technologies will be a substantial challenge in the future, as can be seen in [18–20].

However, it should be noted that most research concerning data generation or augmentation has primarily involved GPT-3 and GPT-2 models. Additionally, prompt engineering could become much more important in the coming months. In this article, we tested multiple methods of generating data with GPT-3.5-turbo.

2.1.2. Prompt Engineering

Prompt Engineering represents a crucial aspect of working with large language models such as GPT-3.5 and GPT-4. It involves designing and optimizing prompts given to these models to generate the desired responses, as shown in Figure 1. This section provides an overview of recent research developments in prompt engineering. We also point towards areas that require further investigation, thus highlighting the ongoing challenges and opportunities in this rapidly evolving field.

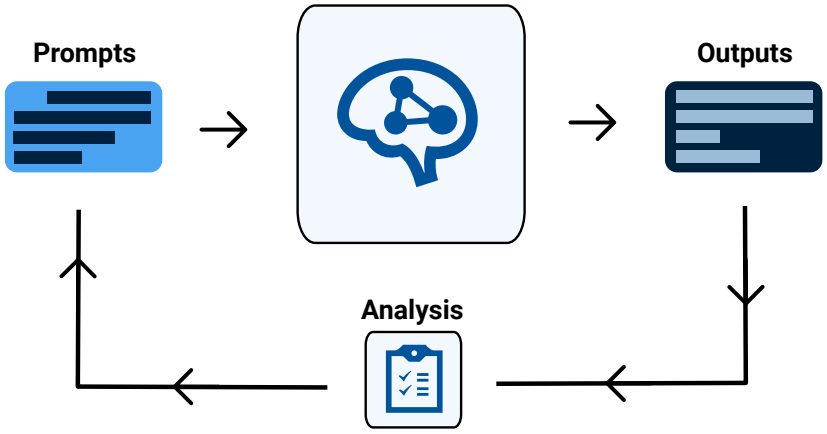


Figure 1. Prompt engineering is a process of continuous improvement based on output analysis.

There is already interesting research on multiple types of prompt patterns collected as a comprehensive catalog [21]. These patterns are useful for researchers and users of LLM chatbot technologies. An example of a pattern called "The Persona Pattern" can be seen in Figure 2. It gives the LLM a 'persona' that assists in determining the output format and the specific details to prioritize. Some pattern catalogs are more specialized, as in White et al. [22], where the focus was on prompts for tasks like code generation, code refactoring, or software design.

The Persona Pattern example:

Prompt:

You will pretend to be a Linux terminal for an attacker who has compromised a computer. When I type in a command, you are going to output the corresponding text that the Linux terminal would produce

Sample output after typing "ls -alt":

total 56
5 user user 4096 Feb 16 14:31 .
1 user user 18881 Feb 16 14:31 .bash_history
43 user user 4096 Feb 16 14:30 ..
1 user user 1178 Aug 15 2022 .bashrc
1 user user 220 Aug 15 2022 .bash_logout
3 user user 4096 Feb 16 14:25 .gnupg
1 user user 4041 Feb 15 17:52 README.md
2 user user 4096 Feb 1 11:35 project

Figure 2. An example of 'Persona' pattern from work by White et al. [21]. This pattern can simulate a 'code reviewer' or represent inanimate or nonhuman entities. An interesting aspect of this pattern is that LLMs may make assumptions or 'hallucinations' regarding the context, as seen in the example.

Out of these patterns, a new paradigm in natural language processing is emerging, which is widely known as prompt-based learning. In [23], the authors argue that this framework is very powerful, as it allows the language model to be pretrained on massive

amounts of raw text. Next, by defining a new prompting function, the model can perform a few-shot or even zero-shot learning, thus adapting to new scenarios with few or no labeled data. This approach was used for multiple NLP tasks shown in the work [15]. Some more specific use cases include relation classification [24] or dialogue systems [25].

Another crucial area in prompt engineering that directly relates to our work is data synthesis methods based on prompting strategies. For instance, Meyer et al. [26] explored whether prompt-driven synthetic data generation can improve conversational agents. In Shanshan et al. [27], the authors employed prompting strategies to generate synthetic data for on-device language models, thus achieving relative improvements of 19.0% and 22.8% in next-word prediction accuracy compared to the baseline model. Most research indicates that prompt-based data synthesis cannot replace standard data gathering, but it can serve as a valuable tool in data-scarce settings [28] or as a data augmentation technique [29]. Prompts have also shown promising results in tasks such as relation extraction [30], chain-of-thought demonstrations [31], and synthetic dialogues [32]. Most studies in this area focus on either developing highly specific prompting strategies or applying very general prompting strategies to new tasks. We believe there is significant value in comparing how different prompting strategies influence model results.

Although the opportunities for progress in prompt engineering are vast, the field also presents unique challenges that must be addressed. It is a difficult task to identify the prompts that lead to the desired outputs. Even if the prompt seems to work, there is an inherent challenge of unpredictability of model responses. Even with well-constructed prompts, there is a non-negligible chance that a model might generate off-topic or incoherent responses, thus complicating the task of ensuring quality control. This issue became particularly apparent with prompts that specified the output as a Python list. These prompts obtained much worse outputs than prompts specifying the output format as JSON.

2.1.3. General Concept of Knowledge Distillation

Knowledge Distillation was introduced in the research paper by Hinton et al. [33] as a procedure where the knowledge of a large model is distilled into a smaller student model. Of course, there were works before and after, where a small student model was generally supervised by a large teacher model [34–36]. This technique should result in a lightweight and computationally efficient student model that mimics the teacher's behavior by assimilating the extracted knowledge. It is a broad and crucial field of artificial intelligence.

An excellent survey on knowledge distillation is available by Gou et al. [37], which provides a comprehensive look at current developments and challenges in this field. The scheme for classic knowledge distillation framework can be seen on Figure 3. It discusses different types of knowledge to distill, such as response-based [33,38] that use logits from a large model, feature-based [39–41] that use features of intermediate layers, and relation-based [42–44] that use relationships between different activations, neurons, or pairs of samples. The survey discusses different distillation schemas, teacher–student architectures, and algorithms used for knowledge distillation.

Some additional notable examples of this technique involve the use of the distillation process for commonsense models [45] or the improvement of the few-shot ability of small language models [46]. In the first example, the authors used prompt engineering and a separately trained critic model to distill high-quality causal commonsense from GPT-3 selectively. They surpassed a human-authored commonsense knowledge graph with their automatically distilled variant. The second study demonstrated that prompt-based distillation can improve performance on few-shot datasets.

In the context of our article, the concept of knowledge distillation is more akin to the extraction of knowledge from the GPT-3.5 model. But, as already shown, prompt engineering can and will most certainly impact this research area in the future. That is why taking a closer look at this process after recent breakthroughs in large language models may be beneficial.

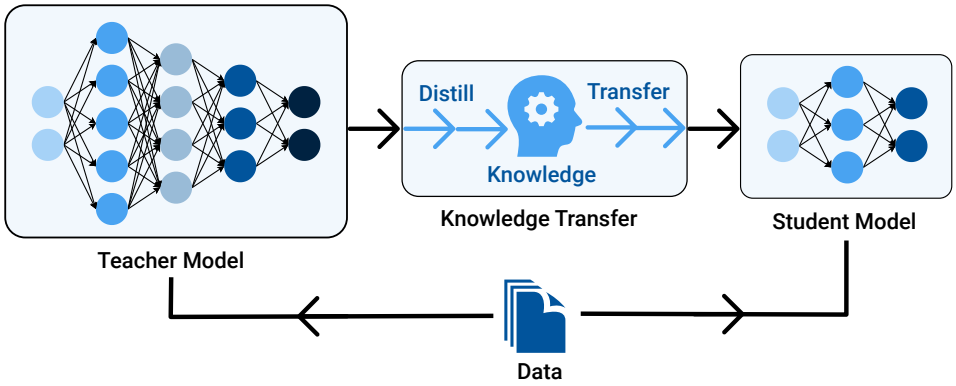


Figure 3. Scheme for classic knowledge distillation framework based on [37].

2.2. Three New Methods for Knowledge Distillation Using Prompts

Our work utilizes the IMDB [47] movie reviews dataset, which we describe in detail in Section 2.3.1. We generated synthetic movie reviews using the official OpenAI API specifically by employing the GPT-3.5-turbo model used by ChatGPT. We set the temperature of the chosen model to 0.8, as it is commonly assumed that ChatGPT has a model temperature set to be around 0.6–0.8. The procedure consisted of dataset analysis, prompt engineering, prompt generation, querying the API, results processing, and dataset rebalancing, as shown in Figure 4.

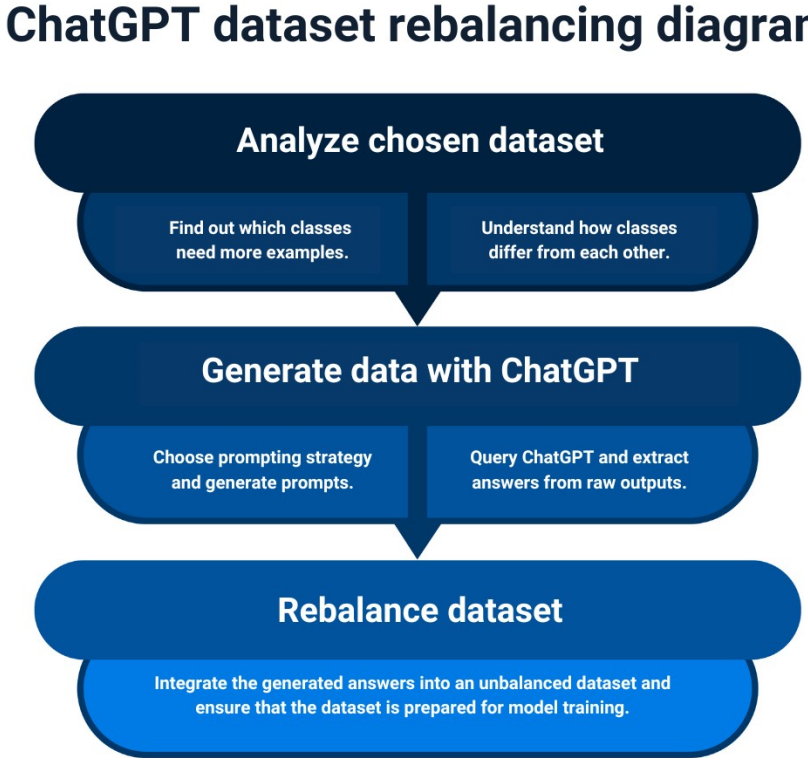


Figure 4. Flow diagram showing the three stages of dataset rebalancing with ChatGPT: (1) analyzing the dataset and choosing classes for synthetic data generation; (2) choosing prompting strategy, querying (prompting) the ChatGPT service, and extracting answers; (3) rebalancing the dataset by integrating extracted answers.

During prompt engineering, we carefully crafted the queries to elicit the desired responses from the model. During the prompt generation phase, we prepared the prompt in the JSON format for basic prompts, or we generated the prompts and saved them to a CSV file for composite and similarity prompts. We queried the API by sending requests

with prompts to the OpenAI API and retrieving the model's answers. Finally, we processed and prepared the resulting synthetic reviews for inclusion in the dataset.

We proposed and considered three different prompting methods:

1. Basic prompts—a simple, single prompt that generates three negative reviews about an unspecified movie.
2. Composite prompts—different prompts generated from sentences and probabilities specified in Listing 1, thus providing better diversity in the generated dataset.
3. Similarity prompts —different prompts that ask for reviews similar to one from the original dataset.

Listing 1. Sentences for prompt generation and their probabilities.

```
film_sentences = [
("Create three film reviews about the same
film: \"{}\".", 0),
("Create three film reviews about the same
film. Film can be real or imaginary.", 1)
]
film_sentences_pp = [0.2, 0.8]
attitude_sentences = [
" Reviews must be negative.",
" Reviews must be negative, but ambiguous.",
" Reviews must be negative and sarcastic."
]
attitude_sentences_pp = [0.5, 0.4, 0.1]
length_sentences = [
"",
" Reviews must be short.",
" Reviews must be long."
]
length_sentences_pp = [0.7, 0.1, 0.2]
additional_sentences = [
"",
" Focus on terrible director.",
" Focus on bad cinematography.",
" Focus on bad acting.",
" Focus on bad CGI effects.",
" Focus on boring plot.",
" Focus on bad writing.",
" Focus on film being very generic.",
" Focus on film being very boring."
]
additional_sentences_pp = [0.2, 0.1, 0.1, 0.1,
0.1, 0.1, 0.1, 0.1,
0.1]
```

2.2.1. Basic Prompts

We designed a basic prompt as a simple and straightforward way to generate data. We show its structure, which aligns with the OpenAI API requirements, in the JSON Listing 2.

The system message helps set the behavior of the assistant, even if, in the OpenAI documentation, it was written at the time that the GPT-3.5-turbo model does not pay strong attention to system messages. The user message is the main instruction that explicitly defines the task for the model and outlines how it should structure the output. Examples of two reviews generated by using basic prompts can be seen in Figure 5.

It is crucial to note that the reviews created by this method have pretty obvious similarities. For instance, in Figure 5, we can see that the sentence "The special effects were cheap and unconvincing" is identical in both reviews, despite them being generated in separate API calls.

The described method undeniably provides a quick, cheap, and straightforward way to generate synthetic data. Unfortunately, it has limitations. Given its simplicity, it does not allow for a high degree of control over the content of the generated reviews. Furthermore, as we represent in Figure 5, the basic prompt method can lead to repetition in the content

of the reviews. A lack of diversity in the training data could massively limit the robustness of machine learning models trained on such datasets.

Example of negative movie reviews from basic prompt:

Negative review 1:

The special effects were cheap and unconvincing. The dialogue was clichéd, and the pacing was slow. I checked my watch constantly, waiting for the film to end.

Negative review 2:

The acting was terrible, **the special effects were cheap and unconvincing**, and the script was full of plot holes. Save your money and skip this film.

Figure 5. An example of movie reviews generated by using basic prompt shown in Listing 2. We can see that both reviews have the same sentence.

Listing 2. Basic prompt in JSON format used as the input for OpenAI API.

```
[
{
"role": "system",
"content": "You are a negative review
generator. Return your output as a
json,where every review is one element.
Output should look like this:
{\"review1\" : \"\", \"review2\" : \"\",
\"review3\" : \"\"}.
Do not explain yourself."
},
{
"role": "user",
"content": "Create three film reviews
about the same film. Film can be real
or imaginary. Reviews must be negative.
Return your output as a json, where
every review is one element. Output
should look like this:
{\"review1\" : \"\", \"review2\" : \"\",
\"review3\" : \"\"}.
Do not explain yourself."
}
]
```

2.2.2. Composite Prompts

We proposed composite prompts as a natural progression from the basic prompting method. Unlike basic prompts, which typically consist of straightforward, singular requests, composite prompts integrate various elements within a single instruction set. The schema for the composite prompt method can be seen in Figure 6, thus illustrating how these components form a cohesive and effective prompt.

Composite prompt structure:

{film sentence} {attitude sentence} {additional sentences} {length sentences}
Return your output as a Python list, where every review is one element in that list.
Output should look like this: ['review1', 'review2', 'review3']
Do not explain yourself.

Figure 6. A schema that was used for the composite prompts generation. Listing 1 presents specific sentences.

The core of this method involved choosing sentences based on probability to parse into variables of the schema structure. Sentences and their associated probabilities were presented in Listing 1. It is also important to note that one of the film sentences had an empty variable that we filled with randomly chosen movie titles from the IMDB dataset. This element introduced an element of randomness and unpredictability, thus further mirroring the natural variability found in human-generated content.

These prompts offer better control over the content of generated reviews. The ability to incorporate specific sentence structures and film titles into the prompts adds an extra layer of realism to the generated reviews. Due to that, this method provides a greater level of diversity in the generated training dataset. Two examples of prompts and model outputs can be seen in Figures 7 and 8.

Composite prompt example 1:
Descriptive part of the prompt:
Create three film reviews about the same film. Film can be real or imaginary. Reviews must be negative and sarcastic. Focus on bad cinematography.
Negative review:
The film was so badly shot that I wondered if the cinematographer was a blindfolded toddler. The camera movements were so shaky that I thought I was watching an earthquake simulation. The lighting was so terrible that I wanted to donate a lamp to the production team.

Figure 7. First example showing a generated part of a prompt and one of resulting reviews.

Composite prompt example 2:
Descriptive part of the prompt:
Create three film reviews about the same film. Film can be real or imaginary. Reviews must be negative but ambiguous. Focus on bad CGI effects.
Negative review:
The film had potential, but unfortunately, the CGI effects were distracting and took away from the overall experience. It's a shame because the storyline was intriguing, but the poorly executed effects were a major letdown.

Figure 8. Second example showing a generated part of a prompt and one of the resulting reviews.

We gathered composite prompt data before basic prompt data. This is why the output structure of the composite prompts is specified as a Python list. During testing, it became apparent that the JSON format is easier for GPT models to understand and format correctly. This change demonstrates the iterative nature of developing methods for synthetic data generation as improvements and optimizations are identified and implemented over time.

Although the composite prompt method offers improved control and diversity compared to the basic prompt method, it has limitations. For example, the generated reviews may still have some repetitiveness due to the predefined set of sentences used in the prompts. Future improvements could explore methods for dynamically generating or selecting diverse sentences to incorporate into prompts, thus reducing the risk of repetitive content. By refining the method for choosing probabilities, movie titles, or even some specific actors, we could enhance the overall quality and realism of the generated reviews.

2.2.3. Similarity Prompts

The similarity prompt method represents another evolution in strategies that we developed for synthetic data generation. The main feature of this method is providing an

example of a negative review from the under-represented class to the generative model. We illustrate the prompts and the resulting output in Figure 9.

Similarity prompt example:

Prompt:

Create three film reviews about the same film. Film can be real or imaginary. Reviews must be negative. Base your reviews on this review:""Ned aKelly is such an important story to Australians but this movie is awful. It's an Australian story yet it seems like it was set in America. Also Ned was an Australian yet he has an Irish accent...it is the worst film I have seen in a long time"". Return your output as a JSON, where every review is one element. Output should look like this: {""review1"" : """", ""review2"" : """", ""review3"" : """"}. Do not explain yourself.

Negative review:

The movie Ned aKelly is a perfect example of how not to make a historical film. It fails to capture the essence of Australian history and its characters. The film looks like an American Western movie with kangaroos, and the portrayal of Ned with an Irish accent is completely disrespectful to Australian culture.

Figure 9. An example of a prompt and a resulting movie review generated by the model.

This method provides a way to generate synthetic reviews that more closely align with the original dataset's domain. Providing an example review of the model allows it to mimic the sentiment of the chosen review and its style, tone, and vocabulary. In this way, generating data that better retain the complexity of the original reviews may be possible.

Using this method, we chose samples from an already artificially reduced dataset to generate data. This approach mitigated the risk of data leakage and is the closest way to simulate real-world scenarios. Additionally, we chose JSON as the output format for this method, thus aligning with the findings from previous methods that showed that JSON is a more effective format for GPT models to understand and format correctly.

Despite some undeniable advantages, this method also presents some challenges and limitations. Most importantly, it heavily depends on the quality of the original data used as the basis for generation. Furthermore, similarity prompts could potentially limit the diversity of the synthesized data. It heavily depends on the number of original reviews available and the amount of synthetic reviews needed.

2.3. Experimental Setup

This section presents a detailed description of the experimental setup we used to gather results for this article. Implementation was coded in the `Python` language using various libraries to speed up the development process. We provide an overview of the entire experimental process used in this article. It should give the reader a comprehensive understanding of how these experiments were conducted, thus enabling them to better understand the results presented in the following sections. Figure 10 shows the general experimental setup. Data with movie reviews (IMDB) were exploited as the input dataset; see Section 2.3.1. Both positive and negative reviews examples from IMDB dataset can be seen on Figure 11. We tested three classification models: BERT, Random Forest, and Naive Bayes. See Section 2.3.2. They were trained on data, which were rebalanced using our three prompting methods applied to ChatGPT: basic, compisite, and similarity. See Section 2.2. Additionally, different contributions of enhanced data were considered: 10%, 20%, and full (100%); see Section 2.3.2.

The choice of BERT, Random Forest, and Naive Bayes for comparing experimental results is due to their diverse characteristics and representative nature in machine learning. BERT excels in understanding context and semantics, thus making it ideal for NLP tasks. Random Forest is an ensemble method that reduces overfitting and improves generalization. Naive Bayes is a simple probabilistic classifier that offers fast training and prediction, thus making it suitable for baseline comparisons.

Figure 10. Overview of general experimental setup. It consists of five consecutive steps: selecting the primary dataset, data generation using ChatGPT for unbalanced classes, dataset rebalancing with synthetic data, classification model training, and performance evaluation.

Examples of movie reviews:

Positive review:

Julie Andrews and Rock Hudson were great in this movie/musical. The opening song by Ms. Andrews, "Whistling Away the Dark", will always be in the back roads of my mind. The plot line during World War I is great and suspenseful. If you are a romantic, you will love this movie. This is a movie that I have always enjoyed seeing again and again.

Negative review:

This fanciful horror flick has Vincent Price playing a mad magician who realizes his vocational talents have been sold to another. He devises ways of avenging all those that have wronged him. His master scheme seems to backfire on him. Price is a little below par compared to his masterpieces, but it is still the only reason to watch this thriller. The supporting cast includes Patrick O'Neal, Mary Murphy, Eva Gabor, and Jay Novello.

Figure 11. Examples of both positive and negative reviews from IMDB dataset.

Result variation between models can be attributed to the inherent differences in how each model processes input data and learns from them. BERT benefits significantly from context-rich prompts that leverage its ability to understand nuanced language patterns; that is why composite prompts may be superior to similarity prompts in this setup, as they provide more examples that are out of standard dataset distribution. That is also why similarity prompts provide better results for Random Forest and Naive Bayes, which ultimately lack the pretraining knowledge of BERT. One relies on how well the prompts highlight distinguishing features, and another is more sensitive to prompt structures that align well with its probabilistic assumptions. Both of these conditions are fulfilled better by similarity prompts, as data generated by them are based on distinguishing features and probabilistic distribution of original training dataset.

2.3.1. Dataset

IMDB [47] movie reviews dataset is a collection of English reviews from the Internet Movie Database (IMDB) website labeled as positive or negative. On this website, users can write an opinion on a film they have seen and rate it on a scale from 1 to 10. In this specific dataset, reviews with a score of 4 or less are labeled negative, while those with a score of 7 or higher are considered positive. No more than 30 reviews are included per film, because reviews for the same movie tend to have correlated ratings.

The dataset contains 50,000 labeled reviews that are evenly split between 25,000 negative and 25,000 positive reviews. It was divided into training and test datasets, with each comprising 25,000 entries. The train and test sets contain a disjoint set of movies.

2.3.2. Experimental Design

In this work, we divided experiments into three main groups: experiments on a complete dataset, experiments on an artificially unbalanced dataset, or experiments on a dataset that was artificially unbalanced and replenished to a balanced state with synthetically generated data. We used all these tests to see whether the designed methods of prompt-based synthetic data generation can be used to balance skewed datasets.

We assembled each experiment in a controlled way. Initially, we prepared the training dataset. The first 10%, 20%, or all original negative review samples, depending on the chosen parameter, were used to create the given dataset. For the unbalanced dataset, we took the same negative reviews into the training dataset for each experiment. This way made it easier to compare results from all experiments. Furthermore, we prepared five stratified folds (balanced or unbalanced) from the training dataset to ensure that every method was reliably tested. Next, for each experiment, we performed a 5-fold stratified crossvalidation. We also evaluated the classifiers from each crossvalidation iteration on the original IMDB testing dataset.

In the following listing, we describe different experimental scenarios used to obtain the results shown in Table 1. Each was repeated for BERT, Random Forest, and Naive Bayes models:

Table 1. Table that shows results of carried-out experiments. The table is divided into three main parts, with each holding results for a specified model. Experiments per model are divided into three groups and are described with a keyword. "Full" means that a full dataset was used in this experiment. "Unbalanced" means that an unbalanced dataset was used. "Basic", "Composite", and "Similar" mean that an unbalanced dataset was replenished with data generated by a specified type of prompt. The 10% or 20% after the keyword specifies how many negative reviews from the original dataset were left in the training dataset. Values significantly better than unbalanced are shown in bold.

Model	Experiment	F1	Recall	Precision	Accuracy
BERT	Full	0.922 ± 0.013	0.905 ± 0.043	0.941 ± 0.025	0.923 ± 0.01
	Unbalanced 10%	0.842 ± 0.024	0.988 ± 0.005	0.735 ± 0.040	0.814 ± 0.035
	Basic 10%	0.857 ± 0.022	0.985 ± 0.009	0.759 ± 0.040	0.835 ± 0.030
	Composite 10%	0.865 ± 0.020	0.981 ± 0.009	**0.775 ± 0.038**	0.846 ± 0.028
	Similar 10%	**0.863 ± 0.026**	0.983 ± 0.009	0.770 ± 0.045	0.842 ± 0.037
	Unbalanced 20%	0.901 ± 0.013	0.969 ± 0.015	0.842 ± 0.035	0.893 ± 0.017
	Basic 20%	0.887 ± 0.024	0.978 ± 0.013	0.814 ± 0.049	0.875 ± 0.032
	Composite 20%	0.896 ± 0.026	0.965 ± 0.017	0.839 ± 0.056	0.887 ± 0.035
	Similar 20%	0.891 ± 0.015	0.978 ± 0.007	0.819 ± 0.030	0.880 ± 0.019
Random Forest	Full	0.841 ± 0.001	0.85 ± 0.003	0.832 ± 0.005	0.842 ± 0.001
	Unbalanced 10%	0.669 ± 0.001	0.503 ± 0.001	1.000 ± 0.000	0.506 ± 0.001
	Basic 10%	0.670 ± 0.001	0.504 ± 0.001	0.998 ± 0.000	0.509 ± 0.003
	Composite 10%	0.675 ± 0.003	0.514 ± 0.004	0.985 ± 0.001	0.526 ± 0.008
	Similar 10%	**0.681 ± 0.004**	**0.521 ± 0.005**	0.986 ± 0.001	**0.539 ± 0.010**
	Unbalanced 20%	0.692 ± 0.001	0.529 ± 0.001	0.999 ± 0.000	0.555 ± 0.002
	Basic 20%	0.682 ± 0.008	0.518 ± 0.009	0.997 ± 0.001	0.535 ± 0.017
	Composite 20%	0.692 ± 0.013	0.535 ± 0.016	0.983 ± 0.002	0.563 ± 0.028
	Similar 20%	0.700 ± 0.015	**0.544 ± 0.019**	0.983 ± 0.003	**0.578 ± 0.032**
Naive Bayes	Full	0.811 ± 0.002	0.857 ± 0.003	0.77 ± 0.004	0.821 ± 0.002
	Unbalanced 10%	0.667 ± 0.000	0.500 ± 0.000	1.000 ± 0.000	0.500 ± 0.000
	Basic 10%	0.671 ± 0.002	0.505 ± 0.002	1.000 ± 0.000	0.509 ± 0.004
	Composite 10%	0.679 ± 0.005	0.514 ± 0.006	0.999 ± 0.000	0.527 ± 0.011
	Similar 10%	**0.687 ± 0.007**	**0.524 ± 0.008**	0.998 ± 0.001	**0.546 ± 0.015**
	Unbalanced 20%	0.667 ± 0.000	0.500 ± 0.000	1.000 ± 0.000	0.500 ± 0.000
	Basic 20%	0.680 ± 0.007	0.515 ± 0.008	0.999 ± 0.001	0.529 ± 0.015
	Composite 20%	0.694 ± 0.013	0.532 ± 0.016	0.997 ± 0.001	0.559 ± 0.029
	Similar 20%	**0.709 ± 0.019**	**0.550 ± 0.023**	0.996 ± 0.002	**0.590 ± 0.040**

- Full: Baseline tests on a complete and unchanged IMDB training dataset.

- Unbalanced 10%/20%: Baseline tests on IMDB training dataset with 10% or 20% of original negative class samples. It means that our methods were not applied, but the contribution of minority class is the same as for the below scenarios with our methods.
- Basic 10%/20%: Setup with 10% or 20% of the original negative reviews left in training dataset. Negative reviews used for training were extended with samples generated using the basic prompting method described in Section 2.2.1.
- Composite 10%/20%: Setup with 10% or 20% of original negative reviews left in training dataset. Negative reviews used for training were extended with samples generated using the composite prompting method described in Section 2.2.2.
- Similar 10%/20%: Setup with 10% or 20% of original negative reviews left in training dataset. Negative reviews used for training were extended with samples generated using the similarity prompting method described in Section 2.2.3.

2.3.3. Model Parameters

We created a BERT classifier using the `BertForSequenceClassification` class from the `transformers` library. We initialized it with pretrained parameters from 'bert-base-uncased' with a number of labels equal to 2. We trained each classifier for four epochs with the learning rate parameter set to 2×10^{-5}. We chose Adam for an optimizer from the `Pytorch` [48] library.

Implementing the Random Forest and Naive Bayes Classifiers was imported from the `sklearn` [49] library with default training parameters.

3. Results

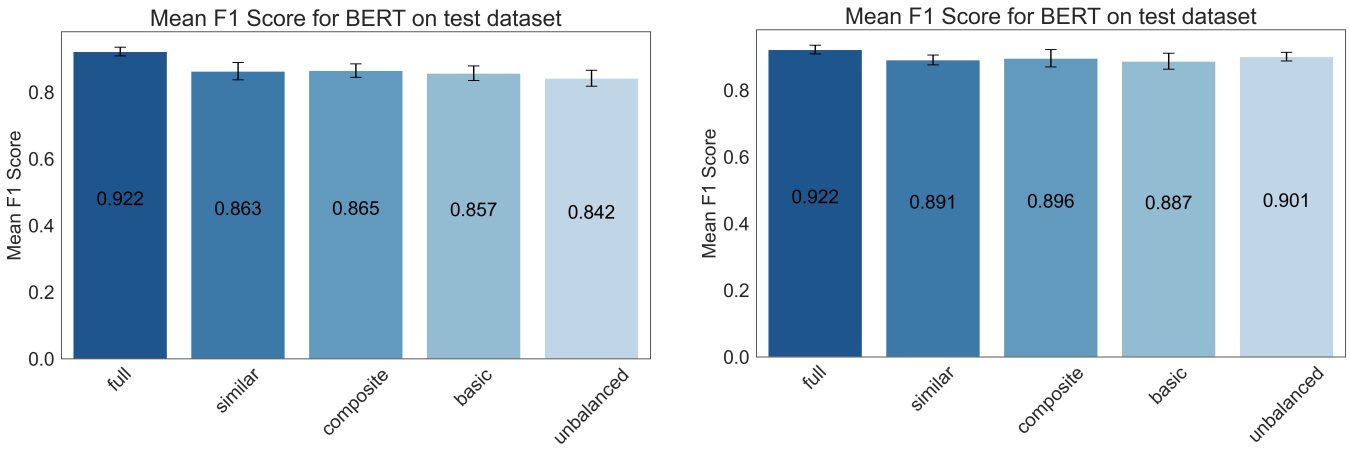
In this section, we provide a detailed analysis of the experimental results presented to evaluate the effectiveness of prompt-based data generation strategies for rebalancing datasets. The focus is on determining which of our new tested methods (basic, composite, or similarity prompts) are best by comparing models evaluated on the IMDB test dataset. All main results are depicted in Table 1. Significantly better values for each metric have been highlighted in bold. The significance of the data was tested by assessing the value to determine whether it was greater than the unbalanced value. Additionally, the samples were checked for normal distribution through the Shapiro–Wilk test [50]. For samples that met the criteria for normality, the Student's t tests [51] with level of significance $\alpha = 0.05$ were employed. For samples that did not meet the normality criteria, we used the Wilcoxon test [52]. The data and code utilized for this task, along with its comprehensive results, are provided as Supplementary Files. We additionally illustrate the F1 scores in Figures 12–14.

The F1 score was chosen for visual representation, because it has been widely used in the natural language processing literature, e.g., in [53], and is often considered the most robust metric in reflecting the balance between precision and recall. This ensures that our evaluation focuses on both false positives and false negatives, thus providing a more holistic view of model performance.

3.1. BERT

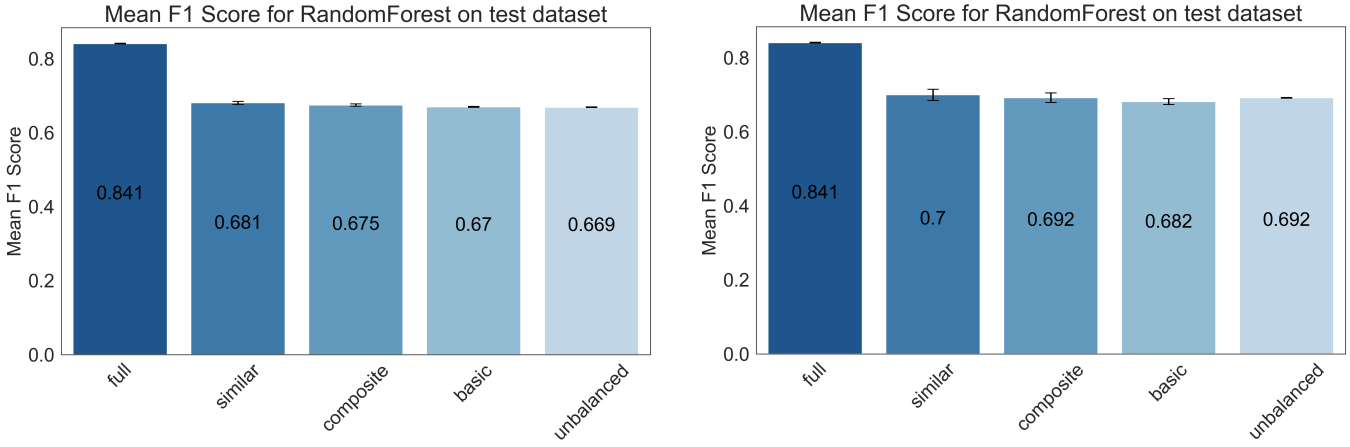
The BERT model, known for its strong performance on various natural language processing tasks, lived up to expectations, thus demonstrating considerable efficacy on the complete dataset. This is evidenced by its highest scores on all metrics: F1 score, accuracy, and precision.

When considering the 10% unbalanced experiments, the "Composite 10%" configuration emerged as the top performer in the F1 score, as illustrated in Figure 12a. This suggests that composite prompts might be valuable when dealing with a significant class imbalance. However, a peculiar finding is that in the 20% unbalanced experiments, the unbalanced dataset itself, ironically, outperformed the other methods. This result raises an intriguing hypothesis: beyond a certain threshold, allowing the dataset to remain unbalanced might be preferable to introducing potential noise through synthetic data generation.



(**a**) (**b**)

Figure 12. F1 score results for the BERT classifier and three our methods: basic, composite, and similar. (**a**) Plot for tests with 10% of train dataset. (**b**) Plot for tests with 20% of train dataset. Label full denotes 100% of enhanced data and refers to the greatest performance achieved for BERT. Label unbalanced is the baseline—our methods were not applied, but the contribution of negative labels were preserved.



(**a**) (**b**)

Figure 13. F1 score results for the Random Forest classifier and three our methods: basic, composite, and similar. (**a**) Plot for tests with 10% of train dataset. (**b**) Plot for tests with 20% of train dataset. Label full denotes 100% of enhanced data and refers to the greatest performance achieved for Random Forest. Label unbalanced is the baseline—our methods were not applied, but the contribution of negative labels were preserved.

Although the differences between the various methods in both the 10% and 20% unbalanced setups are relatively small, it should be noted that the composite prompts performed well, thus emerging as the preferred choice for generating synthetic data when using the BERT model.

3.2. Random Forest Classifier

The performance of the Random Forest classifier diverged noticeably from the full dataset compared to the other unbalanced experimental setups. Interestingly, for both the 10% and 20% unbalanced datasets, the similarity prompts approach outperformed the other methods, which is particularly evident in the F1 score, as shown in Figure 13.

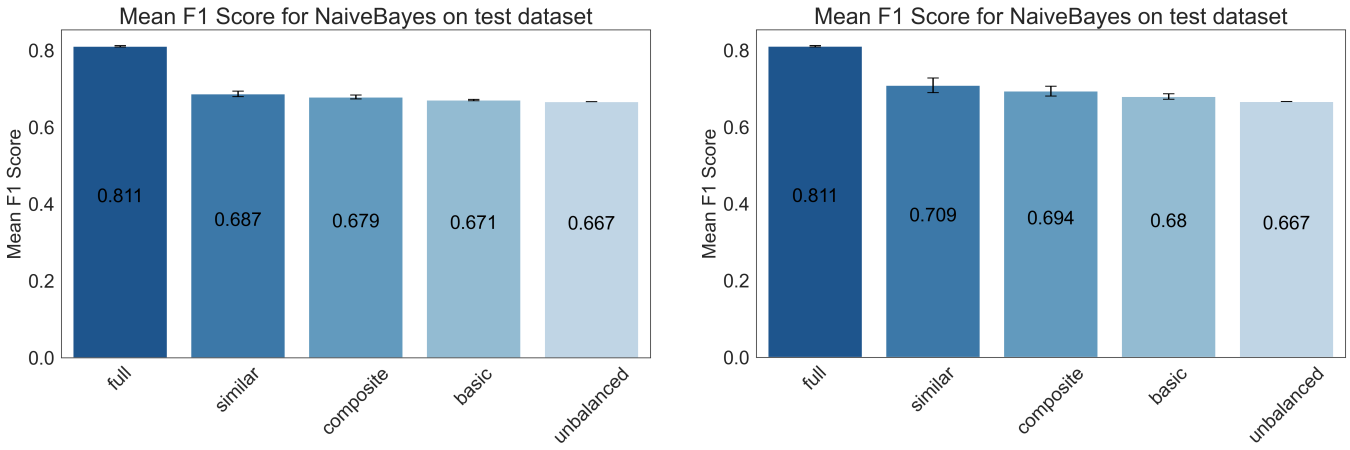


Figure 14. F1 score results for the Naive Bayes classifier and three our methods: basic, composite, and similar. (**a**) Plot for tests with 10% of train dataset. (**b**) Plot for tests with 20% of train dataset. Label full denotes 100% of enhanced data and refers to the greatest performance achieved for Naive Bayes. Label unbalanced is the baseline—our methods were not applied, but the contribution of negative labels were preserved.

The advantage of the similarity prompt method also extended to the accuracy and recall metrics, as highlighted in Table 1. On the contrary, the basic prompts method barely impacted the results in the 10% setup and degraded performance in the 20% setup. These results highlight the potential benefits of incorporating similar instances into the minority class when dealing with a class imbalance in a Random Forest model. On the other hand, the basic method barely had any impact and made the results much worse.

3.3. Naive Bayes Classifier

The Naive Bayes classifier performed similarly to the Random Forest model outcomes. Again, the similarity prompts method achieved the highest scores across all metrics, thus suggesting the consistent effectiveness of this strategy across different types of classifiers.

The composite prompts approach is second in terms of performance, followed by basic prompts. The experiment with an unbalanced dataset yielded the worst results, thus reinforcing the previous observation that leaving the dataset unbalanced negatively impacted performance compared to data generation methods.

4. Discussion

The experiments evaluated the effectiveness of the proposed prompt-based synthetic data generation techniques by checking how well they helped unbalanced datasets in the sentiment analysis task. We tested three proposed methods—basic prompts, composite prompts, and similarity prompts—in multiple configurations on three models: BERT, Random Forest classifier, and Naive Bayes classifier.

For the BERT model, in the case of 10% imbalance, composite prompts provided the best results compared to the unbalanced baseline. Unfortunately, its effectiveness was still far from the full dataset results. Interestingly, when the dataset was in 20% imbalance, the unbalanced baseline produced the best results, thus suggesting that beyond a certain threshold, it may be better to leave the data unbalanced rather than add the generated samples.

The performance of the Random Forest classifier on the full dataset was far better than in other unbalanced experimental setups. Both unbalanced experiments—10% and 20%—showed that the similarity prompts provided the best synthetic data. The composite prompts were slightly worse, and the basic prompts had a minimal impact on the 10% imbalance, thus even degrading performance in the 20% imbalance scenario.

The Naive Bayes classifier showed the most consistent results. Data generated with the similarity prompts achieved the highest results across all metrics, with the composite prompts being the second-best approach. The basic prompts had a better result than the unbalanced dataset.

In general, the composite and similarity prompts showed the potential to generate synthetic data when dealing with class imbalances. The choice of method may depend on the specific model used and the degree of imbalance in the dataset. However, care should be taken not to introduce potential noise through synthetic data generation, as this could degrade performance.

Finally, it is worth noting that prompt-based synthetic data generation could be used in other machine learning domains beyond sentiment analysis. This method has enormous potential in image recognition, spam detection, and more. Future research should explore these possibilities in more depth.

However, it is also crucial to consider the potential risk of ChatGPT generating incorrect data, as previously demonstrated in Lynch et al. [54]. In our case, we have generated over fifty thousand negative reviews using diverse and complex prompts. Before proceeding with further research, we have manually verified a significant number of them to minimize the possibility of using the inaccurate data. The data generated for this research are available in this public Github https://github.com/mateuszkochanek/Improving-Training-Dataset-Balance-with-ChatGPT-Prompt-Engineering (accessed on 6 May 2024) repository.

5. Conclusions

This article presents an overview of three new prompt-based synthetic data generation methods applicable to large language models such as GPT-3.5-turbo: basic prompts, composite prompts, and similarity prompts. We found a formatting improvement from Python list to JSON during the synthetic data gathering process with the composite prompts. We conducted some experiments to check to what extent generated synthetic data can be used to rebalance a given sentiment analysis dataset. For that reason, other language models like BERT, Random Forest and Naive Bayes were trained on the data enhanced with new data for the minority class. These new examples were generated by ChatGPT using our prompting methods.

During the experiments, it became apparent that none of the proposed methods is advanced enough to attain results close to the full dataset. The similarity prompts method provided the best results, but it still scored 8 to 20 percentage points lower than the full dataset. The composite prompts looked promising, and the basic prompts were useless, with only minor possible advantages, such as a simple structure and low cost. The output diversity of the basic prompts appears to be too low. Even if the generative model does not repeat generated data word for word, the outputs are still very similar. Anyway, even with only those three enhancement methods, there are always performance improvements compared to original unbalanced datasets.

These promising results encourage further studies. Firstly, the synthetic data generation methods can be greatly improved and combined. Composite prompts can easily improve their output diversity by adding new variables, and similarity prompts could be combined with composite prompts to generate diverse output that is better embedded in the task domain. Secondly, there are multiple ways to deal with unbalanced datasets that do not involve data generation. However, we can combine our prompt-based approaches with classic methods. Thirdly, the prompts may generate data that are out of domain. But what if those data were not used as rebalancing samples and instead for regular dataset augmentation? It may improve the robustness of models and even increase the results achieved on different test datasets.

Another direction is the usage of other LLMs like GPT-4, Gemini, Falcon, Mistral, Phi, Cohere, RWKV [55], and many others, as they are likely to provide improved performance and greater versatility in generating synthetic data. As language models become more sophisticated and their understanding of context and nuances deepens, more coherent and

diverse synthetic data could be expected. Our general idea of rebalancing training data using LLMs can be further tested on many NLP tasks with unbalanced data like emotion recognition or hate speech detection [56–61].

Supplementary Materials: The following supporting information can be downloaded at https://www.mdpi.com/article/10.3390/electronics13122255/s1 ; Table S1: Results of statistical analysis (T-student and Wilcoxon) in both .csv and .xlsx format; Code S1: Code used for statistical analysis in jupyter notebook format; Data S1: Data, consisting of the results of our experiments, used for statistical analysis.

Author Contributions: Conceptualization, M.K., J.K., P.K., I.C., O.K., D.S., M.M. and D.J.; methodology, M.K., J.K. and P.K.; software, M.K., I.C., O.K., D.S., M.M. and D.J.; validation, M.K.; formal analysis, M.K.; investigation, M.K., J.K. and P.K.; data curation, M.K.; writing—original draft preparation, M.K., J.K. and P.K.; visualization, M.K.; funding acquisition, J.K. and P.K.; supervision, J.K. and P.K.; writing—review and editing, I.C., O.K., D.S., P.K., M.M. and D.J. All authors have read and agreed to the published version of the manuscript.

Funding: This work was financed by (1) the National Science Centre, Poland, project no. 2021/41/B/ST6/04471 (JK, PK); (2) the Polish Ministry of Education and Science, CLARIN-PL; (3) the statutory funds of the Department of Artificial Intelligence, Wroclaw University of Science and Technology; (4) the Polish Ministry of Education and Science within the program "International Projects Co-Funded"; and (5) the European Union under the Horizon Europe, grant no. 101086321 (OMINO). However, the views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Executive Agency. Neither the European Union nor European Research Executive Agency can be held responsible for them.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The original data presented in the study is openly available at: https://github.com/mateuszkochanek/Improving-Training-Dataset-Balance-with-ChatGPT-Prompt-Engineering (accessed on 6 May 2024). Original data is built on the basis of publicly available dataset: https://ai.stanford.edu/~amaas/data/sentiment/ (accessed on 6 May 2024).

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. OpenAI. Introduction of ChatGPT Chatbot. Available online: https://openai.com/blog/chatgpt (accessed on 4 April 2024).
2. Manyika, J. An Overview of Bard: An Early Experiment with Generative AI. Available online: https://ai.google/static/documents/google-about-bard.pdf (accessed on 4 April 2024).
3. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.U.; Polosukhin, I. Attention is All you Need. In *Proceedings of the Advances in Neural Information Processing Systems*; Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R., Eds.; Curran Associates, Inc.: New York, NY, USA, 2017; Volume 30, pp. 6000–6010.
4. Radford, A.; Narasimhan, K.; Salimans, T.; Sutskever, I. *Improving Language Understanding by Generative Pre-Training*; OpenAI: San Francisco, CA, USA, 2018.
5. Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; Sutskever, I. *Language Models Are Unsupervised Multitask Learners*; OpenAI: San Francisco, CA, USA, 2019.
6. Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. Language models are few-shot learners. In Proceedings of the Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, Virtual, 6–12 December 2020; Volume 33, pp. 1877–1901.
7. Ouyang, L.; Wu, J.; Jiang, X.; Almeida, D.; Wainwright, C.L.; Mishkin, P.; Zhang, C.; Agarwal, S.; Slama, K.; Ray, A.; et al. Training language models to follow instructions with human feedback. *arXiv* **2022**, arXiv:2203.02155. [CrossRef]
8. OpenAI. GPT-4 Technical Report. *arXiv* **2023**, arXiv:2303.08774. [CrossRef]
9. OpenAI. Introduction of ChatGPT Plugins. Available online: https://openai.com/blog/chatgpt-plugins (accessed on 6 April 2024).
10. Hämäläinen, P.; Tavast, M.; Kunnari, A. Evaluating Large Language Models in Generating Synthetic HCI Research Data: A Case Study. In Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems, CHI '23, New York, NY, USA, 23–28 April 2023. [CrossRef]
11. Puri, R.; Spring, R.; Patwary, M.; Shoeybi, M.; Catanzaro, B. Training Question Answering Models From Synthetic Data. *arXiv* **2020**, arXiv:2002.09599. [CrossRef]

12. Chintagunta, B.; Katariya, N.; Amatriain, X.; Kannan, A. Medically Aware GPT-3 as a Data Generator for Medical Dialogue Summarization. In Proceedings of the 6th Machine Learning for Healthcare Conference, PMLR, Virtual, 6–7 August 2021; Jung, K., Yeung, S., Sendak, M., Sjoding, M., Ranganath, R., Eds.; Volume 149, pp. 354–372.
13. Bird, J.J.; Pritchard, M.; Fratini, A.; Ekárt, A.; Faria, D.R. Synthetic Biological Signals Machine-Generated by GPT-2 Improve the Classification of EEG and EMG Through Data Augmentation. *IEEE Robot. Autom. Lett.* **2021**, *6*, 3498–3504. [CrossRef]
14. Ben, V.A.P.; Minh, V.N.; Bonan, M.; Huu, N.T. Augmenting Open-Domain Event Detection with Synthetic Data from GPT-2. In *Machine Learning and Knowledge Discovery in Databases*; Nuria, O., Fernando, P.C., Stefan, K., Jesse, R., Antonio, L.J., Eds.; Research Track: Cham, Switzerland, 2021; pp. 644–660.
15. Kocoń, J.; Cichecki, I.; Kaszyca, O.; Kochanek, M.; Szydło, D.; Baran, J.; Bielaniewicz, J.; Gruza, M.; Janz, A.; Kanclerz, K.; et al. ChatGPT: Jack of all trades, master of none. *Inf. Fusion* **2023**, *99*, 101861. [CrossRef]
16. Qin, C.; Zhang, A.; Zhang, Z.; Chen, J.; Yasunaga, M.; Yang, D. Is ChatGPT a General-Purpose Natural Language Processing Task Solver? *arXiv* **2023**, arXiv:2302.06476. [CrossRef]
17. Lamichhane, B. Evaluation of ChatGPT for NLP-based Mental Health Applications. *arXiv* **2023**, arXiv:2303.15727. [CrossRef
18. Zhuo, T.Y.; Huang, Y.; Chen, C.; Xing, Z. Exploring AI Ethics of ChatGPT: A Diagnostic Analysis. *arXiv* **2023**, arXiv:2301.12867. [CrossRef]
19. Rahimi, F.; Talebi Bezmin Abadi, A. ChatGPT and Publication Ethics. *Arch. Med. Res.* **2023**, *54*, 272–274. [CrossRef]
20. Ray, P.P. ChatGPT: A comprehensive review on background, applications, key challenges, bias, ethics, limitations and future scope. *Internet Things Cyber-Phys. Syst.* **2023**, *3*, 121–154. [CrossRef]
21. White, J.; Fu, Q.; Hays, S.; Sandborn, M.; Olea, C.; Gilbert, H.; Elnashar, A.; Spencer-Smith, J.; Schmidt, D.C. A Prompt Pattern Catalog to Enhance Prompt Engineering with ChatGPT. *arXiv* **2023**, arXiv:2302.11382. [CrossRef]
22. White, J.; Hays, S.; Fu, Q.; Spencer-Smith, J.; Schmidt, D.C. ChatGPT Prompt Patterns for Improving Code Quality, Refactoring, Requirements Elicitation, and Software Design. *arXiv* **2023**, arXiv:2303.07839. [CrossRef]
23. Liu, P.; Yuan, W.; Fu, J.; Jiang, Z.; Hayashi, H.; Neubig, G. Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing. *arXiv* **2021**, arXiv:2107.13586. [CrossRef]
24. Zhang, W.; Song, X.; Feng, Z.; Xu, T.; Wu, X. LabelPrompt: Effective Prompt-based Learning for Relation Classification. *arXiv* **2023**, arXiv:2302.08068. [CrossRef]
25. Madotto, A.; Lin, Z.; Winata, G.I.; Fung, P. Few-Shot Bot: Prompt-Based Learning for Dialogue Systems. *arXiv* **2021**, arXiv:2110.08118. [CrossRef]
26. Meyer, S.; Elsweiler, D.; Ludwig, B.; Fernandez-Pichel, M.; Losada, D.E. Do We Still Need Human Assessors? Prompt-Based GPT-3 User Simulation in Conversational AI. In Proceedings of the 4th Conference on Conversational User Interfaces, CUI '22, Glasgow, Scotland, 26–28 July 2022. [CrossRef]
27. Wu, S.; Xu, Z.; Zhang, Y.; Zhang, Y.; Ramage, D. Prompt Public Large Language Models to Synthesize Data for Private On-device Applications. *arXiv* **2024**, arXiv:2404.04360. [CrossRef]
28. He, R.; Sun, S.; Yu, X.; Xue, C.; Zhang, W.; Torr, P.; Bai, S.; Qi, X. Is synthetic data from generative models ready for image recognition? *arXiv* **2023**, arXiv:2210.07574. [CrossRef]
29. Wang, Y.; Xu, C.; Sun, Q.; Hu, H.; Tao, C.; Geng, X.; Jiang, D. PromDA: Prompt-based Data Augmentation for Low-Resource NLU Tasks. *arXiv* **2022**, arXiv:2202.12499. [CrossRef]
30. Chia, Y.K.; Bing, L.; Poria, S.; Si, L. RelationPrompt: Leveraging Prompts to Generate Synthetic Data for Zero-Shot Relation Triplet Extraction. *arXiv* **2022**, arXiv:2203.09101. [CrossRef]
31. Shao, Z.; Gong, Y.; Shen, Y.; Huang, M.; Duan, N.; Chen, W. Synthetic Prompting: Generating Chain-of-Thought Demonstrations for Large Language Models. In Proceedings of the 40th International Conference on Machine Learning, PMLR, Honolulu, HI, USA, 23–29 July 2023; Krause, A., Brunskill, E., Cho, K., Engelhardt, B., Sabato, S., Scarlett, J., Eds.; Volume 202, pp. 30706–30775.
32. Steindl, S.; Schäfer, U.; Ludwig, B. Generating Synthetic Dialogues from Prompts to Improve Task-Oriented Dialogue Systems. In *KI 2023: Advances in Artificial Intelligence*; Seipel, D., Steen, A., Eds.; Springer: Cham, Switzerland, 2023; pp. 207–214.
33. Hinton, G.; Vinyals, O.; Dean, J. Distilling the Knowledge in a Neural Network. *arXiv* **2015**, arXiv:1503.02531. [CrossRef]
34. Buciluundefined, C.; Caruana, R.; Niculescu-Mizil, A. Model Compression. In Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '06, Philadelphia, PA, USA, 20–23 August 2006; pp. 535–541. [CrossRef]
35. Ba, L.J.; Caruana, R. Do Deep Nets Really Need to be Deep? *arXiv* **2014**, arXiv:1312.6184. [CrossRef]
36. Urban, G.; Geras, K.J.; Kahou, S.E.; Aslan, O.; Wang, S.; Caruana, R.; Mohamed, A.; Philipose, M.; Richardson, M. Do Deep Convolutional Nets Really Need to be Deep and Convolutional? *arXiv* **2017**, arXiv:1603.05691. [CrossRef]
37. Gou, J.; Yu, B.; Maybank, S.J.; Tao, D. Knowledge Distillation: A Survey. *Int. J. Comput. Vis.* **2021**, *129*, 1789–1819. [CrossRef]
38. Mirzadeh, S.; Farajtabar, M.; Li, A.; Ghasemzadeh, H. Improved Knowledge Distillation via Teacher Assistant: Bridging the Gap Between Student and Teacher. *arXiv* **2019**, arXiv:1902.03393. [CrossRef]
39. Zagoruyko, S.; Komodakis, N. Paying More Attention to Attention: Improving the Performance of Convolutional Neural Networks via Attention Transfer. *arXiv* **2016**, arXiv:1612.03928. [CrossRef]
40. Heo, B.; Lee, M.; Yun, S.; Choi, J.Y. Knowledge Transfer via Distillation of Activation Boundaries Formed by Hidden Neurons. *arXiv* **2018**, arXiv:1811.03233. [CrossRef]

41. Ahn, S.; Hu, S.X.; Damianou, A.C.; Lawrence, N.D.; Dai, Z. Variational Information Distillation for Knowledge Transfer. *arXiv* **2019**, arXiv:1904.05835. [CrossRef]
42. Yu, L.; Yazici, V.O.; Liu, X.; van de Weijer, J.; Cheng, Y.; Ramisa, A. Learning Metrics from Teachers: Compact Networks for Image Embedding. *arXiv* **2019**, arXiv:1904.03624. [CrossRef]
43. Tung, F.; Mori, G. Similarity-Preserving Knowledge Distillation. *arXiv* **2019**, arXiv:1907.09682. [CrossRef]
44. Liu, Y.; Cao, J.; Li, B.; Yuan, C.; Hu, W.; Li, Y.; Duan, Y. Knowledge Distillation via Instance Relationship Graph. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 7089–7097. [CrossRef]
45. West, P.; Bhagavatula, C.; Hessel, J.; Hwang, J.D.; Jiang, L.; Bras, R.L.; Lu, X.; Welleck, S.; Choi, Y. Symbolic Knowledge Distillation: From General Language Models to Commonsense Models. *arXiv* **2021**, arXiv:2110.07178. [CrossRef]
46. Wang, Y.; Liu, C.; Chen, K.; Wang, X.; Zhao, D. SMASH: Improving SMAll Language Models' Few-SHot Ability with Prompt-Based Distillation. In Proceedings of the Findings of the Association for Computational Linguistics: EMNLP 2022, Online Event, 16–20 November 2020; pp. 6608–6619.
47. Maas, A.L.; Daly, R.E.; Pham, P.T.; Huang, D.; Ng, A.Y.; Potts, C. Learning Word Vectors for Sentiment Analysis. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Portland, OR, USA, 19–24 June 2011; pp. 142–150.
48. Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. Pytorch: An imperative style, high-performance deep learning library. *Adv. Neural Inf. Process. Syst.* **2019**, *32*, 8026–8037.
49. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. Scikit-learn: Machine learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830.
50. SHAPIRO, S.S.; WILK, M.B. An analysis of variance test for normality (complete samples). *Biometrika* **1965**, *52*, 591–611. [CrossRef]
51. Dietterich, T.G. Approximate statistical tests for comparing supervised classification learning algorithms. *Neural Comput.* **1998**, *10*, 1895–1923. [CrossRef] [PubMed]
52. Woolson, R.F. Wilcoxon signed-rank test. *Encycl. Biostat.* **2005**, *8*. [CrossRef]
53. Derczynski, L. Complementarity, F-score, and NLP Evaluation. In Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16), Portorož, Slovenia, 23–28 May 2016; Calzolari, N., Choukri, K., Declerck, T., Goggi, S., Grobelnik, M., Maegaard, B., Mariani, J., Mazo, H., Moreno, A., Odijk, J., et al., Eds.; pp. 261–266.
54. Lynch, C.J.; Jensen, E.J.; Zamponi, V.; O'Brien, K.; Frydenlund, E.; Gore, R. A Structured Narrative Prompt for Prompting Narratives from Large Language Models: Sentiment Assessment of ChatGPT-Generated Narratives and Real Tweets. *Future Internet* **2023**, *15*, 375. [CrossRef]
55. Peng, B.; Alcaide, E.; Anthony, Q.; Albalak, A.; Arcadinho, S.; Biderman, S.; Cao, H.; Cheng, X.; Chung, M.; Derczynski, L.; et al. RWKV: Reinventing RNNs for the Transformer Era. In Proceedings of the Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, 6–10 December 2023; Bouamor, H., Pino, J., Bali, K., Eds.; pp. 14048–14077. [CrossRef]
56. Kazienko, P.; Bielaniewicz, J.; Gruza, M.; Kanclerz, K.; Karanowski, K.; Miłkowski, P.; Kocoń, J. Human-centered neural reasoning for subjective content processing: Hate speech, emotions, and humor. *Inf. Fusion* **2023**, *94*, 43–65. [CrossRef]
57. Mieleszczenko-Kowszewicz, W.; Kanclerz, K.; Bielaniewicz, J.; Oleksy, M.; Gruza, M.; Wozniak, S.; Dziecioł, E.; Kazienko, P.; Kocon, J. Capturing Human Perspectives in NLP: Questionnaires, Annotations, and Biases. In Proceedings of the The ECAI 2023 2nd Workshop on Perspectivist Approaches to NLP. CEUR Workshop Proceedings, Kraków, Poland, 30 September 2023.
58. Kanclerz, K.; Bielaniewicz, J.; Gruza, M.; Kocoń, J.; Woźniak, S.; Kazienko, P. Towards Model-Based Data Acquisition for Subjective Multi-Task NLP Problems. In Proceedings of the 2023 IEEE International Conference on Data Mining Workshops (ICDMW), Shanghai, China, 4 December 2023; pp. 726–735.
59. Bielaniewicz, J.; Kazienko, P. From Generalized Laughter to Personalized Chuckles: Unleashing the Power of Data Fusion in Subjective Humor Detection. In Proceedings of the 2023 IEEE International Conference on Data Mining Workshops (ICDMW), Shanghai, China, 4 December 2023; pp. 716–725.
60. Kanclerz, K.; Karanowski, K.; Bielaniewicz, J.; Gruza, M.; Miłkowski, P.; Kocon, J.; Kazienko, P. PALS: Personalized Active Learning for Subjective Tasks in NLP. In Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, Singapore, 6–10 December 2023; Bouamor, H., Pino, J., Bali, K., Eds.; pp. 13326–13341. [CrossRef]
61. Miłkowski, P.; Gruza, M.; Kazienko, P.; Szołomicka, J.; Woźniak, S.; Kocoń, J. Multiemo: Language-agnostic sentiment analysis. In *International Conference on Computational Science*; Springer: Cham, Switzerland, 2022; pp. 72–79.