SCHOOL OF COMPUTATION, INFORMATION
AND TECHNOLOGY - INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Master's Thesis in Informatics

# Investigating Data-to-Text Approaches to Achieve Diversity of Generated Marketing Text in the Music Industry

**Alexandre Mercier**

SCHOOL OF COMPUTATION, INFORMATION AND TECHNOLOGY - INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Master's Thesis in Informatics

# Investigating Data-to-Text Approaches to Achieve Diversity of Generated Marketing Text in the Music Industry

# Untersuchung von Ansätzen zur Umwandlung von Daten in Text, um Vielfalt der in der Musikindustrie Erstellten Marketingtexte zu Erreichen.



| | |
|---|---|
| Author: | Alexandre Mercier |
| Supervisor: | Prof. Dr. Florian Matthes |
| Advisor: | Anum Afzal |
| Submission Date: | 15th of January 2024 |

I confirm that this master's thesis in informatics is my own work and I have documented all sources and material used.

Munich, 15th of January 2024 Alexandre Mercier

# Acknowledgments

I take this opportunity to thank all those who have supported, motivated and helped me during my academic journey, specifically during the composition of this thesis. I am especially thankful to Prof. Dr. Florian Matthes for enabling me to work on this captivating topic. Additionally, I would like to thank M.Sc. Anum Afzal and Dr. Maximilian Blaschke for their guidance, advice and continued support during this project. Lastly, I would like to thank my family and friends for their unwavering support and encouragement throughout my studies.

# Abstract

Online platforms are increasingly interested in using Data-to-Text technologies to generate content and help their users. Unfortunately, traditional generative methods often fall into repetitive patterns, resulting in monotonous galleries of texts after only a few iterations. This thesis thus investigates LLM-based data-to-text approaches to automatically generate marketing texts that are both of sufficient quality and diverse enough for broad adoption. Our case study utilizes a platform designed for musicians and event organizers to create machine-generated band descriptions that could be used to compete for contracts. Different data-to-text approaches and techniques are investigated to generate engaging and diverse texts from a limited dataset. Multiple Language Models such as T5, GPT-3.5, GPT-4, and LLaMa2 are leveraged in conjunction with fine-tuning, few-shot, and zero-shot approaches to set a baseline for diverse marketing texts. After developing a metric to measure the diversity of a set of texts and using G-eval as a quality and engagingness metric, the approaches can be modified and compared to improve diversity while maintaining a similar quality. We propose solutions both at the prompting stage and the decoding stage and evaluate their impact on diversity. This research extends its relevance beyond the music industry, proving beneficial in various fields where repetitive automated content generation is prevalent.

# Contents

**Acknowledgments** **iii**

**Abstract** **iv**

**1. Introduction** **1**
1.1. Motivation 1
1.2. Problem Statement 2
1.3. Research Questions 5
1.4. Tasks 6
1.5. Outline 6

**2. Fundamentals** **7**
2.1. Data-to-Text over the Years 7
2.1.1. Transformers 9
2.1.2. Pre-trained Language Models 10
2.2. Evaluation of Text Generation 11
2.2.1. Defining Quality 12
2.2.2. Diversity in Text Generation 13

**3. Related Work** **14**

**4. Dataset** **17**
4.1. Data Preparation 18
4.2. Data Exploration 20
4.3. Parameter Dataset 22
4.4. Diversity Dataset 22
4.5. Data Format and Encoding 23

**5. Implementation** **25**
5.1. Models 25
5.2. Evaluation 25
5.2.1. Pipeline 25
5.2.2. G-Eval 26
5.2.3. Diversity Analysis 28
5.3. Experiments 29
5.3.1. Parameter Experiments 30
5.3.2. Diversity Experiments 31

5.3.3. Task2: Adding Gigs . . . 31
5.3.4. Task3: Regional Summaries . . . 32

**6. Results** **33**
6.1. RQ1: Choice of Technologies . . . 33
6.2. RQ2: Comparing Similar Text/Evaluating the Diversity . . . 34
6.3. RQ3: Diversity of Generated Text . . . 35
6.4. RQ4: NLG Compared to Human Written Text . . . 40
6.5. Task2: Adding Gigs . . . 41
6.6. Task3: Regional Summaries . . . 42

**7. Discussion** **44**
7.1. RQ1: Choice of Technologies . . . 44
7.2. RQ2: Comparing Similar Text/Evaluating the Diversity . . . 45
7.3. RQ3: Diversity of Generated Text . . . 45
7.4. RQ4: NLG Compared to Human-Written Text . . . 47
7.5. Task 2 and Task 3: Pilot Experiments . . . 48
7.6. Thoughts on implementing a Data-to-Text system in industry . . . 48
7.7. Future research . . . 49

**8. Conclusion** **51**

**A. Addenda** **52**
A.1. Example Prompt . . . 52
A.2. Geval . . . 53
A.2.1. Geval_engagingness . . . 53
A.2.2. Geval_fluency . . . 54
A.2.3. Geval_naturalness . . . 55
A.2.4. Geval_informativeness . . . 56
A.2.5. Geval_quality . . . 57
A.3. Sample Output . . . 59

**List of Figures** **62**

**List of Tables** **63**

**Acronyms** **64**

**Bibliography** **65**

# 1. Introduction

## 1.1. Motivation

In an age where digital transformation is reshaping industries and artificial intelligence is seemingly introduced into every aspect of life, online platforms are adapting their services to simplify their users' interactions with them. As most platforms' attractiveness relies on user-generated content, they must provide users with the tools to create relevant content. Recent advances in Large Language Models (LLMs) have made it easier to generate text of sufficient quality for mass adoption [1, 2] as shown by the incredibly rapid adoption speed of chatGPT, which reached 100 million users in under two months [3]. Integrating such technologies into platforms can help users create content more easily and efficiently but they also risk affecting the quality and richness of said content.

In fact, despite improving over older models, even the most advanced LLMs still suffer from repetitions or replication of training data without necessarily understanding the underlying context [4, 5]. Therefore, careful consideration is necessary when integrating these technologies, which might explain why many user-facing systems are designed as chatbots with human input. For maximal ease-of-use and a seamless integration, it would however be preferable to include known data and reduce required user input to a minimum, and in the case of automated content generation, no user input can be considered at all. Reducing human input and relying increasingly on structured data can lead to a loss of diversity in the generated content.

Diversity is crucial in the context of marketing on platforms, where competing products and services are often listed side-by-side and where descriptions for said products or services are usually the most distinguishing factor between competitors. In this case, the repetition of similar texts becomes apparent and can lead to a loss of trust in the platform. For this reason, basic systems like templates are of limited use. Jentzsch et al. [4] have also shown that for the complex domain of humour, ChatGPT [6] repeats variations of the same 25 jokes, which is a high enough number to fool users into thinking that each joke is unique, but probably not diverse enough when marketing descriptions are generated for a platform with possibly hundreds or thousands competitors.

In Figure 1.1 a simplified example is shown for a platform [7] aimed at connecting musicians with event organizers. The platform is designed to be used by both parties, with the content or, more abstractly, the product being generated by the musicians. One of the first steps from a band's point of view is to create a profile, which includes a self-description. Since writing about themselves is a time-consuming process, many bands postpone this step or write a short text, which is usually not very appealing. This is where a virtual assistant could help, by generating a description based on the data provided by the band during the profile creation process. Figure 1.1 shows two possible approaches to this problem. The first is to use templates, which are pre-written texts that are filled with the band's data. This approach is simple and can be implemented with little effort, but its capabilities are also minimal. The second approach is to use a data-to-text system, which can generate text based on the data provided. This approach is more complex and requires more effort to implement, but it is also more flexible and can generate more diverse texts.

As the sign-up process collects similar data for each band, generating diverse texts from a limited set of data is one of the main challenges here. Ideally, such a system can lower the entry bar for new users to join a platform, increase the quality of service and, in turn, improve the platform's attractiveness to all participants.

This thesis delves into modern applications of Natural Language Generation (NLG) technologies and their transformative impact on marketing strategies, specifically focusing on a platform designed for musicians and bands. It is an exploration at the intersection of technology, creativity, and diversity, aiming to lay the groundwork for relevant and engaging generation of content from structured data.

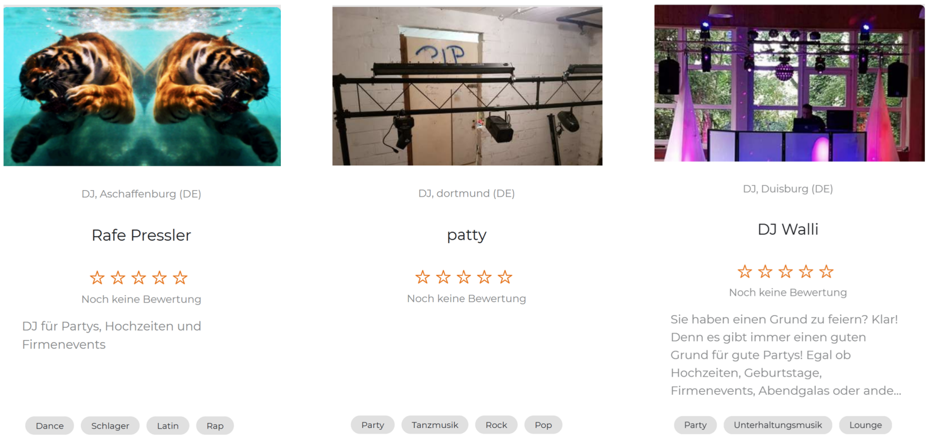
Most studies so far have concentrated on reducing repetitions or controlling text diversity inside a single generated sample. In contrast, this study aims to avoid similar structures and increase text diversity in between samples. By comparing and contrasting various approaches, the study aims to identify the most effective techniques for achieving diverse and engaging marketing content within the music industry context.

## 1.2. Problem Statement

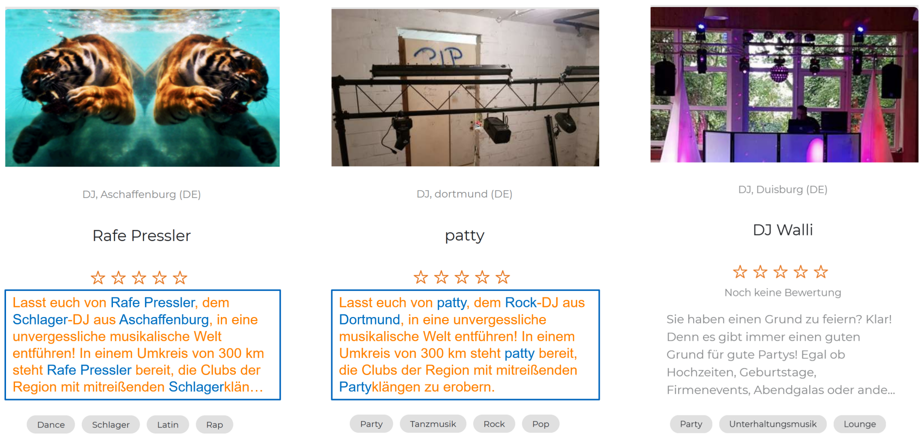
This thesis is at the intersection of NLG, marketing and data science. It aims at advancing diversity-promoting algorithms in the context of marketing and promotion. It relies on a limited dataset of structured data, which will serve as an example during experiments that should be transferable to similar tasks in other contexts where the available data is scarce.

**Problem 1 - Data Limitations** Data-to-text approaches obviously rely on the provided data. Structured data is usually limited in size and scope, which will limit the generated content to a subset of the possible variations. This is especially true for the music industry, where the history of a band is usually not recorded in a database but plays a pivotal role to judge a band's experience. Adding onto that, our dataset is not curated as it is taken straight from a platform's database. This means that the time-consuming step of dataset annotation and sorting is required before any fine-tuning can be done. If the refined dataset becomes too small and complicated, fine-tuning might not be possible at all or simply worsen the quality of the generated text. Finally, how the data is presented to models will impact the generation process. This thesis will explore two different data formats and their impact on the generated text.

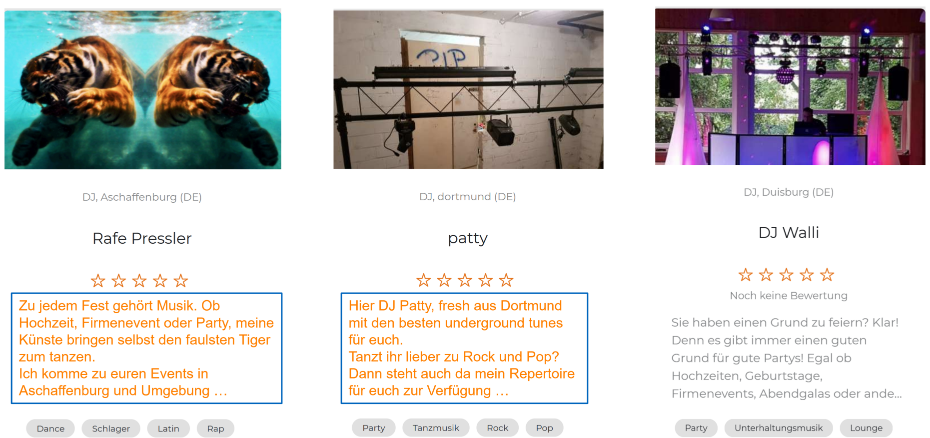
**Problem 2 - Diversifying Content** While the first problem is a limitation of the dataset, the second problem is a limitation of the models. Since the input might be similar for many bands, it is to be expected that their output will contain repetitions. At different stages of




(a) Original descriptions: only the third profile is informative and engaging. Only a subset of the bands are considered by event organizers so there is less competition.




(b) Descriptions generated from templates can increase a formation's attractiveness but quickly become repetitive.




(c) Descriptions generated by GPT-3.5 [1] ideally become indistinguishable from human-written content in terms of engagingness and information content. Event organizers have more choices, and all bands are relevant.

Figure 1.1.: Comparing three competing DJ profiles with *Original Descriptions* (a), *Template Generated Descriptions* (b) and *AI Generated Descriptions* (c). Modified screenshot from [7]

the generation process, measures can be taken to influence text diversity, from modifying the input, over changing parameters, down to modifications in the decoding stage. This thesis will explore the impact of some of these techniques on the diversity and quality [8] of the generated text. Another approach is to artificially add datapoints by generating knowledge [9] or implementing Chain-of-Thought (CoT) techniques.

**Problem 3 - NLG Evaluation** This is a two-pronged problem, as a set of generated texts should be not only diverse with limited repetitions but also engaging and informative as users might not use it otherwise. Many metrics exist to evaluate a text's quality or other aspects, but most rely on a reference text. This applies to most tasks and even works for some NLG tasks, but reference texts would be too different for longer and free-form text. Some steps, like verifying the factuality of a text, can be used with the data at hand, but others, like the engagingness of a text, do not have a reference. Therefore reference-free metrics or manual evaluation will have to be used for the quality evaluation.
Unlike previously, the diversity evaluation sets the focus on a group of generated texts. This requires a measure of similarity between samples. Many metrics exist to compare two sentences or paragraphs but they will have to be adapted as a base to build a metric for a group of texts.

**Scope:**

This thesis focuses on investigating the impact of methods to diversify generated text in a context of constrained data. Throughout the research process, multiple approaches will be compared and evaluated. The results will be used to identify the most effective techniques for achieving diverse and engaging marketing content within the music industry context.

To achieve this, multiple stages are necessary, starting with the data collection, annotation, and preparation, followed by the creation of two evaluation pipelines, one for diversity and one for quality. The third major step is selecting relevant models and methods, crafting experiments, and finally, evaluating and comparing the results.

Once a good and diverse system is found, we wish to expand the dataset by adding events or gigs each band participated in to test the abstraction capabilities of the models and test whether they are able to prioritize the most important events. Finally, we want to test if the models can identify patterns in the data and use them to generate more engaging regional summaries.

The overarching goal of this research is to contribute to the development of a virtual band manager or assistant - a tool that not only aids in administrative tasks but also plays a crucial role in marketing and promotion. By harnessing the power of data-to-text technologies, this virtual assistant will hopefully provide tailored, diverse, and engaging descriptions for bands and musicians, elevating their presence on digital platforms and ensuring they resonate with a broader audience. Event organizers also benefit from this increased choice, leading to increased platform usage and possibly growth.

## 1.3. Research Questions

### RQ1: Choice of Technologies

*Question 1: Which generative data-to-text approach yields the best overall results?*
Data-to-text is a research field with a long history and many different approaches. This study focusses on the use of LLMs and aims to compare models, techniques, data formats and parameters to find the best approach for the given task. This step is essential to the thesis as some methods offer more control over the generation, while others attempt to become indistinguishable from human-written text. We will conduct a literature review to identify the most promising approaches and use some of them in the experiments.

### RQ2: Comparing Similar Text/Evaluating the Diversity

*Question 2: How can we compare similar generated text? How can we measure diversity in generated content?*
We will have to quantify how similar not just two texts are, but expand it to a set of them. By developing a diversity measuring metric we will be able to evaluate the effectiveness of diversity-promoting techniques and compare them to each other. Ultimately, it will lead to a better understanding of required changes on current data-to-text techniques.

### RQ3: Diversity of Generated Text

*Question 3: How can creativity/variety of generative models be controlled and promoted?*
Through a series of experiments, we will be able to judge a technique's impact on the diversity of the generated text. This will allow us to identify how to properly promote diversity from models that are not inherently designed to do so.

### RQ4: NLG Compared to Human Written Text

*Question 4: Are generated texts as fluent and coherent as human written text?*
Ideally, our results should be indistinguishable from human-written text. Due to limitations in the available data, the performance of the models and possible influences of diversity-promoting techniques this is not a given. We will have to investigate how to evaluate the quality of the generated text and compare it to human-written text and references.

The code for this thesis is available on GitHub: `https://github.com/AleMer97/DivGen.git`. The repository contains the code for the evaluation pipelines, the experiments, and the visualizations used in thesis itself. The dataset is not public, but a short synthetic version is available.

## 1.4. Tasks

In addition to the main research on diversity-promoting techniques in NLG, we also integrated two more tasks into the thesis, although they are discussed in less detail. Here is a quick overview of all three tasks and their goals:

| | Description | Research goal |
|---|---|---|
| Task1 | Generate band description for their profile. | Quantify and promote form diversity in text while maintaining quality. |
| Task2 | Create articles for the band by incorporating recent gigs. | Test the abstraction capabilities of models by grouping gigs. |
| Task3 | Create regional summaries, grouping bands. | Take the abstraction a level higher, combining many datapoints |

Table 1.1.: The three tasks we worked on, with task1 being the main one.

Task 2 is quite similar to Task 1, but adding a varying number of events to the input data will test the abstraction capabilities of the models. While the underlying data is quite limited, we want to encourage models to filter out and prioritise just the key concerts or in case a band has a routine, their regular event locations. This task also has some importance for the bands, as local newspaper with limited resources generally request the band to write short articles about themselves.

Task 3 had to be scaled back due to the current dataset limitations, as only Bavaria had enough bands to aggregate. Here, diversity is less important, but in contrast, the abstraction of large numbers of recent events moves into focus. The model should be encouraged to identify favorite event locations, music genres, or other patterns about a region.

## 1.5. Outline

This thesis is structured as follows: chapter 2 provides an overview of the relevant background information, including the history of data-to-text, evaluation metrics, and diversity in text generation. In chapter 5, details about the dataset, specific implementation details, and the evaluation pipelines are provided. Chapter 6 presents and chapter 7 discusses the results of the experiments and circles back to the research questions and the problem statement. Finally, chapter 8 summarizes the findings and provides an outlook on future work.

# 2. Fundamentals

This chapter lays out the foundations necessary to understand and navigate the rest of the thesis. It introduces the NLG research field and delves into the subcategory of data-to-text generation. Important steps in the evolution of the field are showcased, as some of these methods were later used in the thesis. Some of these methods will be explained with more detail, notably transformers - the leading advance behind most modern LLMs. Moreover, the concept of diversity is explained with its intricacies, challenges and its importance in the context of NLG. An introduction to evaluation metrics is also provided, as they are crucial to the comparisons in our research. Finally, the chapter concludes with a brief overview of the existing solutions and related work.

## 2.1. Data-to-Text over the Years

Most information in this section is drawn from [10] and [11].

**The Origins**
Ever since computers were invented, researchers have been trying to make them understand human language. The first ideas, before the so called AI winter, were to use a set of rules to rearrange sentences, but there was no understanding nor learning on the computer's part [12, 13]. Many advances in compute power, algorithms and statistics were necessary to move past manually crafted rules and into the realm of statistics-dominated approaches. A major advantage of these approaches is that they can leverage the large amounts of data emanating from the then nascent internet. These pioneering approaches started out with structured data, just as we will in this thesis.

While a consensus exists on the output format of NLG systems being text, the input format has been debated for a long time. As the systems' complexity increased, so did the input format vary. While early template- and rule-based models still relied on numerical data, flat semantic representations and other structured data, modern systems can now work with unstructured data, normal text, images or even videos. For our setting, the structured data obtained in the dataset lent itself well to experiment with multiple input formats. We used two main methods of formatting the input: flat semantic representations and natural language through prompts (details in section 4.5).

Another concept that emerged early on, was the requirement for tokens, a concept where words are broken down into smaller units that are usually combined. These units form the vocabulary of the models. Most models use a different format or size of tokens and vocabulary, so they require their own tokenizers. Since tokens are still in human-readable format, modern systems transform them into embeddings - a vectorized representation of the tokens. This allows semantic meanings and relationships between words to be encoded and input to the model. These methods already enable many tasks both for understanding and generating text, from ngram-based solutions to neural language models.

**The Renaissance**
However, there is still a missing component to understanding modern Language Models (LMs): "language is a sequence that unfolds in time" [10]. This implies that information from the past (or future) words is necessary to understand or predict the current word. Ngram-based methods for example, rely on a sliding context-window and cannot carry information for more than a few words. Recurrent neural networks (RNNs) were the first to solve this problem, by using a feedback loop in the hidden layer to carry information from previous words.

Following the advent of RNNs, Long Short-Term Memory (LSTM) networks, an advanced variant of RNNs, emerged as a significant improvement, especially in handling long-term dependencies in text. LSTMs were designed to remember information over extended sequences more effectively than traditional RNNs through a complex gating mechanism that allowed them to selectively remember and forget information. This made them well-suited for sequential data where context spread across a long sequence is crucial.

Despite their success and popularity over almost two decades, RNNs and, by extension, LSTMs had two significant drawbacks. First, they used a sequential architecture, significantly reducing the computational efficiency. As a consequence, training for long inputs was slow, limiting scalability. Second, long-term dependencies were still getting diluted over many steps, effectively limiting the amount of information that could be carried over long sequences.

**Modern Times** The required leap in performance came in 2017 with the introduction of the Transformer architecture by Vaswani et al. [14]. While the next subsection will explain transformers and their concepts in more detail, the revolutionizing concept was the (self-)attention mechanism, which could process the input sequences in parallel, solving the bottleneck of sequential architectures. It also permitted direct access to the context of any word in the sequence, allowing the model to learn long-term dependencies more effectively. This made transformers incredibly scalable, and since then, virtually all state-of-the-art LMs have switched to transformers.

In the specific context of data-to-text, most systems of the early 2000s included multiple stages such as content determination, text structuring, sentence aggregation, lexicalization and a realization stage [15, 16]. Although the methods and stages evolved over time, the general idea of planning-then-generating text remained until recently when data-driven end-to-end approaches blurred and combined the stages [17]. The plan-then-generate approach nevertheless remains popular in data-to-text because it is easily interpretable and controllable compared to end-to-end approaches [18]. On the other hand, the sequence-to-sequence (seq2seq) architecture has the advantage of removing intermediate steps, which add complexity and increase the potential for errors.

### 2.1.1. Transformers

As this thesis used exclusively transformer-based models for text generation, it is important to understand how they work and why they have become so popular. They were initially introduced in “Attention is all you need” [14] and have since become the de facto standard for LMs. Unlike previous methods, each token obtained a positional encoding that was combined with its embedding, enabling parallelization while retaining the sequence information. This gain in efficiency enabled the next step, where the self-attention mechanism computed the relevance of all tokens around it. This approach allowed transformers to capture complex dependencies and nuances in language data even over long distances.

Through these innovations, transformers could be trained on huge datasets sourced from the internet using self-supervised learning. With some adaptations, this led to the creation of foundational models or pretrained LMs containing a broad statistical language proficiency. Those models are meant to be fine-tuned with smaller, task-related datasets on specific use cases.

The main components of the traditional transformer are the encoder and decoder. Generally, the encoder stage captures context and condenses meaning, while the decoder’s role is to process the input from the encoder as well as his own output to sequentially generate the next item in the sequence.
Three types of transformers can be outlined:

**Auto-Encoding Transformer:** This category only uses the encoder component of the transformer. It is used to understand the input with common tasks being classification or named entity recognition. The most popular model-family in this category is BERT [19].

**Auto-Regressive Transformer:** This category only uses the decoder component as its aim is not understanding the input but generating the next item in the sequence. Here the most popular model-family is GPT [1].

**Sequence-to-Sequence Transformer:** Also called encoder-decoder transformer, this category uses both the encoder and decoder components. It is used for tasks that require understanding the input and generating a new sequence, such as translation or summarization. Here the most popular model-families are BART [20] and T5 [21].

While transformers can theoretically consider all elements in a sequence without a context window limitation, in practice, there is a constraint on the maximum sequence length they can process. This limitation is due to memory constraints, as the self-attention mechanism requires computational resources that grow quadratically with the length of the input sequence. Therefore, strategies like chunking the text are used for very long sequences. Current state-of-the-art API-based models such as GPT-4 can push these limits to 32K tokens, but it is important to keep these limitations in mind for our third task, which will require a long input sequence.

### 2.1.2. Pre-trained Language Models

In this thesis five of such models are used for the text generation tasks. In this subsection, we will briefly introduce them and their main characteristics.

#### T5 - Finetuned

T5 [21] is a LLM that was released in 2019 by Google. T5 stands for "**T**ext-**T**o-**T**ext **T**ransfer **T**ransformer" and uses the sequence-to-sequence transformer architecture. As its name implies, both its input and output are in text format, with the goal of using the same pretrained weights and loss function for a variety of tasks under the assumption that learning on one task is transferable to another similar task. It uses the encoder-decoder architecture as first described by [14]. It was trained on the C4 dataset, also released in the same paper. It contains around 750GB of text references from the internet on a variety of tasks, among them language translation, summarization, question answering and inference. The task is usually specified in a prefix at the beginning of the input sequence. With a maximum size of 11B parameters, it is by far the smallest model used in this thesis. It was used in the initial stages of research to establish the feasibility of the project and a baseline. Due to performance constraints at the time, we fine-tuned their smallest model (T5-small, 60M parameters) on our task and dataset, using our own prefix "generate" and following the tutorial from [22]. Additionally, the output size was limited to around 200 tokens.

#### Flan-T5 - Finetuned

Flan-T5 [23] is the evolution of T5. Its largest variant uses the same 11B parameters but was fine-tuned on an additional 1.8K tasks, significantly improving zero-shot, few-shot and CoT abilities. By switching to a better hardware in Google colab [24], we were able to choose a larger model than with T5. Since Flan-T5 is already a significant improvement over T5 and model size significantly impacts the output performance, this is a huge step up from the first attempts. We used Flan-T5-Base (250M parameters) and fine-tuned it on our task and dataset, using the same procedure as with T5. We switched to other models after the initial stages of the thesis, so the zero-shot performance was not evaluated.

#### GPT-3.5-turbo-1106

GPT-3.5 [1] is our first closed-source, API-based model from OpenAI. It is a variant of the model that powers ChatGPT [6] and is among the latest models of the GPT family. GPT is a text-to-text auto-regressive transformer, meaning it only uses the decoder component of the transformer, and both the in- and output must be in text form. It has around 175 billion parameters, although this has not been officially confirmed. Its training dataset is also unpublished. Due to its size, it needs to be trained and used on multiple GPUs, thus requiring the API for general availability. Over the course of the thesis, the formats of the API, the available models and the prices changed multiple times. This is the latest GPT-3.5 model we used; all results generated by a GPT-3.5 model are from this model. While it is possible to fine-tune this model, our zero-shot and few-shot results were already satisfactory for our analysis.

#### GPT-4-1106-preview

This is OpenAI's newest model [25], with significant improvements over GPT-3.5. The parameter size is rumored to be around a trillion, once again without confirmation. Compared to GPT-3 models it is slower and pricier, but it is also more powerful. OpenAI claims improvements in instruction following, reliability, creativity and more. While both this model and GPT-3.5 are optimized for chat/conversation purposes, GPT-4 is able to handle multimodal input, such as images.

#### LLama2-13b-chat

Since we wanted to compare the performance of closed-source models with at least another "open innovation" model, we chose LLama2-13b-chat [26] as a contender. This is one of the newest available models, as its paper was published after the official start of this thesis. It is free to use for research and commercial use, although not strictly open-source. The foundation models range from 7B to 70B parameters and after experimenting with the 7B models, we chose the 13B model for our experiments as it seemed a lot more reliable. The models were pretrained on publicly available online sources and then refined using Reinforcement Learning with Human Feedback (RLHF). Special attention was given to the safety of the output. In common benchmarks, it is rated at a similar level as GPT3.5, and using GPT-4 as a judge, it was consistently rated as more helpful than ChatGPT.

## 2.2. Evaluation of Text Generation

Evaluating generative models and their texts is a difficult and multi-faceted task. In this section, we will introduce the challenges we faced and the choices we made. We will also introduce the metrics we used and why we chose them. Specific implementations are described in section 5.2.

There are many aspects to consider when evaluating generated texts depending on the task at hand or the desired outcome. Accordingly, many methods exist to address these aspects, but each has its drawbacks and limitations. The most comprehensive method is still a manual **human evaluation**, and almost all automated metrics use a human correlation index for validation and comparison. Typically, human evaluations are run as a survey, where human evaluators are asked to rate generated texts according to specific instructions [27]. However, this method is often critiqued for being expensive, time-consuming, and biased. Survey methods are also hard to reproduce [28] and lack standardization [29, 30], leading to unreliable metrics. Nevertheless, they are the most flexible when it comes to evaluating all the required aspects of NLG tasks.

By trading some fidelity for less effort and time, automated evaluation metrics are able to process high volumes of samples [31]. According to Sai et al. [32], we can distinguish between context-free and context-dependent metrics and, in these categories, between trained and untrained methods. Context-free metrics do not require any additional information about the task or the generated text, while context-dependent metrics do and are usually not transferable to other tasks. Most importantly, virtually all metrics described in this survey require a reference text to compare generated samples to. This is a major drawback for cases where datasets are not curated, as is the case for our dataset.

In recent years, many attempts at changing the evaluation metrics have been made [33] with prominent figures in the field calling for replacements of outdated methods like **BLEU**, **METEOR** or **ROUGE** [34, 32] which are still widely used because of their simplicity and despite their poor human correlation. The proposed replacements leverage transformers and even LLMs [35, 36, 37] to achieve scores with better human correlation and sometimes even better consistency than manual evaluations. However, these methods are still in their infancy and need to be scientifically validated. Prof. Ehud Reiter, also mentions that "we need to use LLMs which are fixed and do not change (many commercial LLMs such as GPT4 are constantly being updated, which improves performance but makes replicability hard)" [34].

One of these new methods leveraging LLMs is **G-Eval** [38], as this framework is easily adapted to all required evaluation aspects (subsection 2.2.1) and outperforms many other modern metrics such as **BERTScore** [35], **BARTScore** [39] and **UniEval** [40] at least on the SummEval benchmark [41]. It is also one of the few metrics that can be used without a reference text, which is a major advantage for our dataset. G-Eval uses CoT techniques and formulates the evaluation as a form-filling problem, expecting a score. The metric relies entirely on GPT-4's ability to understand context and language and relies on carefully crafted prompts. The authors of G-Eval nevertheless warn about potential biases of the model, as the source of the text (human or different models) can influence the score. It should also be noted, that using an LLM for evaluation purposes is both slower and more expensive than local metrics, although it is still cheaper than human evaluation.

### 2.2.1. Defining Quality

Having chosen G-eval as an adequate automated metric, we need to define the quality of a generated band description entails. The aim of a marketing text is to capture the audience's interest and attention in the hopes of incentivizing a desired behavior. Therefore, the *engagingness* [42] of the text is a crucial aspect of its quality. The text should also be informative, accurate and concise. We thus chose common evaluation aspects defined in the literature. For one, *informativeness* [40] is a key aspect of text quality. Additionally to being interesting, the text should be easy to read and follow, and be free of grammatical and spelling errors. These points can be aggregated in a metric called *fluency* [41]. Finally, the text should sound natural and human-like. We thus define *naturalness* [40] as a key aspect of text quality.

### 2.2.2. Diversity in Text Generation

On top of quality metrics, our evaluation requires a measure of diversity. The key difference to the quality evaluation is that we now operate at an inter-sample level, meaning that multiple samples are compared to each other. While G-Eval can compare texts to each other, much more cost-effective methods exist for this use-case. We need to keep in mind, that diversity can be separated into form diversity and content diversity [43]. Form diversity is when the text itself is different, without necessarily changing the meaning. Content diversity is when the meaning of the text is different. The latter is much harder to measure, as it requires a semantic understanding of the text, and even human evaluation is sometimes ambiguous for that task. As our band descriptions will be generated from similar data and have the same purpose, the content diversity should be low. We thus focus on form diversity and argue that a metric picking up on the semantic meaning would negatively impact the metric's performance, as different band names would suffice to influence a score based on content diversity.

Despite the drawbacks of such basic methods, we chose the **Jaccard-similarity** coefficient, an n-gram-based method for comparing two texts:

$$Jaccard(U, V) = \frac{|U \cap V|}{|U \cup V|}$$

The Jaccard coefficient measures similarity between sample sets, here n-grams. It is calculated by dividing the size of the intersection of the two sets by the size of the union of the two sets. The coefficient is symmetrical, meaning the order of the texts doesn't matter, it is quick to compute and the result is a score between 0 and 1, with lower values signifying more diverse texts. By obtaining the pairwise Jaccard-similarity of all generated samples in a set, we can get an average similarity score. As long as the sets contain the same sampled bands, the scores can be compared to each other. This method is not perfect, as it does not take into account the length of the texts, but it is a good approximation and combined with a manual step detailed in subsection 5.2.3.

# 3. Related Work

This chapter will give an overview of previous work in the various fields our research touches upon. The primary domain of our research is definitely Data-to-Text, which is actively being researched and developped. This thesis dives especially into the domain of generational diversity, where many different approaches are explored in literature. As our application is in the marketing domain, we will briefly look at research in the field of advertising. Finally we will look at the evaluation of generated text, which is a challenging task in itself.

**Data-to-Text** This research field and NLG at large have seen a lot of evolution in the last few years, especially with the advent of transformer-based solutions. Over the years, many different approaches have been proposed, from (Neural) templates [44] over pipeline approaches to end-to-end solutions. A recurring problem is the controllability of the models, which leads to most papers proposing a multi-step approach. To test their systems, most papers rely on the WebNLG dataset [45] that uses a linearized knowledge graph in the form of triplets as input.

One family of approaches relies on building a content plan, which is then used to generate the text. They can be separated into micro-planning architectures [17] to plan a single sentence and into macro-planning [18, 46, 47] that usually applies the plan to paragraphs or whole documents. Puduppully et al. [46] used a sequential approach, generating a new content plan for each paragraph of sports summaries. Each plan considers the previous plans to ensure coherence and reduce repetitions. These methods usually require training or few-shot approaches for the content planners, so Kasner et al. [48] proposed a zero-shot approach by using multiple pre-trained models in four steps. First the input triplet are formed into facts that are then ordered using a variation of BART-base. RoBERTa is then applied to decide which facts can be aggregated with the last step using BART-base again to aggregate and compress the paragraphs.

The newest approaches tend to use end-to-end solutions such as GPT [49] or T5 either in zero-shot mode [50] or in combination with additional steps such as disambiguation [51], reasoning [52] or CoT prompting [53, 54] or even tree of thought reasoning [55] when the tasks become more complicated.

**Diversity of Generation** Although these methods do try to achieve diverse texts, their research focus is concentrated on the quality of the text and controllability [56] of the models. In literature less papers focus on achieving diversity of generation. Nevertheless, some approaches try to tackle this problem by adding relevant words that a model is then supposed to include [57]. Using transformers, methods are modifying the prompts [58], the encoder [59] and the decoder [60]. Alternatively they use a more manual approach like human-in-the-loop [8]. For the specific case of question generation, Cho et al. [61] achieved diverse questions by using multiple selectors that apply a mask on the input data, such that the model can focus on different parts of the input. However these methods are used to generate multiple outputs from a single input or push a model to create new expressions. Indeed no dataset nor method seems to focus on sequential generation, with the goal of reducing previously used expressions and formulations which is partially the aim of our research.

**NLG in Marketing** In the domain of marketing, AI in general is already being used, especially in cases such as Search Engine Optimization (SEO) and at all stages of advertising campaigns. In the context of product descriptions, tools like Alibaba's Luban [62] use AI to generate graphical banners, and a common practice for large international commercial platforms is to use automatic translations in descriptions. Since our research will generate descriptions for users of a platform, the closest topic in marketing is generating descriptions for products in e-commerce [63, 64]. The advertisement sector of NLG (AdNLG) seems to have received less scholarly attention than other domain-specific areas of NLG [65]. Nevertheless, this survey defines three categories of programmatic online advertising methods in NLG: template-based, extractive and abstractive. The extractive methods aim to select the most relevant information from data, but they are generally reserved for title and slogan-generation. The abstractive methods are more flexible and greatly benefit from recent advances in LLMs. DeepGen [66] is an example of a system that uses abstractive methods to generate a precomputed database of small text snippets that are then stitched together at query time. While these methods share a lot of common ground with the common data-to-text task, they are focused on advertisement and thus not directly applicable to our research. Their evaluation criteria are however very relevant to us, as advertisers constantly battle for the attention of users and need to prevent *ad fatigue* [67].

**Evaluation of Generated Text** Evaluating generative tasks is an ongoing challenge and a topic of debate among researchers. Most papers rely on comparative metrics such as BLEU [68] that depend on references. These metrics are often critiqued for being antiquated and only focusing on surface-level features with n-grams [34, 43, 32, 33]. According to Gehrmann et al. [69] evaluation practices have problems at every level: datasets prevent measuring tail effects, cover only English, metrics only measure similarity to references and human evaluation has high variance and requires rigorous standard [70]. Nevertheless, there are many newer metrics, often utilizing neural methods with the goal of better capturing the core of the content. Methods like SummEval [41], USR [42], UniEval [40] or G-eval [38] promise to be more robust and move away from potentially problematic reference-based methods. These metrics need to be validated [30, 71] to prove their accuracy and usefulness, as most metrics based on pretrained models have biases [72].

Researchers agree that there is no perfect metric yet, that reference-free metrics are limited because of biases and missing context [30] and that a combination of metrics is often the best solution. To this end, certain aspects can be targeted depending on the use case. For example, in the case of AdNLG, advertising performance, diversity, faithfulness, fluency, and relevance [65] are usually considered. G-eval [38] mentions engagingness, naturalness and fluency as relevant metrics for NLG.

In the aspect of diversity evaluation, [43] distinguishes between form- and content-diversity. Form-diversity is the diversity of the surface-level features such as words, phrases and sentences, whereas content-diversity is the diversity of the core content. While n-gram based methods can measure the former, the latter is difficult to measure even with neural methods and even human evaluation can be ambiguous. Measuring content diversity requires a semantic understanding of the texts. Different approaches have been tried over the years [73, 74, 75] utilizing various embedding techniques.

Luckily for us, we are interested in the form-diversity, which can be measured with metrics such as Self-BLEU, Jaccard- or Cosine-similarity. Metrics are also starting to emerge to measure the diversity inside a dataset. The Vendi Score [76] comes to mind, which uses the Shannon entropy to obtain a score.

Going in the other direction, some attempts try to combine quality and diversity measurements into a single score [77].

# 4. Dataset

We obtained our dataset from our industry partner [7] by querying a snapshot of their database from the beginning of 2023. We used this source to train or query models and evaluate the results. Later on, we obtained a newer snapshot in October 2023, with updated data that was used for gigs played by the bands. The newer snapshot was not used for retraining, as many changes in between both datasets meant that the manual annotations mentioned in 4.1 would have had to be redone.

While the data is not public, we can describe the database structure and the data type we used. Our source is a relational database containing personal and public information about individual musicians, grouped in formations, event organizers that will interact and hire the formations, and information about the events themselves. We wanted to use as much relevant data as possible without risking revealing personal information. Therefore, we designed two custom SQL queries, the result formats of which are described below. The data was exported as a csv file and imported into a local database for further processing.

**Formations:** With the intention of generating descriptions for marketing purposes, we formed a dataset containing the following information about each formation:

- *Formation name*
- *Formation type:* Formations can be categorized as duet, band or dj.
- *Formation description:* Existing description of the formation.
- *Formation members:* When listed, the members of a formation.
- *Homebase:* The city where the formation is based.
- *Radius:* The distance a band is willing to travel to play a gig.
- *Genres:* The music genres the formation plays. These are tags from a predefined list.
- *Event types:* The types of events the formation plays at. These are tags from a predefined list.

**Gigs:** Another query was used as active formations have multiple gigs. Despite some events containing price information, we opted not to use them. The following information was used:

- *Formation name*
- *Event start:* Date and time.
- *Event end:* Date and time.
- *Event type:* The type of event(wedding, etc.).
- *Event genre:* The requested music type for an event.
- *Public event name:* The name of the event as defined by the organizer.
- *Address:* Only the city and region are used.

- *Event description:* Sometimes contains additional information or requests by the organizers.

The formation dataset is the main source for our research. It was used to generate the descriptions, evaluate the results, and run the diversity-improving experiments. The second dataset was only used in later tasks to add more information. A third query containing individual band members and their instruments was also created but not used in the end because the relations between the members and the formations were only partially available and by reducing the data size, we simplified the evaluation task.

## 4.1. Data Preparation

The dataset was not immediately ready to be used for training or evaluation. Roughly half the formations didn't have any description; some more were just test data, and almost all were in German. Band descriptions being in German is problematic for some models as they often struggle with non-English text.

Preprocessing steps were therefore required to make the dataset usable. First, we removed all the formations without description. This still leaves non-usable descriptions, which we treated after all of the preprocessing steps, as detailed in 4.1.

For the sake of comparability, we decided to use exclusively English for all our experiments. The descriptions were translated into English using the Deepl API [78]. Deepl was chosen for its reliability and overall quality of translation when compared with other services.

The last automated preparation step was to add regional information on the bands homebase. This allows us to group and filter the dataset by region, as musical genres, bandstyles and event types are often region specific. Furthermore, regional information could be used in the future to add dialects to the generated text. Promising results were obtained during prompt engineering in ChatGPT but not used for our research as this requires the generation to be in German.

Unfortunately, it was quickly realized that these processing steps were not enough for evaluation, fine-tuning or comparative studies. We therefore decided to manually evaluate the dataset and filter out unusable descriptions and finally creating sub-datasets in 4.3 and 4.4.

#### Manual Annotation

The manual evaluation was done by the author of this thesis. The goal was to filter out unusable descriptions, rate the quality of the remaining ones and select good texts to use as references or templates when generating new descriptions. The results were used to select sub-datasets, training new models and also for few-shot generation. They were also crucial for the evaluation steps (5.2), especially the manual ones.

During the preparation of the annotation process it was realized that we would need metrics to not only assess the quality of descriptions but also their usability and possibly the lexical diversity. For that we needed to define what we meant by these terms. The descriptions also varied wildly in length, with the shortest ones being only a few words long and the longest one at just under 500 words.

For our specific case, the quality of a description is be defined by the amount of information about the band contained within, it's lexical quality, how interesting and engaging it is. Overall a high quality description should give it's reader a good idea about the fit of the band for their purposes and from the band's point of view it should nudge an event organizer to contact and recruit them.

The usability should describe how well the description fits the purposes described above. For example, some supposed bands wrote in the description that they were a lighting and audio engineering company. While the description was good quality-wise, it wouldn't fit our task of writing a musicians marketing text. Others were written very personally, with the band describing their history and anecdotes. Unfortunately, this data is not contained in our database and will not be available for a model later. We therefore had to keep the available data in mind when grading the usability.

Lexical diversity in the context of the manual annotation was not a statistical or mathematical metric but rather a subjective measure of how much variety in the word choice or whether uncommon expressions, turn of phrases or even word plays were used. This aimed to push models to be more creative and less repetitive by teaching fine-tuned models some new possibilities. It is important to note, that we focused on form diversity and not content diversity, as the content of the descriptions is very similar despite each band having different underlying data. We will talk more about this in 5.2.3.

From these three goals, we finally settled on two metrics for the manual evaluation: the first is called quality, and the second group's usability and linguistic diversity and was called uniqueness, for lack of a better word. Both metrics are scored between 0 and 10, with 0 being the worst and 10 the best. The scores are subjective and might be seen as arbitrary, but follow the guidelines described above. Text length, linguistic quality, spacing, readability and tabular data being used (or not) were factored in for the quality score. In contrast, unique formulations, dialects, and anecdotes were used to judge the uniqueness score.

We called anecdotes any part of text that didn't rely on the given tabular data. Two types were distinguished: personal anecdotes that require knowledge and data-agnostic anecdotes. Personal anecdotes should be avoided when possible as models wouldn't be able to accurately guess those with hallucinations. We decided that the presence of personal anecdotes would negatively impact the uniqueness score despite technically increasing the uniqueness of the text. On the other hand, data-agnostic and diversity-promoting anecdotes should have a positive impact. Links also negatively impacted the uniqueness score as the tabular data did not provide replacements. We hoped for primarily data-agnostic formulations that gave some form of context on the event, band, or DJ, which might generalize well or push the model to add sentences that do not rely on the tabular input data.

In later parts of the research, we noticed some bands would add multiple event types to their profile, but their descriptions would focus purely on weddings. In hindsight, this makes sense for search engine optimization, but it hinders our ability to generate descriptions that the band might actually use.

These two metrics helped filter inadequate descriptions and build diversity-increasing datasets that would remain factual. Restricting ourselves to them also proved time-efficient, as only the descriptions needed to be read without checking the tabular data. The annotation process still took a long time, especially as the file got corrupted once, just after finishing the first time.

While we did not know it when we manually annotated our dataset, [43] showed that humans are biased by quality and they attempted to reduce this bias by first asking humans to rate the quality of a sample then explicitly asked to ignore that in diversity questions. Luckily we noticed this effect ourselves and tried to evaluate the diversity while ignoring the quality of the text. In the next section, we will nevertheless try to verify the correlation between quality and diversity in our annotations.

## 4.2. Data Exploration

Now that we have a usable dataset, let's explore it to better understand the data we are working with.

The initial formation dataset has 880 samples/formations of which roughly 440 have no description at all. Some of the rows are also duplicates or development artifacts that are unusable for our research. We therefore decided to select, sort, and annotate the dataset, a process described in 4.1. After this process, we kept 359 samples. Not all of them are of high quality, but they are at least not test data or duplicates and stem from real musicians.

Inspecting their length, the shortest text is only 214 characters long, or 28 words and the longest is over 3000 characters long. The average length is about 863 characters with a standard variation of 512 characters. Figure 4.1 shows the distribution of the description lengths. Due to some models' limited attention windows and for comparability, we will try to choose texts of similar lengths in the sub-datasets.

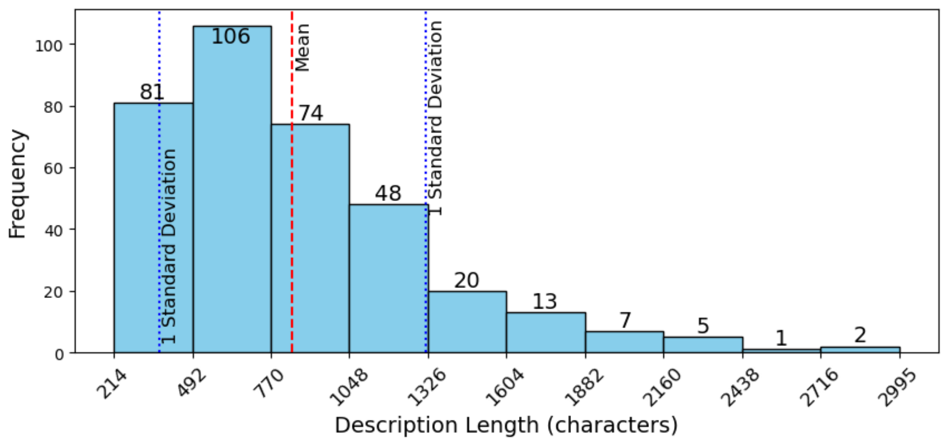
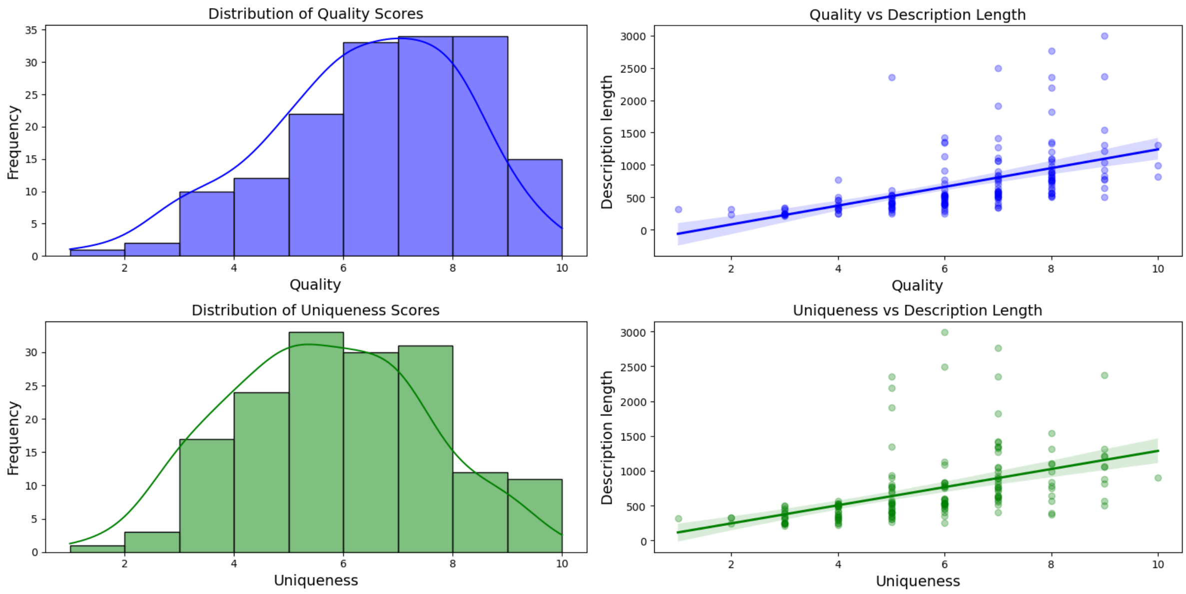



Figure 4.1.: Distribution of description lengths.

Next, we looked at the regional distribution of the bands, as this might indicate some




(a) Distribution of scores as histogram and as kernel density estimation.

(b) Regression analysis of scores.

Figure 4.2.: Exploring the manually annotated scores reveals that the quality scores are skewed towards higher values while the uniqueness score is more centered. The linear regression analysis shows a medium positive correlation between the description length and both scores, whereby texts around 1000 characters were scored the highest on average.

regional variances. Figure 4.3 shows the distribution of the bands in the dataset. The bands are mostly from Germany, with a few from Austria and Switzerland. In fact, over half come from Bavaria. This is unsurprising, as the company is based in Munich, and the dataset is a snapshot from early 2023. Since the data is heavily skewed towards Bavaria, we must consider this for the sub-dataset creation, as some event types and music categories might be overrepresented.

Lastly, we inspected the manual scores for biases. While a human can reliably score the quality of a sample, Tevet et al. [43] showed in a pilot experiment with a group of NLP graduate students, that "humans are biased by quality: if a generated set has high diversity but low quality, humans will rate diversity low". Despite knowing of this human bias during the annotation process, this also seems to be at least partially the case with our annotations, as we obtained a Pearson correlation score of 0.73 between quality and uniqueness. Therefore, one score may have influenced the other as unique formulations likely increased the quality score. The manual scores should therefore be examined critically and used as indicators rather than absolute values.

As a similar effect could have occurred dependent on text length, Figure 4.2 shows the distribution of scores as well as a regression analysis of the scores. The regression analysis shows a medium positive correlation between the description length and both scores. With

Pearson correlations of 0.51 and 0.46 respectively, the quality and uniqueness scores are both moderately correlated with the description length.

A moderate correlation is not surprising, as longer texts have more space to convey information and are therefore more likely to be of higher quality. Bands writing uninteresting descriptions would also be more likely to write shorter texts.

## 4.3. Parameter Dataset

During the initial trials, we quickly realized that some experiments would require smaller test datasets. Therefore, we created the parameter dataset, called `task1_para` in the code. From the high-quality descriptions, we selected 10 formations split equally between bands and DJs, all with a high-quality score. We also verified that the tabular data was as distinct as possible for better coverage. Despite being a minimal dataset, the results should be larger as the plan was to run each formation through each model multiple times while varying parameters. This should allow us to check each parameter's impact on the results while keeping some statistical relevance.

We came to this solution after generating descriptions for a single band while varying the temperature parameter. We noticed that for high-temperature values, the output would become unreadable. By adding multiple formations of different types, we have a more reliable way of checking the temperature parameter and any other one going forward.

This dataset should never be used to check the diversity of models, because it will obviously have similar results as each formation is represented multiple times.

## 4.4. Diversity Dataset

To compare and ascertain the diversity between models and experiments, we selected a dataset of 50 bands. They will be used as a benchmark for different experiments and models in order to evaluate their quality and, most importantly, diversity. The dataset is called `task1_div` in the code. As we aimed to increase output diversity, we decided to reduce input diversity by selecting only bands from Bavaria and no DJs. The idea behind this approach is that a method able to provide great diversity from little input variety will be even more diverse once given more input variety. Obviously, this sub-dataset will not be used for finetuning or few-shot prompting.

Taking bands from similar locations has a secondary benefit as these bands probably have to compete for the same gigs anyway, and someone reading their description would, therefore come across multiple of them. It is precisely in this situation that we want to have highly diverse texts.

## 4.5. Data Format and Encoding

In the field of data-to-text there are many ways of formating the input for models. During the initial trials, we chose a triplet format inspired by WebNLG's [45] structure consisting of *(subject, predicate, object)*, which we used on T5 and FlanT5 experiments. This format allows us to encode knowledge graphs linearly from which a model can learn. It has the advantage of forming structured connections between data points and is therefore easier to learn from.

As we switched to more performant models, this rigid structure repeated much of the same information and limited additional instructions. We therefore switched to text-based templates and prompts for all subsequent LLMs. The prompts are a mix of natural language and placeholders for the tabular data. Depending on the experiment, variations of the prompts were used to achieve our goals. An example of a prompt is shown in section A.1.

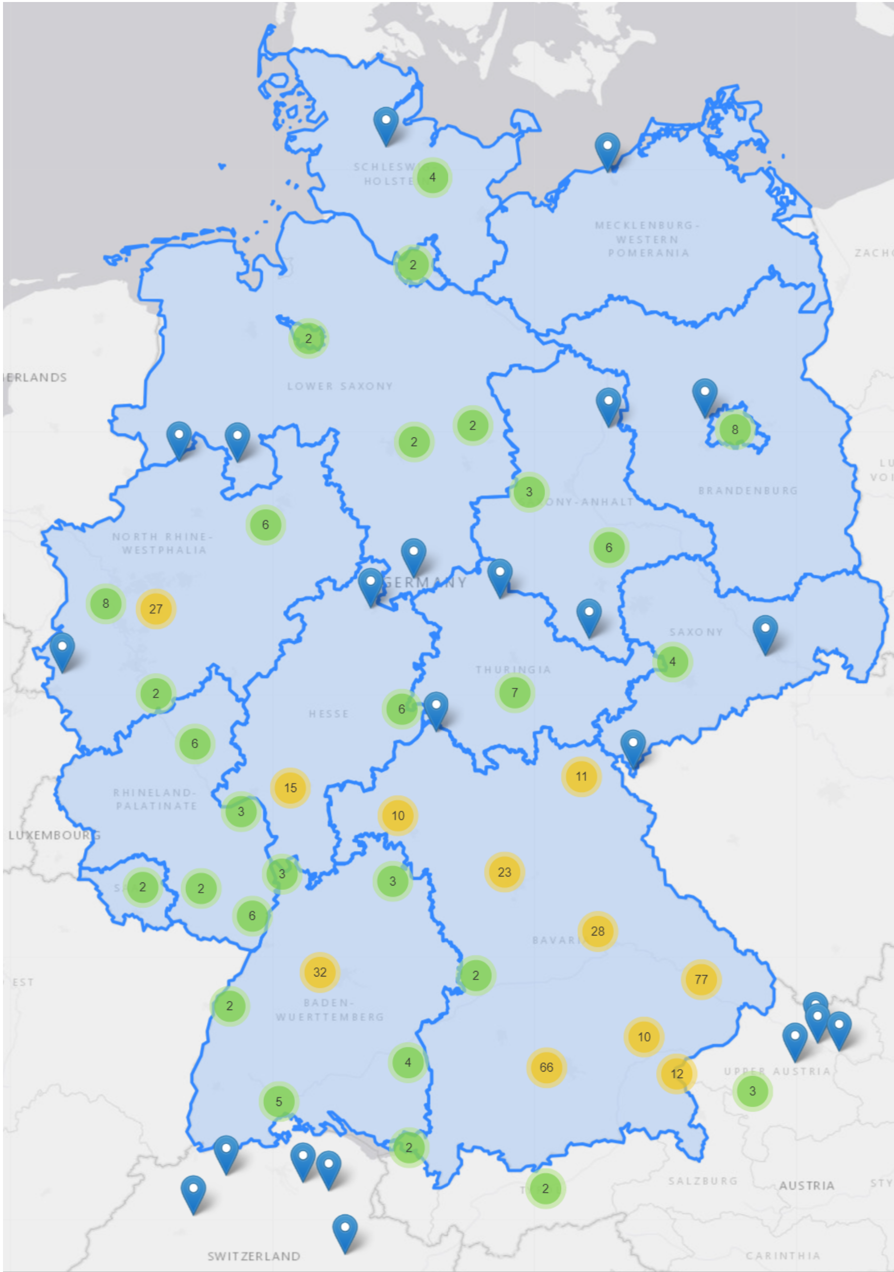



Figure 4.3.: Distribution of bands in the dataset. Colors represent cluster sizes. Most bands are located in Bavaria. Map from [79]

# 5. Implementation

## 5.1. Models

Introductions to the models can be found in 2.1.2. Here we briefly mention how we used each of them.

We mainly used five different foundation models from three families. During initial work we focused on T5-small then FlanT5-base models from Google and fine-tuned them to our task using the Adafactor optimizer [80]. Later on, we switched to GPT-3.5 and GPT-4 from OpenAI for their performance in zero-shot and few-shot generation. Due to their ease of use through the API, those were later used for our industry partner's website. Towards the end, we added LLama2-13b from Meta by running it through llama.cpp [81], a framework for running large models on CPUs efficiently. Here, we aimed to see whether open models could be used as well with zero-shot generation and how they would perform compared to commercial models.

## 5.2. Evaluation

### 5.2.1. Pipeline

After we generated our first texts, the most important part was to evaluate them. We built up a series of evaluation steps to give us different insights. Given our research questions and future experiments, we mainly wanted to assess their quality and their similarity, or rather diversity. This led to two types of evaluation levels: we defined the first one as the intra-sample level, looking at a single sample at a time, and the second one as the inter-sample level, where we compared samples to each other.

The intra-sample level is one of the standard ways of evaluating text and generation models. Many services both in- and outside of research domains can analyze, grade or improve single texts with more or less human correlation. The difficult part for us is that we work in a reference-free domain and the most accurate method is still human evaluation.

The inter-sample level is less common, but we believe it is crucial for our research. It allows us to compare the diversity achieved by different models and experiments. While it is quite common to compare multiple texts, it is rarely done to assess the generational diversity of models. We therefore had to adapt existing methods to our purposes and ended up with an n-gram based similarity matrix from which we judged the diversity of the results.

The evaluation pipeline is shown in Figure 5.1 and starts with automatic metrics based on G-eval [38] (details in subsection 5.2.2) and leads to manually evaluating the most and least

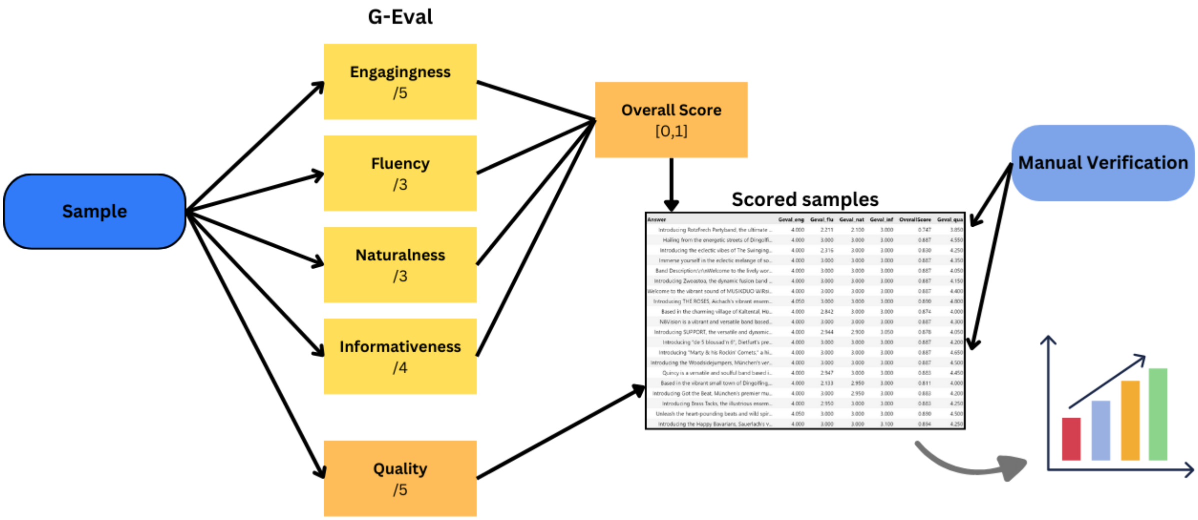



Figure 5.1.: Evaluation Pipeline for our Quality metrics

promising results. These quality checks help determine whether the results are usable and whether we can proceed with a similarity analysis.

Quality is a broad term we employ to summarize the metrics related to a single text's content. In fact, we used five different automated metrics to assess the engagingness, fluency, naturalness, informativeness and overall quality of generated content.

Before moving to said similarity analysis we performed experiment-specific checks, such as verifying the quality scores by temperature for the temperature experiments. These usually led to manually inspecting interesting or suspicious results.

For our diversity analysis, we used a similarity matrix based on n-grams (details in subsection 5.2.3). As the topic of every text was similar anyway, we did not pursue a more advanced semantic similarity analysis or embedding-based metric. The similarity matrix allowed us to calculate a diversity score that is even comparable between experiments and when using the `task1_div` dataset (section 4.4). Other datasets also benefitted from the matrix, as it enabled us to compare the most similar and different samples graphically by highlighting the ngrams found in the compared texts.

With automated and manual ways of assessing the quality and diversity of our results, we were now able to run all of our experiments and compare them to each other. The next two subsections will delve deeper into the metrics used for the quality and diversity analysis.

### 5.2.2. G-Eval

We chose G-eval as our main automatic evaluation metric because we required a reference-free method and this is the most modern and adaptable one available. It is also task agnostic and can be used for any text generation task. According to Liu et al., "LLM-based metrics generally outperform reference-based and reference-free baseline metrics in terms of correlation with human quality judgments, especially for open-ended and creative NLG tasks, such as dialogue response generation."[38].

G-eval uses an LLM, in our case, first GPT-3.5, then later on GPT-4, to generate a score based on an elaborate prompt. The prompts leverage chain-of-thought [53] with a task introduction and evaluation criteria to push the evaluation model to reason before returning a score. Liu et al. [38] realized that current LLMs have problems with decimal numbers and thus limit the score to integers on a defined scale. To eliminate equal scores, the temperature parameter of GPT-4 is set to its maximum value of 2.0 and the prompt is run 20 times. The final score is the average of the 20 runs.

The paper also proposes some metrics and illustrates how to use them. We decided to use their templates, as well as metrics defined by Fabbri et al. [41], Zhong et al. [40] and Mehri et al. [42]. We used the four metrics below for our evaluation and combined them into an overall score with normalization and weights. The Quality score was used to compare the step-wise evaluation to a direct method. Unfortunately, its results varied less than the overall score as it almost always gave high scores for descriptions generated by LLMs without explanations. The overall score was more explicit and understandable, making it the primary decision factor during the experiments.

- **Engagingness:** Engagingness is a metric that measures the ability of a text to engage a reader and judges how interesting a text might be from the way it is written. [42] [38]
- **Fluency:** Fluency is a metric that measures the grammatical correctness of a text. [41]
- **Naturalness:** Naturalness is a metric that is used to "determine whether the utterance could plausibly have been produced by a human." [40].
- **Informativeness:** Informativeness is a metric that is used to "determine whether the utterance contains all the information in the given content." [40]

- **Overall Score:** The overall score is a combination of the four metrics above. It is the average of the normalized scores and ranges from 0 to 1.
- **Quality:** Regroups the other four metrics as well as some other aspects into a single prompt. Unfortunately, it was not as varied as combining the separate scores and thus less useful. It ranges from 0 to 5.

The exact prompts for each metric can be found in the appendix section A.2

While G-eval globally performed well, it took many code iterations to get it to work properly. One of the problems was OpenAI's API, where gaining access to GPT-4 models only worked after spending credits and waiting for around a month. The authors mentioned that the human correlation index went up when using GPT-4 so it was paramount for us to use that model. Another, more persistent problem was that it was a slow and expensive process compared to other metrics that work locally. It also took a significant amount of time to get it to work properly, as with the high temperature setting, the API regularly returned non-integer responses, which required tweaking of prompts and additional logic in the code. While it is possible for the model to explain a single score and why it was given, the final results are not very transparent due to the 20 runs that get averaged. Finally, the scores might be biased as the same LLM was used to generate the sentences that would later be evaluated. The authors of G-eval caution against such a bias and we noticed this problem with the results of our fourth research question.

### 5.2.3. Diversity Analysis

The aim of our diversity analysis is essentially to quantify repetitions in a set of generated texts. In Tevet et al. [43] they distinguish between form and content diversity. Form diversity is the diversity of the sentence structure and words used, while content diversity is the diversity of the meaning. We focussed on form diversity due to the nature of our task, where the content was very similar in every sample, unlike with QA tasks with multiple possible outcomes. The major difference to their research is that they evaluated the evaluation metrics themselves, while we wished to use the metrics on our models. Additionally they created their sets from a single prompt, while we generated a diverse set stemming from different data and prompts. The pipeline we built for the diversity analysis is depicted in Figure 5.2.

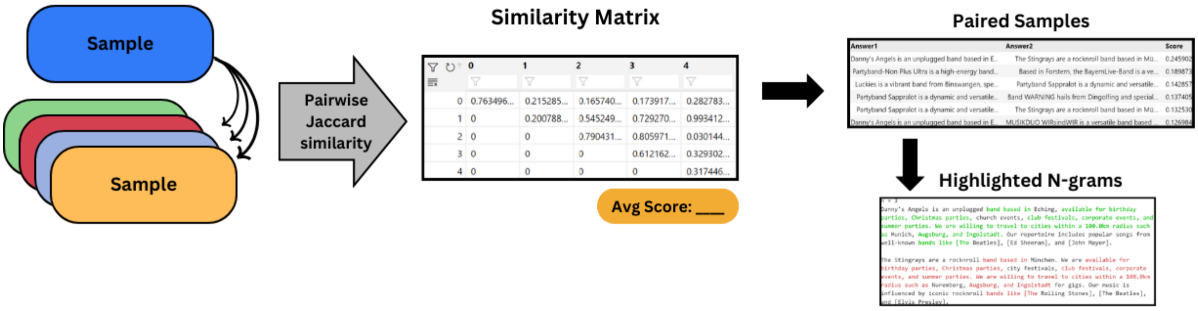



Figure 5.2.: Evaluation Pipeline for the diversity analysis

In their research, Tevet et al. [43] point out, that n-gram based methods work well for form diversity. Therefore, we built our own metric based on n-grams. Starting from a set of generated texts, usually created using the `task1_div` dataset (section 4.4), the algorithm first removes stopwords and punctuation before stemmatizing the words. We can then choose to generate bi-, tri- or 4-grams for each text from a parameter. The next step then compares all texts to each other by calculating the pairwise Jaccard similarity from the n-grams. The resulting values are stored in an upper diagonal matrix, as the similarity is symmetrical.

From there, we can average all the values to get an approximate diversity score. We note, that the Jaccard similarity metric does not properly account for text length, but during development, all texts inside an experiment were generally of similar length, such that the impact of text length on the score was negligible.

The averaged score could then be used to compare experiments and methods, but the similarity matrix had more importance to us, as from it we could build text pairs with their similarity scores. We extracted the most similar and most different text pairs for a manual evaluation. Through another function, we could highlight the n-grams in the compared texts, enabling an informed manual analysis of the results. While the manual review did not provide a metric, it was crucial for the analysis of the results as patterns and repetitions were easily visible in the highlighted n-grams. An example of this method is shown in Figure 6.2.

## 5.3. Experiments

Once the evaluation pipeline had been assembled and the datasets and the models were prepared, we started running experiments. The first step was to get a sense of how each model performed, which input format (section 4.5) was the most promising and how much additional training would be required. Both T5-small and FlanT5-base required fine-tuning and used the triplet-based input without additional instructions, while GPT-3.5, GPT-4 and LLaMa2 created decent results in zero-shot and few-shot mode, and improved with a text-based input format that added instructions. The smaller models (T5 and FlanT5) were abandoned after baseline experiments, as both their input and their output were rigid and inferior to the larger models. As LLaMa2 ran slowly, it was only used for some of the experiments with the goal of comparing it to commercial models.

The next step for the chat-based LLMs was to improve the prompt through prompt engineering [82] in OpenAI's playground. After many iterations, we ended up with the prompt shown in section A.1 as well as variations. Note that information of the band members and their instruments was not included to keep the texts shorter and because many bands didn't fill out this information in our dataset. During this stage, it was found that the models were generally able to correctly identify and use large cities close to the band's location. This is a typical example of how the instructions were modified to improve the results.

Once a satisfying prompt had been engineered, baseline tests were run using the `task1_div` dataset (section 4.4). This dataset contains 50 similar bands from Bavaria with the goal of creating a repeatable and comparable set of results. Quality and diversity were evaluated using the evaluation pipeline (subsection 5.2.1).

After the baseline tests, we started running experiments on the parameters of the models. To this end, we used the `task1_para` dataset (section 4.3), which is small with only ten entries. As each entry was run multiple times with different values for the tested parameter, the output quickly ballooned in size. The reason why we used an entire dataset for these experiments and not just a single entry, was to stay statistically relevant. The results of these experiments were not compared to the baseline results but were used to find the best parameters for the subsequent experiments. We ran these experiments for most of our attempts to improve diversity, as it enabled us to compare the results of different values on the same input.

To then compare the techniques to each other, we resorted to the `task1_div` dataset (section 4.4). We ran four experiments specifically to measure and rate diversity improvements, namely `Data-ordering`, `Alternate-instructions`, `Fewshot` and `Logit Bias`. They were run in GPT-3.5, GPT4. LLaMa2 was run with two of these experiments. The results were evaluated using the evaluation pipeline (subsection 5.2.1) and then the scores of each run were compared to each other.

### 5.3.1. Parameter Experiments

#### Temperature

The temperature experiments were run on ten values between 0.0 and 2.0 for GPT3.5 and GPT4. The experiment was not run for LLaMa2 because of resource limitations. As the temperature parameter adds some randomness to the generation process, we expect it to significantly impact diversity. Unfortunately, high temperature values can lead to nonsensical results, so we expect the quality to decrease with higher temperatures and a balance needs to be struck.

#### Top-p

We ran the same setup as with the temperature experiments, but using different values for the nucleus sampling parameters. This parameter focuses the next prediction on the smallest subset of the vocabulary, where the cumulative probability is above 'p'. The advantage of Top-p is that it strikes a balance between randomness and coherence. As the default value is already 0.9 on a scale from 0 to 1, we hoped to reduce this parameter in conjunction with a high temperature to increase diversity without losing quality. As the temperature parameter didn't affect the generation negatively, this parameter was not changed for the diversity experiments.

#### Logit Bias

The logit bias is not a single parameter but rather a list of tokens (not words!) for which the output probabilities can be positively or negatively influenced. The logit bias is a powerful tool to influence the output of an LLM, as it can be used to force the model to use or stop using certain words or phrases. This is the closest we can get to directly influencing the use of words with API-based systems. The bias has a range of -100 to 100. After each generation, the occurrences of each token are counted, and we select the top 100 most used tokens. Additional refinement steps are performed to select relevant tokens and not stopwords or punctuation. A negative bias is applied to each of them based on how often they have appeared in the top 100 list. The new biases are then applied to the next generation task. Initially, the bias values were fixed to -50, but we later experimented with an adaptive method that assigned a bias according to the following formula: $(\#in\_top\_100 - \#not\_in\_top\_100) * 10$ limited to a range of -100 to 0. By plotting some of the token's bias evolution over the generations, we were able to determine their influence on the models.

#### Data Ordering

During the initial experiments, it was noted that the order of the data in the input had a significant impact on the text generation. As an example, if the music genres were specified as *Pop, Jazz, Lounge* the models would by default use the same order when describing each genre. This led to repetitive texts especially when the generated text was short. To counteract this, we shuffled the data but kept it's structure before feeding it to the model. The order inside each prompt was also shuffled to increase the effect. By rerunning the same experiment with different shuffles, we hoped to simulate the same effect as using different content plans from pipeline approaches and to verify how the order in the output was influenced by its input.

#### Alternative Instructions

Obviously changing the instructions should have an impact on the output. We therefore created a set of alternate instructions with the intention of replacing both the beginning and the end of our prompt (section A.1). To inspect the effect of modifying the instructions, we first ran a small experiment with the parameter dataset. For each band, three runs were made using different instructions each time. The intention was to observe different formulations in the output based on subtle differences in the input prompt, mainly in the form and objectives of additional sentences.

#### Fewshot

Lastly we experimented with few-shot implementations, were we used other existing descriptions from the curated dataset as additional input. The goal was to see whether the largely free LLMs would incorporate existing expressions into their output. Unfortunately, this method relied on references which were in German or had been translated to English through Deepl, which would loose some of the language specific intricacies and expressions.

### 5.3.2. Diversity Experiments

To rate, compare and rank the influence of these approaches on quality and diversity on a dataset-level, we reran some of the experiments with the `task1_div` dataset (section 4.4). Combined with the base case mentioned earlier, we compared GPT3.5, GPT4 and LLaMa2 outputs from four diversity-enhancing techniques. `Data order`, `Alternative instructions`, `Fewshot` and `Logit Bias` were evaluated using the same evaluation pipeline and rated both automatically and manually and their effect on diversity related to each other as our implementation is dataset specific and could not be compared to existing metrics.

### 5.3.3. Task2: Adding Gigs

In a step to expand band descriptions towards automated news generation, we modified the input prompts to include recent gigs for each band. The aim of this is quite different to the previous task, as this requires the user to have had events planned and accepted through the platform, whereas previously, we mainly used data added during the sign-up process. The goal was to see whether the models would be able to abstract and relate additional information, similar to how sport summaries need to extract the most relevant information from games. Unfortunately, the gigs data didn't include information about event size or budget, which would have been informative to rank the importance of each event. On the other hand, not having this information meant, that the models could not leak it during generation, a common problem for black-box systems.

### 5.3.4. Task3: Regional Summaries

A pilot experiment was run for the region of Bavaria as most other regions didn't have enough relevant data for the bands we were working on. In this experiment, all events in the region in a 12 months window were fed to the GPT-4 model. The goal was to see whether the model would be able to abstract the data, assemble similar events, and music genres, and talk about upcoming events and most active bands. We used the following prompt in combination with a linearized version of the events data:

```
Prioritize recent events and abstract from the data to create a summary of the
    ↪ region's music scene.
Mention planned events or popular locations and bands.
The summary should be under 400 tokens.
```

The limitations here are the context window of the model and how well it can abstract from the data. In a way, it will test if GPT-4 can be overwhelmed with data and requires additional steps to extract the most relevant information.

# 6. Results

## 6.1. RQ1: Choice of Technologies

In this section we will briefly discuss the methods we have used or encountered to generate text and then present the results of our experiments. Our focus was set on transformer-based models, as they are the most advanced method that have replaced many of the older approaches but their lack of controllability is still a challenge for most non-creative tasks. A few text samples are provided in the appendix A.3.

The first model we experimented on was T5-small. Through finetuning on the dataset, we were able to achieve some basic text, although when we retrained the model later on with the now cleaned and annotated data, the results deteriorated, such that the model became unusable. After a sentence or two the model would start repeating itself and become incoherent. We suspect that multiple factors have come together to overwhelm the finetuning: the model was small, the dataset was small and the dataset cleaning process removed mainly short descriptions, such that the model had to, and failed at learning from longer descriptions.

The next model we attempted to use was FlanT5, an updated and improved version of T5. Here we were able to choose a slightly larger model and the results were much better, although still not perfect as can be seen by the G-eval score in Figure 6.1. About half the samples still suffered from word looping, where the model repeats the same few words or concepts over and over, but it was a big step up from T5-small, where virtually every sample suffered from this problem.

It should be noted that FlanT5 was able to generate short band descriptions without finetuning. Due to ambiguous naming in the input triplets and no context to rely on they were unfortunately not usable. In most cases, the model would describe the band as being music genres/events types/etc., instead of using the proper verbs.

It was time for another step up, so we switched to the GPT family of models which is an order of magnitude larger than T5. We started with GPT-3.5 and later got access to the preview version of GPT-4. Both models were able to generate text of good quality without finetuning, such that we could modify our input from triplets to a proper prompt to be more flexible and add instructions to improve the generation. The results were very promising, as can be seen in 6.1, where the scores are much higher than for the T5 models. Both models used paragraph for added readability and were globally coherent. The major difference between GPT-3.5 and GPT-4 was that GPT-4 was able to generate longer texts, talking about each music genre, event type etc. in detail, while GPT-3.5 usually wrote shorter descriptions with less details and squarely repeating the input data. The shorter text had the advantage of being faster to read, whereas the GPT-4's exceeded the data-to-text task by adding details about almost every datapoint. Its descriptions were more engaging and gave a good overview

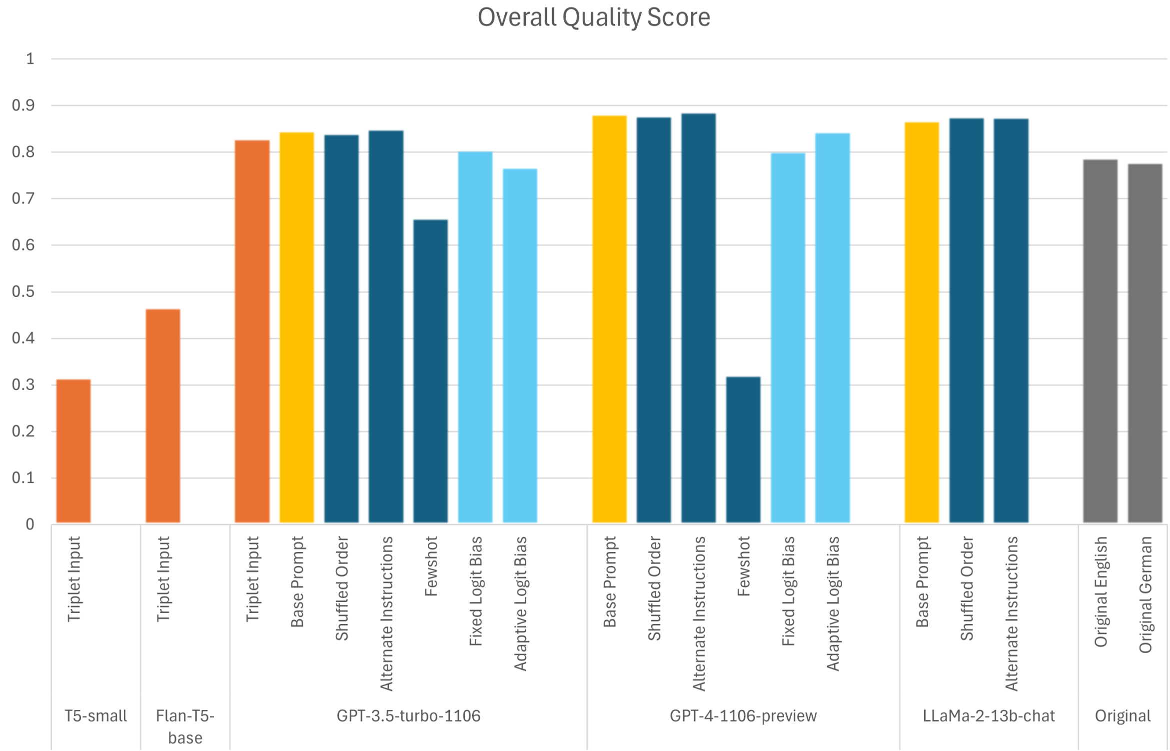



Figure 6.1.: Overall quality scores by G-eval by model and experiment. In orange and yellow are the scores for the baseline experiments depending on their input format, in blue are experiments with diversity augmenting techniques.

of what to expect from the band. In some cases, the longer texts felt tedious and regularly surpassed the token limit we had imposed on the model.

Lastly, we experimented with LLaMa2-13b, the intermediate version of LLaMa2. The quality of its generation was on par with GPT-4, although some differences can be observed. For one, LLaMa2 regularly uses emoji's, whereas GPT-4 only adds them when explicitly asked to do so. Another key difference is the diversity in the generation. We suspect that LLaMa2's great performance despite a smaller model size comes at the cost of diversity as around 90% of our base samples start with "`Introducing` <band name> `- the ultimate ...`". A more detailed analysis of the diversity is presented in the next section.

For now we only inspected the baseline experiments shown in orange and yellow on 6.1, as the other experiments are not aimed at improving the quality of the generated text and can be applied to almost any model.

## 6.2. RQ2: Comparing Similar Text/Evaluating the Diversity

We built a pipeline to compare the similarities of generated text using a pairwise approach. An n-gram based metric was deliberately chosen to focus the similarity analysis on the form

```
n = 3
Danny's Angels is an unplugged band based in Eching, available for birthday
parties, Christmas parties, church events, club festivals, corporate events, and
summer parties. We are willing to travel to cities within a 100.0km radius such
as Munich, Augsburg, and Ingolstadt. Our repertoire includes popular songs from
well-known bands like [The Beatles], [Ed Sheeran], and [John Mayer].

The Stingrays are a rocknroll band based in München. We are available for
birthday parties, Christmas parties, city festivals, club festivals, corporate
events, and summer parties. We are willing to travel to cities within a 100.0km
radius such as Nuremberg, Augsburg, and Ingolstadt for gigs. Our music is
influenced by iconic rocknroll bands like [The Rolling Stones], [The Beatles],
and [Elvis Presley].
```

Figure 6.2.: Manual diversity analysis: For a given n, the n-grams found in both texts are highlighted. This method allows us to quickly identify reoccurring expressions or patterns. It is important to also read non-highlighted parts of the texts as the simple n-grams do not pick up on more complex patterns.

of text, as the task and the experiments were not designed to change the meaning of the descriptions. The Jaccard similarity score was useful to interpret two texts' similarity at a glance as its scale ranges from 0 to 1, with 1 signifying identical texts. Then highlighting the common n-grams for a manual evaluation (as shown in Figure 6.2) was a very effective and time-efficient way of identifying reoccurring expressions or patterns, as the visual cues were easy to spot. Nevertheless, non-highlighted parts of the text also had to be read as the simple n-grams were not able to pick up on more complex patterns, word order changes or insertions.

Moving away from a single comparison towards a more global analysis, we used a similarity matrix generated by applying the pairwise comparison on the generated datasets. This analysis was used to extract the most common ngrams in the dataset, and calculate an average similarity score of the dataset. As this measure is not entropy-based and doesn't account for length of text, it was paramount to use the same input dataset for all experiments. The results are easily comparable, as shown in Figure 6.3. Each model stayed consistent in length throughout the experiments, but there were important variations between the models, with LLaMa2 and GPT-4 usually writing longer texts than GPT3.5. While using the same input didn't solve the problem of text length when comparing multiple models to each other, it allowed us to compare the experiments to each other and to at least approximate the diversity of the generated text by model.

## 6.3. RQ3: Diversity of Generated Text

Knowing from the get-go that even the largest models were prone to repetitions, we set out to find ways to improve the diversity of generated text. We experimented with different

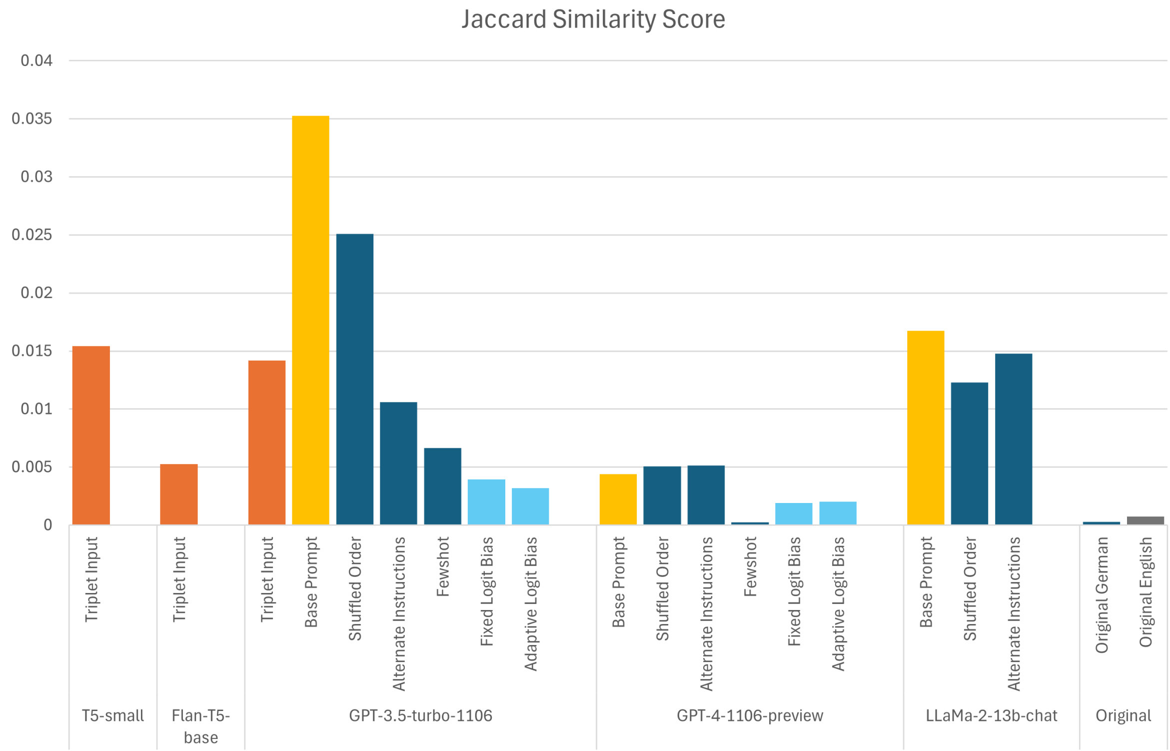



Figure 6.3.: Text similarity scores by experiment. The scores are calculated by averaging the pairwise Jaccard similarity scores of the generated text and are used to show the diversity of an experiment.

parameters and techniques to improve the variety of generated text.

Skipping the T5 models, as they were not able to generate text of sufficient quality, the first steps were actually in the **prompt engineering** phase, where we tried to push model's to augment the input data. As an example, we asked to include nearby cities or possible bands and songs a band might cover. The thought was that by using CoT techniques, models would have more data to work with and thus be able to generate more diverse output. Compared to the original approach consisting of **data-triplets**, this approach surprisingly reduced diversity as shown by the scores for GPT-3.5 in Figure 6.3. Manually inspecting the results showed that GPT3.5 with data-triplets created more diverse formulations that were not all on topic because the model had been given few instructions. Meanwhile, the prompt-based approach was more precise, pushed the model to use more data like city names and correctly used the distance data. However, since the prompt was always the same, the output followed its structure and was thus less diverse. We can see that the base prompt was followed, just by examining the most common tri-grams (stemmatized):

```
('to', 'travel', 'to'),
('travel', 'to', 'citi'),
('citi', 'within', 'a')
```

All of them are related to the part in the prompt where the model is asked to include nearby cities. Their presence had a significant impact on the similarity metric as they belong to the same expression that appeared in over 30 of the 50 samples: 'to travel to cities within a'. What happened is that in the process of bettering the relevancy and quality of the output, we had to unexpectedly compromise on diversity, thus warranting and necessitating further research in the domain.

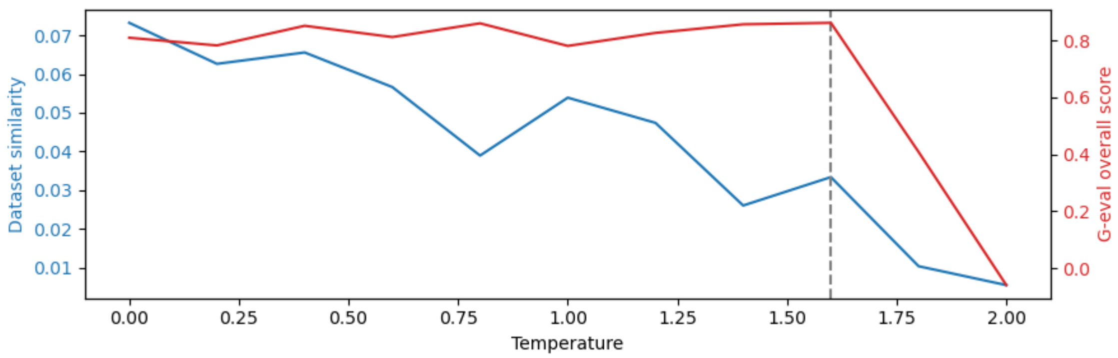



Figure 6.4.: The temperature's effect on similarity and overall quality for a small number of samples. Samples were generated with GPT3.5.

To increase the diversity again, we experimented with parameters such as the temperature which can have a moderate impact on the diversity of samples by introducing randomness. For the GPT models, modifying the **temperature** did not impact the quality, up to a threshold of around 1.6, where catastrophic randomness took over according to the G-eval metrics

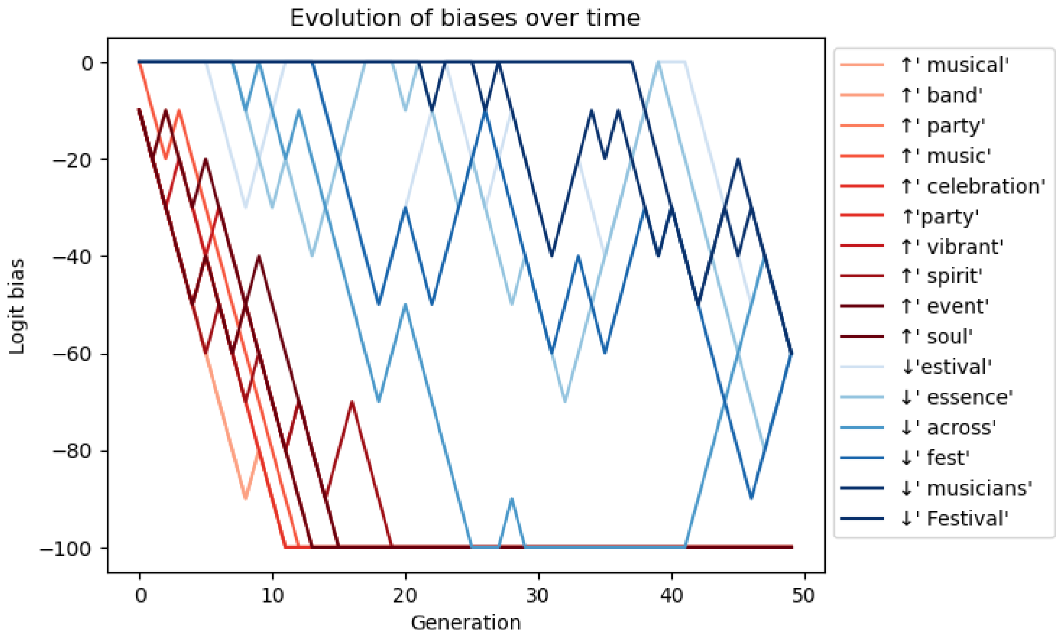



Figure 6.5.: The evolution of the logit bias values for some tokens over 50 generations using an adaptive score (5.3.1). The most used tokens (red) did not get eliminated despite the biases quickly reaching their maximum value. Less used tokens among the top100 (blue) were successfully limited, as shown by the fluctuating bias values.

and manual verification. Figure 6.4 shows the effect of temperature on quality and diversity during a small parameter experiment on ten samples. The similarity decreases slightly while the quality remains high until the output collapses for high temperatures. This shows that we should aim to maximize the temperature. We chose a temperature of 1.4 for subsequent experiments, as an exploration in OpenAI's playground produced nonsensical output with a temperature of as low as 1.5 in rare cases. With careful tweaking, the **top-p** parameter could have been used to push the threshold of nonsensical output slightly further. However, since the effect was negligible and we could use a high temperature value, we discarded this parameter. Lowering the top-p value would have had a negative impact on diversity, so we kept its default value.

Instead, we experimented with **Logit Bias**, which is a more direct way of controlling the output. Unfortunately, its use is poorly documented on OpenAI's website, and it was more challenging than expected to use effectively. For one, there are multiple ways to encode the same word using tokens, but spaces, special characters, and capitalization also modify a word's tokenization. This makes it difficult to use the logit bias to target specific words. In Figure 6.5, we can see the changes of the logit bias values when we used the adaptive method described in 5.3.1. The most used tokens (red) had their bias values quickly reach the maximum and stay there, meaning they were still present in subsequent generations. Lesser used tokens that were still in the top 100 (blue) seem to have been successfully limited in their usage as shown by the fluctuating bias values. The Logit bias experiments had a strong effect on the diversity as shown by 6.3, however in some cases words would simply miss in the middle of sentences. Thus strong logit biases affect the quality of generation (Figure 6.1), leading us to conclude that Logit Bias is a good way to improve diversity but should be used sparingly and implemented carefully.

Moving over to prompt-level mechanisms, we experimented with shuffling the input data. We noticed earlier in our research, that most generated texts kept the order of the input for similar data such as event types and music genres. Thus, shuffling this data would also modify the generated text. This method had a significant impact on the diversity for GPT3.5 and LLaMa2. For GPT-4, the diversity seems to have been reduced slightly instead, although it is still significantly lower than for the other models. An explanation can be found when manually analyzing the results, as the model was already reordering parts of the input data while adding some text in between. These snippets did not change with shuffled input, they were merely reordered. All models had in common that shuffling the order of the input didn't impact quality.

Utilizing alternate instructions had a different impact on each model. Despite having the same meaning, the different formulations of the instructions were supposed to lead to other focus points and thus generate more diverse samples. In the case of LLaMa2, just changing the first sentence in the prompt lead to differences in the output, such that the model started using other expressions at the beginning, unlike the baseline results briefly addressed in 6.1. Of course, each set of instruction had its own beginning, but at least it was possible to influence the model's output. For the GPT3.5 model, the effect on diversity was very pronounced, having an increased effect compared to just shuffling. With GPT-4 it was once more the inverse, similarly to the shuffling experiment.

Lastly, we experimented with fewshot techniques, by passing an existing band description with a simpler prompt. The goal was for the model to adapt to another description, originating from a much larger pool, possibly learning from other users of the platform. Unfortunately, this experiment was a failure, as both GPT versions generated very low-quality texts. In the case of GPT-4, the score even dropped to T5-small levels because at some point in the answers, it would just start spewing random words, similarly to when extremely high temperature values were used. The reason for this is not entirely evident, but it is possible, that the fewshot samples were of lower quality or non-sense (we used the translated versions of our references), used different structure than the models could handle or that the models somehow got confused by the prompt in general. It is evident, that with such a drop in quality the extremely good diversity scores are not representative of fewshot's potential.

Overall, all models and experiments suffered from repetitions and were not able to become as even remotly as diverse as the original references. Nevertheless, our experiments were able to successfully improve diversity and reduce repetitions while maintaining an overall level of quality (except fewshot approaches). Ranking the models performance, GPT-4 was the best for being consistently diverse with no improvements through our experiments except Logit Bias. For LLaMa2 and GPT3.5, the results are more nuanced, as the baseline of LLaMa2 had more variety than GPT3.5's, but through simple experiments, the latter was able to

| Model | Fluency Avg. | #Scores >=2.9/3 | Naturalness Avg. | #Scores >=2.9/3 | #samples |
|---|---|---|---|---|---|
| Original DE | 2.48 | 4 | 2.75 | 18 | 50 |
| Original EN | 2.42 | 22 | 2.74 | 28 | 50 |
| GPT-3.5 | 2.74 | 41 | 2.98 | 49 | 50 |
| GPT-4 | 2.94 | 46 | 2.98 | 49 | 50 |
| LLaMa2 | 2.99 | 20 | 2.97 | 19 | 20 |

Table 6.1.: Fluency and Naturalness scores from G-eval.

surpass the former on diversity. Quality-vise GPT3.5 scored consistently lower than GPT4 and LLaMa2 but virtually all experiments with these three models scored higher than the human-written references.

## 6.4. RQ4: NLG Compared to Human Written Text

Our last research question is about comparing generated texts to human-written texts, especially in terms of fluency and coherence. Despite having automated fluency and naturalness metrics, we found it difficult to rely on those to compare generated text to the original human-written descriptions. This is because the metrics weren't perfect and could be biased towards LLMs, but also because the human-written texts were not of uniform quality. This bias is exacerbated by the fact, that the original descriptions were in german while the evaluation was done using an english prompt. Running the same evaluation on translated versions had its own caveats, as the translations performed by Deepl were not always perfect and could have produced lower quality text.

Table 6.1 demonstrates the difficulty of comparing human-written text to machine-generated text using automated metrics. The human-written answers are consistently rated worse, with around half of the samples achieving a score of 2.9 or higher out of 3 when translated to english. In contrast, the machine-generated are all highly rated on these two metrics, achieving the high-scores in almost all cases. This pattern extended to the other metrics as well as seen in Figure 6.1.

In the absence of reliable automated metrics, we had to investigate manually by reading through the generated texts and comparing them to the original human-written texts. During this process, we noticed that some of the translations were of poor quality, despite the original descriptions coming from the curated `task1_div` dataset. Ignoring diversity and engagingness, the generated texts were of similar fluency and naturalness as the original texts. This is expected as the models we used are known for their fluency, naturalness and grammatical correctness.

Comparing the texts on other factors also proved complicated as the main difference between machine-generated and human-written samples was in the content. Whereas the machine texts were limited to datapoints, many of the human-written texts included personal anecdotes or experiences that were not included in the structured data. The original descriptions usually didn't use all of the datapoints such as cities they play in. Their responses also were more nuanced, especially in the context of music genres, compared to the rigid options specified in the dataset.

## 6.5. Task2: Adding Gigs

In an effort to explore other ways to employ data about bands, we ran a pilot experiment by adding gigs and writing news articles about their latest events. Below is a sample of such an article, with gigs added to the input data. It seems that the model was able to use the new data, yet both the query and the data need to be improved before this approach can be used in production. First of all, the data contains no information about the importance of each gig nor the relevance for a news article. Secondly, the model (GPT4 in the example provided below) mentions each event individually, and doesn't aggregate them with the current prompt.

```
Holzhausen's Very Own Tromposaund Enchants Audiences Across Bavaria

In the heart of Bavaria, one band has been making waves with its eclectic mix
of genres and feel-good performances. Meet Tromposaund, Holzhausen's homegrown
ensemble that has perfected the blend of blues, brass music, folk, jazz, and
swing. Their performances, as varied as their music, range from intimate
wedding parties to lively fairgrounds, ensuring there's never a dull moment
when they're on stage.

Tromposaund has recently dazzled at several significant events, showcasing
their ability to not only entertain but also to connect with diverse crowds. On
 May 28, 2022, they were the heart and soul of a wedding party in Mamming. The
event was a unique challenge, as the couple sought a band capable of
celebrating their love and impressing a Musikerfamilie. Tromposaund rose to the
 occasion, engaging international guests in multiple languages and even
incorporating a traditional Bavarian 'Brautstehlen' with class and humor. Their
 skills shone as they integrated the musical talents of the bride and groom's
family and tactfully curated a playlist filled with a touch of the '90s
nostalgia.

Less than a month earlier, on April 9, 2022, Tromposaund brought their
signature swing to a joyous wedding in Aindling. Their performance was
described as the heartbeat of the Bayerische Hochzeit, keeping guests laughing
and dancing throughout the day.

Their calendar of events continued unabated as Tromposaund took their talents
to a fairground in Weßling on April 22, 2023. Their ability to adapt and
produce a festive atmosphere was evident as they catered to visitors' diverse
```

```
tastes and interests.

Moving forward into the summer, Tromposaund is scheduled to appear at a
weddingparty on June 24, 2023, in Arnstorf, anticipating yet another
opportunity to display their musical prowess and their flair for making every
occasion unforgettable.

Tromposaund stands out not just as a band but as a vital thread in the fabric
of local celebrations, stitching together memories with every note they play.
It's clear that, in Bavaria, when there's a special moment to be commemorated,
Tromposaund is the name on everyone's lips. Join them as they continue their
journey, bringing harmony and high spirits to every performance.
```

## 6.6. Task3: Regional Summaries

Our second pilot project attempts to generate regional summaries of the music scene in a given area. This type of report could be used by all users on a website or by the platform itself for internal use. We only had enough data for the state of Bavaria, and in general, diversity is less important compared to the ability to summarize large amounts of structured data. We provided GPT4 with data about 261 events in a time window of 6 months before and after the querydate. We also provided the current date, as GPT4 doesn't have access to the current date. We were worried, that the query would be too large for GPT-4's context window, as the query was 15260 tokens long, but in November 2023, OpenAI upgraded the context window for gpt-4-1006-preview to 128K tokens, so this was not an issue. Below is a sample output of such a report.

```
The music scene in Bayern is vibrant and diverse, attracting audiences with a
variety of tastes and hosting events across numerous towns and cities. Among
the recent and upcoming events, wedding parties emerge as popular occasions for
 live music, with bands such as Partyband Hi!light, LUPOS, and Lets-Dance-
Partyband frequently providing entertainment ranging from classical to country
and party genres. Notably, the town of Hauzenberg appears to be a hub for
wedding celebrations with alternative and rockabilly music being in demand, as
seen with bands like BayernLive-Band and MATRIX being lined up for performances
.

Fairgrounds also serve as key venues, showcasing genres like classical, folk,
and dance music. Events such as "Donau in Flammen" in Vilshofen an der Donau,
and traditional fairs in locations like Ortenburg and Straubing, are attracting
 bands including Notausstieg Partyband that cater to the loungemusic and
classical tastes of the local populace.
```

```
Club festivals and corporate events are not left behind. Forstenstein club
festival booked Notausstieg Partyband, pointing towards an appetite for rock
music in more informal settings. Meanwhile, corporate events in Deggendorf
anticipate performances from bands like Hitbox, catering to the need for
diverse music styles at company gatherings.

In summary, Bayern's music scene is characterized by a strong tradition of live
 bands at weddings and local fairs, with upcoming events signifying a continued
 preference for genres that blend the traditional with the contemporary. This
regional summary captures a snapshot of a vibrant, music-loving community that
supports a wide array of genres and bands.
```

# 7. Discussion

The structure of this chapter mirrors that of chapter 6, with each section corresponding to one of the research questions. We will discuss the results of our experiments, contextualize them, talk about their implications and their limitations. At the end, we will discuss the implications of our research for the field of Data-to-Text generation and the thoughts on implementing such a system in an industrial context.

## 7.1. RQ1: Choice of Technologies

On the search for the best possible approach for our data-to-text task, we explored multiple avenues, such as finetuning with T5-small and Flan-T5, as well as more monolithic approaches with the LLMs GPT3.5, GPT4 and LLaMa2. We compared different input formats and their caveats, as precise prompts improved quality at the price of creativity and diversity. We then explored ways to keep the high quality while increasing diversity.

First of all, our smallest models, T5-small and FlanT5-base were not able to produce quality text despite finetuning. It is possible, that the finetuning data was either not sufficient or of poor quality, as it had to be automatically translated from German. In the case of T5-small, it is more likely, that the model got overwhelmed during finetuning, because the reference texts were quite long and contained both a lot of data and simultaneously not all of the input data. As the fewshot experiments also failed on bigger models, we believe the dataset to be at fault. We therefore recommend carefully curating datasets before finetuning and making sure that the input data covers most of the references, unlike in our case. In the case of T5-small and FlanT5-base, utilizing pipeline approaches like PlanGen [18] or similar could probably improve the results by breaking up the task into smaller subtasks.

In the case of our larger models, zero-shot approaches performed very well, with minor differences in terms of grammatical quality, fluency and engagingness. There were more pronounced differences in text length and content, with GPT3.5 falling short of the other two. While all models were provided with the same input, GPT4 and LLaMa2 could extrapolate more details, suggesting to a potential event planner why this band would be a good fit and what they could expect. Meanwhile GPT3.5 provided a more generic and concise description of the band.

Based on the results, the choice of model has the most significant impact on the overall results, with the size of the model being one of the most important factors, but its focus or goal playing a huge role too. LLaMa2 surprised us with its high quality despite being over ten times smaller than the other LLMs, which goes to show that a well designed model can outperform larger models. The drawback is its reduced diversity in generation, which is one of the main motivations of this thesis. Our findings are mostly in line with the literature and common rankings of LLMs [83], although we found LLaMa2 to be more capable than GPT3.5 for our use-case.

We conclude that GPT-4 is the best monolithic model we experimented with for our use case, partially its high quality and partially for its output consistency, a finding from 6.3 and recommend comparing these results to non-monolithic approaches, especially in the case of smaller models.

## 7.2. RQ2: Comparing Similar Text/Evaluating the Diversity

On our quest to generate diverse datasets, we had to design and implement ways to compare texts and measure diversity. We opted for a simple approach based on n-gram comparison. This approach might not work for all use cases and lacks precision, but it had distinct advantages for us. It was efficient, allowing large quantities of comparisons in a short amount of time. It was easy to use and even easier to understand and visualize.

With values that could be roughly compared, we were able to measure the diversity of the datasets and see the impact of changes in implementation. Through the visualization function, we were able to identify patterns in real-time and adjust the implementations and even detect situations an n-gram based method would normaly miss by looking outside of the highlighted text. This shows that despite its simplicity, the method is effective and can be used to improve the diversity of generated text. It furthers the thought that in some cases, a simple solution is as effective as advanced metrics and possibly explains why simple metrics like BLEU are still used in research despite its shortcomings and critique.

A drawback of our process was obviously that the visuals were a manual and finegrained process, which didn't improve on the underlying diversity metric. The metric itself can not be taken as a precise or universal measure of diversity because for it to be effective and comparable, the datasets need to be of the same size and the text length does impact the Jaccard similarity score. We found that short unprecise text such as those generated by Flan-T5 with a triplet input rivaled in diversity with the longer and more precise texts of GPT4. The visualization also had its shortcomings, as the highlights could distract from the overall text and overarching patterns could be missed when skimming over the colors instead of reading the texts.

All in all, the metric we implemented worked well for our use case, but we recommend moving away from such a niche metric for future research and instead use concepts like entropy [76] to develop a more universal measure of diversity. Furthermore, our topic only handled form diversity, while content diversity [43] is also an important factor in most tasks.

## 7.3. RQ3: Diversity of Generated Text

Our main research topic was trying to understand how to generate diverse text with LLMs. To this end, we implemented many experiments and used multiple models to explain how different parts interact and how models can be steered toward more diverse samples. One of the first insights was that the format of our input data hugely impacted the diversity of the output. In trying to leverage CoT to augment the dataset, we gave more instructions to the model, which inadvertently reduced the output diversity compared to freer inputs. Since these additional instructions were useful for the quality and relevance of the output, we had to keep them and investigate other methods to reach the diversity levels of an unconstrained prompt while maintaining a similar quality. This validates one of our initial claims that data-to-text systems must be designed carefully to reach the desired output, and especially to reduce repetitions over multiple samples. Further attention should be paid to methods that open queries instead of restricting them, while keeping the generation on the task.

To mitigate the negative side effects of precise prompts on diversity, we inspected parameters and then more advanced methods. Our first experiment was a simple temperature optimization which also served as an initial test for our evaluation metrics. With increasing temperature, the diversity of the output increased as would be expected, but the quality metric stayed the same until the output became non-sensical. This was surprising but a good sign that the models were able to handle introduced randomness well. Thus we proceeded with two experiments modifying the logits of the models. In a way, these two experiments transferred information about previous generations to the upcoming one in a memory-efficient way, as a maximum of 100 values can be passed to the GPT models. The results dramatically increased the diversity of the output despite saturating biases in some cases and negatively impacting the quality. Other approaches exist to pass previous data to the model, with the obvious one being to keep the old outputs in memory or in the case of GPT in the chat history and, modifying the prompt to tell the model not to reuse previous expressions. In the long run, this approach would however cost more, require the whole message chain to be stored up to the context window size, and depend on the model's adherence to the prompt.

Our approach, while working on a token level instead of word- or event expression-level is more direct, as the biases are applied at the end of the decoder stage, right before the softmax function that converts logits to probabilities. Our methods only applied negative biases on the most common token, but more advanced and fine-grained control is possible to achieve the best possible output. Here, choosing a method is a trade-off between controllability, ease-of-use and efficiency. We recommend using the logit bias sparingly and paying careful attention when designing compute algorithms for adaptive biases, as their impact can quickly disrupt a model's output. When used correctly, they are a powerful and precise tool to steer our models.

Next, we explored algorithms modifying the input, such as reordering the data inside the prompt, randomizing some of the additional instructions or adding description examples in few-shot prompting. The results were mixed, with the fewshot approach basically failing without us understanding why. A manual inspection showed that the translations passed as templates were usually of inferior quality to their original counterparts, but they were still written in correct english and should not have caused the models to start spewing out random words. Luckily, our other experiments had more success, with GPT3.5 significantly benefiting from shuffling the input order and changing the instructions slightly. Interestingly, GPT4 was not influenced at all by these changes, and LLaMa2 less than GPT3.5. This shows that the models are not equally affected by the same changes and that the same methods can not be applied to all models. A manual inspection of the results showed that GPT3.5 wrote short texts, densely packed with the provided information, but with little creativity, merely inserting a few adjectives to make the text more interesting. Meanwhile, GPT4 and LLaMa2 were more creative and verbose, trying to add a small anecdote to each data point. Usually, these anecdotes explained why the band would be a good fit for such an event, but we didn't have enough information on the actual bands to fact-check any claims in the generated texts.

Circling back, each model's reaction to added randomness lets us reflect on the model's inner workings. Unfortunately, not a lot of details are known about the training procedures of GPT3.5 and GPT4, so we can only analyze our results and make conjectures as to what led to their different reactions. GPT3.5 being strongly affected by small changes could indicate that it was trained on text-to-text tasks such as ours with the objective of being sensitive to small changes in the input. Meanwhile, GPT4 and, to some extent, LLaMa2, being less affected by the same changes, could indicate that the models "understand" the underlying task or intentions passed on through the prompt and react less to a reformulation of the same task. This is a very interesting finding because it highlights how different and powerful models can be when trained rigorously and with specific objectives, such as understanding. It is also interesting to see that a small public model such as LLaMa2 is essentially able to outperform a much larger model such as GPT3.5, both in quality and diversity (only without alternative instructions).

To conclude, we have shown that diversity in-between samples can be loosely controlled by introducing randomness into the generation process, either through the temperature or through the input prompt. We have also shown that each model reacts differently to prompt modifications based on their understanding of the task. Due to this, we recommend verifying a model's reaction to perturbations in the input before using it in production. Instead, our most influential method for diversity, logit bias modification, could be used to control a model, or alternatively, previous outputs could be kept in memory, although the effectiveness of this technique was not tested in this thesis. We understand that this is not an exhaustive list of methods to control diversity, but we believe it is a good starting point for future research. Another unaddressed task is to implement and compare our approaches to pipeline methods, which claim to have more targeted control over the data-to-text generation, as only a few of them attempt to master diversity over multiple samples.

## 7.4. RQ4: NLG Compared to Human-Written Text

Trying to compare our generated material to the references proved difficult due to underlying differences between the input data we used and the content of the references. The references were written by humans, who had access to more information about their own band than we did. Usually, they did not include all the database information in their texts because parts of that data were displayed elsewhere on their profile. Additionally, our automated metrics only worked reliably for english texts, requiring translations for the German references. This translation probably introduced a few errors, which affected the scoring metrics. We also suspect the G-eval metrics to be slightly biased toward other LLM-generated content since the references had been selected from the dataset for their quality, yet they still scored lower than the generated texts on most quality metrics.

These problems pushed us to manually evaluate and compare the samples to the references, which was a time-consuming process. We found that the generated texts were of similar grammatical quality, but due to the limited input data, the content expectedly different. In the case of GPT-4 we obtained longer texts, that sometime felt more hollow than engaging. Nevertheless, most generated texts were fluent and enjoyable to read, with GPT-3 texts being easily identifiable due to the short length and strict adherence to the input data.

## 7.5. Task 2 and Task 3: Pilot Experiments

We conducted two pilot experiments showcasing additional tasks a data-to-text system could perform on a platform. While initial results were promising, as is often the case with LLMs, we would require at least one human written reference for each task to properly judge the results. They nevertheless showcase what is possible in zero-shot scenarios and how far NLG has come in just a few years. Since these tasks require larger amounts of data, particular care is required to verify the factuality of each claim in the output as hallucinations are very likely to occur.

## 7.6. Thoughts on implementing a Data-to-Text system in industry

We implemented a first system for the industry-partners platform using our research in this thesis. Our aim was to conduct a statistical analysis by comparing our generated propositions to the descriptions chosen and modified by the artists. While the system was not released before the end of the thesis, we can already draw some lessons from the process of implementing such a system in the industry. A screenshot of a possible frontend implementation is displayed below: 7.1.

We intentionally limited our research to purely generating new text, while in reality, improving existing texts would probably be more useful to the user, and yield more diverse results as the model would be able to rely on an existing text. In a broader context of automated product descriptions, our research could also be applied to generate new descriptions for products that do not have one yet. In the context of multiple languages, most models still struggle when it comes to mixing them: As an example we once generated German descriptions but our input data was in English. In the German output, we had the word "brasmusic" instead of the correct "blasmusik". When we instead generated an English description and then asked the model to translate it, all words were in their correct language.

From an efficiency perspective, using these huge models has a few drawbacks. One is the time of generation, which was of around six seconds for GPT3.5, 20-30 seconds for GPT-4 and LLaMa2 running on the own hardware would vary between 20 seconds for powerful GPUs and three entire minutes running through the LLama.cpp framework on a laptop CPU. This is a long time for users to wait and needs to be considered when designing the system. Another disadvantage is the cost of running these models, with locally run models requiring expensive GPUs and cloud-based models requiring a subscription. Especially GPT-4 is expensive to use, with a single description costing up to 0.10$ to generate.

A problem we encountered during the months of research, was the fast paced changes in API systems, from obsolete models being removed, to unexpected changes both in the API and the models performance [84]. This makes it difficult to recommend API systems in a production environment, as they can change without notice and require constant care.

Systems in contact with users, also need to be thoroughly inspected for potential abuse and misuse. If we were to use fewshot techniques or keep old conversations to increase diversity, we would need to ensure, that no information could leak out. In case human input is integrated, we would need to sanitize it.

## 7.7. Future research

In the future we would like to see our experiments reproduced on a simpler dataset to properly compare the results to other research, smaller models and pipelined approaches and methods. Especially our smaller models could have benefitted from content plans or breaking up the task into smaller subtasks. Furthermore, we see potential in breaking up the single query using CoT into multiple queries, as the final query could then be written in a less restrictive way, with previous queries serving as data augmentation.

More research is also required to increase diversity in generational tasks, as methods that rely on other factors than randomness need to save and pass on information from previous generations. Our method was to use logit biases, but other methods exist and they should be investigated for their effectiveness and efficiency.

In the realm of evaluation, advanced metrics like G-eval were very promising, but their reliance on a constantly changing API-based LLM has significant drawbacks. Further investigation of biases of such models is required. For diversity evaluation metrics, we don't recommend our method for general use, but believe that an algorithm based on entropy could be a good starting point for future research.

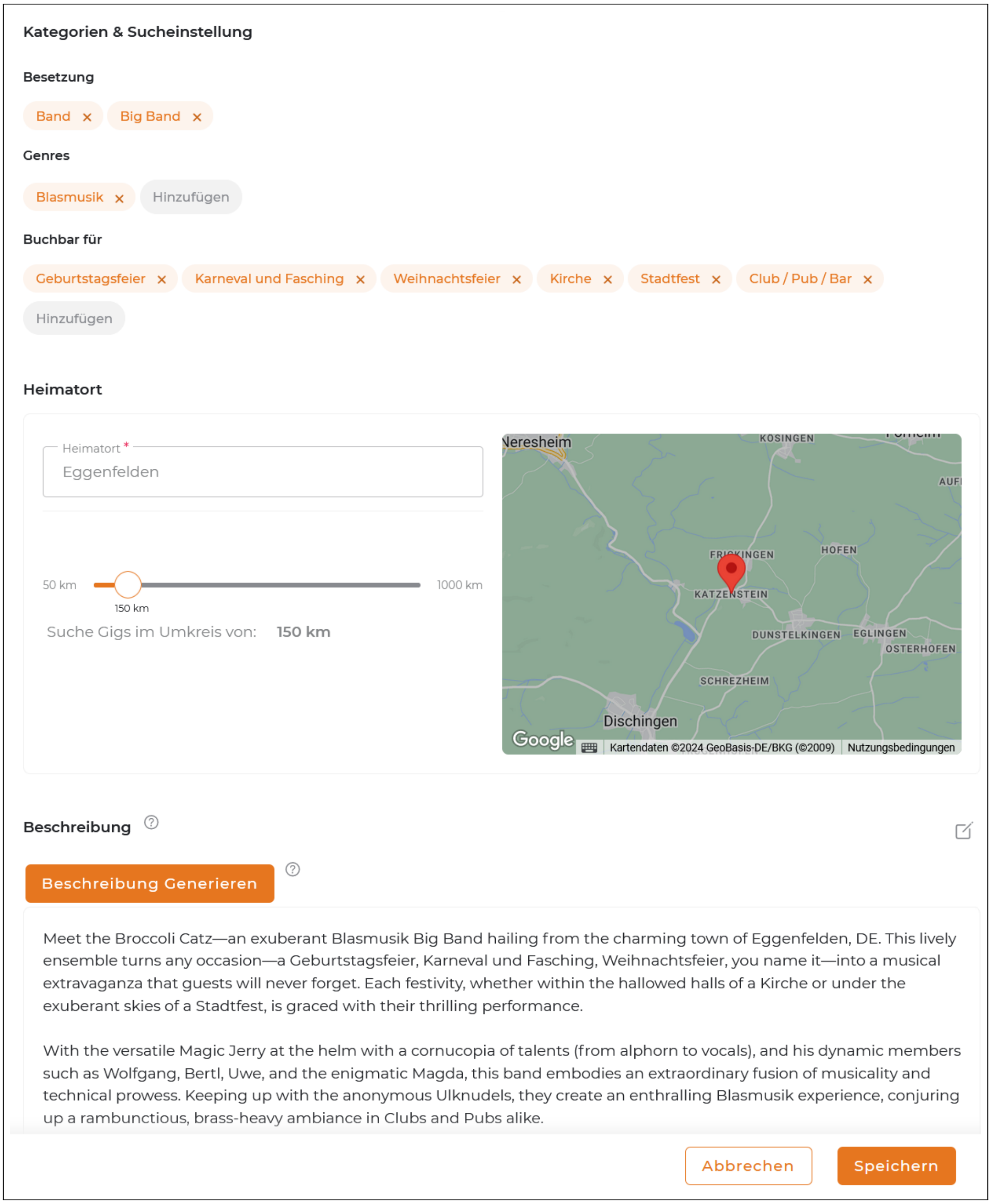



Figure 7.1.: Screenshot of a potential implementation of our system on Connactz's platform. Note the absence of an additional instruction field, as the data-to-text approach aims to reduce the necessity for human input.

# 8. Conclusion

This research implemented a robust evaluation pipeline relying on an LLM to judge different aspects of text quality, all without using references. We designed and implemented a straightforward but effective way of measuring, ranking and visualizing text similarity, which enabled us to compare the diversity of texts to each other, as well as entire datasets and models. We then experimented with different models, methods and input formats to see how each affected the quality and more importantly the diversity of generated descriptions. We found that randomness, induced by query shuffling, temperature parameters and prompt modification, had different impacts on each model and was thus unreliable at improving the diversity. For a more consistent and reliable diversity-enhancing technique, we found that transferring previously used tokens to new prompts was a very effective, controllable and reliable way. This technique has the drawback, that its scalability is limited by the amount of information a model can ingest. Future research on diversity should start with simpler datasets, as our's didn't have reliable references and concentrate on the most effective way of transferring relevant information between prompts.

We also found during our prompt engineering phase, that inlcuding CoT techniques in prompts was very effective at improving the quality and relevance of the output and augmenting the existent data with new insights, like geographical knowledge. On the flip side, this restricted the diversity of the output, as the model was boxed in with additional instructions. To tackle this tradeoff, we suggest researching ways to augment the data without restricting the model's creativity.

We hope our insights will provide a good base for potential implementations of data-to-text systems in production environments, as well as inspire future research on the topic of diversity in NLG.

# A. Addenda

## A.1. Example Prompt

This is the main prompt and the variations we ended up with after prompt engineering. Words in double brackets were replaced by the corresponding data from the database. Some of the alternate beginnings and endings are also shown below, each separated by a semicolon.

```
Create a captivating and comprehensive band description with the following
    ↪ details:
Name: {{Name}}
Location: {{Location}}
Music Genres: {{Genres}}
Event Type(s): {{Events}}
We are a {{Type}}
Mention in the description some cities within a {{Radius}}km radius (without
specifying the distance) to which we can travel for gigs, and also mention
suitable well-known bands or songs if you believe we play them. Bands and songs
should be enclosed in [].
The text should be concise but include all the important information.
```

**Alternate Beginnings:**

```
Create a short yet captivating band description using the information below:;
Order, select and use the data below to write a band description that will get
    ↪ the attention of event organizers.;
Present the following band or dj. The target reader is an event organizer.;
```

**Alternate Endings:**

```
Ensure the description is short but engaging and clearly portrays the band's
    ↪ style and versatility.;
The text should be easy to skim through but also motivate event organizers to
    ↪ hire the group.;
Use the event types and music genres that you think are most relevant for this
    ↪ band, you can leave out the others.;
```

## A.2. Geval

### A.2.1. Geval_engagingness

This is the prompt used by Geval to judge the engagingness of the text. {{Description}} is replaced by the generated text.

```
You will be given one description written in german or english for a band or DJ
    ↪ .

Your task is to rate the descriptions on one metric.

Please make sure you read and understand these instructions carefully.

Please keep this document open while reviewing, and refer to it as needed.


Evaluation Criteria:

Engagingness (1-5) Is the text dull/interesting?

- -1: This is not a description of a band or DJ.
- 1: (Low Engagingness) Dull, lacks creativity and excitement, no captivating
    ↪ details. Uninspiring and unlikely to hire.
- 2: (Below Average Engagingness) Somewhat interesting, but lacks uniqueness.
    ↪ Basic information, bland language. Might consider if no better options.
- 3: (Average Engagingness) Decent level of engagement, informative but not
    ↪ exciting. Might be considered if criteria align.
- 4: (High Engagingness) Exciting and enthusiastic. Unique selling points,
    ↪ captivating language. Highly likely to hire.
- 5: (Exceptional Engagingness) Extremely engaging, persuasive, and creative.
    ↪ Creates strong desire to hire without hesitation.


Evaluation Steps:

1. Read the description carefully to get a sense of its overall content and
    ↪ style. If it is not a description of a band or DJ, rate it -1 and skip
    ↪ the other steps.
2. Consider the language used, the information within and the structure of the
    ↪ sentences. Is it engaging and attention-grabbing, or does it lack
    ↪ creativity?
3. Rate the description on a scale of 1 to 5, with 1 being dull and 5 being
    ↪ highly interesting.
```

```
Remember to focus on the engagement factor of the text, rather than the
    ↪ accuracy of the description or personal preferences for music genres.


Description:

{{Description}}


Evaluation Form (scores ONLY):
```

### A.2.2. Geval_fluency

This is the prompt used by Geval to judge the fluency of the text. {{Description}} is replaced by the generated text. The metric is based on fluency as defined in [41].

```
You will be given one description written in german or english for a band or DJ
    ↪ .

Your task is to rate the description on one metric.

Please make sure you read and understand these instructions carefully. Please
    ↪ keep this document open while reviewing, and refer to it as needed.


Evaluation Criteria:

Fluency (1-3): the quality of the description in terms of grammar, spelling,
    ↪ punctuation, word choice, and sentence structure.

- -1: This is not a description of a band or DJ.
- 1: Poor. The description has many errors that make it hard to understand or
    ↪ sound unnatural.
- 2: Fair. The description has some errors that affect the clarity or
    ↪ smoothness of the text, but the main points are still comprehensible.
- 3: Good. The description has few or no errors and is easy to read and follow.


Evaluation Steps:

1. Read the text. If this is not an artist description, assign a fluency score
    ↪ of -1 and skip the other steps.
```

```
2. Check for grammar (1-3).
3. Assess spelling (1-3).
4. Evaluate punctuation (1-3).
5. Review word choice (1-3).
6. Analyze sentence structure (1-3).
7. Assign an overall fluency score (1-3) by averaging the scores from steps
    ↪ 2-6.

Assign a score for each step and report only the overall fluency score.


Description:

{{Description}}


Evaluation Form (scores ONLY):
```

### A.2.3. Geval_naturalness

This is the prompt used by Geval to judge the naturalness of the text. {{Description}} is replaced by the generated text.

```
You will be given one description written in german or english for a band or DJ
    ↪ .

Your task is to rate the description on one metric.

Please make sure you read and understand these instructions carefully. Please
    ↪ keep this document open while reviewing, and refer to it as needed.


Evaluation Criteria:

Naturalness (1-3): Whether the utterance could plausibly have been produced by
    ↪ a human.

- -1: This is not a description of a band or DJ.
- 1: Poor. The description lacks human-like quality and appears artificial or
    ↪ robotic.
- 2: Fair. The description is moderately human-like but has noticeable areas
    ↪ for improvement.
- 3: Good. The description is highly human-like in quality and very natural
```

```
Evaluation steps:

1. Read the text. If it is not a description of a band or DJ, rate it -1 and
   ↪ skip the other steps.
2. Assess fluency (grammar).
3. Evaluate coherence (logical flow).
4. Check idiomatic usage.
5. Ensure contextual relevance.
6. Rate overall human-like quality by averaging the scores from steps 2-5.

Assign a score for each step and report only the overal naturalness score.

Description:

{{Description}}

Evaluation Form (scores ONLY):
```

### A.2.4. Geval_informativeness

This is the prompt used by Geval to judge the informativeness of the text. {{Description}} is replaced by the generated text. {{Data}} is replaced by data from the database. In our case, it is data about the band given in a structured way.

```
You will be given one description written in german or english for a band or DJ
   ↪  as well as the data we used to generate it.

Your task is to rate the description on one metric.

Please make sure you read and understand these instructions carefully. Please
   ↪ keep this document open while reviewing, and refer to it as needed.


Evaluation Criteria:

Informativeness (1-4): Judges whether the description provides meaningful and
   ↪ relevant information to the reader.

- -1: This is not a description of a band or DJ.
- 1: (Low - Minimal Information) The description lacks essential information
   ↪ and leaves the reader with an incomplete understanding of the band/dj.
```

```
- 2: (Basic Information) The description provides fundamental details but
    ↪ remains relatively concise, offering a basic understanding.
- 3: (Good - Comprehensive) The description offers a good amount of information
    ↪ , including essential details, providing a comprehensive understanding
    ↪ of the band/dj.
- 4: (Excellent - Above and Beyond) The description goes above and beyond,
    ↪ delivering a wealth of meaningful and relevant information, ensuring a
    ↪ thorough and in-depth understanding that exceeds the provided data.


Evaluation steps:

Short Evaluation Steps for Informativeness (1-4):

1. Read the German description. If the text is not a description of a band or
    ↪ DJ, assign a score of -1 and skip the other steps.
2. Evaluate how much of the provided data is in the description.
3. Assess the depth of provided information and how much additional data is
    ↪ provided.
4. Rate based on informativeness: 1 (Minimal), 2 (Basic), 3 (Comprehensive), or
    ↪  4 (Above and Beyond).

Assign a score for each step and report only the overal informativeness score.


Description:

{{Description}}


Data:

{{Data}}


Evaluation Form (scores ONLY):
```

### A.2.5. Geval_quality

This is the prompt used by Geval to judge the quality of the text. {{Description}} is replaced by the generated text.

```
You will be given one description written in german or english for a band or DJ
    ↪ .

Your task is to rate the descriptions on one metric.

Please make sure you read and understand these instructions carefully.

Please keep this document open while reviewing, and refer to it as needed.


Evaluation Criteria:

Quality (1-5): how good is the description?


Evaluation Steps:

1. Read Description
2. Clarity of Message: Assess clarity in communicating the band/DJ's identity
    ↪ and offerings.
3. Relevance: Evaluate how well the description matches potential listeners' or
    ↪  organizers' interests.
4. Engagement: Determine how engaging and interesting the description is.
6. Language and Tone: Analyze the appropriateness of language and tone for the
    ↪ band/DJ's image and audience.
7. Grammar and Syntax: Inspect grammar and sentence structure for correctness
    ↪ and clarity.
8. Brevity and Conciseness: Evaluate if the description is concise and to the
    ↪ point.
9. Overall Impression: What is the overall impression and how is memorable is
    ↪ it.
10. Scoring Guide:
    - -1: Unacceptable or not a band description
    - 1: (Poor) Lacks in most areas, needs significant improvement.
    - 2: (Fair) Adequate in some aspects but with considerable weaknesses.
    - 3: (Good) Satisfactory in most criteria, may lack slightly in some areas.
    - 4: (Very Good) Strong in most areas, minor flaws only.
    - 5: (Excellent) Outstanding across all criteria, very effective in
        ↪ presentation and appeal.


Description:
```

```
{{Description}}


Evaluation Form (number ONLY):
```

## A.3. Sample Output

A few samples for a band with the following data:

```
Name: Tromposaund
Location: Holzhausen
Music Genres: blues,brassmusic,folkmusic,jazz,swing
Event Type(s): birthday,fairground,weddingparty
We are a band,brassband
```

This section is meant to showcase each model's style for the same prompt and does not reflect the diversity of generation.

#### GPT3.5 - Triplet style prompt

```
Hey there, we're Tromposaund! Hailing from the small town of Holzhausen, we're
a brass band with a love for blues, folk, jazz, and swing music. Whether you
need some lively tunes for a birthday celebration, fairground event, or wedding
 party, we're ready to bring our unique sound and energetic vibe to your
special occasion. With a distance of 500.0, we're willing to travel to deliver
our uplifting music wherever it's needed. Book us for an unforgettable musical
experience!
```

#### GPT3.5 - Default prompt

```
Tromposaund is a dynamic brass band based in Holzhausen, specializing in blues,
 brass music, folk music, jazz, and swing. Whether it's a birthday, fairground,
 or wedding party, we bring the energy to get everyone on their feet. We are
available for gigs in cities within a 500.0km radius, such as Berlin, Munich,
and Prague. Our repertoire includes classics from well-known bands like [Louis
Armstrong and His Hot Five] and [Duke Ellington], ensuring a lively and
entertaining performance for all.
```

**GPT4 - Default prompt**

```
Introducing Tromposaund, the eclectic ensemble hailing from the quaint village
of Holzhausen, renowned for their vibrant fusion of musical styles that span
blues, brass music, folk music, jazz, and swing. As a brass band with a unique
flair, Tromposaund delivers an unforgettable auditory experience that resonates
 with the rustic charm of their origins.

Whether you're planning a birthday bash, a lively fairground event, or dreaming
 up your ideal wedding party, Tromposaund is poised to bring the musical
heartbeat to your celebration. The band's versatility ensures that each
performance is tailored to the festive atmosphere of your event, promising to
leave your guests enchanted and your special occasions imbued with a toe-
tapping rhythm.

Tromposaund is no stranger to the road, their music a passport to cities within
 a 500.0km radius of Holzhausen. Whether it's the vibrant cultural hubs of
Frankfurt or Munich, or the historical richness of Prague and Vienna,
Tromposaund is ready to journey forth and bring their soulful sounds to your
doorstep.

Fans of time-honored tunes and legendary acts will delight in Tromposaund's
repertoire, which often includes nods to iconic bands and songs. Imagine the
smooth, sultry tones of [Miles Davis]'s trumpet, the foot-stomping energy of [
Glenn Miller]'s swing classics, or the heartfelt narratives woven through [Bob
Dylan]'s folk ballads. And, for those who revel in the power of brass, expect
homages to the likes of [The Dirty Dozen Brass Band], echoing the festive
streets of New Orleans right to your event.

Prepare for a melodic journey with Tromposaund - where passion meets tradition,
 and every performance is a celebration of musical heritage.
```

**LLaMa2 - Default prompt**

```
Introducing Tromposaund - the unparalleled brass band that's sweeping the
nation with their intoxicating blend of blues, jazz, swing, and folk music!
Hailing from the charming town of Holzhausen, these sonic visionaries are ready
 to transport audiences to a world of euphoria and enchantment.

With a dynamic repertoire that boasts an impressive array of genres,
Tromposaund seamlessly weaves together the rich textures of blues, the soulful
```

```
melodies of jazz, the lively energy of swing, and the heartfelt melancholy of
folk music. Whether it's a birthday celebration, a fairground spectacle, or a
luxurious wedding party, this band is guaranteed to leave a lasting impression
on any audience.

As a quintessential brass band, Tromposaund is comprised of talented musicians
who have honed their craft in some of the most prestigious musical institutions
 in the land. Their mastery of brass instruments, combined with their
unparalleled showmanship, ensures that every performance is an unforgettable
experience.

If you're searching for a band that can bring the house down with a medley of
classic tunes, look no further than Tromposaund! Their repertoire includes
timeless anthems like Louis Armstrong's "When the Saints Go Marching In," Duke
Ellington's "Take the A Train," and Count Basie's "One O'Clock Jump," all of
which are sure to get your party started! And if you're looking for something a
 little more contemporary, Tromposaund can also deliver with their own soulful
takes on modern hits.

As for their travel radius, Tromposaund is more than happy to bring their
infectious beats and melodies to any location within a 500km radius. They've
already captivated audiences in nearby cities like Frankfurt, Cologne, and
Heidelberg, and they're ready to do the same in your hometown!

So if you're in search of a brass band that's guaranteed to leave your guests
dancing, singing, and begging for more, look no further than Tromposaund!
```

# List of Figures


1.1. Motivational example . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 3

4.1. Distribution of description lengths. . . . . . . . . . . . . . . . . . . . . . . . . . 20
4.2. Analysis of annotations . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 21
4.3. Map of bands . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 24

5.1. Evaluation Pipeline: Quality . . . . . . . . . . . . . . . . . . . . . . . . . . . 26
5.2. Evaluation Pipeline: Diversity . . . . . . . . . . . . . . . . . . . . . . . . . . 28

6.1. Quality scores by G-eval . . . . . . . . . . . . . . . . . . . . . . . . . . . . 34
6.2. Manual diversity analysis . . . . . . . . . . . . . . . . . . . . . . . . . . . . 35
6.3. Text similarity scores by experiment . . . . . . . . . . . . . . . . . . . . . . . 36
6.4. Temperature's effect on similarity . . . . . . . . . . . . . . . . . . . . . . . . 37
6.5. Adaptive Logit Bias Values . . . . . . . . . . . . . . . . . . . . . . . . . . . 38

7.1. Implementation Frontend . . . . . . . . . . . . . . . . . . . . . . . . . . . . 50

# List of Tables

1.1. Tasks . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 6

6.1. Fluency and Naturalness scores from G-eval. . . . . . . . . . . . . . . . . . . . . . 40

# Acronyms

**CoT** Chain-of-Thought. 4, 10, 12, 14, 37, 46, 49, 51

**LLM** Large Language Model. 1, 5, 7, 10, 12, 23, 29, 44, 51

**LM** Language Model. 8, 9

**NLG** Natural Language Generation. 2, 4, 6, 7, 11, 14, 48, 51

# Bibliography


[1] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. 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. M. 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*. en. May 2020. URL: https://arxiv.org/abs/2005.14165v4 (visited on 01/04/2024).

[2] J. Wei, Y. Tay, R. Bommasani, C. Raffel, B. Zoph, S. Borgeaud, D. Yogatama, M. Bosma, D. Zhou, D. Metzler, E. H. Chi, T. Hashimoto, O. Vinyals, P. Liang, J. Dean, and W. Fedus. *Emergent Abilities of Large Language Models*. 2022. arXiv: 2206.07682 [cs.CL].

[3] K. Hu. *ChatGPT sets record for fastest-growing user base - analyst note*. https://www.reuters.com/technology/chatgpt-sets-record-fastest-growing-user-base-analyst-note-2023-02-01/. [Accessed 14-01-2024].

[4] S. Jentzsch and K. Kersting. "ChatGPT is fun, but it is not funny! Humor is still challenging Large Language Models". In: *Proceedings of the 13th Workshop on Computational Approaches to Subjectivity, Sentiment, & Social Media Analysis*. Toronto, Canada: Association for Computational Linguistics, July 2023, pp. 325–340. DOI: 10.18653/v1/2023.wassa-1.29. URL: https://aclanthology.org/2023.wassa-1.29 (visited on 09/13/2023).

[5] R. T. McCoy, P. Smolensky, T. Linzen, J. Gao, and A. Celikyilmaz. "How Much Do Language Models Copy From Their Training Data? Evaluating Linguistic Novelty in Text Generation Using RAVEN". In: *Transactions of the Association for Computational Linguistics* 11 (June 2023), pp. 652–670. ISSN: 2307-387X. DOI: 10.1162/tacl_a_00567. URL: https://doi.org/10.1162/tacl_a_00567 (visited on 09/20/2023).

[6] *Introducing ChatGPT — openai.com*. [Accessed 06-01-2024]. URL: https://openai.com/blog/chatgpt#OpenAI.

[7] *The virtual agent for bands, djs and musicians — connactz.com*. https://www.connactz.com/. [Accessed 06-01-2024].

[8] J. Chung, E. Kamar, and S. Amershi. "Increasing Diversity While Maintaining Accuracy: Text Data Generation with Large Language Models and Human Interventions". In: *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Toronto, Canada: Association for Computational Linguistics, July 2023, pp. 575–593. DOI: 10.18653/v1/2023.acl-long.34. URL: https://aclanthology.org/2023.acl-long.34 (visited on 09/13/2023).

[9] J. Liu, A. Liu, X. Lu, S. Welleck, P. West, R. L. Bras, Y. Choi, and H. Hajishirzi. *Generated Knowledge Prompting for Commonsense Reasoning*. arXiv:2110.08387 [cs]. Sept. 2022. DOI: 10.48550/arXiv.2110.08387. URL: http://arxiv.org/abs/2110.08387 (visited on 09/28/2023).

[10] D. Jurafsky and J. H. Martin. *Speech and language processing : an introduction to natural language processing, computational linguistics, and speech recognition*. 2023. URL: https://web.stanford.edu/~jurafsky/slp3/.

[11] *How do Transformers work? - Hugging Face NLP Course — huggingface.co*. https://huggingface.co/learn/nlp-course/chapter1/4. [Accessed 06-01-2024].

[12] N. Chomsky. *Syntactic Structures*. 1957. ISBN: 9783110172799. URL: https://books.google.de/books?id=a6a_b-CXYAkC.

[13] J. Weizenbaum. "ELIZA—a computer program for the study of natural language communication between man and machine". In: *Commun. ACM* 9.1 (Jan. 1966), pp. 36–45. ISSN: 0001-0782. DOI: 10.1145/365153.365168. URL: https://dl.acm.org/doi/10.1145/365153.365168 (visited on 01/04/2024).

[14] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. *Attention Is All You Need*. en. June 2017. URL: https://arxiv.org/abs/1706.03762v7 (visited on 01/05/2024).

[15] E. Reiter and R. Dale. *Building Natural Language Generation Systems*. Studies in Natural Language Processing. Cambridge University Press, 2000.

[16] A. Gatt and E. Krahmer. "Survey of the State of the Art in Natural Language Generation: Core tasks, applications and evaluation". en. In: *Journal of Artificial Intelligence Research* 61 (Jan. 2018), pp. 65–170. ISSN: 1076-9757. DOI: 10.1613/jair.5477. URL: https://jair.org/index.php/jair/article/view/11173 (visited on 01/04/2024).

[17] R. Eisenstadt and M. Elhadad. "Neural Micro-Planning for Data to Text Generation Produces more Cohesive Text". In: *Proceedings of the Workshop on Discourse Theories for Text Planning*. Dublin, Ireland: Association for Computational Linguistics, Dec. 2020, pp. 6–9. URL: https://aclanthology.org/2020.dt4tp-1.2 (visited on 09/13/2023).

[18] Y. Su, D. Vandyke, S. Wang, Y. Fang, and N. Collier. *Plan-then-Generate: Controlled Data-to-Text Generation via Planning*. arXiv:2108.13740 [cs]. Aug. 2021. DOI: 10.48550/arXiv.2108.13740. URL: http://arxiv.org/abs/2108.13740 (visited on 09/13/2023).

[19] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. 2019. arXiv: 1810.04805 [cs.CL].

[20] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer. *BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension*. 2019. arXiv: 1910.13461 [cs.CL].

[21] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu. *Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer*. arXiv:1910.10683 [cs, stat] version: 4. Sept. 2023. DOI: 10.48550/arXiv.1910.10683. URL: http://arxiv.org/abs/1910.10683 (visited on 01/04/2024).

[22] M. Alexander. *Data to Text generation with T5; Building a simple yet advanced NLG model — towardsdatascience.com*. https://towardsdatascience.com/data-to-text-generation-with-t5-building-a-simple-yet-advanced-nlg-model-b5cce5a6df45. [Accessed 07-01-2024].

[23] H. W. Chung, L. Hou, S. Longpre, B. Zoph, Y. Tay, W. Fedus, Y. Li, X. Wang, M. Dehghani, S. Brahma, A. Webson, S. S. Gu, Z. Dai, M. Suzgun, X. Chen, A. Chowdhery, A. Castro-Ros, M. Pellat, K. Robinson, D. Valter, S. Narang, G. Mishra, A. Yu, V. Zhao, Y. Huang, A. Dai, H. Yu, S. Petrov, E. H. Chi, J. Dean, J. Devlin, A. Roberts, D. Zhou, Q. V. Le, and J. Wei. *Scaling Instruction-Finetuned Language Models*. 2022. arXiv: 2210.11416 [cs.LG].

[24] *Google Colaboratory — colab.research.google.com*. https://colab.research.google.com. [Accessed 07-01-2024].

[25] *GPT-4 — openai.com*. https://openai.com/research/gpt-4. [Accessed 04-01-2024].

[26] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, D. Bikel, L. Blecher, C. C. Ferrer, M. Chen, G. Cucurull, D. Esiobu, J. Fernandes, J. Fu, W. Fu, B. Fuller, C. Gao, V. Goswami, N. Goyal, A. Hartshorn, S. Hosseini, R. Hou, H. Inan, M. Kardas, V. Kerkez, M. Khabsa, I. Kloumann, A. Korenev, P. S. Koura, M.-A. Lachaux, T. Lavril, J. Lee, D. Liskovich, Y. Lu, Y. Mao, X. Martinet, T. Mihaylov, P. Mishra, I. Molybog, Y. Nie, A. Poulton, J. Reizenstein, R. Rungta, K. Saladi, A. Schelten, R. Silva, E. M. Smith, R. Subramanian, X. E. Tan, B. Tang, R. Taylor, A. Williams, J. X. Kuan, P. Xu, Z. Yan, I. Zarov, Y. Zhang, A. Fan, M. Kambadur, S. Narang, A. Rodriguez, R. Stojnic, S. Edunov, and T. Scialom. *Llama 2: Open Foundation and Fine-Tuned Chat Models*. arXiv:2307.09288 [cs]. July 2023. DOI: 10.48550/arXiv.2307.09288. URL: http://arxiv.org/abs/2307.09288 (visited on 01/04/2024).

[27] C. van der Lee, A. Gatt, E. van Miltenburg, and E. Krahmer. "Human evaluation of automatically generated text: Current trends and best practice guidelines". In: *Computer Speech & Language* 67 (2021), p. 101151. ISSN: 0885-2308. DOI: https://doi.org/10.1016/j.csl.2020.101151. URL: https://www.sciencedirect.com/science/article/pii/S088523082030084X.

[28] L. Watson and D. Gkatzia. "Unveiling NLG Human-Evaluation Reproducibility: Lessons Learned and Key Insights from Participating in the ReproNLP Challenge". In: *Proceedings of the 3rd Workshop on Human Evaluation of NLP Systems*. Ed. by A. Belz, M. Popović, E. Reiter, C. Thomson, and J. Sedoc. Varna, Bulgaria: INCOMA Ltd., Shoumen, Bulgaria, Sept. 2023, pp. 69–74. URL: https://aclanthology.org/2023.humeval-1.6.

[29] D. M. Howcroft, A. Belz, M. Clinciu, D. Gkatzia, S. A. Hasan, S. Mahamood, S. Mille, E. Van Miltenburg, S. Santhanam, and V. Rieser. "Twenty years of confusion in human evaluation: NLG needs evaluation sheets and standardised definitions". In: *13th International Conference on Natural Language Generation 2020*. Association for Computational Linguistics. 2020, pp. 169–182.

[30] D. Deutsch, R. Dror, and D. Roth. *On the Limitations of Reference-Free Evaluations of Generated Text*. arXiv:2210.12563 [cs]. Oct. 2022. DOI: 10.48550/arXiv.2210.12563. URL: http://arxiv.org/abs/2210.12563 (visited on 10/27/2023).

[31] D. Çavuşoğlu. *Evaluation Metrics: Assessing the quality of NLG outputs — towardsdatascience.com*. https://towardsdatascience.com/evaluation-metrics-assessing-the-quality-of-nlg-outputs-39749a115ff3. [Accessed 07-01-2024].

[32] A. B. Sai, A. K. Mohankumar, and M. M. Khapra. *A Survey of Evaluation Metrics Used for NLG Systems*. 2020. arXiv: 2008.12009 [cs.CL].

[33] J. Novikova, O. Dušek, A. Cercas Curry, and V. Rieser. "Why We Need New Evaluation Metrics for NLG". In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Copenhagen, Denmark: Association for Computational Linguistics, Sept. 2017, pp. 2241–2252. DOI: 10.18653/v1/D17-1238. URL: https://aclanthology.org/D17-1238 (visited on 09/13/2023).

[34] E. Reiter. *Future of NLG evaluation: LLMs and high quality human eval?* https://ehudreiter.com/2023/05/22/future-of-nlg-evaluation/. [Accessed 07-01-2024].

[35] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi. "Bertscore: Evaluating text generation with bert". In: *arXiv preprint arXiv:1904.09675* (2019).

[36] P. Ke, H. Zhou, Y. Lin, P. Li, J. Zhou, X. Zhu, and M. Huang. "CTRLEval: An Unsupervised Reference-Free Metric for Evaluating Controlled Text Generation". In: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Dublin, Ireland: Association for Computational Linguistics, May 2022, pp. 2306–2319. DOI: 10.18653/v1/2022.acl-long.164. URL: https://aclanthology.org/2022.acl-long.164 (visited on 09/13/2023).

[37] T. Kocmi and C. Federmann. *Large Language Models Are State-of-the-Art Evaluators of Translation Quality*. 2023. arXiv: 2302.14520 [cs.CL].

[38] Y. Liu, D. Iter, Y. Xu, S. Wang, R. Xu, and C. Zhu. *G-Eval: NLG Evaluation using GPT-4 with Better Human Alignment*. arXiv:2303.16634 [cs]. May 2023. DOI: 10.48550/arXiv.2303.16634. URL: http://arxiv.org/abs/2303.16634 (visited on 10/11/2023).

[39] W. Yuan, G. Neubig, and P. Liu. *BARTScore: Evaluating Generated Text as Text Generation*. 2021. arXiv: 2106.11520 [cs.CL].

[40] M. Zhong, Y. Liu, D. Yin, Y. Mao, Y. Jiao, P. Liu, C. Zhu, H. Ji, and J. Han. *Towards a Unified Multi-Dimensional Evaluator for Text Generation*. arXiv:2210.07197 [cs]. Oct. 2022. DOI: 10.48550/arXiv.2210.07197. URL: http://arxiv.org/abs/2210.07197 (visited on 10/11/2023).

[41] A. R. Fabbri, W. Kryściński, B. McCann, C. Xiong, R. Socher, and D. Radev. *SummEval: Re-evaluating Summarization Evaluation*. arXiv:2007.12626 [cs]. Feb. 2021. DOI: 10.48550/arXiv.2007.12626. URL: http://arxiv.org/abs/2007.12626 (visited on 12/27/2023).

[42] S. Mehri and M. Eskenazi. "USR: An Unsupervised and Reference Free Evaluation Metric for Dialog Generation". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, July 2020, pp. 681–707. DOI: 10.18653/v1/2020.acl-main.64. URL: https://aclanthology.org/2020.acl-main.64 (visited on 09/13/2023).

[43] G. Tevet and J. Berant. "Evaluating the Evaluation of Diversity in Natural Language Generation". In: *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*. Ed. by P. Merlo, J. Tiedemann, and R. Tsarfaty. Online: Association for Computational Linguistics, Apr. 2021, pp. 326–346. DOI: 10.18653/v1/2021.eacl-main.25. URL: https://aclanthology.org/2021.eacl-main.25 (visited on 12/28/2023).

[44] S. Wiseman, S. Shieber, and A. Rush. "Learning Neural Templates for Text Generation". In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium: Association for Computational Linguistics, Oct. 2018, pp. 3174–3187. DOI: 10.18653/v1/D18-1356. URL: https://aclanthology.org/D18-1356 (visited on 09/13/2023).

[45] C. Gardent, A. Shimorina, S. Narayan, and L. Perez-Beltrachini. "The WebNLG Challenge: Generating Text from RDF Data". In: *Proceedings of the 10th International Conference on Natural Language Generation*. Ed. by J. M. Alonso, A. Bugarín, and E. Reiter. Santiago de Compostela, Spain: Association for Computational Linguistics, Sept. 2017, pp. 124–133. DOI: 10.18653/v1/W17-3518. URL: https://aclanthology.org/W17-3518.

[46] R. Puduppully, Y. Fu, and M. Lapata. "Data-to-text Generation with Variational Sequential Planning". In: *Transactions of the Association for Computational Linguistics* 10 (June 2022), pp. 697–715. ISSN: 2307-387X. DOI: 10.1162/tacl_a_00484. eprint: https://direct.mit.edu/tacl/article-pdf/doi/10.1162/tacl_a_00484/2029954/tacl_a_00484.pdf. URL: https://doi.org/10.1162/tacl%5C_a%5C_00484.

[47] Y. Su, Z. Meng, S. Baker, and N. Collier. "Few-Shot Table-to-Text Generation with Prototype Memory". In: *Findings of the Association for Computational Linguistics: EMNLP 2021*. Punta Cana, Dominican Republic: Association for Computational Linguistics, Nov. 2021, pp. 910–917. DOI: 10.18653/v1/2021.findings-emnlp.77. URL: https://aclanthology.org/2021.findings-emnlp.77 (visited on 09/13/2023).

[48] Z. Kasner and O. Dusek. "Neural Pipeline for Zero-Shot Data-to-Text Generation". In: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Ed. by S. Muresan, P. Nakov, and A. Villavicencio. Dublin, Ireland: Association for Computational Linguistics, May 2022, pp. 3914–3932. DOI: 10.18653/v1/2022.acl-long.271. URL: https://aclanthology.org/2022.acl-long.271.

[49] OpenAI, J. Achiam, S. Adler, et al. *GPT-4 Technical Report*. en. Mar. 2023. URL: https://arxiv.org/abs/2303.08774v4 (visited on 01/04/2024).

[50] A. Axelsson and G. Skantze. *Using Large Language Models for Zero-Shot Natural Language Generation from Knowledge Graphs*. 2023. arXiv: 2307.07312 [cs.CL].

[51] J. Xiang, Z. Liu, Y. Zhou, E. P. Xing, and Z. Hu. *ASDOT: Any-Shot Data-to-Text Generation with Pretrained Language Models*. 2022. arXiv: 2210.04325 [cs.CL].

[52] S. Saha, X. V. Yu, M. Bansal, R. Pasunuru, and A. Celikyilmaz. *MURMUR: Modular Multi-Step Reasoning for Semi-Structured Data-to-Text Generation*. 2022. arXiv: 2212.08607 [cs.CL].

[53] J. Wei, X. Wang, D. Schuurmans, M. Bosma, B. Ichter, F. Xia, E. Chi, Q. Le, and D. Zhou. *Chain-of-Thought Prompting Elicits Reasoning in Large Language Models*. arXiv:2201.11903 [cs]. Jan. 2023. DOI: 10.48550/arXiv.2201.11903. URL: http://arxiv.org/abs/2201.11903 (visited on 09/28/2023).

[54] T. Kojima, S. S. Gu, M. Reid, Y. Matsuo, and Y. Iwasawa. *Large Language Models are Zero-Shot Reasoners*. arXiv:2205.11916 [cs]. Jan. 2023. DOI: 10.48550/arXiv.2205.11916. URL: http://arxiv.org/abs/2205.11916 (visited on 09/28/2023).

[55] S. Yao, D. Yu, J. Zhao, I. Shafran, T. L. Griffiths, Y. Cao, and K. Narasimhan. *Tree of Thoughts: Deliberate Problem Solving with Large Language Models*. arXiv:2305.10601 [cs]. May 2023. DOI: 10.48550/arXiv.2305.10601. URL: http://arxiv.org/abs/2305.10601 (visited on 09/28/2023).

[56] H. Zhang, H. Song, S. Li, M. Zhou, and D. Song. "A survey of controllable text generation using transformer-based pre-trained language models". In: *ACM Computing Surveys* (2022). Publisher: ACM New York, NY.

[57] H. Elder, S. Gehrmann, A. O'Connor, and Q. Liu. "E2E NLG Challenge Submission: Towards Controllable Generation of Diverse Natural Language". In: *Proceedings of the 11th International Conference on Natural Language Generation*. Ed. by E. Krahmer, A. Gatt, and M. Goudbeek. Tilburg University, The Netherlands: Association for Computational Linguistics, Nov. 2018, pp. 457–462. DOI: 10.18653/v1/W18-6556. URL: https://aclanthology.org/W18-6556.

[58] V. Puranik, A. Majumder, and V. Chaoji. "PROTEGE: Prompt-based Diverse Question Generation from Web Articles". In: *Findings of the Association for Computational Linguistics: EMNLP 2023*. Ed. by H. Bouamor, J. Pino, and K. Bali. Singapore: Association for Computational Linguistics, Dec. 2023, pp. 5449–5463. DOI: 10.18653/v1/2023.findings-emnlp.362. URL: https://aclanthology.org/2023.findings-emnlp.362.

[59] T. Zhao, R. Zhao, and M. Eskenazi. "Learning Discourse-level Diversity for Neural Dialog Models using Conditional Variational Autoencoders". In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Ed. by R. Barzilay and M.-Y. Kan. Vancouver, Canada: Association for Computational Linguistics, July 2017, pp. 654–664. DOI: 10.18653/v1/P17-1061. URL: https://aclanthology.org/P17-1061.

[60] G. Zhou and G. Lampouras. *Informed Sampling for Diversity in Concept-to-Text NLG*. 2021. arXiv: 2004.14364 [cs.CL].

[61] J. Cho, M. Seo, and H. Hajishirzi. *Mixture Content Selection for Diverse Sequence Generation*. 2019. arXiv: 1909.01953 [cs.CL].

[62] *Alibaba Luban: AI-based Graphic Design Tool — alibabacloud.com*. https://www.alibabacloud.com/blog/alibaba-luban-ai-based-graphic-design-tool_594294. [Accessed 12-01-2024].

[63] T. Zhang, J. Zhang, C. Huo, and W. Ren. "Automatic Generation of Pattern-Controlled Product Description in E-Commerce". In: *The World Wide Web Conference*. WWW '19. San Francisco, CA, USA: Association for Computing Machinery, 2019, pp. 2355–2365. ISBN: 9781450366748. DOI: 10.1145/3308558.3313407. URL: https://doi.org/10.1145/3308558.3313407.

[64] Q. Chen, J. Lin, Y. Zhang, H. Yang, J. Zhou, and J. Tang. "Towards Knowledge-Based Personalized Product Description Generation in E-Commerce". In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. KDD '19. Anchorage, AK, USA: Association for Computing Machinery, 2019, pp. 3040–3050. ISBN: 9781450362016. DOI: 10.1145/3292500.3330725. URL: https://doi.org/10.1145/3292500.3330725.

[65] S. Murakami, S. Hoshino, and P. Zhang. *Natural Language Generation for Advertising: A Survey*. 2023. arXiv: 2306.12719 [cs.CL].

[66] K. Golobokov, J. Chai, V. Y. Dong, M. Gu, B. Chi, J. Cao, Y. Yan, and Y. Liu. *DeepGen: Diverse Search Ad Generation and Real-Time Customization*. 2022. arXiv: 2208.03438 [cs.CL].

[67] Z. Abrams and E. Vee. "Personalized Ad Delivery When Ads Fatigue: An Approximation Algorithm". In: Dec. 2007, pp. 535–540. ISBN: 978-3-540-77104-3. DOI: 10.1007/978-3-540-77105-0_57.

[68] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. "Bleu: a Method for Automatic Evaluation of Machine Translation". In: *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*. Ed. by P. Isabelle, E. Charniak, and D. Lin. Philadelphia, Pennsylvania, USA: Association for Computational Linguistics, July 2002, pp. 311–318. DOI: 10.3115/1073083.1073135. URL: https://aclanthology.org/P02-1040.

[69] S. Gehrmann, E. Clark, and T. Sellam. *Repairing the Cracked Foundation: A Survey of Obstacles in Evaluation Practices for Generated Text*. 2022. arXiv: 2202.06935 [cs.CL].

[70] K. Ethayarajh and D. Jurafsky. "The Authenticity Gap in Human Evaluation". In: *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, Dec. 2022, pp. 6056–6070. DOI: 10.18653/v1/2022.emnlp-main.406. URL: https://aclanthology.org/2022.emnlp-main.406 (visited on 09/13/2023).

[71] T. He, J. Zhang, T. Wang, S. Kumar, K. Cho, J. Glass, and Y. Tsvetkov. "On the blind spots of model-based evaluation metrics for text generation". In: *arXiv preprint arXiv:2212.10020* (2022).

[72] T. Sun, J. He, X. Qiu, and X. Huang. "BERTScore is unfair: On social bias in language model-based metrics for text generation". In: *arXiv preprint arXiv:2210.07626* (2022).

[73] R. Mihalcea, C. Corley, and C. Strapparava. "Corpus-based and knowledge-based measures of text semantic similarity". In: *Proceedings of the 21st national conference on Artificial intelligence - Volume 1*. AAAI'06. Boston, Massachusetts: AAAI Press, July 2006, pp. 775–780. ISBN: 978-1-57735-281-5. (Visited on 10/16/2023).

[74] H. Liu and P. Wang. "Assessing Sentence Similarity Using WordNet based Word Similarity". In: *JSW* 8.6 (June 2013), pp. 1451–1458. ISSN: 1796-217X. DOI: 10.4304/jsw.8.6.1451-1458. URL: http://ojs.academypublisher.com/index.php/jsw/article/view/9168 (visited on 10/16/2023).

[75] A. Kashyap, L. Han, R. Yus, J. Sleeman, T. Satyapanich, S. Gandhi, and T. Finin. "Robust semantic text similarity using LSA, machine learning, and linguistic resources". In: *Language Resources and Evaluation* 50.1 (Mar. 2016), pp. 125–161. ISSN: 1574-0218. DOI: 10.1007/s10579-015-9319-2. URL: https://doi.org/10.1007/s10579-015-9319-2.

[76] D. Friedman and A. B. Dieng. "The Vendi Score: A Diversity Evaluation Metric for Machine Learning". In: (Sept. 2022). URL: https://openreview.net/forum?id=dF0g-5k05h_ (visited on 09/21/2023).

[77] E. Montahaei, D. Alihosseini, and M. S. Baghshah. *Jointly Measuring Diversity and Quality in Text Generation Models*. arXiv:1904.03971 [cs, stat]. May 2019. DOI: 10.48550/arXiv.1904.03971. URL: http://arxiv.org/abs/1904.03971 (visited on 09/21/2023).

[78] *DeepL Translate: The world's most accurate translator — deepl.com*. https://www.deepl.com/translator. [Accessed 09-01-2024].

[79] OpenStreetMap contributors. *Germany retrieved from https://server.arcgisonline.com/ArcGIS/rest/services/Canvas/World_Light_Gray_Base/*. URL: https://www.openstreetmap.org.

[80] N. Shazeer and M. Stern. *Adafactor: Adaptive Learning Rates with Sublinear Memory Cost*. 2018. arXiv: 1804.04235 [cs.LG].

[81] *GitHub - ggerganov/llama.cpp: Port of Facebook's LLaMA model in C/C++ — github.com*. https://github.com/ggerganov/llama.cpp. [Accessed 09-01-2024].

[82] *Prompt Engineering Guide*. en. URL: https://www.promptingguide.ai/introduction/examples (visited on 09/28/2023).

[83] *LMSys Chatbot Arena Leaderboard - a Hugging Face Space by lmsys — huggingface.co*. https://huggingface.co/spaces/lmsys/chatbot-arena-leaderboard. [Accessed 13-01-2024].

[84] L. Chen, M. Zaharia, and J. Zou. *How is ChatGPT's behavior changing over time?* arXiv:2307.09009 [cs]. Oct. 2023. DOI: 10.48550/arXiv.2307.09009. URL: http://arxiv.org/abs/2307.09009 (visited on 01/03/2024).