DUTCH DATA AUGMENTATION FOR LOW-RESOURCE TASKS

CAN LARGE LANGUAGE MODELS BE USED FOR DUTCH DATA AUGMENTATION?

Daniël Johannes van der Weijden
5733111

Daily Supervisor:	Dr. Saskia Lensink
First Supervisors:	Dr. Dong Nguyen & Yupei Du
Second Supervisor:	Dr. Albert Gatt

Utrecht University TNO innovation for life

44 ECTS
Master Artificial Intelligence
Graduate School of Natural Science
Utrecht University
The Netherlands
August 2023

Abstract

Recent advancements in language models years have resulted in increased performance on various language task, mostly due to the availability of large datasets for pre-training and fine-tuning. However, acquiring labelled datasets for fine-tuning remains a significant challenge, particularly for low-resource tasks, due to the high cost and need for expert knowledge.

This thesis explores data augmentation techniques as a potential solution to this problem, with a focus on Dutch low-resource tasks. A newly proposed data augmentation method leverages Large Language Models (LLM's) to generate synthetic data, by providing the LLM with a description of the given task, and a small amount of examples.

The generated data is used to expand two original low-resource dataset, a sentiment analysis task and a natural language inference task. It is evaluated intrinsically by comparing the synthetic data to the original data on several linguistic metrics. The extrinsic evaluation is performed by fine-tuning a Dutch pre-trained language model for each synthetic dataset generated using one of the augmentation methods, and comparing the performance of the fine-tuned model.

The intrinsic evaluation of the augmented datapoints produced by the LLM highlighted their quality, though label accuracy couldn't be fully ensured. Despite enhancing diversity, the generated text seemed repetitive, leaning heavily on training data rather than provided samples.

Regarding extrinsic evaluation, outcomes for the augmentation method were mixed. The fine-tuned model's performance using LLM-generated data varied significantly between tasks. While sentiment analysis saw improved results surpassing the original dataset and alternative methods, this wasn't mirrored in the natural language inference task. The proposed method performed poorly with smaller subsets, and although it helped with larger subsets, it did not outperform alternative methods.

Although the proposed augmentation method is too unreliable in its current state, future research could provide improve on this method, by controlling for label distribution and testing the method on a variaty of alternative tasks.

The code used for this thesis is available online, and could be used to further develop the newly proposed augmentation method.[1]

[1] https://github.com/daanvdweijden/thesisAI

Internship at TNO

This thesis is conducted as a graduation internship at TNO[2]. TNO is a Dutch independent research organisation, which specialises in technical research and innovation. During my internship I am part of the Data Science team, within which I will operate in the Explainable and Responsible Data Science (XRDS) team.

My research will be part of an ongoing project called STARLIGHT[3], a large collaboration between European law enforcement agencies and research organisations, to enhance Europe's strategic autonomy in the field of artificial intelligence. One of the research interests of TNO within this initiative is synthetic data generation, for both computer vision and natural language tasks.

I will explore the possibilities of synthetic data generation for natural language tasks. While my research focuses on Dutch, it can contribute to the project not only for Dutch law enforcement agencies, but possible also for other European organisations or agencies, as they might encouner similar problems.

My aim for this internship is to not only contribute to the scientific community, but also to explore the possibilities for partners at STARLIGHT, as well as the XRDS team at TNO.

[2] https://www.tno.nl/nl/
[3] https://www.starlight-h2020.eu/

Contents

1 Introduction

In recent years, language models have seen a significant increase in performance due to advancements in machine learning techniques, increased computational resources and the availability of large amounts of data. These improvements have led to models that have increased performance on tasks including answering complex questions, understanding natural language inference and classifying texts correctly. This increased capability has led to a wide range of applications across several domains, including the medical industry [Szo19], law enforcement [Raa19], and the financial sector[Buc19], and continues to expand.

One of the key factors in the increased performance is the availability of large amounts of data, which are used to train these models. These large datasets are used in the first part of the training process of these large language models, where the model is pre-trained using self-supervised learning to learn the statistical distribution of the data, meaning they do not require the data to be labelled. While these large pre-trained language models learn the general distribution of the data, they often still require training to perform well on specific tasks.

The main challenges of these models comes with this second part of training, called fine-tuning, which require labelled datasets to train the language models for a specific task. These tasks (i.e. text classification), require labelled data to show the model how it should perform. This phase thus requires different datasets than in the pre-training phase, where unlabelled datasets could be used. For many tasks, the necessary training data for fine-tuning is not available or is costly to produce, which presents a significant challenge for the development of high-performing models.

While new approaches like zero- or few-shot learning provide alternatives to these large datasets, some are outperformed by fine-tuned models [Bro+20], or are too large to be run locally [Bro+20; Ope23] making them unable to be usable for sensitive data. They can also become rather expensive, as the user has to pay a fee for every request it makes. For many cases, it would thus be preferred to train ones own model, in order to decrease the reliance on these models.

Acquiring labelled datasets is thus a hurdle which hinders the expansion of possibilities for which language models could be employed. Although there is a lot of data available online, this data often needs to be curated, filtered and labelled for it to be used as training data. For certain tasks expert knowledge is even required for labelling, which further increases the costs. Addressing the problem of labelling datasets is therefore critical in increasing the capabilities of language models, especially for low-resource tasks. Solving this issue can thus increase the number of tasks in which language models can be employed with success, and could lead to more domains which can make use of this technology.

A potential solution to this problem is to use data augmentation methods, in order to expand existing datasets to a sufficient size. Data augmentation is an umbrella term for different methods for augmenting the amount of data in a dataset or increasing the diversity of the data in a given

Figure 1.1: Flipping the image of a flower will alter the pixels, but will remain semantically similar (i.e. still showing a flower).

dataset, using the original dataset as a reference. This is often done to either increase the variety of the data, or simply to increase the size of the dataset, in order to increase the performance of a model that is trained on the dataset [Whi21; Yoo+21]. But it can also be used for other purposes, like balancing a dataset by increasing the amount of datapoints of a certain class [Cha+02; WZ19], to generate a similar dataset that does not contain personal sensitive information [Yue+23], or to diversify the dataset to boost underrepresented groups [Sha+20; Zha+18]. The goal of data augmentation is to provide additional datapoints that are similar to the original dataset in order to provide help in the current task, but different enough so that they do not increase the amount of overfitting of the model.

These methods are in use in several machine learning domains, but have been used extensively in the computer vision domain [PW17; SK19]. Within this domain, several data augmentation methods are universally applied when training, and have been adopted in most pipelines by default. The main reason for adapting these models is to prevent overfitting on the dataset, and to diversify the pool of images that the model trains on. These techniques include randomly cropping, rotating, zooming and other simple translations of the images, often using a combination of these techniques. As can be seen in Figure 1.1, the flipping of an image does not alter the interpretation of the image itself. This allows for the model to learn how that a given subject of an image (i.e. a flower) can be illustrated in different ways. This makes the model invariant to these changes, and should in theory result in a more robust model.

In the field of natural language processing, data augmentation is not as straightforward when compared to computer vision, as it is hard to apply universal transformation, like flipping an image, that would leave the quality of the textual data intact [Kob18]. Flipping a sentence by flipping all the words or characters in a sentence, would impact the sentence in such a way that it would become nonsensical and ungrammatical, as can be seen in the following examples.

(1) "John is walking his dog."

(a) "Dog his walking is John."

(b) ".god sih gniklaw si nhoJ"

Data augmentation for language thus requires alternative methods, in order to preserve the semantics and labelling of the data. Common methods use techniques like synonym replacement[ZZL15], which replaces a random number of words with synonyms from WordNet [Mil95] to generate augmented sentences. Wei [WZ19] proposes EDA (Easy Data Augmentation), by combining synonym replacement with simple methods like random deletion, random swapping of words and random insertion. Although this can result in non-grammatical sentences, they show their augmentation method did increase performance over several language tasks. These methods can be extended to use contextual augmentation [Kob18], leveraging word embeddings to find suitable replacements.

With increasingly capable language models like BERT [Dev+18] and RoBERTa [Liu+19], previously successful augmentation methods that apply simple transformations (i.e. synonym replacement) do not show increased performance for these models [LWD20], meaning that the augmentation methods are not effective at providing extra data that is useful for training the model. Longepre et al. [LWD20] hypothesise that the small replacements do not alter the representation space that these models use for the data enough to increase the performance. Data augmentation can, according to Longpre et al., only be useful for these language models if they introduce new linguistic patterns, which require more sophisticated augmentation methods.

One of the more promising approaches is to use pretrained large language models [Bro+20; Rad+18; Rad+19] for data augmentation [Eva21; Yoo+21]. These models are capable of generating new text data that is similar in context to the original data, but does not augment the original data directly (i.e. replacing some words in the sentence). This can not only increase the amount of data in the datasets, but could also diversify the dataset which is used for fine-tuning a model, which could lead to better performing models. Using these models for synthetic data generation could thus lead to improvements over previous methods [Yoo+21], but as these models have only recently been made available, research into the capabilities of these models for this task has remained underexplored.

An interesting application of data augmentation using pretrained large language models is for low resource tasks, which are tasks for which there are few to no labelled datasets available. One set of these tasks involves use cases for the Dutch language, as it can be difficult to find large amounts of labelled data in languages other than English. Even though pre-trained large language models like GPT-3 [Bro+20] are predominantly trained on English text (92%), other languages are not filtered out of the training data, thus allowing it to learn how to generate sentences in other languages as well. GPT-3 is trained on 934.788 Dutch documents, containing over 660 million tokens, which is roughly 0.3% of the entire dataset.

While there are Dutch pre-trained models that can be used for language tasks, like RobBERT [DWB20; DWB22] and BERTje [De +19], there are a limited amount of labelled datasets available [VV19; WM21]. Creating tools for Dutch Data augmentation could lead to the expansion of Dutch datasets, thus allowing more tasks to be performed by these language models, and for newer models to be better evaluated on a larger set of tasks.

In this thesis the datasets that are available [VV19; WM21] will be used to mimic low-resource tasks, in order to investigate the effect of the data augmentation methods. This could show the potential of using these methods to increase the amount of labelled Dutch datasets available, at limited costs and effort. Using existing datasets is useful as it can give insights into difference in

performance compared to a full dataset, and can thus provide a better understanding of the range of performance increases that this method is capable of.

Evaluating synthetically generated data is not a trivial task, as it is hard to evaluate the quality of the generated text. There are many metrics which all evaluate the generated text in various ways, differing in complexity and ease of implementation. These metrics can be split into two categories: intrinsic methods, which evaluate the quality of the generated text itself, and extrinsic methods, which evaluate how well the generated data performs when used as training data for a model. While I will focus on the extrinsic evaluation to test our data augmentation method, I will also use several intrinsic methods to determine the quality of the synthetic data itself, in order to get better insight into the possibilities that this method has to offer.

1.1 Research Questions

In order to research this topic further, the following research questions and subquestions have been formed:

RQ "How does using LLM's as an generative augmentation method for Dutch language tasks compare to other augmentation methods?"

SQ1 "How does the synthetic data generated by GPT-3 increase the performance on on low-resource tasks of differing sizes, compared to other augmentation methods?"

SQ2 "How does the synthetic data generated by GPT-3 differ from other augmentation methods in generating diverse data, evaluated intrinsically?"

To answer **RQ**, I will use a similar methodology to Yoo et al. [Yoo+21], comparing the generative capabilities of LLM's for Dutch data generation to other data augmentation methods. For this research I will use a state of the art model, namely GPT-3.5, an updated version of GPT-3 [Bro+20], which is a multilingual large language model. The performance of the model will be tested on two Dutch datasets, DBRD [VV19] and SICKNL [WM21], which are a sentiment analysis task and a natural language inference task respectively.

The methods which I will use to compare the generative approach will be back-translation [Cou18], EDA [WZ19] and embedding replacement using a Dutch variant of BERT [Dev+18] called RobBERT [DWB22].

The three sub questions will be answered by comparing these augmentation methods on data-scarce tasks (**SQ1**), which will be mimicked by taking only a small percentage of an existing (data-abundant) dataset, and using that to generate synthetic sentences to increase the size back to the size of the original dataset. While **SQ2** will be answered by evaluating the generated datasets for diversity, as this is an important aspect for data augmentation methods.

2 Relevant Work

This section will dive into some of the relevant work for this research. Starting off with previous work in textual data augmentation in 2.1.1, looking into different methods that have been applied in this domain. After this I will focus on work that has been done in this field using large pre-trained language models in 2.1.2, and how this method differs from other techniques. The next section will highlight the Dutch context, providing relevant work for Dutch data augmentation in 2.2.2 as well as the representation of the Dutch language in the current Large Language models space in 2.2.1. Finally the evaluation methods will be discussed in 2.3, highlighting different intrinsic and extrinsic evaluation methods.

2.1 Data Augmentation

2.1.1 Text Data Augmentation

Although data augmentation methods for natural language processing are non-trivial, several methods have been proposed. These methods are on a spectrum of complexity, where complex models are often better performing but harder to implement, scale or generalise.

The simplest methods that are used for text data augmentation disregard the sensibility of the generated sentences, and apply random transformations to the source sentences. This can be done at a character level [BB18], where random letters in a word can be swapped, in order to make the model more robust to spelling variation, which often occur in datasets that are scraped from the internet. Random transformation can also be performed at a word level [Xie+17], by either replacing a word or leaving it blank, introducing noise in the data.

A more common approach is the replacement of words in a sentence with related words, either by using synonym replacement [KBM11; WZ19] or by using embeddings to find similar words [WY15]. These methods often maintain their syntactical structure, as the are being replaced by words in the same grammatical category. Since the word should be semantically similar given the embedding similarity, the meaning of the sentence as a whole should also remain similar. This is important given the fact that the augmented sentence should maintain its label. As can be seen in the example below, the word "love" in the original sentence can be replaced by either "like" or "hate", which could both have similar embeddings to "love", given their similar distributions. Replacing "love" with "like", would leave the semantics of the sentence fairly intact, while replacing it with "hate" would change the semantics completely. This is important in tasks like sentiment analysis, as the label for the original sentence would be positive, and would be maintained for example *a*, it should not be maintained for *b*.

(1) "I love this new show on tv!"

(a) "I *like* this new show on tv!"

(b) "I *hate* this new show on tv!"

Kobayashi et al. [Kob18] takes this concept even further, by not only providing the context in which a word appears, but also providing the label of the source sentence, resulting in a replacement that preserves the labelling explicitly. This approach is further explored and extended to using BERT [Dev+18] to get contextual replacements, while keeping labelling of the data into account [Wu+19].

Although these replacement approaches seem to provide increased performance on older language models, Longpre et al. [LWD20] states that modern large language models [Bro+20; Dev+18; Liu+19] do not benefit from these augmentation methods anymore. They hypothesise that this is due to the way these models represent language, and that the methods described above can do not provide enough variance to the original dataset, thus not having the desired effect. These models represent the input in a latent space, and changing a single word in the input does not change this representation enough for the model to reach new insights. In the worst case, it could even lead to overfitting, as the sentences generated are too similar to the original dataset. In order to increase the performance of these models, other data augmentation methods are required that change the underlying structure of the sentences, and add additional linguistic elements that might not be present in the original dataset.

Backtranslation, or round-trip-translation, is an augmentation method that uses language models for translation. It takes a sentence from the dataset, translates it into a foreign language using the language model, and then translates it back to the original language, getting a alternate version of the original sentence. This results in paraphrases of the original sentences, by exploiting the slight differences in language use of the different languages. Yu et al. [Yu+18] use a single translation cycle from English to French and back, which is also used by Coulombe [Cou18]. Aroyehun et al. [AG18] used multiple intermediate language to generate new sentences, to diversify the types of sentences they get. Using this method, they increased their dataset to 5 times the size of the original dataset, and increasing the performance of the model.

2.1.2 Augmentation using Large Language Models

One of the promising approaches to resolve this issue, is by using large pre-trained language models for data augmentation. These models are capable of generating synthetic data that does not augment the source sentence directly, thus by merely changing some tokens in the sentence but leaving the structure intact. These models can be used to augment entire paragraphs by using generative methods [Ana+20; He+22; Whi21; Yoo+21] to generate entirely new data.

Generative methods use large language models to generate new synthetic datapoints, either by fine-tuning a language model [Ana+20; He+22; Whi21], or by zero- or few-shot learning and providing natural text prompts [He+22; Yoo+21]. Most methods use GPT-2 [Rad+19] as their language model, finetuning it for the task of generating synthetic labelled data by providing the model with examples of desired augmentations. Anaby et al. [Ana+20] introduce a novel method called LAMBADA, Language Model Based Data Augmentation, which proposes several steps for synthesising data of high quality. They first train a baseline classifier on the original dataset,

which can later be used as a filter for the synthetically generated data, to serve as a minimal quality threshold. They than finetune a language model, in their case GPT-2, and synthesise a set of labelled sentences, which are passed through the filter mentioned before to be left with a filtered synthetically generated dataset. They show clear improvements over the baseline classifier, and also in comparison to other augmentation methods like EDA [WZ19] and CBERT [Wu+19]. Whitfield [Whi21] uses a similar method, but does not filter the dataset as a final step, while still improving the model's performance.

He et al. [He+22] extends this approach, by not only finetuning GPT-2, but also prompting language with examples from the original dataset. This technique is called few-shot learning, and has become an alternative to finetuning since the introduction of large language models. Instead of explicitly training a model for a specific task, the model is provided a natural text prompt describing the task at hand, accompanied with 1 or more examples of the intended result. This method thus requires much less data than finetuning a model, while still achieving impressive results [Bro+20; Rad+19]. They combine this data augmentation technique with knowledge distillation, increasing performance of DistilBERT on the GLUE [Wan+18] evaluation metric.

Few-shot learning performance has increased significantly for GPT-3 compared to its preceding version (GPT-2) [Bro+20], which is why Yoo et al. [Yoo+21] use this updated model for their data augmentation framework GPT-3Mix. They construct prompts according to a template that gets filled in based on the task at hand, the label that should be produced and corresponding examples from the original dataset. They test the performance of their proposed augmentation method by taking a small percentage of the original dataset, 0.1%, 0.3% and 1.0%, and expanding this dataset using GPT-3, as well as other augmentation methods EDA [WZ19] and back translation [Yu+18]. Their method show impressive results, increasing the accuracy of the models by up to 18.6% compared to the baseline.

2.2 Dutch Context

2.2.1 Large Language Models

While Large Language Models have emerged in recent years as seen in the previous section, most of this work has focused on the English language primarily. However, given the large amounts of data that is gathered to train these models, these models also take in some other languages, resulting in limited multilingual capabilities. The training data for GPT-3 contains data from a total of 118 different languages, but as can be seen in Table 2.1, the distribution of the amount of data is heavily skewed towards the English language.

Multilingual Large Language models have been proposed, with BLOOM being the largest open-source model [Sca+22] and LLaMa [Tou+23] being able to provide self-hosting, but these models either do not contain Dutch (as is the case with BLOOM) or do not publicise their training data, as is the case with GPT-4 [Ope23]. At this moment, there have been no monolingual Dutch large language models.

This does not mean that these models cannot be used for the Dutch language, as Table 2.1 shows, the training data for GPT-3 contains well over 600 million Dutch words, making it the 6th largest language in the dataset.

Language	Number of Words	Percentage of Total Words
en	181,014,683,608	92.647%
fr	3,553,061,536	1.819%
de	2,870,869,396	1.469%
es	1,510,070,974	0.773%
it	1,187,784,217	0.608%
pt	1,025,413,869	0.525%
nl	669,055,061	0.342%
ru	368,157,074	0.188%
ro	308,182,352	0.158%
pl	303,812,362	0.155%

Table 2.1: Top 10 languages in training data of GPT-3

2.2.2 Data augmentation for the Dutch Language

Data augmentation methods have, like many other natural language processing research, mainly revolved around the English language. There has been some research into multilingual data augmentation frameworks [JPL19; Liu+21], but most of these methods work by translating labelled sentences from a source language to a target language, thus not requiring any labelled data from the target language . Other approaches also turn to generative methods, using multilingual models like mBART [Liu+20] for the generation process.

In this work I will only focus on the Dutch language, to get a better sense of the shortcomings that data augmentation methods might have. To the best of my knowledge, this is the first work that dives into creating Dutch data augmentation methods. By looking at the performance of several data augmentation methods for a specific language, error analysis can highlight the potential hurdles that are specific for a given language, in order to better facilitate the needs that a given augmentation methods has for the target language, while also providing tools for Dutch data augmentation that could be used to increase the number of Dutch datasets available.

2.3 Evaluation

In order to evaluate the performance of data augmentation methods, it is important to determine what metrics I want to use. The most important thing to measure is the performance of a model trained on the augmented dataset on a downstream task. This type of evaluation is known as *extrinsic evaluation* and is usually measured in the form of increased performance compared to a model trained on another dataset, which can be the original dataset or a dataset augmented using a different approach. The change in performance is usually expressed as a change in metrics like accuracy, precision, recall or F1-score, but can also be task specific.

Another form of evaluation is known as *intrisic evaluation*, and is used to evaluate the quality of a text itself, instead of its performance on a downstream task. The most common method for evaluating generated text is human evaluation [CCG20], letting them rate the text either by its overall quality, or along one or more dimensions. These dimensions could relate to the grammatically of a text, the fluency or how creative the text is.

One of the problems with the evaluation of generative augmentation methods, is that many of these metrics rely in some form of comparing the generated text to a golden, often human-written, standard, which is been adopted from the translation domain. Given that many tasks that require generation are more open ended than translation, it is often difficult to compare the generated text to a 'golden standard'. Especially for the data augmentation task, where the goal is to generate alternative data, that should diversify the dataset.

Although many automated metrics compute a similarity of the generated text with a human written reference, there are other metrics that measure the performance in an alternative way. These metrics focus more on the diversity of the generated text, which could be compared to the diversity in the original data. Lexical diversity is important for data augmentation, as we want our dataset to be diverse in order for our model to capture a wide variety of expressions. Although these metrics do not capture the quality of a generated sentence, they do capture an important requirement for data augmentation methods, namely lexical diversity.

In section 3.4 I will describe the evaluation metrics that will be used to measure the performance of the different augmentation methods.

3 Methods

In order to answer my research questions, several experiments will be conducted. In this section I will dive deeper into the datasets that I will be using for these experiments (section 3.1), how I will form the natural language prompts that will be used for prompting the natural language model (section 3.2), and lastly a description of the experiments I will conduct (section 3.3).

3.1 Datasets

As discussed in section 2.3 I will evaluate the augmentation methods for two Dutch datasets, namely DBRD [VV19] and SICKNL [WM21] as these are publicly available datasets that have been used to measure the performance of Dutch language models in the past. The two datasets that have been chosen are for a sentiment analysis classification task [VV19] and for a natural language inference task [WM21].

Sentiment Analysis

Sentiment analysis is the task to classify the sentiment of a given text, or the *tone* of a text. This task can take multiple forms, but is most often framed as a classification task, with its most standard form with binary labelling 'positive' and 'negative'. It is used in many data augmentation research [BKR22], as it is not as sensitive to specific word changes, thus allowing relatively easy label preservation, compared to other tasks. One of the most used English datasets is the Large Movie Review Dataset [Maa+11], a binary classified dataset containing highly polar movie reviews.

A Dutch dataset inspired by the Large Movie Review Dataset is the Dutch Book Review Dataset (DBRD) [VV19], which contains 110k Dutch book reviews with its associated binary sentiment polarity labels. Only a part of these labels (22k) has been labelled by humans, which is the part that will be used in this research. Current state of the art models like RobBERT [DWB20] achieve an F1-score of 94.4, showing of the capabilities of these models.

Natural Language Inference

Natural language inference tasks target the ability of a model to classify the inference of a given premise sentence (or sentences) to a hypothesis sentence. The classification is often binary, with given labels 'entailment' and 'non-entailment', but this can also be extended by splitting the 'non-entailment' label into 'neutral' and 'contradiction'. There are multiple English datasets available like SICK [Mar+14] and SNLI [Bow+15].

SICKNL [WM21] is the Dutch variant of the SICK dataset [Mar+14], consisting of roughly 6k sentences forming almost 10k inference pairs. The dataset contains many forms of inference, most of which are heavily reliant on specific words, like hypernym relations or monotonicity. This

could make the dataset vulnerable to data augmentation methods, as the entailment of a inference pair can easily change if a word is replaced by a related word or synonym.

While there are some data augmentation techniques developed specifically for this task like Monalog [Hu+20], these are not only task-specific but also rely on (natural) logic and are often template based. It will thus be interesting to see how large language models are able to capture the essence of inference in their generative process.

3.2 Prompt Design

One of the key aspects of using large language models effectively, is by providing good prompts. As previously mentioned, prompts are the way to interact with the LLM, by providing instructions written in natural language that tell the model how it should behave, and optionally by providing it with some examples. Providing a sub-optimal prompt can drastically impact the outcome of the model [Zha+21], thus effecting the usefulness of the model as a data augmentation technique. A prompt should give enough information about the task for which it is supposed to generate new samples, provide a number of examples including their label, and give a template for the intended output format. It is thus important to create prompts in a structured way, while allowing the necessary task specific information to be inserted.

The proposed prompt template is based on the templates used in GPT3-Mix [Yoo+21], while adjusted to fit Dutch tasks and for the requirements for the GPT-3.5 model, which is conversation based. This resulted in the template that can be seen here:

> **User:** Provide a Dutch <TEXT TYPE> and the respective <LABEL TYPE>. <LABEL TYPE> is one of '<LABEL TOKEN 1>', ..., or '<LABEL TOKEN N>'.
>
> **Assistant:**<TEXT TYPE>: <EXAMPLE TEXT 1> (<LABEL TYPE>: <EXAMPLE LABEL 1>)
> ...
> <TEXT TYPE>: <EXAMPLE TEXT K> (<LABEL TYPE>: <EXAMPLE LABEL K>)
>
> **User:** Provide another <TEXT TYPE> in similar format and style.

As can be seen in the provided prompt template, the prompt is divided into messages from a user and from the assistant. The user first asks the model to provide an example of a given TEXT TYPE, and desired LABEL TYPE, as determined by the task at hand. A reply by the assistant (i.e. the model) is mimicked, using one or more of the examples from the original dataset. The user replies by asking the model to provide a similar output again, in the same format and style. The used template tokens for both tasks can be seen in Table 3.1.

In practice, an example prompt for the sentiment analysis task will look like this:

> **User:** Provide a Dutch *book review* and the respective *sentiment label*. *Sentiment label* is one of *'positive'* or *'negative'*.

Assistant: *Book review*: "Ik zou dit boek zeker aanraden!" (*sentiment label*: *positive*)

User: Provide another *book review* in similar format and style.

The output of the model will follow the same format, allowing the use of a regular expression to extract the necessary information required to add the synthetic data into the dataset. The following regular expression will be used: `'(?i)^Book review:\s(.*)\(sentiment:\s(\w+)'`, with group 1 containing the content of the review, and group 2 containing the associated label.

Task	<TEXT TYPE>	<LABEL TYPE>	'<LABEL TOKEN N>'
SA	Book Review	Sentiment label	Positive, Negative
NLI	Sentence Pair	Entailment label	Neutral, Entailment, Contradiction

Table 3.1: Template tokens used for the Sentiment Analysis (SA) task and the Natural Language Inference (NLI) task

3.3 Experimental design

The experimental design for this research will consist of two parts: the generation of the augmented datasets and the finetuning of the model based on these augmented training datasets.

3.3.1 Generating Datasets

The experiments that will be conducted will test the capabilities of using LLM's as a data augmentation technique, by generating data based on a subset of one of the two datasets described in section 3.1. In order to evaluate the effectiveness of this method on low-resource tasks, different size subsets will be taken, following the method of GPT3-mix [Yoo+21], as can be seen in Table 3.2.

Subset	Original Dataset		Augmented Dataset	
	DBRD	SICKNL	DBRD	SICKNL
0.1%	20	8	200	80
0.3%	60	24	600	240
1.0%	200	80	2000	800

Table 3.2: Size of subsets and augmented datasets for given tasks. Datasets are: DBRD [VV19] and SICKNL [WM21]

This subset of the original datasets will be used as a pool of data, from which random samples can be taken as example sentences for the provided prompt. The size of the subset thus determines the variety of examples that the model sees, which could determine the variety of outputs it provides.

For each of these sample sizes, three settings will be tested for the data augmentation method using LLM's, by providing a different number of examples from the subset to the prompt. I will

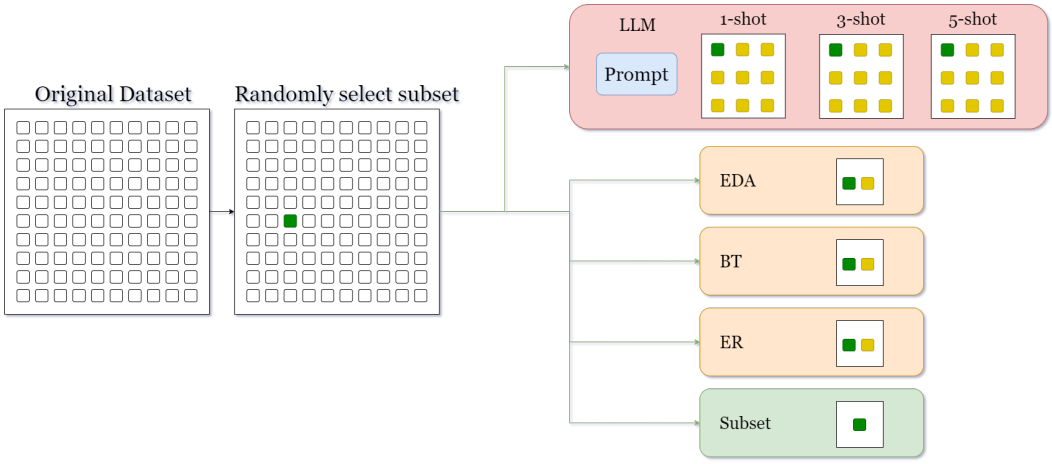

Figure 3.1: Different augmentation methods for generating augmented datasets.

use a one-shot setting, a 3-shot setting and a 5-shot setting, meaning I will provide either 1, 3 or 5 examples to the prompt according to the template described in section 3.2.

The Large Language Model that I will use for generating the data is GPT-3.5 - or specifically GPT-3.5-TURBO - from OpenAI[1]. This is an updated model from GPT-3, and is the model behind ChatGPT. To the best of my knowledge, this model has not yet been used in research before, which is a gap this thesis will fill. I will connect to this model using the provided API, similar to older GPT models. The hyperparameter settings will be similar to the method of GPT3Mix [Yoo+21], which used top-p of 1, a temperature of 1 and a frequency penalty of 0.02. I will increase the size of each subset by a factor of 10, resulting in various sizes of augmented datasets, as can be seen in Table 3.2.

Alternative Data augmentation Methods

Next to these three settings, I will also use these subsets to generate additional data using the other data augmentation methods, in order to compare their performance to the proposed method of this thesis. The alternative augmentation methods are EDA, embedding replacement and back-translation.

The implementation of EDA will be based on the the implementation in [WZ19], for which the code is provided online.[2]. EDA combines several simple data augmentation methods, most of which can remain intact from the original code, as these are not affected by the change in language. For the synonym replacement and random insertion however, the English wordnet[Mil95] is replaced with the Dutch variant, as provided by the NLTK package[3]. An alpha value of 0.1 is used for all provided methods.

[1] https://openai.com/
[2] https://github.com/jasonwei20/eda_nlp
[3] https://www.nltk.org/

Embedding replacement will be done using the embeddings from Dutch Word2vec model [TED16], and will be applied according to a similar methodology from Wang [WY15]. An alpha value of 0.1 is used to calculate the amount of replacements made.

For back-translation, the DEEPL python library is used, which implements the DeepL API[4], which provides translations to several languages. Given that this API is not deterministic, translating the same sentence multiple times can lead to different translations, allowing for back-translation from the same language to result in different outcomes. For this method, translation from Dutch to French back to Dutch is used. This method will only be used for the DBRD, as the texts used in the SickNL dataset only consists of a single sentence, which leads to an identical result when back-translation is used as an augmentation method.

Taking these different data augmentation methods together, results in 6 different augmented datasets: 3 datasets generated by the LLM, and 3 datasets by the alternative data augmentation methods. The results of the generated datasets will be compared to 2 subsets of the original dataset: the original subset without data augmentation applied, used as a lower bound baseline, and a subset with the same size as the resulting generated datasets, used as an upper bound, resulting in 8 total datasets per subset, as can be seen in Figure 3.1. This will be done for all 3 subset-sizes as seen in Table 3.2, which will yield 24 different datasets per task.

3.3.2 Finetuning Model

For both tasks I will fine-tune a language model to each of the augmented datasets, to perform the downstream task. For both tasks I will fine-tune RobBERT [DWB20], a Dutch pre-trained language model based on the RoBERTa architecture [Liu+19]. This model has been used before on both of the tasks, and has shown to be capable of performing these tasks - sentiment analysis and natural language inference - given that it is fine-tuned on the respective datasets. The model will be implemented using HuggingFace's TRANSFORMERS library [5]. Following GPT-3mix, the default hyperparameters will be used for training, with a batch-size of 8, a learning rate of 5e-5, while training for 10 epochs.

The model will be fine-tuned for each of the augmented datasets, resulting in 24 fine-tuned models per task, for a total of 48 models.After being fine-tuned, the performance of these models will be tested on the held out test set, which contains only original data, and is thus not augmented. These results will be compared by different performance metrics, in order to determine which data augmentation provides the best results.

3.4 Evaluation

3.4.1 Extrinsic Evaluation

In order to measure the performance of the augmentation methods, the performance metrics of the fine-tuned models on the given tasks will be compared. The metrics used will be accuracy, precision, recall and F1-score, and will be compared to the other data augmentation methods, as well as against the performance using just the subset.

[4] https://www.deepl.com/translator
[5] https://huggingface.co/docs/transformers/index

3.4.2 Intrinsic Evaluation

While the main focus of the evaluation of the data augmentation methods will be the performance on the downstream tasks, some intrinsic evaluation will also be performed to determine the quality of the generated text. The most common method for intrinsic evaluation is by human evaluators, who score the generated text on several dimension like grammar, fluency and creativity [CCG20]. Since this form of evaluation is not within the scope of this thesis, a small manual evaluation will be conducted, to get an idea of the grammatically and fluency of the generated texts.

Furthermore, a small subset will be analysed for label correctness, in order to manually check if the generated labels are in line with the generated texts. For each of the datasets, a sample of 50 sentences will be taken from the largest generated subset using the GPT-based method.

Next to this human evaluation a number of untrained automatic metrics will be used, looking into the most common words and sentences, the Type-token Ratio [Ric87], and the representation of the generated datasets in vector space using t-SNE plots [VH08].

Most common words and sentences

The analysis of most common words and sentences provides a quantitative measure to evaluate the diversity and variety both within, but also between the different datasets. Comparing the difference between the original dataset and the augmented version, can indicate how distinct the augmented dataset is. The analysis for most common sentences will only be performed for the DBRD, as the SICKNL dataset only contains single sentence texts.

Type-Token Ratio

The Type-token ratio (TTR) [Ric87] measures the ratio of types, which are unique words, to tokens, which is the total number of words, of a given text as can be seen in Equation 3.1. This measure can be easily computed, and can provide insight into the diversity of the lexicon of a given dataset. While originally used as a measure to determine the lexical diversity of children's language, it can also be used to compare different datasets generated by augmentation methods.

$$TTR = \frac{\#oftypes}{\#oftokens} \tag{3.1}$$

Because of the different sizes of datasets that need to be compared, a normalised version of this ratio will be used, the Moving-Average Type-Token ratio [CM10]. This calculates the TTR for a sliding window of 50 words, and takes the average over these windows. While the Normal TTR gives a general idea of overall lexical richness, the Moving-Average TTR provides a finer-grained analysis, and is thus preferred.

Vector distribution

Another method of measuring the difference between the generated datasets, is by looking at their representation in vector space. In order to visualise these datasets, all datapoints are vectorized using a simple tf-IDf vectorizer, after which a dimensionality reduction algorithm is used over the vector representation for each datapoint in all datasets. I will use t-Distributed Stochastic

Neighbor Embedding, or t-SNE [VH08] as the dimensionality reduction method, to reduce the dimensionality to two, in other to plot the results. For this, the dimensions will be reduced to 2 with a perplexity of 30, calculated over a maximum of 1000 iterations.

This visualisation can help in determining the overlap between the different augmented datasets, while not relying on direct word or sentence overlap, but rather by its representation in vector space. This can give additional insight into the diversity of the generated datasets.

4 Results

This section presents the results of the experiments conducted, starting of in section 4.1 with the intrinsic evaluation, which will provide insight into the generated datasets independent of their results when trained for a specific task. The following section, section 4.2, the results of the extrinsic evaluation will be presented, showing the performances of the fine-tuned models for the augmented datasets. Further interpretation of the results will be provided in section 5.1.

4.1 Intrinsic Evaluation

First, the results of an intrinsic evaluation of the generated datasets are presented, which represents a comprehensive analysis of the quality and diversity of the generated datasets for the needs of the specific tasks. The evaluation strategy employed takes a multifaceted approach, looking at different metrics which help in gaining a better understanding of the strengths and flaws of the generated datasets.

4.1.1 Manual Analysis

A manual exploration of the quality of the generated texts indicate the capability of the LLM based approach, as it generates texts that are coherent, grammatical and relevant for the task at hand. There are no spelling errors or syntactically strange phrases found in the generated datapoints. This is also the case for the back-translation datasets, but does not hold for the datasets generated by EDA or embedding replacement. These datasets are full of mistakes, both grammatical and semantically, although this is to be expected, given the nature of these methods.

Furthermore, a manual analysis was performed to determine label correctness. For both datasets, a subset of 50 sentences were chosen randomly, and manually assessed to determine their correspondence between the generated text and given label by the GPT-based method.

DBRD

The generated datapoints for the DBRD dataset, all showed correct labels for the generated texts that they were accompanied by. The generated reviews were one-sided, and did not contain the nuance that some of the original reviews contained, as can be seen in Table 4.1 below.

The first sentence of all generated reviews are enough to determine the given label, which is not the case for the original reviews, were the review might highlight both the positive and the negative points of a book. Since this nuance is absent for the generated reviews, the labels given to these reviews are sound.

Label	Generated Review
Positive	"Wat een fantastisch boek! Ik werd volledig meegesleept in het verhaal en kon niet stoppen met lezen. De personages waren zo levendig en het plot zat vol verrassingen. Ik raad dit boek zeker aan aan iedereen die van spanning en avontuur houdt."
Positive	"Ik ben echt ontzettend enthousiast over dit boek van Herman Koch! Het plot is zo intrigerend en zit vol met onverwachte wendingen. Koch heeft een meesterlijke manier van schrijven waardoor je als lezer volledig wordt meegesleept in het verhaal. De personages zijn goed uitgewerkt en voelen ontzettend realistisch aan. Ik kon me makkelijk inleven en voelde echt een connectie met hen. Daarnaast weet Koch ook maatschappelijke vraagstukken subtiel te verwerken in het verhaal, waardoor het ook nog eens relevant is. Kortom, een briljant boek dat ik zeker zou aanraden aan anderen!"
Negative	"Ik had hoge verwachtingen van dit boek, maar helaas heeft het me teleurgesteld. De schrijfstijl is saai en de personages blijven oppervlakkig. Het verhaal kabbelt voort zonder enige spanning of verrassing. Ik had gehoopt op een meeslepende plot, maar het blijft allemaal erg voorspelbaar. Het is hartstikke zonde van mijn tijd geweest om dit boek te lezen. Ik kan het niet aanraden aan anderen."
Negative	"Wat een verschrikkelijk boek! Ik kon gewoon niet in het verhaal komen. De personages waren saai en oninteressant. Het verhaal zelf was erg voorspelbaar en clichématig. Ik heb me enorm verveeld tijdens het lezen. De schrijfstijl was ook niet mijn smaak, het voelde geforceerd en onnatuurlijk aan. Ik zou dit boek zeker niet aanraden aan anderen."

Table 4.1: Sample of generated reviews using the GPT-based method for the DBRD dataset

SickNL

The SickNL dataset analysis provides a different view on the label correctness of the proposed method. From the 50 datapoints that were investigated, nine were found to be incorrect, and thus did not follow the labelling as proposed by the original dataset, which can be seen in Table 4.2.

Most of the incorrect labelling were caused by an incorrect contradiction label, were a neutral label would be correct. As seen in Table 4.2, the model incorrectly assumes a contradiction between the sentence pair, were the events are unrelated in practice. Although thematically the sentences share some content, this does not result in a contradiction, as given by the model.

Most sentence pairs revolve around a cat performing an activity, while the other sentence contains a dog doing an activity, with the label of the sentence pair being a contradiction. The sentences are unrelated, and the subjects of both sentences are different, with the correct label thus being neutral.

The only incorrect neutral sentence pair also has a cat as a subject, with the cat being asleep on the windowsill in the first sentence, and the cat jumping around in the garden in the paired sentence. Since these sentences both revolve around the same subject, performing different activities in two distinct locations, this should have been labelled as a contradiction.

The sentence pair labelled as an entailment is not as clearly incorrect as the other sentence pairs, but does not follow the proper logic. The first sentence states that the sun is shining bright, while

Sentence 1	Sentence 2	Label
De kat ligt te slapen op de vensterbank	De hond blaft luid en rent rond in de tuin	Con
De kat springt op de tafel	De hond blaft in de tuin	Con
Ik eet graag pizza	Ik ben allergisch voor kaas	Con
De zon schijnt fel	Het is overdag	Ent
De lucht is blauw op een heldere dag	De zon komt op in het oosten	Con
De hond eet zijn voer in de keuken	De kat jaagt op een muis in de tuin	Con
De kat ligt te slapen op de vensterbank	De kat springt rond in de tuin	Neu
Ik ben dol op chocolade	Ik ben allergisch voor chocolade	Con
De hond rent door het park	De kat ligt lui in de zon te slapen	Con

Table 4.2: Incorrect labels for the SickNL dataset. Con=Contradicion, Ent=Entailment, Neu=Neutral.

the second sentence says that it is daytime. This is not strictly entailed by the first sentence, although it might be the case most of the time. The correct labelling should have been neutral.

4.1.2 Most used Words

An analysis of the most common words for both datasets has been conducted for the largest subset size of 1.0%, as the size of these datasets provides the most information about the patterns that emerge. The most common words for the DBRD datasets can be seen in Table 4.3, while the results for the SickNL datasets can be found in Table 4.4.

DBRD

The most common words across all DBRD datasets, as seen in Table 4.3 are 'boek' (*book*) and 'verhaal' (*story*), which is to be expected given the fact that the dataset contains Dutch book reviews. Moving further down the list, the differences between the different datasets starts to show, as the most common words for the original datasets is comparable to the EDA- and embedding replacement dataset. The most common words for EDA contain the words 'boekje' and 'boeken' in the top 5, which are the diminutive and the plural form of 'boek', indicating that these were probable replacements for the most common word. The words for the dataset augmented using embedding replacement is similar to the original dataset, with the top 5 being identical.

The most common words for these 3 datasets all differ considerably from the most common words for the GPT-augmented datasets. The majority of the most common words in the GPT-augmented datasets are not present in the other datasets, while the difference between the GPT-augmented datasets itself is minor, with the most common words of the 3-Shot and 5-Shot datasets being nearly identical. The top 5 most common words, aside from 'boek' and 'verhaal', are not found in the original dataset.

SickNL

For the SickNL dataset, a similar difference between the GPT-augmented datasets and the other datasets can be seen. The most common words for the original dataset and the augmented datasets for EDA and embedding replacement are virtually identical, as seen in Table 4.4. For both the

Original	EDA	ER	1-Shot	3-Shot	5-Shot
boek	boek	boek	boek	boek	boek
verhaal	verhaal	verhaal	verhaal	verhaal	verhaal
lezen	lezen	lezen	personages	personages	personages
wel	boekje	wel	schrijfstijl	schrijfstijl	schrijfstijl
goed	boeken	goed	helaas	verwachtingen	verwachtingen
leven	eerste	heel	echt	helaas	helaas
eerste	gaat	leven	waardoor	vond	echt
gaat	heel	eerste	plot	echt	vond
echt	goed	gaat	oppervlakkig	oppervlakkig	oppervlakkig
heel	komt	erg	verwachtingen	voorspelbaar	voorspelbaar

Table 4.3: Top 10 of most common words of the largest subset size of 1.0% for the DBRD dataset

EDA augmented datasets and the embedding replacement augmented dataset, the word 'kat' (*cat*) is the ninth most common word, while not appearing in most common words of the original dataset.

The top 10 of these datasets differs substantially from the most common words in the GPT-augmented datasets, similar to the difference seen in the DBRD datasets. Not only does the top 10 differ from the original, the words 'zon' (*sun*), 'schijnt' (*shines*), 'park' (*park*), 'fel' (*bright*) and 'vensterbank' (*windowsill*) do not appear in the original subset once.

Original	EDA	ER	1-Shot	3-Shot	5-Shot
man	man	man	kat	kat	kat
vrouw	vrouw	vrouw	zon	zon	hond
hond	hond	hond	hond	hond	park
meisje	speelt	meisje	schijnt	park	vrouw
speelt	snijdt	speelt	speelt	schijnt	vrolijk
snijdt	meisje	snijdt	ligt	ligt	speelt
kijkt	kijkt	water	park	fel	zon
kind	houdt	kijkt	fel	speelt	rent
houdt	kat	kat	vensterbank	vensterbank	vensterbank
grote	springt	jongen	slapen	rent	ligt

Table 4.4: Top 10 of most common words of the largest subset size of 1.0% for the SickNL dataset

4.1.3 Most used Sentences

The analysis for the most common sentences is only performed for the Dutch Book review dataset as the SickNL dataset only contains datapoints consisting of single sentences. The analysis for the Dutch Book review dataset, which can be seen in Table 4.5, shows the difference between the top 5 sentences for each augmented dataset for the largest subset size of 1.0%. The analysis of the original subset is not given, as there were no sentences repeated more than twice in the

entire dataset. This is reflected by the most common sentences for the alternative augmentation methods, which have sentences that occur a maximum of 17 times in the dataset. The relatively short sentences that get repeated are due to the nature of these augmentation methods, as the methods randomly augmented certain parts of a text, and can thus leave parts of the text to stay unaltered, resulting in repetition.

Comparing the most common sentences of these datasets to the sentences for the GPT-augmented datasets, a clear distinction is visible. There is no overlap in the most common sentences when compared to the most common sentences of the alternative methods, while the occurrences of the most common sentences is as high as 194. These sentences also overlap between the different GPT-augmented sentences, and seem to follow similar sentence structures. Further analysis has shown that none of these sentences occur in this form in the original subset, although 'Een absolute aanrader!' (*An absolute must!*) and 'Wat een geweldig boek!' (*What an amazing book!*) both appear once as sub-phrases, and 'De personages waren oppervlakkig en ik kon me niet echt met ze identificeren.' (*The characters were superficial and I couldn't really identify with them.*) appeared once as the paraphrase: 'De personages waren plat en oppervlakkig, ik kon me totaal niet in hen inleven.' (*The characters were flat and superficial, I could not empathise with them at all.*)

4.1.4 Type Token Ratio

In order to get an understanding of the diversity of the generated datasets in contrast to the original subsets, the MATTR is calculated for each individual dataset, as can be seen in Table 4.6. The ratio for the DBRD datasets are higher than for the SickNL datasets, as is to be expected given the type of text that is used in these datasets.

DBRD

For the DBRD datasets, the overall MATTR for the augmented datasets does not differ much from the results of the original datasets. Only for the largest subset size does the difference between the ratio of the original datasets differ from the ratio of the generated datasets, especially for the alternative augmentation methods like EDA, Embedding replacement and Back-translation. No noticeable difference is seen between the different GPT-augmentation methods, as their ratio stays consistent for all subset sizes.

SickNL

A significant difference is seen in the ratio for the SickNL datasets, both between the augmented datasets and the original subset, and between the GPT-augmented methods and the alternative augmentation methods. The MATTR for the original dataset is significantly higher than for all augmented datasets, for all of the different subset sizes. It especially outperforms the ratios of the alternative augmentation methods, which achieve a ratio as low as 0.347. This difference is also notable when comparing the ratios of the GPT-augmented methods, which themselves do not differ much, with the alternative methods. This is in line with the expected results for these alternative methods, which generate more repetitive data given the small size of the original texts., resulting in lower ratios.

Form	Sentences	Count
1-Shot	Wat een teleurstelling!	194
	Wat een geweldig boek!	73
	Een absolute aanrader!	56
	Wat een fantastisch boek!	41
	De personages waren oppervlakkig en ik kon me niet echt met ze identificeren.	26
3-Shot	Wat een teleurstelling!	139
	De personages waren oppervlakkig en ik kon me niet echt met ze identificeren.	53
	Ik had hoge verwachtingen van dit boek, maar helaas heeft het me teleurgesteld.	51
	Wat een teleurstelling was dit boek!	43
	Ik had hoge verwachtingen van dit boek, gezien de lovende recensies die ik had gelezen.	34
5-Shot	Wat een teleurstelling!	102
	Ik had hoge verwachtingen van dit boek maar helaas heeft het me teleurgesteld.	64
	De personages waren oppervlakkig en ik kon me niet echt met ze identificeren.	57
	Ik had hoge verwachtingen van dit boek, gezien de lovende recensies die ik had gelezen.	53
	Wat een teleurstelling was dit boek!	36
EDA	Meestal duurt het veel langer voor ze me ‘grijpen’.	11
	Jammer	10
	Nee.	10
	Of toch?	10
	Gelukkig!	10
ER	Missie geslaagd in dit geval.	17
	Het verhaal slabakt ook nergens.	13
	Wat komt er uit de hoed?	13
	Meestal duurt het veel langer voor ze me ‘grijpen’.	12
	(Ik kwam tijdens het lezen een viertal taal- en spellingfouten tegen.)	11
BT	Nee.	10
	Meestal duurt het veel langer voor ze me ‘grijpen’.	9
	Wat komt er uit de hoed?	8
	Missie geslaagd in dit geval.	7
	Of toch?	7

Table 4.5: The top 5 most common sentences for the DBRD augmented datasets for subset size 1.0%

	DBRD			SickNL	
Size	Form	Result	Size	Form	Result
	Original	0.843		Original	0.753
	1-Shot	0.824		1-Shot	0.602
	3-Shot	0.827		3-Shot	0.642
0.1%	5-Shot	0.825	0.1%	5-Shot	0.612
	EDA	0.878		EDA	0.348
	ER	0.859		ER	0.403
	BT	0.837			
	Original	0.823		Original	0.792
	1-Shot	0.825		1-Shot	0.666
	3-Shot	0.812		3-Shot	0.631
0.3%	5-Shot	0.834	0.3%	5-Shot	0.663
	EDA	0.812		EDA	0.363
	ER	0.816		ER	0.416
	BT	0.836			
	Original	0.859		Original	0.769
	1-Shot	0.812		1-Shot	0.629
	3-Shot	0.797		3-Shot	0.610
1.0%	5-Shot	0.802	1.0%	5-Shot	0.639
	EDA	0.743		EDA	0.347
	ER	0.779		ER	0.406
	BT	0.764			

Table 4.6: The Moving-Average Type-Token Ratio for all datasets.

4.1.5 Vector Space

Analysing the vector space of the generated datasets provides a visual representation of the similarities of the datasets, by mapping the vector representations of the texts in the datasets to two dimensions to allow this visualisation. The resulting t-SNE plots for both tasks can be seen in Figure 4.1 and Figure 4.2.

DBRD

The t-SNE plot, as seen in Figure 4.1, shows a clear distinction between the GPT-augmented dataset and the datasets generated using the alternative methods. The datapoints for the GPT-augmented dataset form a cluster in the centre of the plot, while the datapoints for the alternative datasets are pushed to the periphery of the vector space. Within this larger cluster two smaller clusters can be identified, a dense cluster on the right, containing a combination of all datapoints from the different GPT-augmented datasets, and a more spread out cluster, consisting of mostly datapoints from the 3-Shot and 5-Shot datasets.

In the border of the vector space lie the datapoints from the alternative methods, forming small clusters, separated from eachother, but containing several datapoints from each of the alternative

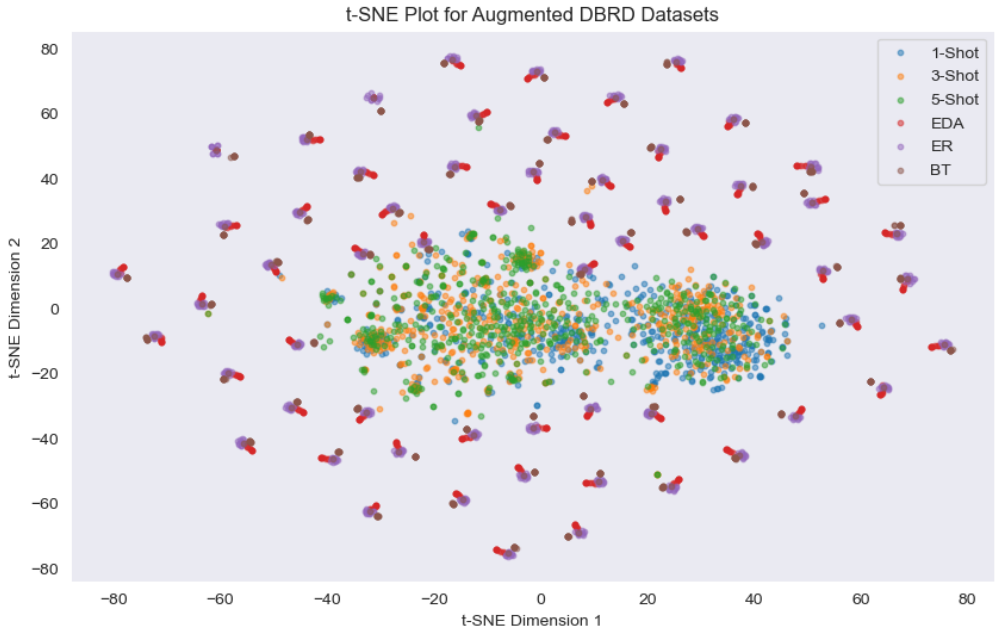

Figure 4.1: t-SNE plot for the augmented DBRD datasets.

methods. This shows the strong similarity that exists within smaller groups of datapoints, while a strong dissimilarity is seen outside of these small clusters. These individual clusters likely resemble the generated datapoints for the same source datapoint from the original subset.

SickNL

Figure 4.2 shows the t-SNE plot for the augmented datasets generated for the SickNL task. In this visual representation of the vector space, no major cluster can be identified. The datapoints of the GPT-augmented datasets are concentrated on the right side of the plot, without forming a dense cluster. While the datapoints for the alternative methods appear more on the left side of the plot, there are some clusters appearing throughout the entire vector space, overlapping with the datapoints of the GPT-augmented datasets.

Between the GPT-augmented datasets, the 1-Shot dataset appear in dense clusters, with individual outliers spread throughout the vector space. The 3-Shot and 5-Shot datasets are more spread out, while not overlapping with the datapoints from the alternative datasets.

4.2 Extrinsic Evaluation

4.2.1 DBRD

The results on the sentiment analysis task, as can be seen in Table 4.7 and Figure 4.3, show an increase in performance for all data augmentation methods when compared to the baseline performance of the original subset. None of the augmentation methods outperform the upper-bound.

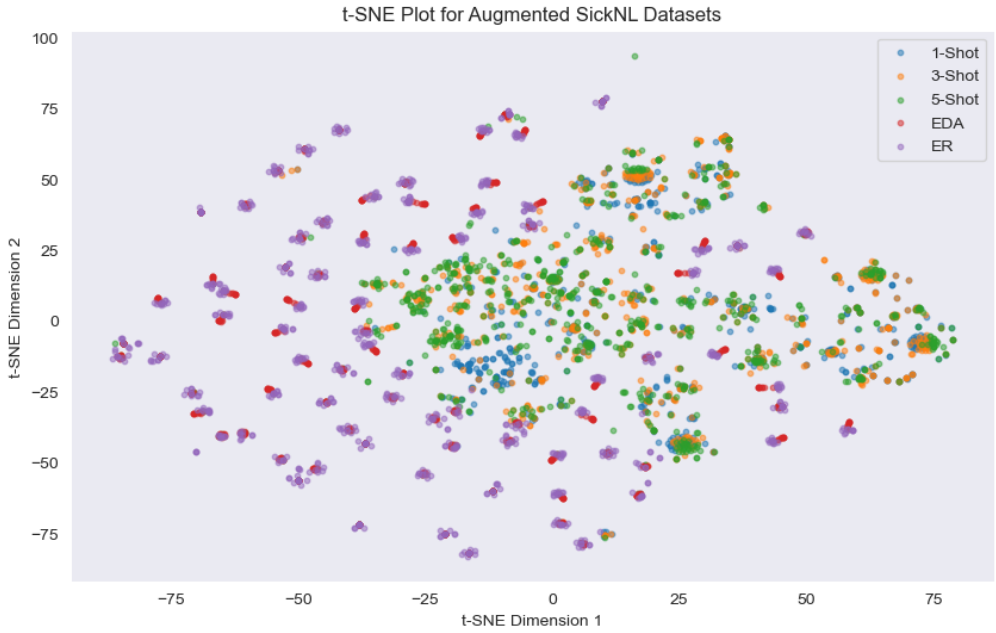

Figure 4.2: t-SNE plot for the augmented SickNL datasets.

The difference in performance is most notable in the smallest subset category, only using 0.1% of the full dataset.

For this subset size, the proposed data augmentation method outperforms not only the original subset, but also all of the alternative data augmentation methods substantially, with accuracy scores for LLM augmentation ranging between 0.81 and 0.83, while the alternative methods only achieve a maximum accuracy of 0.72 using embedding replacement. For this subset size, the 1-shot method achieves a higher performance than the 3- or 5-shot methods.

The increased subset size of 0.3% show less of a difference in performance between the different data augmentation methods, all ranging between 0.84 and 0.85, outperforming the original subset accuracy of 0.76. The 1-shot method still achieves higher performance than the 3- and 5-shot methods, but is followed up by the EDA method, achieving a nearly identical accuracy score of 0.85.

For the largest subset size, all methods outperform the original subset by almost 0.1 absolute accuracy score, while no significant difference can be seen between the different methods themselves.

4.2.2 SickNL

The results of the natural language inference experiment can be seen in Table 4.8 and Figure 4.4, showing mixed results over the different subset sizes. While the upper-bound remains the model with the highest accuracy scores, similar to the DBRD results, the original subset outperforms some of the models trained on the augmented datasets.

Form	Size	Accuracy	Precision	Recall	F1
Original	0.1%	0.67311	0.71337	0.67311	0.65693
	0.3%	0.76169	0.75751	0.78107	0.76169
	1.0%	0.81205	0.81170	0.81439	0.81205
1-Shot	0.1%	0.82644	0.82674	0.82644	0.8264
	0.3%	0.85477	0.85525	0.85477	0.85472
	1.0%	0.83993	0.84281	0.83993	0.83959
3-Shot	0.1%	0.82464	0.83002	0.82464	0.82392
	0.3%	0.84442	0.84712	0.84442	0.84412
	1.0%	0.84937	0.85558	0.84937	0.84871
5-Shot	0.1%	0.8134	0.81368	0.8134	0.81336
	0.3%	0.83813	0.83841	0.83813	0.8381
	1.0%	0.85926	0.85935	0.85926	0.85925
EDA	0.1%	0.68345	0.75073	0.68345	0.66069
	0.3%	0.84982	0.84963	0.85154	0.84982
	1.0%	0.86421	0.86513	0.86421	0.86412
ER	0.1%	0.72032	0.72522	0.72032	0.71879
	0.3%	0.83678	0.83681	0.83678	0.83678
	1.0%	0.85612	0.85709	0.85612	0.85602
BT	0.1%	0.68345	0.68641	0.68345	0.68219
	0.3%	0.84712	0.84858	0.84712	0.84696
	1.0%	0.84104	0.84945	0.84723	0.84754
Upper	0.1%	0.86061	0.86061	0.86061	0.86061
	0.3%	0.86331	0.87648	0.86331	0.8621
	1.0%	0.89793	0.90124	0.89793	0.89772

Table 4.7: Results of DBRD models

For the smallest subset size of 0.1%, the models trained on the GPT-augmented datasets underperform dramatically, achieving accuracy scores from 0.16 only up to 0.35, while the original subset achieves an accuracy score of 0.57. Further analysis shows that this is due to the fine-tuned models trained on the GPT-augmented datasets only predicting a single class, given the imbalanced dataset it is trained on. The other data augmentation methods do not suffer from a similar problem, and thus outperform the models fine tuned on the GPT-augmented datasets. as well as the model trained on the original subset.

For the following subset size of 0.3% the difference in performance between the GPT-augmented datasets decreases, while still achieving accuracy scores below the scores of the baseline model. The difference in performance between the 3 different GPT-augmented models shifts, as the 5-shot augmentation method only marginally increases its performance compared to the smaller subset size, while the 1-shot and 3-shot models boost their performance towards the baseline scores.

The largest subset size of 1.0% of the full dataset is the only subset size that sees all models trained on augmented datasets outperform the model trained on the original subset. The differences between the different augmentation methods is reduced, while the 1-Shot method and EDA

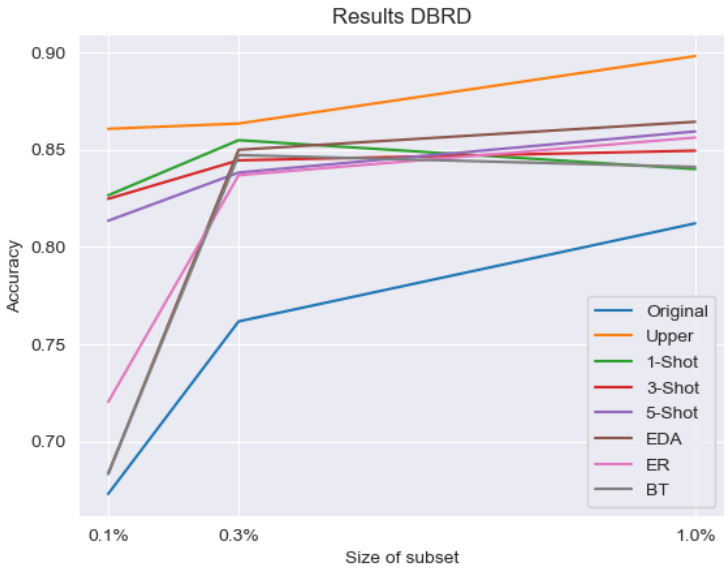

Figure 4.3: Plot of accuracy scores of models fine-tuned on the augmented DBRD subsets

both achieve the highest accuracy scores of 0.70, and the other augmentation methods achieving a score of 0.65.

Form	Size	Accuracy	F1	Precision	Recall
Original	0.1%	0.56869	0.24168	0.18956	0.33333
	0.3%	0.56869	0.24168	0.18956	0.33333
	1.0%	0.5903	0.30945	0.36555	0.36499
1-Shot	0.1%	0.16327	0.10895	0.20155	0.33943
	0.3%	0.52956	0.25043	0.21964	0.32051
	1.0%	0.70118	0.65328	0.71802	0.62141
3-Shot	0.1%	0.26559	0.20449	0.21426	0.32793
	0.3%	0.54872	0.25161	0.23024	0.3286
	1.0%	0.65695	0.59744	0.61263	0.58953
5-Shot	0.1%	0.35406	0.25906	0.2407	0.36618
	0.3%	0.41113	0.25929	0.25876	0.28133
	1.0%	0.64941	0.52394	0.60873	0.50998
EDA	0.1%	0.5689	0.24398	0.37483	0.33404
	0.3%	0.56217	0.37198	0.50953	0.38391
	1.0%	0.69874	0.66161	0.72636	0.62785
ER	0.1%	0.56869	0.24168	0.18956	0.33333
	0.3%	0.52303	0.41101	0.47252	0.40031
	1.0%	0.65817	0.61447	0.62089	0.60917
Upper	0.1%	0.56971	0.24563	0.38038	0.33487
	0.3%	0.72157	0.72594	0.72345	0.74242
	1.0%	0.77762	0.77064	0.79197	0.75353

Table 4.8: Results of SickNL models

Figure 4.4: Plot of accuracy scores of models fine-tuned on the augmented SickNL subsets

5 Discussion

In this final section I will conclude my thesis, starting of by giving an interpretation of the results in section 5.1, following this up with a comparison with previous research in section 5.2. Continuing with the identified limitations in section 5.3 and providing some ideas for further research in section 5.4. Finally, the conclusion of the thesis is given in section 5.5.

5.1 Interpretation of Results

5.1.1 Intrinsic Evaluation Results

The intrinsic evaluation of the generated datasets provides insights into the quality and diversity of the texts that different augmentation methods offer. The use of various metrics allows for a comprehensive analysis of the generated datasets, and all aspects will be considered in interpreting the results.

The manual analysis of the quality of the generated data shows that the GPT-based methods are capable of generating coherent texts, which containing grammatically correct sentences, in contrast to the generated sentences using the alternative methods like EDA and embedding replacement. While the label correctness for the DBRD dataset is also in line with expectations, this is not the case for the SickNL dataset. Although the model is able to generate correct labels for its generated sentence pairs most of the time, the mistakes it makes seem to indicate a lack of language knowledge.

The analysis of the most common words and sentences, presented in Table 4.3, Table 4.4, and Table 4.5, reveals similarities between the different tasks, highlighting the differences between GPT-augmentation methods and alternative methods. The alternative methods show comparable results to the original dataset, which is to be expected as they alter specific elements of the original text to achieve augmentation. Consequently, the analysis aligns well with the original dataset, as evidenced in the results for both the DBRD and SickNL datasets.

Conversely, the GPT-augmented datasets yield contrasting results, with their most common words and sentences showing less overlap with the original dataset. This indicates that GPT-based augmentation methods rely less on the provided samples and draw more from their original training data to generate new datapoints. This diversification enriches the augmented dataset with new information not present in the original data. This diversification is also evident in the MATTR results in Table 4.6, where the alternative methods show lower values compared to the GPT-augmentation methods, especially for the SickNL dataset.

However, this deviation from the original dataset may present challenges, as exemplified by the most common sentences for the DBRD, as shown in Table 4.5. The most common sentences in the GPT-augmented datasets are distinct from the original dataset but frequently occur in the

generated datapoints. This could suggest that while the GPT-augmented dataset introduces diversity not directly reliant on the original dataset, the generated datapoints still revolve around the data on which the model was trained. This repetitiveness is evident from the analysis of most common sentences.

The apparent reliance on its original training data rather than the given samples from the dataset being augmented is further highlighted by visualisations of the vector space of the augmented datasets, as shown in Figure 4.1 and Figure 4.2. The alternative augmentation methods form small, individual clusters, indicating limited variance centred around the original datapoints. In contrast, the GPT-augmented datasets do not exhibit a similar pattern, especially evident in the DBRD dataset, where the large cluster formed by GPT-augmented datapoints shows a narrow vector space and significant deviation from the original sentences, further emphasising the repetitive nature of this method.

This issue is less prominent for the SickNL dataset, where only a single sentence needs to be generated, making sentence repetition less problematic. Consequently, the representation of the vector space, as seen in Figure 4.2, is not as distinct as for the DBRD datasets.

The diversity brought by GPT-augmented datasets, as indicated by most of the previously discussed metrics, appears to have limitations. While they provide a variety of new datapoints that differ from the original dataset and may improve model performance, this diversity seems to remain constrained by the boundaries of the model's previous training data. This limitation could potentially lead to overfitting when applied to larger datasets.

5.1.2 Extrinsic Evaluation Results

The results of the sentiment analysis task, as can be seen in Figure 4.3 and Table 4.7, indicate a significant performance boost achieved through data augmentation methods compared to the baseline performance of the original subset. All the data augmentation techniques lead to improved results across different subset sizes.

Among the various data augmentation methods, the GPT-augmentation method stands out as the most effective. Regardless of the subset size, the models fine-tuned on the GPT-augmented datasets consistently outperforms all alternative methods. Notably, the performance improvement is most pronounced when dealing with the smallest subset, comprising only 0.1% of the full dataset. For this subset size, the GPT-augmented models achieves accuracy scores ranging from 0.81 to 0.83, while the alternative methods fall short, reaching a maximum accuracy of only 0.72 using embedding replacement.

While the difference between the performance of the models fine-tuned on the GPT-augmented datasets is substantial for the 0.1% dataset, this difference becomes negligible for the larger subset sized, even outperforming the GPT-based methods slightly for the largest subset size. This could to indicate the stagnation of the boost in performance that is achieve by augmenting the dataset using the GPT-based methods, compared to the boost the alternative methods provide.

The results of the natural language inference experiment, presented in Figure 4.4 and Table 4.8, reveal mixed outcomes across different subset sizes. Similar to the sentiment analysis task, the upper-bound model maintains the highest accuracy scores, indicating the best possible performance achievable on the dataset. However, the performance of models trained on augmented datasets differs from the sentiment analysis results.

The GPT-augmented datasets significantly underperform on the smallest subset of 0.1%, achieving accuracy scores ranging from a mere 0.16 to 0.35 compared to the original subset score of 0.57. The underlying issue of an imbalanced dataset, provides insight into a weakness of the GPT-based method over the alternative methods, as can be seen in Table 5.1. While the generation of augmented datasets for the alternative methods simply copy the label provided by the original sample, the GPT-based method generates its own labels based on the text it has generated. This has resulted in imbalanced augmented datasets, that do not follow the label distribution of the original subset it is augmenting. There is no clear reason why the GPT-based method is heavily skewed towards generating contradictions, eventhough it does not receive many contradiction examples. This might be due to the original data it is trained upon, or could be caused by the prompt it is given.

	Label	Original	1-Shot	3-Shot	5-Shot
0.1%	Neutral	6	15	17	26
	Contradiction	1	61	52	37
	Entailment	1	4	11	17
0.3%	Neutral	17	49	54	170
	Contradiction	4	167	148	40
	Entailment	3	24	38	30
1.0%	Neutral	53	132	162	161
	Contradiction	7	599	545	554
	Entailment	20	69	93	85

Table 5.1: Label Distribution for generated datasets using GPT-based method

A different issue is prominent for the alternative augmentation methods, which also do not increase the performance over the original dataset, except for the largest subset size. Given that these methods use the same label for their augmented datapoints as is provided by the sample, and the fact that the datapoints only consist of 2 sentences, changing a few words can lead to a switch in entailment. This is not an issue in the DBRD datasets, as these do not rely as much on individual words, as is required for natural language inference.

For the 1.0% subset size, all augmentation methods outperform the original subset. The issue of label distribution becomes less apparent with the increase of dataset size, thus resulting in an increased performance of the GPT-based models.

Overall, the natural language inference task is in general a harder task to perform than the DBRD sentiment analysis task, as is apparent by the overall lower scores for this task. It relies more on the intricacies of human language, which are more vulnerable to changes as provided by these augmentation methods. This shows that the deployment of augmentation methods should not be seen as universal as it is in other domains, and the performance increase is much more reliant on the specific task and dataset.

5.1.3 Overall Interpretation

Using LLM's as a method for augmenting low-resource Dutch datasets can provide a increase in overall size and diversification of the given dataset, but this unfortunately does not always lead to

better performance. The potential over reliance of the LLM method on its training data causes issues when the dataset size increases, and can cause even more problems for language tasks that it is unfamiliar with. For the augmentation method to work, the GPT model needs to have some previous understanding of the problem, as it seems like it does not gain enough information from the simple prompt and examples given.

Another large issue can be found in the independent nature of this method when it comes to label distribution. The method does not follow the distribution as given by the dataset it needs to augment, which could lead to unbalanced label distributions. Given the independent generation for each datapoint, it is not able to rely on the labels of the datapoints it has already generated, and can thus not be controlled during the generation process.

The results, both for the intrinsic evaluation as well as the extrinsic evaluation, indicate that although this method could be deployed in a successful matter, it is too unreliant to be deployed for any given dataset. The understanding of the model of the Dutch language appears to be too narrow to provide a wide variety in texts it can produce. It could however be used in a supervised manner, allowing to increase the size of a dataset, while keeping manual control over the quality of the datapoints generated.

5.2 Comparison with Previous Studies

While no similar studies have been performed for Dutch data augmentation, a comparison can be made with similar methods for the English language. It should to be noted however, that the model used in this study is mostly trained on English data, which can have an impact on the results.

Comparing the results of my experiments with the results of GPT3Mix [Yoo+21], shows similar results for the sentiment analysis task, as the GPT models outperform both the original dataset as well as the alternative methods used. However, the performance of EDA and back-translation is much more similar to the performance of the original dataset, while my findings indicate that these methods can also provide a significant performance boost. The study does not perform testing on a natural language inference task, nor does it provide any intrinsic evaluation.

The results for the DBRD experiments also provide a counter-argument for the hypothesis stating that the alternative data augmentation methods could not provide enough augmentation to boost the performance of the model [LWD20]. Eventhough the intrinsic evaluation shows that the datapoints are very similar to the datapoints from the original dataset, they still contribute to an increase in overall performance when used to fine-tune a language model.

5.3 Limitations

5.3.1 Dataset Generation

The augmentation of the datasets using GPT is done by providing a prompt, which provides limited context on the task at hand, together with providing 1, 3 or 5 examples. This prompt design, based on [Yoo+21], is used as it provides a general framework which should make it easier to apply the method for a variety of datasets for specific tasks. However, given the generality of the prompts, and the possible lack of understanding of the Dutch language of GPT, this could have also led to too general of outputs.

The choice of using the specific model GPT-3.5-turbo, could have also had an effect on the generated datasets. The newest GPT model, GPT-4 [Ope23], is stated to be better suited for multilingual use, and could thus have been a better suit for the experiments. This model has only recently been made available, and has thus not been used.

Unfortunately, no Dutch monolingual LLM's are available as of the current time. A Dutch monolingual LLM is specifically designed and trained to understand and generate Dutch text, making it inherently more accurate and contextually appropriate for Dutch data augmentation methods. In contrast, GPT-3.5, although a powerful language model, is mostly trained on the English language, and can thus not provide the knowledge and nuance required.

5.3.2 Tasks

The selection of the two datasets used in this thesis is based on their widespread adoption as standard benchmarks for evaluating the performance of Dutch language models[De +19; DWB20; DWB22]. However, they do not cover the entire array of possible natural language tasks. This study is limited to the experiments on a sentiment analysis task and a natural language inference task, which can both be identified as classification tasks.

The two datasets used in this thesis, have been used to generate different subsets of these datasets in order to mimic a low-resource datasets. This was done in order to get an understanding of the performance of the augmentation method on different dataset sizes. However, the method has not been tested on an actual low-resource dataset. This could thus have implications on the ecological validity of the experiments conducted, as it is unclear how well the method would perform when used on a real low-resource dataset, especially given the differing performance of both experiments.

5.3.3 Models

The performance of fine-tuned models can be influenced by several factors, both related to the dataset it is fine-tuned on as well as external variables. Due to the considerable number of models that had to be trained for both of the experiments, no hyper-parameter tuning was performed. Although this is in line with prior conducted research [Yoo+21], this could still lead to an increase in performance, and is thus a limitation of the current study. Despite this limitation, the results and conclusions drawn from the experiments remain valuable and informative for the research context.

Given the limited size of the datasets that were used for fine-tuning the model, generating a multitude of datasets for each subset size and augmentation method, and taking the average of the performance of these models could give an indication of the variability of the performance. Averaging out the performance metrics can give a better impression of usability of the method, as high variance would indicate an unreliable method. As a result, the study's findings might not fully capture the potential variability in the model's performance.

5.4 Future Work

The study conducted in this thesis has its limitations, as mentioned in section 5.3. To combat these identified issues, future research into this topic could be conducted, to gain a better understanding of the possibilities of Dutch Data Augmentation using LLM's. I will provide possible directions for this research in the following subsections.

5.4.1 Controlling Label Distribution

To combat the unbalanced label distribution as seen in Table 5.1, that can result when using the LLM-based augmentation method, further research should be conducted into better control mechanisms. This could be done by exploring different prompting techniques, limiting the prompt to only allow the generation of certain labels to maintain an identical label distribution. This control could result in more reliable methods for data augmentation, as the label distribution from the original dataset would be left intact.

5.4.2 Alternative Models

Using GPT-3.5 as the model to use for the data augmentation methods, has proven to yield mixed results. Although the choice for this model is well-considered, as the largest available LLM that was trained on Dutch data, the landscape of LLM's is evolving fast. As previously mentioned, an updated version of GPT has been released to the public, in the form of GPT-4 [Ope23]. Even-though no information is given on the training data that was used to train this model, the report from the creators of OpenAI has stated increased support for multilingual models.

Furthermore, research into the usability of open-source models could benefit this approach even further, as this would allow organisations to self-host their own LLM's, and thus wouldn't have to sent over their dataset to OpenAI. With alternative models like LLaMa [Tou+23] and BLOOM [Sca+22] being created, we should pay attention to their capabilities to generate Dutch text, and should test the possibility of implementing these alternatives for data augmentation goals.

5.4.3 Alternative Language Tasks

In order to gain a better understanding of the capabilities of the proposed data augmentation method, alternative datasets should be tested. While classification tasks are commonplace within the natural language task domain, alternative tasks require different aspects for the generation of additional datapoints, and should thus be evaluated. This would require a change in the prompt template, thus requiring further exploration.

Possible examples of alternative datasets, could be a Part of Speech tagging task [Van+13] or a Named Entity Recognition task [Tjo02]. These tasks are often used to evaluate English data augmentation methods as well, which could be used to compare the difference in performance. A dataset that would be specifically interesting is the die/dat disambiguation task [ALM20], as this ambiguity is typical for the Dutch language, and would thus test the capabilities of the LLM specifically for this language.

5.4.4 Deployment in Low-Resource Setting

The setup of the experiments for this thesis has been controlled, by mimicking low-resource scenarios by taking different sized subsets from full datasets. Although this approach is required to get a better grasp on the performance of the smaller datasets and the impact of the data augmentation method on this performance, future research could look into testing out this approach in a real low-resource setting. Not only could this highlight unseen issues, that are not evident in a mimicked scenario, but this could also lead to direct results that can impact a real life scenario.

5.5 Conclusion

This thesis aimed to tackle the main research question: *"How does using LLM's as an generative augmentation method for Dutch language tasks compare to other augmentation methods?"*. To answer this question, two aspects of this proposed augmentation method were researched, the intrinsic aspect, looking into the quality and diversity of the generated data itself, as well as the extrinsic aspect, which looked into the performance difference when the augmented datasets were used to fine-tune a language model.

The proposed method was tested using three possible approaches, by providing the Large Language Model with either one, three or five example sentences, in order to gain insight into the impact this difference in prompting could make. To compare this new approach to alternative methods for Dutch data augmentation, three methods were used: Easy Data Augmentation, Embedding Replacement and Back-Translation. These resulting six data augmentation methods were tested on two different Dutch classification task, a sentiment analysis task and a natural language inference task. For both datasets belonging to each task, three different subsets were taken, in order to mimic a low-resource setting in which data augmentation methods would be used.

The intrinsic evaluation highlighted the quality of the augmented datapoints generated using the LLM, although the label correctness could not be fully guaranteed. It also showed that even though the generated datapoints provided more diversity to the original dataset, the generated text appeared to be of repetitive nature, and would potentially rely too much on its underlying training data, and less on the provided samples.

The results of the extrinsic evaluation show mixed results for the proposed augmentation method, as the performance of the fine-tuned model on the augmented dataset using the LLM differed substantially between the two tasks. While the results for the sentiment analysis task indicate that the proposed method not only outperforms the original dataset, but for the smallest subset size also outperforms the alternative methods, this is not the case for the natural language inference task. The proposed method heavily underperforms for the smaller subsets, and eventhough it boosts the performance for the largest subset size, it does not outperform the alternative methods.

In conclusion, using LLM's as a generative augmentation method for Dutch language tasks does not outperform alternative methods in every scenario. While it is able to generate qualitative texts, it is not able to provide the required control that the other methods provide. However, with the current rate of acceleration in the field of Large Language Models, this conclusion is not set in stone. While the results indicate both promising aspects and limitations, the field of Large Language Models remains dynamic, leaving room for future advancements and discoveries that could potentially unlock the full potential of LLMs for Dutch data augmentation purposes.

Bibliography

[AG18] S. T. Aroyehun and A. Gelbukh. "Aggression Detection in Social Media: Using Deep Neural Networks, Data Augmentation, and Pseudo Labeling". In: *Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018)*. Association for Computational Linguistics, Santa Fe, New Mexico, USA, 2018, pp. 90–97.

[ALM20] L. Allein, A. Leeuwenberg, and M. Moens. "Binary and Multitask Classification Model for Dutch Anaphora Resolution: Die/Dat Prediction". *CoRR* abs/2001.02943, 2020.

[Ana+20] A. Anaby-Tavor, B. Carmeli, E. Goldbraich, A. Kantor, G. Kour, S. Shlomov, N. Tepper, and N. Zwerdling. "Do not have enough data? Deep learning to the rescue!" In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 34. 05. 2020, pp. 7383–7390.

[BB18] Y. Belinkov and Y. Bisk. "Synthetic and Natural Noise Both Break Neural Machine Translation". In: *International Conference on Learning Representations*. 2018.

[BKR22] M. Bayer, M.-A. Kaufhold, and C. Reuter. "A survey on data augmentation for text classification". *ACM Computing Surveys* 55:7, 2022, pp. 1–39.

[Bow+15] S. R. Bowman, G. Angeli, C. Potts, and C. D. Manning. "A large annotated corpus for learning natural language inference". In: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, 2015.

[Bro+20] T. Brown, 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. "Language Models are Few-Shot Learners". In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin. Vol. 33. Curran Associates, Inc., 2020, pp. 1877–1901.

[Buc19] B. Buchanan. "Artificial intelligence in finance", 2019.

[CCG20] A. Celikyilmaz, E. Clark, and J. Gao. "Evaluation of text generation: A survey". *arXiv:2006.14799*, 2020.

[Cha+02] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. "SMOTE: synthetic minority over-sampling technique". *Journal of artificial intelligence research* 16, 2002, pp. 321–357.

[CM10] M. A. Covington and J. D. McFall. "Cutting the Gordian knot: The moving-average type–token ratio (MATTR)". *Journal of quantitative linguistics* 17:2, 2010, pp. 94–100.

[Cou18] C. Coulombe. "Text Data Augmentation Made Simple By Leveraging NLP Cloud APIs". *CoRR* abs/1812.04718, 2018.

[De +19] W. De Vries, A. van Cranenburgh, A. Bisazza, T. Caselli, G. van Noord, and M. Nissim. "Bertje: A dutch bert model". *arXiv:1912.09582*, 2019.

[Dev+18] J. Devlin, M. Chang, K. Lee, and K. Toutanova. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". *CoRR* abs/1810.04805, 2018.

[DWB20] P. Delobelle, T. Winters, and B. Berendt. "RobBERT: a Dutch RoBERTa-based Language Model". In: *Findings of the Association for Computational Linguistics: EMNLP 2020*. Association for Computational Linguistics, Online, 2020, pp. 3255–3265.

[DWB22] P. Delobelle, T. Winters, and B. Berendt. "RobBERT-2022: Updating a Dutch Language Model to Account for Evolving Language Use". *arXiv:2211.08192*, 2022.

[Eva21] K. Evanko-Douglas. *Can GPT-3 Create Synthetic Training Data for Machine Learning Models?* 2021.

[He+22] X. He, I. Nassar, J. Kiros, G. Haffari, and M. Norouzi. "Generate, Annotate, and Learn: NLP with Synthetic Text". *Transactions of the Association for Computational Linguistics* 10, 2022, pp. 826–842. ISSN: 2307-387X.

[Hu+20] H. Hu, Q. Chen, K. Richardson, A. Mukherjee, L. S. Moss, and S. Kuebler. "MonaLog: a Lightweight System for Natural Language Inference Based on Monotonicity". In: *Proceedings of the Society for Computation in Linguistics 2020*. Association for Computational Linguistics, New York, New York, 2020, pp. 334–344.

[JPL19] A. Jain, B. Paranjape, and Z. C. Lipton. "Entity Projection via Machine Translation for Cross-Lingual NER". In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Association for Computational Linguistics, Hong Kong, China, 2019, pp. 1083–1092.

[KBM11] O. Kolomiyets, S. Bethard, and M.-F. Moens. "Model-portability experiments for textual temporal analysis". In: *Proceedings of the 49th annual meeting of the association for computational linguistics: human language technologies*. Vol. 2. ACL; East Stroudsburg, PA. 2011, pp. 271–276.

[Kob18] S. Kobayashi. "Contextual Augmentation: Data Augmentation by Words with Paradigmatic Relations". In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*. Association for Computational Linguistics, New Orleans, Louisiana, 2018, pp. 452–457.

[Liu+19] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov. "Roberta: A robustly optimized bert pretraining approach". *arXiv:1907.11692*, 2019.

[Liu+20] Y. Liu, J. Gu, N. Goyal, X. Li, S. Edunov, M. Ghazvininejad, M. Lewis, and L. Zettlemoyer. "Multilingual Denoising Pre-training for Neural Machine Translation". *Transactions of the Association for Computational Linguistics* 8, 2020, pp. 726–742. ISSN: 2307-387X.

[Liu+21] L. Liu, B. Ding, L. Bing, S. Joty, L. Si, and C. Miao. "MulDA: A multilingual data augmentation framework for low-resource cross-lingual NER". In: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. 2021, pp. 5834–5846.

[LWD20] S. Longpre, Y. Wang, and C. DuBois. "How Effective is Task-Agnostic Data Augmentation for Pretrained Transformers?" In: *Findings of the Association for Computational Linguistics: EMNLP 2020*. Association for Computational Linguistics, Online, 2020, pp. 4401–4411.

[Maa+11] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts. "Learning Word Vectors for Sentiment Analysis". In: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, Portland, Oregon, USA, 2011, pp. 142–150.

[Mar+14] M. Marelli, S. Menini, M. Baroni, L. Bentivogli, R. Bernardi, R. Zamparelli, et al. "A SICK cure for the evaluation of compositional distributional semantic models." In: *Lrec*. Reykjavik. 2014, pp. 216–223.

[Mil95] G. A. Miller. "WordNet: a lexical database for English". *Communications of the ACM* 38:11, 1995, pp. 39–41.

[Ope23] OpenAI. *GPT-4 Technical Report*. 2023.

[PW17] L. Perez and J. Wang. "The Effectiveness of Data Augmentation in Image Classification using Deep Learning". *CoRR* abs/1712.04621, 2017.

[Raa19] S. Raaijmakers. "Artificial intelligence for law enforcement: challenges and opportunities". *IEEE security & privacy* 17:5, 2019, pp. 74–77.

[Rad+18] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever, et al. "Improving language understanding by generative pre-training". *OpenAI blog*, 2018.

[Rad+19] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, et al. "Language models are unsupervised multitask learners". *OpenAI blog* 1:8, 2019, p. 9.

[Ric87] B. Richards. "Type/token ratios: What do they really tell us?" *Journal of child language* 14:2, 1987, pp. 201–209.

[Sca+22] T. L. Scao, A. Fan, C. Akiki, E. Pavlick, S. Ilić, D. Hesslow, R. Castagné, A. S. Luccioni, F. Yvon, M. Gallé, et al. "Bloom: A 176b-parameter open-access multilingual language model". *arXiv:2211.05100*, 2022.

[Sha+20] S. Sharma, Y. Zhang, J. M. Ríos Aliaga, D. Bouneffouf, V. Muthusamy, and K. R. Varshney. "Data augmentation for discrimination prevention and bias disambiguation". In: *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*. 2020, pp. 358–364.

[SK19] C. Shorten and T. M. Khoshgoftaar. "A survey on image data augmentation for deep learning". *Journal of big data* 6:1, 2019, pp. 1–48.

[Szo19] P. Szolovits. *Artificial intelligence in medicine*. Routledge, 2019.

[TED16] S. Tulkens, C. Emmery, and W. Daelemans. "Evaluating Unsupervised Dutch Word Embeddings as a Linguistic Resource". In: *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*. European Language Resources Association (ELRA), Portorož, Slovenia, 2016, pp. 4130–4136.

[Tjo02] E. F. Tjong Kim Sang. "Introduction to the CoNLL-2002 Shared Task: Language-Independent Named Entity Recognition". In: *Proceedings of the 6th Conference on Natural Language Learning - Volume 20*. COLING-02. Association for Computational Linguistics, USA, 2002, pp. 1–4.

[Tou+23] H. Touvron, 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. *LLaMA: Open and Efficient Foundation Language Models*. 2023.

[Van+13] G. Van Noord, G. Bouma, F. Van Eynde, D. De Kok, J. Van der Linde, I. Schuurman, E. T. K. Sang, and V. Vandeghinste. "Large scale syntactic annotation of written Dutch: Lassy". *Essential speech and language technology for Dutch: results by the STEVIN programme*, 2013, pp. 147–164.

[VH08] L. Van der Maaten and G. Hinton. "Visualizing data using t-SNE." *Journal of machine learning research* 9:11, 2008.

[VV19] B. Van der Burgh and S. Verberne. "The merits of Universal Language Model Fine-tuning for Small Datasets–a case with Dutch book reviews". *arXiv:1910.00896*, 2019.

[Wan+18] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. Bowman. "GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding". In: *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*. Association for Computational Linguistics, Brussels, Belgium, 2018, pp. 353–355.

[Whi21] D. Whitfield. "Using GPT-2 to Create Synthetic Data to Improve the Prediction Performance of NLP Machine Learning Classification Models". *CoRR* abs/2104.10658, 2021.

[WM21] G. Wijnholds and M. Moortgat. "SICK-NL: A Dataset for Dutch Natural Language Inference". In: *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*. Association for Computational Linguistics, Online, 2021, pp. 1474–1479.

[Wu+19] X. Wu, S. Lv, L. Zang, J. Han, and S. Hu. "Conditional bert contextual augmentation". In: *Computational Science–ICCS 2019: 19th International Conference, Faro, Portugal, June 12–14, 2019, Proceedings, Part IV 19*. Springer. 2019, pp. 84–95.

[WY15] W. Y. Wang and D. Yang. "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*. 2015, pp. 2557–2563.

[WZ19] J. Wei and K. Zou. "EDA: Easy Data Augmentation Techniques for Boosting Performance on Text Classification Tasks". In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Association for Computational Linguistics, Hong Kong, China, 2019, pp. 6382–6388.

[Xie+17] Z. Xie, S. I. Wang, J. Li, D. Lévy, A. Nie, D. Jurafsky, and A. Y. Ng. "Data Noising as Smoothing in Neural Network Language Models". In: *Proceedings of the 5th International Conference on Learning Representations, ICLR 2017,* 2017.

[Yoo+21] K. M. Yoo, D. Park, J. Kang, S.-W. Lee, and W. Park. "GPT3Mix: Leveraging Large-scale Language Models for Text Augmentation". In: *Findings of the Association for Computational Linguistics: EMNLP 2021*. Association for Computational Linguistics, Punta Cana, Dominican Republic, 2021, pp. 2225–2239.

[Yu+18] A. W. Yu, D. Dohan, M. Luong, R. Zhao, K. Chen, M. Norouzi, and Q. V. Le. "QANet: Combining Local Convolution with Global Self-Attention for Reading Comprehension". *CoRR* abs/1804.09541, 2018.

[Yue+23] X. Yue, H. Inan, X. Li, G. Kumar, J. McAnallen, H. Shajari, H. Sun, D. Levitan, and R. Sim. "Synthetic Text Generation with Differential Privacy: A Simple and Practical Recipe". In: *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Toronto, Canada, 2023, pp. 1321–1342.

[Zha+18] J. Zhao, T. Wang, M. Yatskar, V. Ordonez, and K.-W. Chang. "Gender Bias in Coreference Resolution: Evaluation and Debiasing Methods". In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*. Association for Computational Linguistics, New Orleans, Louisiana, 2018, pp. 15–20.

[Zha+21] Z. Zhao, E. Wallace, S. Feng, D. Klein, and S. Singh. "Calibrate before use: Improving few-shot performance of language models". In: *International Conference on Machine Learning*. PMLR. 2021, pp. 12697–12706.

[ZZL15] X. Zhang, J. Zhao, and Y. LeCun. "Character-level convolutional networks for text classification". *Advances in neural information processing systems* 28, 2015.