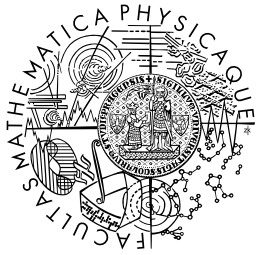

FACULTY
OF MATHEMATICS
AND PHYSICS
Charles University

**MASTER THESIS**

Ivan Kartáč

# Explainable LLM-based evaluation for NLG using error analysis

Institute of Formal and Applied Linguistics

Supervisor of the master thesis: Ing. Mateusz Lango, Ph.D.

Study programme: Language Technologies and Computational Linguistics

Prague 2025

I declare that I carried out this master thesis on my own, and only with the cited sources, literature and other professional sources. I understand that my work relates to the rights and obligations under the Act No. 121/2000 Sb., the Copyright Act, as amended, in particular the fact that the Charles University has the right to conclude a license agreement on the use of this work as a school work pursuant to Section 60 subsection 1 of the Copyright Act.

In ..... date .....  .....

Author's signature

I would like to thank my supervisor, Ing. Mateusz Lango, Ph.D., and my consultant, Mgr. et Mgr. Ondřej Dušek, Ph.D, for their invaluable advice, guidance and feedback. Their expertise and support have been fundamental throughout the process of working on this thesis.

My deepest thanks go to my wife Barbora for her encouragement, support and endless patience.

Title: Explainable LLM-based evaluation for NLG using error analysis

Author: Ivan Kartáč

Institute: Institute of Formal and Applied Linguistics

Supervisor: Ing. Mateusz Lango, Ph.D., Institute of Formal and Applied Linguistics

Consultant: Mgr. et Mgr. Ondřej Dušek, PhD., Institute of Formal and Applied Linguistics

Abstract: Traditional metrics for evaluating natural language generation (NLG) often struggle to capture linguistic complexity or align with human judgment. Recently, approaches based on large language models (LLMs) have been proposed to address these limitations. However, many existing approaches rely on proprietary LLMs or lack sufficient explainability. This thesis explores the potential of open-weight LLMs to develop a robust and explainable method for NLG evaluation. We develop a prompt-based evaluation method that applies an ensemble of LLMs to assess the quality of generated texts. This method is then applied to construct a synthetic training dataset that represents a wide range of tasks, evaluation aspects and systems. Using this dataset, we train a specialized evaluator model through distillation, employing Llama 3.1 8B as the backbone. Evaluation on a number of benchmarks demonstrates that our ensemble approach outperforms both the traditional NLG metrics as well as trained neural models and LLM-based methods. Additionally, the fine-tuned evaluator achieves competitive performance, with substantial improvements over the backbone model.

Keywords: natural language generation, evaluation, large language models, natural language processing

Název práce: Vysvětlitelná evaluace generování textu založená na velkých jazykových modelech a chybové analýze

Autor: Ivan Kartáč

Department: Ústav formální a aplikované lingvistiky

Vedoucí bakalářské práce: Ing. Mateusz Lango, Ph.D., Ústav formální a aplikované lingvistiky

Konzultant bakalářské práce: Mgr. et Mgr. Ondřej Dušek, PhD., Ústav formální a aplikované lingvistiky

Abstrakt: Tradiční metriky pro evaluaci generování přirozeného jazyka (NLG) často nedokážou adekvátně zachytit komplexitu jazyka a mnohdy se neshodují s lidským hodnocením. V poslední době byly navrženy přístupy založené na velkých jazykových modelech (LLM), které si kladou za cíl tyto nedostatky překonat. Nicméně, většina současných přístupů je založena na uzavřených (closed-source) modelech nebo postrádá dostatečnou interpretovatelnost. Tato práce se zaměřuje na využití otevřených LLM k vytvoření robustní a interpretovatelné metody pro evaluaci NLG, a prezentuje přístup založený na promptech, který využívá ensemble několika LLM. Tuto metodu následně využíváme k vytvoření syntetického trénovacího datasetu, který zahrnuje řadu úloh, evaluačních kritérií a typů systémů. Na tomto datasetu trénujeme specializovaný evaluační model založený na Llama 3.1 8B. Evaluace na různých benchmarcích ukazuje, že náš ensemble přístup překonává jak tradiční metriky NLG, tak i metody založené na neuronových sítích a LLM. Dále, náš trénovaný model dosahuje přesvědčivých výsledků a významně překonává svůj základní model.

Klíčová slova: generování přirozeného jazyka, evaluace, velké jazykové modely, zpracování přirozeného jazyka

# Contents


**1 Introduction** **8**

**2 Theoretical Background** **10**
2.1 Natural Language Generation . . . . . 10
2.2 Transformer Architecture . . . . . 10
2.3 Pre-trained Language Models . . . . . 12
2.3.1 Architectures . . . . . 12
2.3.2 Training . . . . . 13
2.3.3 Prompt Engineering . . . . . 14
2.3.4 Quantization . . . . . 14
2.3.5 Parameter-efficient Fine-tuning . . . . . 15
2.4 Explainability for LLMs . . . . . 16
2.5 Evaluation of NLG . . . . . 16
2.5.1 Human Evaluation . . . . . 16
2.5.2 Error Analysis . . . . . 19
2.5.3 Evaluation Aspects . . . . . 21
2.5.4 Automatic Metrics . . . . . 23
2.5.5 LLM-based Evaluation . . . . . 25
2.5.6 Meta-evaluation . . . . . 27

**3 Methods** **30**
3.1 Problem Formulation . . . . . 30
3.2 Ensemble Prompting . . . . . 30
3.2.1 The Approach . . . . . 30
3.2.2 Prompts . . . . . 31
3.3 Dataset . . . . . 32
3.3.1 Summarization . . . . . 34
3.3.2 Data-to-text . . . . . 37
3.3.3 Dialogue Response Generation . . . . . 40
3.3.4 Story Generation . . . . . 41
3.3.5 Question Answering . . . . . 43
3.4 Fine-tuning . . . . . 44
3.4.1 Prompts . . . . . 45
3.4.2 Models . . . . . 45

**4 Experiments** **46**
4.1 Meta-evaluation . . . . . 46
4.2 Baselines . . . . . 47
4.3 Prompt-based Evaluation . . . . . 48
4.4 Fine-tuning . . . . . 48

**5 Results and Analysis** **49**

**6 Conclusion** **54**
6.1 Discussion . . . . . 54
6.2 Future Work . . . . . 55

**Bibliography** **56**

**List of Figures** **71**

**List of Tables** **72**

**A Appendix** **74**
A.1 System Outputs . . . . . 74
A.2 Prompts for Output Generation . . . . . 74
A.3 Evaluation Prompts . . . . . 80

# 1 Introduction

Natural language generation (NLG) is a field of artificial intelligence that focuses on building systems that can produce human-like texts. This involves generating texts from inputs such as structured data, text, or images. NLG encompasses a variety of tasks, including summarization, data-to-text or dialogue response generation.

Human evaluation is widely considered the gold standard for assessing NLG output quality (Gatt and Krahmer, 2018). However, it is resource-intensive, time-consuming, and lacks scalability, which leads researchers and practitioners to rely primarily on automatic metrics. Traditional automatic metrics, such as BLEU (Papineni et al., 2002) and ROUGE (Lin, 2004), are widely used in NLG research practice due to their simplicity and ease of implementation. However, these metrics are known to have significant limitations, particularly in capturing the complexity and contextual details of generated texts (Gehrmann et al., 2023). Additionally, they have been shown to correlate poorly with human judgments (Lowe, Noseworthy, et al., 2017; Reiter and Belz, 2009).

These issues have led to a growing interest in the development of reliable evaluation methods that can effectively capture both linguistic quality and semantic adequacy, especially as the performance of current models on some NLG tasks has been reaching or even exceeding human performance. Building upon these advancements, more sophisticated evaluation methods have been proposed in recent years that better align with human judgment, demonstrating higher correlations with expert annotations. However, these methods are often specialized for a single task and require expensive data collection and training. More recently, large language models (LLMs) have been applied for automatic NLG evaluation, offering a flexible and better-performing alternative (Kocmi and Federmann, 2023b; Liu, Iter, et al., 2023). However, these methods rely on proprietary LLMs, which are typically prohibitively expensive and suffer from transparency and reproducibility issues. An alternative approach involves fine-tuning open-source LLMs, resulting in more transparent and efficient evaluation models (Jiang, Li, et al., 2023; Xu, Wang, et al., 2023). Nevertheless, these models often lack a sufficient degree of explainability, as they typically generate only numeric scores without providing the rationale behind their evaluations. In addition, these approaches typically use GPT-4, a proprietary model, to generate their training data.

This work explores the potential of open-weight large language models (LLMs) to develop a robust and transparent method for evaluating NLG systems, addressing the limitations of traditional metrics and complementing the strengths of human evaluation. To address the lack of explainability inherent in many automatic metrics, our method incorporates fine-grained error analysis alongside numeric scores. The evaluator model not only identifies error spans in the text but also provides detailed explanations and assigns severity levels to each error. This approach serves a dual purpose: it improves the interpretability of the evaluation process, and functions as a chain-of-thought (CoT) mechanism (Wei et al., 2022), which helps the model to ground its decisions in a structured reasoning path. By explicitly reasoning through identified errors, the model both provides explainable outputs and improves its own alignment with human reasoning. Since

reference texts are often not available in many evaluation scenarios, our approach is designed to be reference-free: instead of relying on reference texts, it makes use of knowledge and reasoning capabilities of modern language models.

In this thesis, we address the following research questions:

> **RQ1**: Can an efficient and explainable LLM-based NLG evaluator be trained through distillation from larger open-weight models, while maintaining competitive performance?
>
> **RQ2**: What would be the performance of such approach on unseen domains, tasks and evaluation aspects?
>
> **RQ3**: What level of performance in NLG evaluation can be achieved through prompt engineering with the quantized versions of current open-weight LLMs?

To explore RQ1, we train an evaluator model using Llama 3.1 8B (Dubey et al., 2024) as the backbone, employing LoRA (Hu, Shen, et al., 2021), a parameter-efficient fine-tuning method. Our approach leverages synthetic evaluation data that contain fine-grained error analyses in addition to numeric scores. We generate these data using an ensemble of open-weight LLMs across a range of tasks, source datasets and evaluated systems. The ensemble outputs are aggregated to create a diverse and representative training dataset that combines the strengths of individual LLMs. To ensure comprehensive coverage of various NLG evaluation scenarios, this dataset includes examples from five distinct task types: data-to-text, summarization, dialogue response generation, story generation, and question answering. Additionally, a diverse set of system outputs is either collected or newly generated. The evaluator model is trained to assess multiple aspects of NLG quality, including factual consistency, coherence and relevance. To validate our approach, we conduct meta-evaluations on a number of established benchmarks. We investigate RQ2 by including unseen domains, tasks and aspects in the meta-evaluation data. To address RQ3, the ensemble-based method used for generating our synthetic dataset is evaluated as a standalone zero-shot prompting evaluation approach.

The thesis is structured as follows. Chapter 2 provides theoretical background, explaining key concepts that form the foundations for this work. Chapter 3 presents the methodology of our work. In Chapter 4 describes the experimental setup, while the results and analysis are presented in Chapter 5. Finally, we conclude with the summary of our findings and provide ideas for future work in Chapter 6.

# 2 Theoretical Background

## 2.1 Natural Language Generation

Historically, natural language generation (NLG) has been narrowly defined as the process of generating text from non-linguistic data (e.g., Reiter and Dale, 1997). Even today, there is no universally agreed-upon definition of NLG or the types of input it involves (Gatt and Krahmer, 2018). However, the term is commonly used rather broadly to describe a number of tasks that involve the generation of natural language text, where inputs can include structured data, text, images or even audio (Celikyilmaz et al., 2020). In this broader sense, NLG includes tasks such as summarization, data-to-text, machine translation, story generation, image captioning or instruction following.

NLG was traditionally carried out using rule-based and template-based approaches that relied on predefined rules and manually crafted templates. These systems provided greater interpretability and control over the generated text, but lacked diversity and were difficult to adapt to new tasks and domains. Recent advancements in NLG research have shifted focus to transformer-based models, which will be discussed in Sections 2.2 and 2.3.

## 2.2 Transformer Architecture

The Transformer (Vaswani et al., 2017) is a neural network architecture that has revolutionized the field of natural language processing. Neural networks are computational models typically consisting of multiple layers of interconnected nodes, each connection having an associated weight. These weights are iteratively adjusted during training using *backpropagation* to compute gradients, while optimization algorithms such as SGD (e.g., Ruder, 2016) or Adam (Kingma and Ba, 2014) minimize the error between the model's predictions and the target values. Nonlinear activation functions within each node enable neural networks to model complex, nonlinear relationships in data. For a comprehensive introduction to neural networks, the reader is referred to Goodfellow et al. (2016).

The original architecture of Transformer is based on an encoder-decoder structure. The encoder processes the input sequence into a contextualized representation, while the decoder generates the output sequence using the encoded input and previously generated tokens. Both the encoder and decoder are composed of almost identical stacked layers. Each layer contains two main components: a multi-head self-attention mechanism and a feedforward neural network. The overall architecture is illustrated in Figure 2.1. In the following, we provide a more detailed description of the main components of the architecture.

**Self-attention** The self-attention mechanism computes a weighted representation of the input sequence, taking into account the relationship of each token with all other tokens. For an input sequence $X$, self-attention computes three linear transformations of $X$, referred to as query ($Q$), key ($K$) and value ($V$) matrices:

$$Q = XW_Q, K = XW_K, V = XW_V \tag{2.1}$$

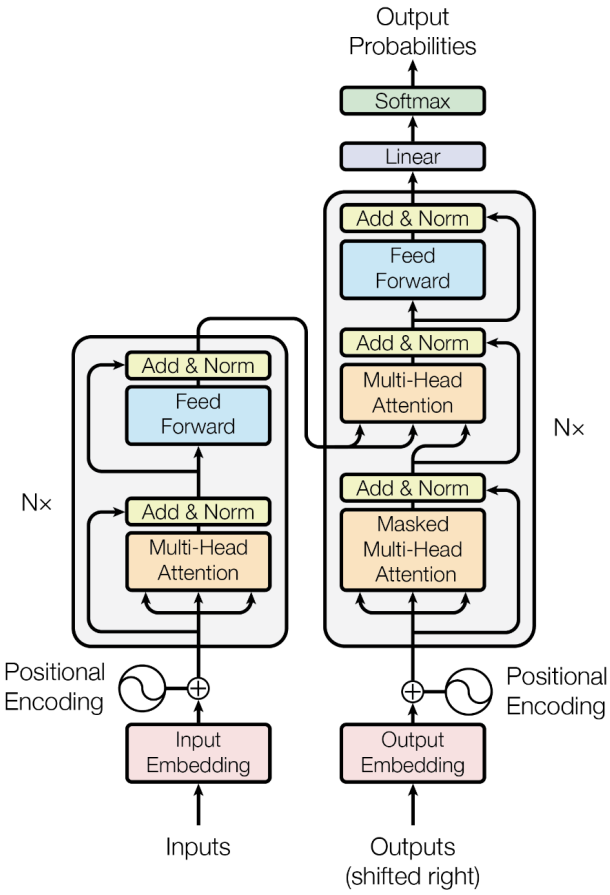


**Figure 2.1** Transformer architecture (source: Vaswani et al. (2017)

Here, $W_Q$, $W_K$ and $W_V$ are learned weight matrices. Then attention scores are computed as follows:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{2.2}$$

where $d_k$ is the dimensionality of $K$ and $\sqrt{d_k}$ serves as a scaling factor to ensure numerical stability during training. The softmax function normalizes the result of the dot-product, ensuring that the attention scores for a specific input token are non-negative and sum up to one. Finally, the result is multiplied by the $V$ matrix.

In the decoder, a mask is applied during self-attention to prevent the model from attending to future tokens in a sequence. The decoder layers additionally include an encoder-decoder attention mechanism, which allows the decoder to attend to the encoder's output. This mechanism involves an extra sublayer between the self-attention and feedforward network blocks. The encoder-decoder attention is similar to self-attention, but uses $K$ and $V$ matrices from the encoder.

The architecture uses multiple self-attention heads, each with its own $Q$, $K$ and $V$ matrices. The outputs of the individual heads are concatenated and multiplied by a learned weight matrix $W_O$.

**Feedforward Network** Following the attention block, each token representation is processed independently through a fully connected feedforward network:

$$\text{FFN}(x) = \text{ReLU}(0, xW_1 + b_1)W_2 + b_2 \tag{2.3}$$

Here, FFN represents a feedforward network applied to input $x$, and $\text{ReLU}(z) = \max(0, z)$ is the activation function. $W_1$ and $W_2$ are learned weight matrices, while $b_1$ and $b_2$ are bias vectors.

**Residual Connections and Layer Normalization** Each component of a layer, such as the self-attention or the feedforward network, has a residual connection (He et al., 2016) around it, followed by layer normalization (Ba et al., 2016):

$$Z = \text{LayerNorm}(x + L(x)) \tag{2.4}$$

where $L$ represents the sublayer, $x$ is the input to $L$ and LayerNorm is layer normalization.

Since the Transformer processes the input sequences in parallel, it does not have any sequential information. For this reason, *positional embeddings* are added to the input embeddings to encode information about the order of tokens. In the original Transformer architecture, positional embeddings are computed using sine and cosine functions of different frequencies.

LLMs, introduced in Section 2.3, differ with respect to which components of the architecture they use, the positions of layer normalization (Xiong et al., 2020), activation functions (Hendrycks and Gimpel, 2016; Shazeer, 2020), types of positional embeddings (Shaw et al., 2018; Su et al., 2024), and other minor variations. Section 2.3.1 provides an overview of the different Transformer variants used by pre-trained language models.

## 2.3 Pre-trained Language Models

### 2.3.1 Architectures

Generally, pre-trained language models are based on the Transformer architecture.[1] While they share this common foundation, they might differ in how they adapt the original Transformer architecture. This section provides an overview of dominant pre-trained language model architectures and presents examples of models based on each architecture. Large language models (LLMs) refer to a large, typically instruction-tuned variant of these models (discussed in Section 2.3.2).

Encoder-only models focus on encoding input text into contextualized representation. These representations are suitable for natural language understanding tasks, such as text classification, named entity recognition and sentiment analysis. Encoder-only models are trained using bidirectional representations, which consider both left and right context of the input sequence. BERT (Devlin, 2018) is an example of an encoder-based language model. BERT is pre-trained on two unsupervised tasks, masked language modeling and next sentence prediction. In

[1]While alternative architectures, such as RWKV (Peng et al., 2023) and Mamba (Gu and Dao, 2023), are being explored, they are out of scope for this thesis.

masked language modeling, a random subset of tokens in the input is masked and the model learns to predict these tokens. This encourages the model to learn bidirectional contextual representations. Next sentence prediction task involves predicting whether a given sentence B follows sentence A in a text. This approach helps the model learn the relationships between sentences. BERT is adapted to downstream tasks by adding task-specific layers on top of the pre-trained encoder. For example, a classification head can be added to predict a class label in text classification tasks.

Decoder-only architectures are designed for language generation tasks. They generate text by predicting the next token in a sequence based on all preceding tokens, which makes them suitable for open-ended generative tasks. GPT (Radford, 2018) is a well-known example of a decoder-only model. Unlike encoder-based models, GPT focuses on autoregressive modeling, where the goal is to maximize the likelihood of the next token given the context of previous tokens. Later iterations, such as GPT-2 (Radford et al., 2019) and GPT-3 (Brown et al., 2020), introduced larger architectures and training data, with GPT-3 reaching 175 billion parameters.

Encoder-decoder architectures aim to combine the strengths of both encoding and decoding. The encoder processes the input sequence into a contextual representation, which the decoder uses to generate an output sequence. T5 (Raffel et al., 2020) is an example of an encoder-decoder architecture that frames all NLP tasks as text-to-text problems. During pre-training, T5 uses an objective where random tokens of input text are masked and replaced with a unique sentinel token. The model learns to reconstruct the original sequence, learning both understanding and generative capabilities.

### 2.3.2 Training

Although pre-trained LLMs can be used as text-completion models, they are usually trained through a multi-stage process that includes **fine-tuning**, a process of adapting the model to a specific task. As discussed in Section 2.3.1, **pre-training** involves next-word prediction or masked language modeling and uses vast amount of text data, from which the model learns the foundational linguistic knowledge. This is typically followed by **instruction-tuning**, which teaches the model to map task-specific instructions to desired outputs. This stage helps in improving the practical usability of the model. To further align the model with human expectations and values, **global training** methods, such as RLHF and DPO are applied. RLHF (Reinforcement Learning from Human Feedback) (Ouyang et al., 2022) involves training a reward model on human preferences collected for the model's outputs. This reward model is then use to optimize LLM outputs to produce outputs aligned with human expectations. DPO (Direct Preference Optimization) (Rafailov et al., 2024) is an alternative method that directly optimizes LLM outputs based on human preferences, without applying reinforcement learning.

The data used to fine-tune the LLMs are typically human-crafted. On the other hand, **distillation** refers to fine-tuning on synthetic data, where a smaller and less complex model learns to mimic the capabilities of a more powerful model. For example, Alpaca (Taori et al., 2023) is a family of models fine-tuned by

distillation from GPT-3 (Brown et al., 2020).

### 2.3.3 Prompt Engineering

Prompt engineering is a technique used to effectively leverage the capabilities of generative LLMs through prompts. Prompts are textual inputs that guide the model to provide desired outputs. Numerous prompt engineering techniques have been proposed since the advent of LLMs. This section briefly describes some of these methods.

**Few-shot** prompting involves providing the model with examples to guide its behavior. By including examples, the model is expected to adhere to the task requirements and output format outlined in the prompt. In contrast, zero-shot prompting relies on the model's pre-training knowledge and its instruction-following capabilities without providing any examples.

**Chain-of-thought** (CoT) (Wei et al., 2022) refers to an approach where the model is encouraged to generate step-by-step reasoning before arriving at the final solution. This typically involves decomposing problems into smaller subproblems. As outlined in Wei et al. (2022), the model is presented with examples of reasoning paths that it is expected to emulate. Zero-shot CoT (Kojima et al., 2022) builds on this idea by prompting the model to generate the reasoning steps independently, using an instruction such as "think step-by-step".

More advanced methods include **self-consistency** prompting (Wang, Wei, et al., 2022) or **tree-of-thought** prompting (Long, 2023; Yao, Yu, et al., 2024). In self-consistency prompting, multiple reasoning paths are sampled to select the answer that is most consistent. Tree-of-thought prompting combines CoT with search algorithms to solve a problem by exploring trees of intermediate thoughts through look-ahead and backtracking.

**ReAct** (Yao, Zhao, et al., 2023) has been proposed for complex tasks that require both reasoning and interaction from an agent interacting with the external environment, such as tools or APIs. This method interleaves reasoning and action steps, which are dynamically updated based on feedback from external sources. **Reflexion** (Shinn et al., 2024) is a related approach that involves generating reasoning–action trajectories, obtaining feedback and generating self-reflection. The agent then refines its trajectory based on this information.

### 2.3.4 Quantization

Quantization is a compression technique that reduces the number of bits used to represent model weights (typically 2 to 8 bits), aiming to minimize accuracy loss (e.g., Dettmers, Lewis, et al., 2022; Dettmers and Zettlemoyer, 2023; Jacob et al., 2018). By decreasing model size, quantization helps to reduce storage requirements, memory use and inference latency. Various quantization techniques have been proposed, which can be categorized into two types. Quantization-aware training (QAT) refers to methods where quantization is applied during training, whereas post-training quantization (PTQ) applies it to pre-trained models. Common data types used in quantization include floating-point or integers, although a range of custom data types also exists (cf. Zhu, Li, et al., 2024). The level of quantization required to maintain the performance of a full-precision model depends on several

factors, including the task, model size and the specific quantization technique (Li, Ning, et al., 2024).

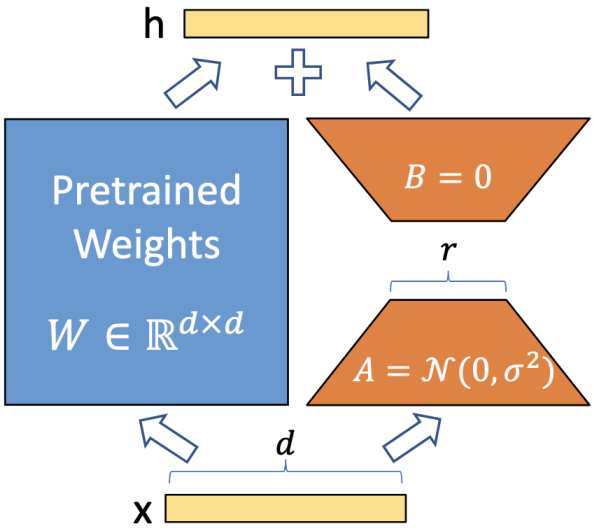


**Figure 2.2** Weight update matrix decomposition used in LoRA (source: Hu and Shen, et al. (2021)

### 2.3.5 Parameter-efficient Fine-tuning

Fine-tuning LLMs can be computationally expensive due to their large number of parameters and the constraints of GPU memory. During backpropagation, the weight update matrix contains the same number of parameters as the original weight matrix. Parameter-efficient fine-tuning is a family of methods that address this issue by adapting LLMs without the need to update all model parameters. Notable approaches include **adapters**, which introduce additional layers into a network while keeping the original layers frozen (Houlsby et al., 2019; Rebuffi et al., 2017), and **prefix-tuning**, which optimizes continuous prompts (Lester et al., 2021; Li and Liang, 2021), trainable embeddings that are prepended to the input sequence.

One such method, **Low-rank adaptation** (LoRA) (Hu, Shen, et al., 2021) decomposes the weight update matrix $\Delta W$ into a lower-rank representation. As illustrated in Figure 2.2, instead of the original large update matrix of size $d \times k$, LoRA uses two significantly smaller matrices, $A$ and $B$ with dimensions $d \times r$ and $r \times k$, respectively. During LoRA fine-tuning, the pre-trained weights remain frozen, and only $A$ and $B$ are updated. This approach can reduce the number of trainable parameters by hundreds to thousands of times, while often providing performance comparable to full fine-tuning (Hu, Shen, et al., 2021; Shuttleworth et al., 2024). The amount of saved memory depends on the rank of the LoRA matrices, with commonly used values ranging from 8 to 256. To further reduce memory usage during training, **QLoRA** (Dettmers, Pagnoni, et al., 2024) applies 4-bit quantization to the model weights.

## 2.4 Explainability for LLMs

In machine learning, explainability refers to the extent to which a model's behavior can be explained in a way that is understandable to humans (Doshi-Velez and Kim, 2017). Although some traditional machine learning models, such as decision trees or linear regression, are inherently explainable, the complexity of deep neural networks in terms of the number of parameters and training data inspired development of novel explanation techniques. LLMs with their enormous scale make explainability even more challenging. However, their emergent abilities, such as in-context learning and CoT prompting, provide new opportunities for explaining their decisions. Generally, explanations can be categorized as either local, focusing on explaining how a prediction for a specific input is made, or global, where the goal is to understand the model's overall behavior and internal mechanisms. In this thesis, we focus on approaches most relevant to NLG evaluation. For a comprehensive overview of explainability for LLMs, refer to (Zhao, Chen, et al., 2024).

Explanation through **adversarial examples** applies various perturbations to the input, which aim to examine the robustness of a model-based metric (Leiter et al., 2022). For example, the input might be modified in a way that changes its form but preserves the meaning to test if the metric remains consistent in its assessment. **Uncertainty estimation** aims to increase explainability by quantifying the uncertainty of model predictions, and includes techniques such as confidence elicitation through prompting (Kadavath et al., 2022), temperature sampling (Xiong et al., 2020) or estimation based on token-level uncertainty (Duan et al., 2023).

**Natural language explanation** involves providing a text that explains the model's decisions. Earlier research focused on training separate modules, where one module generates or extracts natural language explanations, and the other conditions its predictions on those explanations (Lei et al., 2016; Rajani et al., 2019). More recently, LLMs have introduced the capability to provide explanations for their own decisions, either through reasoning, as in the case of CoT prompting (discussed in Section 2.3.3), or through post-hoc rationalizations. In NLG evaluation, this can be used to enhance the interpretability of metrics by providing textual assessments alongside numeric scores. LLM-based evaluation methods that include natural language explanation will be discussed in more detail in Section 2.5.5.

## 2.5 Evaluation of NLG

### 2.5.1 Human Evaluation

Human evaluation scores are widely regarded as the most important and reliable form of evaluation, often serving as the gold standard (Gatt and Krahmer, 2018). However, they are less frequently used compared to automatic evaluation metrics. Van der Lee et al. (2021) surveyed ACL and INLG papers from 2016 to 2019, reporting that by 2019, 95% of these papers used automatic metrics (an increase from 71% in 2016), with 48% relying exclusively on them. Similarly, Schmidtová et al. (2024) found that only around 57% of over 100 analyzed NLG

papers included human evaluation, whereas automatic metrics were utilized in 94% of cases. The lower prevalence of human evaluation is likely due to its time-consuming and expensive nature, especially when domain experts are required or when high-quality annotations are the goal. For example, Thomson and Reiter (2020) describe a human evaluation methodology designed to obtain high-quality annotations for assessing the accuracy of generated texts. While this approach led to high inter-annotator agreement and the identification of fine-grained errors, it came at a significant cost: annotating a single 300-word text required 20–30 minutes and cost approximately $30. Even with simpler annotation protocols, the costs remain relatively high, leading many researchers to prefer automatic evaluation methods.

Crowdsourcing platforms such as Amazon Mechanical Turk[2], Prolific[3], and Appen[4] are the most common tools for obtaining human evaluations. These platforms have an advantage of lower costs, reduced time requirements, and greater scalability compared to directly hiring human experts. Moreover, they provide more diversity, as the crowdworkers often represent broader populations with varied cultural backgrounds (Celikyilmaz et al., 2020). While crowdsourcing might yield annotations of reasonable quality for many tasks—assuming sufficient guidelines and effective quality control mechanisms—certain scenarios require the use of expert annotators. For example, when a system needs to be evaluated with a very specific target audience in mind, crowdsourcing may be impractical (Celikyilmaz et al., 2020). Additionally, tasks requiring specialized domain expertise, such as in medical NLP, or a linguistic background may exceed the skills of general crowdworkers. Thomson and Reiter (2020) noted that their linguistically focused error categories were challenging to understand for a general crowdsourcing audience. Similarly, Freitag and Foster, et al. (2021) found very low agreement between scores assigned by linguists and those assigned by crowdworkers. However, employing expert annotators significantly increases costs and reduces flexibility. Unlike crowdsourcing, expert evaluations cannot be easily scaled or distributed among multiple individuals.

Human evaluation suffers from additional issues, including large variance both between and within annotators (Gehrmann et al., 2023). For instance, Karpinska et al. (2021) observed inconsistent results when conducting the same annotation task on Mechanical Turk across three different days. These can be attributed to several factors, such as the inherent complexity and subjectivity of NLG evaluation, and insufficient, unclear or even missing annotation guidelines (Gehrmann et al., 2023; Howcroft et al., 2020). There is also often a lack of transparency in how methodological details are reported. For example, Van der Lee et al., 2021 found that only 57% of surveyed papers reported the number of annotators involved and only a few of them (3%) include demographic information about the participants. These issues contribute to the limited reproducibility of NLG evaluations.

[2] `https://www.mturk.com/`

[3] `https://www.prolific.com/`

[4] `https://www.appen.com/`

### Extrinsic Evaluation

Extrinsic evaluation measures the effectiveness of a system based on its impact on end users (Celikyilmaz et al., 2020). In machine translation, for instance, it might be useful to measure the time saved compared to manual human translation. Similarly, in dialogue systems, examples of extrinsic evaluation include ratings of user satisfaction or task success rate. Although extrinsic evaluation is traditionally considered very valuable, as it measures real-world effectiveness of a system, it is often expensive and time consuming (Reiter and Belz, 2009). Consequently, it is not employed frequently, as reported by Van der Lee et al. (2021), who found that only 3% of surveyed NLG papers include results from extrinsic evaluations.

### Intrinsic Evaluation

Intrinsic evaluation, a more widely used method than extrinsic evaluation, focuses on assessing the properties of systems and generated texts, typically by evaluating output quality (Belz and Reiter, 2006; Celikyilmaz et al., 2020). This evaluation can target specific aspects such as coherence, faithfulness, or relevance, or it may assess overall quality. The quality of outputs is usually judged by comparing them against reference texts written by humans. In intrinsic evaluation, annotators are usually presented either with one output at a time and asked to rate it using a predefined scale, or they receive multiple system outputs which they are asked to compare with each other and rank. In the following, we present some common approaches to intrinsic evaluation.

Scoring methods are the simplest and most widely adopted approach to intrinsic evaluation. They are cost-effective and require less time investment compared to more complex evaluation protocols. Scoring typically involves assigning categorical or numerical values to outputs based on predefined scales. The most popular scoring methods are Likert scales and rating scales (Van der Lee et al., 2021). These ordinal scales generally include three to five points, although research indicates that seven-point scales maximize reliability, validity, and discriminative power (Cicchetti et al., 1985; Miller, 1956; Preston and Colman, 2000). While not as common across NLG tasks, Direct Assessment (DA) (Graham et al., 2013) is an approach often used in machine translation (MT) evaluation. This method employs a continuous scale, typically implemented in a user interface as a slider with two endpoints, allowing annotators to rate outputs with greater granularity. Binary labels, such as yes/no questions, are the simplest type of scoring approaches as well as a typical representative of categorical scoring methods, which are one of the most commonly used approaches (Van der Lee et al., 2021).

An alternative approach to direct scoring is ranking, where multiple outputs are directly compared against one another instead of being evaluated in isolation. Some research suggests that ranking can be more reliable than Likert or rating scales (Martinez et al., 2014; Yannakakis and Hallam, 2011). This method provides relative quality judgments, helping to better identify differences between systems. However, rankings may lack the diagnostic power needed for error-specific analyses, particularly in complex outputs such as long-form summaries or structured data-to-text generation. Additionally, ranking methods face problems with scaling as the number of systems to compare increases. Additionally, instrinsic evaluation can be carried out through error analysis, discussed in Section 2.5.2.

**Inter-annotator Agreement**

Evaluating NLG outputs is subjective to a certain extent, making it common to have multiple annotators score or annotate the same output to decrease bias. Van der Lee et al. (2021) report that a median of 3 annotators is typically used, based on a sample of surveyed papers from ACL and INLG conferences. As discussed in 2.5.1, high inconsistencies in ratings of different annotators for the same outputs can highlight issues with the annotation guidelines, evaluation design or insufficient annotator qualifications. These inconsistencies may also reflect the inherent subjectivity in the evaluated task or criteria, or a lack of reliable differences in the outputs. To assess (dis)agreement between annotators, several commonly used measures of inter-annotator agreement (IAA) are employed.

**Cohen's** $\kappa$ (Cohen, 1960) is a statistical measure of agreement between two annotators for categorical items. It accounts for agreements that may happen by chance, which makes it more robust than a simple percentage agreement. The formula for Cohen's $\kappa$ is as follows:

$$\kappa = \frac{P_o - P_c}{1 - P_c} \tag{2.5}$$

where $P_o$ is the observed percentage of agreement between the annotators, and $P_c$ is the expected proportion of agreement due to chance:

$$P_c = \sum_{s \in S} P(s \mid a_1) \cdot P(s \mid a_2) \tag{2.6}$$

Here, $P(s \mid a_1)$ refers to the probability that annotator $a_1$ assigns the score s, which is estimated from its observed frequency.

Since Cohen's $\kappa$ is limited to pairwise agreements, **Fleiss'** $\kappa$ (Fleiss, 1971) can be applied when agreement between multiple annotators needs to be measured. Additionally, **Krippendorf's** $\alpha$ is a disagreement measure that allows to accounts for different levels of disagreement between annotators.

Amidei et al. (2019) analyzed 135 NLG papers and found that only 18% of them report some measure of IAA. Among those that did, the agreement levels are generally low. The authors recommend using correlation coefficients together with IAA to address this issue. Furthermore, Van der Lee et al. (2021) highlight that most NLG papers also fail to include statistical significance testing along with their results.

### 2.5.2 Error Analysis

A specialized form of evaluation is error analysis, where annotators identify and categorize errors in system outputs. The efforts to standardize error analysis have been particularly prominent in the machine translation community. Multidimensional Quality Metrics (MQM) (Freitag, Foster, et al., 2021; Lommel et al., 2014) serves as a comprehensive framework for categorizing errors across multiple dimensions, and has been adopted as a standard for human evaluation in recent WMT metrics tasks (Freitag, Foster, et al., 2021). In MQM, error spans are

identified, labeled with severity level and categorized using a hierarchical error taxonomy. A scoring scheme assigns negative weights to errors based on their category and severity, while the final score for a segment is obtained by summing these weights. While MQM provides fine-grained details with respect to output issues and shows greater capability in differentiating between systems compared to DA and SQM (Kocmi, Zouhar, et al., 2024), it suffers from low inter-annotator agreement (Knowles and Lo, 2024) and requires expert annotators trained in the framework. Recently, Error Span Annotation (ESA) (Kocmi, Zouhar, et al., 2024) has been proposed as a faster and less expensive alternative to MQM. ESA focuses on tagging specific problematic spans within a segment and assigning one of two severity levels to each span, after which an overall score is assigned to the segment. By annotating errors in the text first, ESA aims to prime the annotators to assign more accurate overall scores. Note that this is similar to chain-of-thought prompting in LLMs, introduced in Section 2.3.3.

Attempts to define error analysis protocols and taxonomies have been made in other NLG tasks as well. For data-to-text, Thomson and Reiter (2020) propose an annotation protocol for fine-grained span-level annotation of accuracy errors in generated text. Their error taxonomy includes categories such as incorrect number, incorrect named entity, as well as contextual errors where an incorrect inference is present due to context. Inspired by this taxonomy, Kasner and Dušek (2024) use a more high-level approach with four error categories to annotate data-to-text outputs from various domains: Incorrect, Not checkable, Misleading and Other. Dou et al. (2021) introduce a crowd-sourced error annotation framework with ten error categories grouped into language errors (e.g. incoherence or grammar errors), factual errors (e.g. commonsense or encyclopedic errors), and render errors (where external verification or expertise is to understand the text). Goyal et al. (2022) offer a protocol for coherence error detection in narrative summarization, categorizing coherence errors into missing reference to an event or object, new character without introduction, abrupt scene transition and inconsistency. Sachdeva et al. (2024) introduce a dataset containing span-level annotations for long-form question answering with the following five evaluation aspects: *factuality*, *relevance*, *completeness*, *references* and *question misconception* (see 2.5.3 for a discussion of evaluation aspects). The last two aspects address errors related to unhelpful examples, analogies or references, and errors related to false assumptions in questions, respectively.

Despite providing better insights into the aspects of outputs that need improvement compared to simple scoring or ranking, error analysis is underutilized and errors in system outputs are still severely underreported in practice. A survey of INLG papers from 2010, 2015 and 2020 (Van Miltenburg, Clinciu, Dušek, Gkatzia, Inglis, Leppänen, Mahamood, Manning, et al., 2021) revealed that only around 11% of surveyed papers include some error analysis. Similarly, Gehrmann et al. (2023) reported that error analysis is reported in only 23% of analyzed papers from ACL, INLG and EMNLP. In a position paper, Van Miltenburg and Clinciu and Dušek and Gkatzia and Inglis and Leppänen and Mahamood and Schoch, et al. (2023) analyze factors which have an impact on researcher's decisions to not report error analyses in their work. By surveying 49 NLG researchers and practitioners, they identify resource limitations—including time, funding, tooling, error taxonomies and standardized procedures—as major constraints. Additionally,

current research culture was mentioned as an important factor, such as lack of appreciation for error analysis from reviewers, as well as the fact that reporting error analysis is not a requirement. Recently, some LLM-based evaluation approaches that use error span annotation have been proposed. These will be discussed in Section 2.5.4.

### 2.5.3 Evaluation Aspects

Evaluation aspects serve as criteria for assessing an output quality. For instance, a text can be evaluated in terms of fluency or informativeness. The simplest approach to evaluating output quality is to use a single, general *overall quality* criterion. This is sometimes used as the only evaluation criterion (Guan, Zhang, et al., 2021; Kreutzer et al., 2018), often based on observations that different aspects of quality tend to be highly correlated and human judgments often fail to distinguish between them (Manishina et al., 2016; Wen, Gasic, Kim, et al., 2015). However, evaluating text quality on a single dimension can be overly abstract (Hastie and Belz, 2014; Van der Lee et al., 2021), as it obscures which specific aspects of the text may be problematic. Furthermore, the reliability and discriminative power of overall quality scores heavily depends on experimental design. Novikova et al. (2018) and Dušek et al. (2020) show that evaluation aspects can be de-correlated if the judgments for each aspect are collected separately. Dušek et al. (2020) also exclude inputs when assessing naturalness–where only output form is evaluated–to decrease correlations between ratings of different aspects. Consequently, it is common for human evaluations to assess generated texts on multiple aspects of quality.

Evaluation aspects can be broadly categorized into those assessing the form of a text and those focusing on its semantics. Examples of the first category include fluency or grammaticality. While recent state-of-the-art LLMs excel at generating natural and fluent text, formal aspects such as fluency or grammaticality remain critical in multilingual settings, where current LLMs often struggle to produce fluent text in certain languages (Zhang, Li, et al., 2023). Semantic aspects, on the other hand, evaluate the meaning and content of the text and include factuality, relevance or informativeness. However, some aspects cannot be clearly categorized as formal or semantic. Coherence typically involves both the form and the content of a text. Even for primarily semantic aspects, the form will always have a certain effect on the quality with respect to the given aspect. For example, a typo might affect informativeness of an output if it introduces ambiguity otherwise not present in the output.

Even when evaluation designs consider multiple aspects, these aspects are often hierarchical, with their definitions commonly referencing other aspects (Howcroft et al., 2020). For example, fluency is frequently defined in terms of naturalness and grammaticality (e.g., Chen, Chen, et al., 2020; Nan, Hsieh, et al., 2022; Parikh et al., 2020). Other aspects are inherently multidimensional. Faithfulness in summarization, for instance, is often defined as the extent to which the information in the summary is supported by the source text (Maynez et al., 2020). A summary is considered unfaithful if it includes extraneous information not present in the source text, a phenomenon known as hallucination. Maynez et al. (2020) classify hallucinations into intrinsic, which misrepresent information

from the input, and extrinsic, which ignore the input entirely. Additionally, they distinguish between non-factual and factual hallucinations, noting that factual hallucinations may sometimes be beneficial. Cao et al. (2021) propose a method to detect hallucinations by separating factual from non-factual entities.

Belz and Mille, et al. (2020) propose a taxonomy that categorizes evaluation aspects based on three properties. The first, type of quality, differentiates between assessing correctness, goodness or features of outputs. For correctness aspects, necessary conditions can be defined under which an output is maximally correct. For instance, in a table-to-text task, a text is maximally accurate if it contains no inaccuracies with respect to the input table. In contrast, for goodness aspects it is only possible to decide whether an output is better or worse relative to some other output. For example, it does not make sense to assess whether a text is maximally fluent or coherent. The second property is the aspect of system output, distinguishing between evaluations of form, content, or both—examples being grammaticality, factuality, and coherence, respectively. Finally, different frames of reference can be used to evaluate output quality. The outputs can be evaluated in their own right, relative to input, or relative to an external context. External context is often represented by a knowledge base that is not explicitly present in the input. Deciding whether to evaluate outputs against such context or real-world knowledge is a critical design consideration that varies by task. For example, in summarization tasks, it is typically more important for outputs to align factually with the source text than with real-world knowledge, even if the source text contains inaccuracies. Similarly, in data-to-text tasks, reflecting inaccuracies in the input data may be preferable for users. Evaluating against input context alone also simplifies the annotation process, as annotators need only to reference the input. However, with pre-trained LLMs encoding extensive real-world knowledge in their weights (Petroni et al., 2019), tasks such as open-domain question answering or instruction following often require correctness against real-world context.

Howcroft et al. (2020) analyze quality criteria in 165 NLG papers and highlight the lack of consensus on naming and definitions of evaluation aspects. The same term may refer to different concepts and the same concept may use different terms. Belz and Mille, et al. (2020) demonstrate that in three different papers, fluency is used to assess goodness of form, correctness of form and even correctness of content with external reference. After normalizing aspect names, Howcroft et al. (2020) find that over half of the papers lack aspect definitions, with some failing to name the aspects entirely.

Most aspects involve a certain degree of subjectivity. For instance, relevance in summarization is typically defined as covering important information from the source while omitting unimportant details (e.g., Fabbri, Kryściński, et al., 2021). Determining what is important information, however, is more ambiguous than deciding whether an information is factually consistent with the source. Kryściński et al., 2019 find that evaluators agree, on average, on only 0.6 sentences when selecting the most important sentences in a document[5]. Some tasks involve aspects that are even more subjective, such as dialogue or story generation. Mehri and Eskenazi (2020b) collected human judgments for five aspects, including the interestingness of responses. Chhun et al. (2022) evaluate generated stories using aspects such as empathy and engagement, defining empathy as "how well the reader

[5]Average document length was 16.59, with a standard deviation of 5.39.

understood the character's emotion" and engagement as "how much the reader engaged with the story". In story generation, these subjective aspects strongly depend on the reader's interpretation and background, making reliability of ratings particularly sensitive to annotation guidelines and the number of annotators. It is worth noting that even seemingly purely objective aspects of text quality are not entirely free from subjectivity. For example, assessments of grammaticality may vary depending on the annotator's dialect (Gehrmann et al., 2023). This underscores the importance of providing clear and unambiguous aspects definitions and establishing sufficient annotation guidelines.

### 2.5.4 Automatic Metrics

#### Overlap Metrics

Overlap metrics are the most widely used type of automatic evaluation metrics (Schmidtová et al., 2024). They measure the extent to which a text matches one or more human-written reference texts. We describe some of the most prominent in the following:

**F-Score** is a statistical measure of accuracy commonly used in NLG evaluation. Defined as a harmonic mean of precision and recall, it balances false positives and false negatives. In evaluation of generated text, the F-Score can be applied to tokens, n-grams, or more abstract linguistic structures, such as syntactic or semantic units. F-Score ranges from 0 to 1. Precision, recall or F-score are also important components of some other automatic evaluation metrics, as discussed below.

**BLEU** (Papineni et al., 2002) is a precision-based metric originally developed for evaluating machine translation. It computes precision over n-grams of different orders by measuring their overlap with one or more reference texts. To discourage repetitive phrases in the outputs, the count of an n-gram in the output is clipped to its maximum count in reference texts. For example, phrases in output texts such as "is the is the is the" are not additionally rewarded for the extra bigrams if "is the" appears only once in the reference text. BLEU also penalizes overly short outputs with Brevity Penalty (BP), which is defined as:

$$\mathrm{BP} = \begin{cases} 1 & \text{if } o > r, \\ \exp\left(1 - \frac{r}{o}\right) & \text{if } o \leq r \end{cases} \tag{2.7}$$

where $o$ is the length of the output and $r$ is the length of the reference. The overall BLEU score is calculated with the following formula:

$$\mathrm{BLEU} = \mathrm{BP} \cdot \exp\left(\sum_{n=1}^{N} w_n \log p_n\right) \tag{2.8}$$

where $N$ is the maximum N-gram order, $p_n$ is precision of n-grams of order $n$, and $w_n$ is a weight for n-gram of order $n$. The term $p_n$ is computed as the number of matching n-grams divided by the total number of n-grams in the output for a given $n$. Using a higher $N$ value improves the metric's sensitivity to word order, but reduces flexibility with respect to variations in the texts. In practice, weights for different n-gram orders are usually set to uniform.

**ROUGE** (Lin, 2004) is a metric designed for summarization of long texts. Similarly to BLUE, it has been adapted to other NLG tasks as well. The metric has several variants, with ROUGE-N and ROUGE-L being most commonly used. ROUGE-N is a recall-based metric that measures the overlap of n-grams between output and reference texts, using this formula:

$$\text{ROUGE-N} = \frac{\sum_{n \in R} c_m}{\sum_{n \in R} c_r} \tag{2.9}$$

where $n$ is n-gram, $R$ is reference text, $c_m$ is the count of matches for the n-gram in the output, and $c_r$ is the count of the n-gram in the reference text. ROUGE-L is based on both precision and recall and uses longest common subsequence (LCS), the longest sequence of words that appear both in output and reference text in the same order. Precision is then calculated as the LCS length divided by the length of the output. Recall is computed as the LCS length divided by reference length. Precision and recall are then combined to compute an F1-score.

**METEOR** (Banerjee and Lavie, 2005) computes exact match precision and recall, but uses a backoff to non-exact matching if no exact match is found. Non-exact matching involves stemming, synonyms and paraphrases. The score is computed as:

$$\text{METEOR} = F \cdot (1 - p) \tag{2.10}$$

where $F$ is the F-score and $p$ is a penalty that discourages non-contiguous matching fragments. Unlike BLEU, this metric is designed for segment-level comparisons where it provides better correlations with human ratings (Agarwal and Lavie, 2008).

Although still widely used, n-gram overlap metrics face multiple limitations. They focus on surface form rather than semantics and are not able to account for variations in texts, nor do they address issues such as factual inconsistencies in the output (Gehrmann et al., 2023). These metrics also lack interpretability, as they only provide a single number, while evaluation of text is inherently multi-dimensional. Another limitation is that they require reference texts, which are not always available. For example, BLUE was designed to be used with multiple reference texts. Furthermore, overlap metrics, including BLEU and ROUGE have been shown to correlate poorly with human judgements (Kryściński et al., 2019; Liu, Lowe, et al., 2016; Lowe, Noseworthy, et al., 2017; Reiter and Belz, 2009; Schluter, 2017). Based on an analysis of 284 correlations reported in 34 studies, Reiter (2018) suggests using BLEU for diagnostic evaluation for MT, but not for other tasks or for evaluating individual texts.

### Distance Metrics

Distance metrics involve a function that measures similarity between two texts. In NLG, these metrics measure how similar the output is to reference texts (Celikyilmaz et al., 2020). There are two main categories of distance metrics. Edit distance-based metrics measure the dissimilarity of two strings based on the minimum number of edit operations needed to transform one string to another. Embedding metrics use vector embedding representation of texts to assess their similarity. We present examples of metrics of each types below:

**Word Error Rate** (WER) is derived from the edit distance and operates at the word level. It measures the number of operations needed to transform one text to another, calculated using the following formula:

$$\text{WER} = \frac{S + D + I}{N} \tag{2.11}$$

Here, $S$ (substitutions) represents the number of words in the reference text that were incorrectly replaced in the output, $D$ (deletions) indicates the number of reference words that are missing in the output, and $I$ (insertion) represents to extra output words not present in the reference. Finally, $N$ is the total number of words in the reference text. WER has been used for evaluation of speech recognition and machine translation systems.

**BERTScore** (Zhang, Kishore, et al., 2019) is a reference-based metric based on pre-trained contextual embeddings from BERT (Devlin, 2018). BERTScore computes recall of the output sentence as a pairwise cosine similarity between token embeddings of the output and reference sentences. **MoverScore** (Zhao, Peyrard, et al., 2019) was designed as an improvement over BERTScore, using BERT embeddings and Word Mover's Distance (Kusner et al., 2015).

#### Trained Metrics

**BARTScore** (Yuan et al., 2021) uses BART (Lewis, 2019), a pre-trained language model, to compute a score based on the likelihood of the output text conditioned on the reference text.

**UniEval** (Zhong et al., 2022) is a trained metric based on T5 (Raffel et al., 2020), designed to evaluate multiple aspects. UniEval frames NLG evaluation as a boolean question answering task by encoding the aspects, source and target as questions and answers, based on which the evaluation score is computed.

**USR** (Mehri and Eskenazi, 2020b) is a reference-free metric designed specifically for dialogue response generation. It is based on RoBERTa (Liu, 2019) pre-trained language model, and consists of multiple sub-metrics, each specializing in different evaluation aspect.

### 2.5.5 LLM-based Evaluation

#### Prompt-based Approaches

With the advancements in large language models (LLMs), numerous studies have explored their application in natural language generation (NLG) evaluation. One prominent line of research focuses on leveraging proprietary LLMs, particularly those developed by OpenAI. The main advantage of these approaches lies in their lower development costs, as they rely solely on prompt engineering and work effectively in a zero-shot setting. However, proprietary models are often prohibitively expensive for large-scale NLG evaluation and suffer from transparency and reproducibility issues (Chen, Zaharia, et al., 2023).

**GPTScore** (Fu et al., 2024) utilizes an LLM like GPT-3 (Brown et al., 2020) to estimate the probability of a candidate text based on a given task description, aspect, and contextual information, such as source or reference texts and optional few-shot examples. This probability serves as the candidate text's evaluation score, based on the assumption that high-quality texts will be assigned higher probabilities by the LLM. Formally, GPTScore is expressed as:

$$\text{GPTScore}(\mathbf{h} \mid d, a, S) = \sum w_t \log p(h_t \mid \mathbf{h_{<t}}, T, \theta) \tag{2.12}$$

Here, $h = \{h_1, \ldots, h_m\}$ represents the candidate text with $m$ tokens, $d$ is the task description, $a$ is the aspect definition and $S$ is the context information. $T$ denotes a prompt template including $d$, $a$ and $S$, while $\theta$ refers to model parameters and $w_t$ is the weight for the token at position $t$.

**GEMBA** (Kocmi and Federmann, 2023b) applies GPT models with simple prompting to predict a score for overall translation quality, using source text and optionally reference text. In a follow-up work (Kocmi and Federmann, 2023a), the authors prompt GPT models to identify errors in the output, and apply the MQM framework to annotate error spans. Similar approach is adopted by AutoMQM (Fernandes et al., 2023) and EAPrompt (Lu, Qiu, et al., 2023), where the LLM provides a list of errors annotated with error categories and severity levels, which are then utilized to derive the final score.

**G-Eval** (Liu, Iter, et al., 2023) introduces an approach where an LLM generates scores directly, which are then normalized into continuous values using token probabilities. Given a prompt containing a task description and aspect definition, the model generates a CoT with detailed evaluation steps. This is followed by a form-filling approach to produce a score (e.g., on a scale of 1 to 5) based on the prompt, the generated CoT, and the input, which includes both contextual information (e.g., source text) and output text. The final score is computed by weighting the output tokens by their probabilities to obtain a normalized value:

$$\text{score} = \sum p(s_i) \cdot s_i \tag{2.13}$$

This normalization aims to address the issues of a large number of tied scores and low score variance.

### Fine-tuning Approaches

Some studies use proprietary LLMs solely to generate data to train smaller, open-source LLMs as NLG evaluators. We present some of methods in the following:

**InstructScore** (Xu, Wang, et al., 2023) leverages GPT-4 (Achiam et al., 2023) to generate synthetic data, which is then used to fine-tune a specialized model for MT evaluation. After producing synthetic outputs with predefined error types, Llama-7B LLM (Touvron et al., 2023) is iteratively fine-tuned on data that is refined through automatic feedback in each iteration. This method aims to provide explainable outputs that consist of a list of errors annotated with their error type,

location, severity level and explanations. Similar to the MQM framework, the errors are weighted according to their severity, and the final negative score is computed as the sum of all error weights.

**TigerScore** (Jiang, Li, et al., 2023) is a reference-free evaluation metric fine-tuned using the Llama-2-7B and Llama-2-13B LLMs (Touvron et al., 2023), leveraging synthetic data generated by GPT-4 for multiple NLG tasks. In addition to purely synthetic system outputs, the training dataset contains pre-generated outputs from a number of different NLG systems. The output from TigerScore provides an error analysis of the evaluated text, where each error is annotated with its location, the evaluation aspect, an explanation and a penalty score ranging between $-5$ and $-0.5$. Penalty scores for all errors are summed to obtain the final score.

**Prometheus** (Kim, Shin, et al., 2024) provides free-form text along with a numeric score to assess output quality. This method focuses on evaluating general instruction-following capabilities. The model based on fine-tuning Llama-2-Chat models on synthetic data generated by GPT-4. Each example in the dataset consists of an instruction, a score rubric that describes scoring guidelines for a specific evaluation aspect, and the corresponding output with a reference. Prometheus 2 (Kim, Suk, et al., 2024) builds on this work, introducing an ability to process both direct assessment and pairwise ranking.

### 2.5.6 Meta-evaluation

The performance of automatic metrics is evaluated based on their alignment with human ratings, typically by measuring the correlation between automatic and human scores. Depending on how the scores are grouped, correlations can be measured at two distinct levels. Segment-level correlation evaluates the ability of a metric to distinguish between different outputs by grouping scores at the output level. If multiple scores for a single output exist, they are usually aggregated. In contrast, system-level correlation aggregates scores at the system level and evaluates the metric's ability to distinguish between different systems. For a meta-evaluation to be informative, it is important that meaningful differences exist between evaluated outputs or systems.

To formalize the procedure, let there be $S$ systems and $I$ inputs. For each system $s_i$ and input $i_j$, there is an output $o_{ij}$. Applying an evaluation metric yields a matrix $X$ of size $S \times I$, where $x_{ij}$ represents a score for $o_{ij}$. Similarly, human scores can be represented by a matrix $Y$ of the same dimensions, where $y_{ij}$ denotes the human score for $o_{ij}$. Consequently, segment-level correlation has multiple variants, each with different grouping (cf. Gao, Hu, et al., 2024).

**Global-level** correlation treats the scores as a single flat vector without any grouping:

$$Corr_{\text{global}}(X, Y) = c\left((x_{ij})_{i=1,j=1}^{S,I}, (y_{ij})_{i=1,j=1}^{S,I}\right) \tag{2.14}$$

**Input-level** correlation groups the outputs based on the inputs. To achieve this, the correlation coefficient between systems—represented as two $S$ dimensional

vectors—is calculated for each input. These $I$ coefficients are then averaged:

$$Corr_{\text{input}}(X, Y) = \frac{1}{I} \sum_{j=1}^{I} c\left((x_{ij})_{i=1}^{S}, (y_{ij})_{i=1}^{S}\right) \tag{2.15}$$

**Item-level** correlation calculates correlation coefficient for outputs grouped by system and then averages the $S$ coefficients:

$$Corr_{\text{item}}(X, Y) = \frac{1}{S} \sum_{i=1}^{S} c\left((x_{ij})_{j=1}^{I}, (y_{ij})_{j=1}^{I}\right) \tag{2.16}$$

WMT22 metrics shared task Freitag and Rei, et al. (2022) applied this approach as one of three methods for measuring correlations, referring to it as *segment averaging*. In this work, we follow the term used in Deutsch et al. (2023).

### Correlation Coefficients

The $c$ term introduced above might represent one of the following correlation coefficients:

**Pearson's $r$** measures the linear relationship between two variables, quantifying how changes in one variable relate to changes in another variable. Its values range from $-1$ to 1, where positive values indicate positive linear correlation, negative values indicate negative linear correlation, and 0 signifies no correlation. Pearson's $r$ is calculated as covariance of the vectors for the two variables, divided by the product of their variances:

$$r = \frac{\sum_{i=1}^{n}(X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum_{i=1}^{n}(X_i - \bar{X})^2 \sum_{i=1}^{n}(Y_i - \bar{Y})^2}} \tag{2.17}$$

**Spearman's $\rho$** is a nonparametric correlation coefficient used to measure the association between the rankings of two variables. It is particularly suited for monotonic relationships, even when they are nonlinear. Spearman's $\rho$ ranges from $-1$ to 1, where a high positive value indicates strong similarity between the rankings of two variables, while low negative value indicates dissimilarity in their rankings. The formula for Spearman's $\rho$ is as follows:

$$\rho = 1 - \frac{6 \sum_{i=1}^{n} d_i^2}{n(n^2 - 1)} \tag{2.18}$$

where $n$ is the number of observations and $d_i$ is the difference between the ranks of values of the two variables.

**Kendall's $\tau$** measures the ordinal relationship between two quantities, where a similar rank between observations of two variables leads to higher Kendall correlations. The value of $\tau$ is based on calculating the number of concordant and discordant pairs: concordant pairs are pairs with matching order of observations for the two variables, while discordant pairs are those pairs where the order does not match. The formula for $\tau$ is as follows:

$$\tau = \frac{2}{n(n-1)} \cdot (C - D) \tag{2.19}$$

where $C$ is the number of concordant pairs, $D$ is the number of discordant pairs and. The difference is normalized by $\frac{2}{n(n-1)}$, which is derived from the total number of possible pairs $\frac{n(n-1)}{2}$. This coefficient should be interpreted carefully if there is a large number of tied pairs in the observations, since these do not contribute to either concordance or discordance.

### Span Precision and Recall

For the meta-evaluation of error span annotations provided by fine-grained automatic metrics, precision and recall can be used to assess how well the metric aligns with human annotations. In this context, precision measures the overlap between the spans identified by the metric and target spans, while recall measures the percentage of gold spans that are identified as error spans. However, predicted and gold spans often do not align perfectly, even if they refer to the same segment of the output. In such cases, it might be appropriate to relax the alignment criteria. For example, Fernandes et al. (2023) compute precision and recall based on *positional spans* instead. Given a set of annotated spans $S$, positional span is defined as the set that contains the indices of all words within each span in $S$.

# 3 Methods

This chapter outlines our methodology used to develop an LLM-based metric introduced in 1. The approach consists of several stages. First, we develop a prompt-based method to NLG evaluation, using an ensemble of open-weight LLMs. This ensemble is then employed to create a balanced synthetic training dataset that represents a diverse range of NLG tasks, consisting of fine-grained annotations and overall numeric scores. Using this dataset, we train a metric designed to evaluate a range of tasks across multiple aspects, including new tasks and evaluation aspects not seen during training. The entire process, from dataset creation to fine-tuning, uses only open-weight LLMs, which makes it transparent and replicable.

Section 3.1 introduces the problem formulation and defines the scope of the methodology. In Section 3.2, we describe the prompting approach applied to an ensemble of LLMs, which is then used to construct a dataset detailed in Section 3.3. Finally, Section 4.4 present the approach used to fine-tune the evaluator model.

## 3.1 Problem Formulation

As outlined in Chapter 1, the goal of this thesis is to develop an explainable, reference-free, LLM-based evaluation metric that provides both quantitative assessments of evaluated text, as well as natural language explanations in the form of fine-grained error analysis.

More specifically, given an input $x$, an output $y$, and an evaluation aspect $a$, this metric should return a tuple $(z, \{e_1, \ldots, e_n\})$, where $z$ is a numeric score assigned to $x$, and $\{e_1, \ldots, e_n\}$ represents a set of error annotations. Each error annotation includes a span of text corresponding to the problematic segment, an explanation, and a severity level. While the term "error" is used throughout this work, it should be understood in a broader sense as any issue in the text that may require improvement.

## 3.2 Ensemble Prompting

We apply an ensemble of open-weight LLMs to generate a synthetic evaluation dataset, which serves as training data for fine-tuning a smaller NLG evaluation model. However, this could also be used independently as a standalone method for evaluating NLG systems. In this section, we present a detailed description of the prompt-based approach.

### 3.2.1 The Approach

We prompt an ensemble of open-weight LLMs, referred to as *annotators*, to perform evaluations consisting of error span annotations and overall scores. Error annotations serve a dual purpose: first, as a chain-of-thought that provides a detailed analysis of the issues in the output before assigning a final score,

and second, to introduce explainability and informativeness into our approach, compared to simple scoring. Each annotator generates an *annotation set*, which is a list of errors identified and the overall score. After post-processing these outputs, a *supervisor*—a different LLM using its own prompt template—is instructed to merge the annotation sets into a single unified annotation. Inputs to the supervisor exclude overall scores, which are aggregated separately.

The process can be formalized as follows. Let $A = \{a_1, \ldots, a_n\}$ represent the set of annotators, $s$ be the supervisor and $\phi$ the score aggregation function. Suppose $T_e$ is the evaluation prompt template and $T_m$ is the merging prompt template. Let $x$ be the input to evaluate. Applying annotator $a_i$ with prompt template $T_e$ to input $x$ and evaluation aspect $c$ yields the evaluation $(e_i, z_i) = a_i(x, c; T_e)$, where $e_i$ is the error annotation set and $z_i$ is the overall score. $e_i$ consists of zero or more error annotations $\{e_{i1} \ldots, e_{im}\}$. Each annotation is a triple $e_{ij} = (p_{ij}, q_{ij}, r_{ij})$, where $p_{ij}$ is the location of the error, $q_{ij}$ is an explanation for the error and $r_{ij}$ is the severity level. After all annotators process the input, we obtain error annotations $e_1, \ldots, e_n$ and the corresponding scores $z_1, \ldots, z_n$. These outputs are then merged as follows:

$$e_f = s(e_1, \ldots, e_n; T_s) + \phi(z_1, \ldots, z_n) \tag{3.1}$$

where $e_f$ is the final annotation and + denotes concatenation.

### 3.2.2 Prompts

#### Annotator Prompt

Figure A.16 shows an example of the prompt template for data-to-text evaluation. We use this template for illustration, since prompt templates for other tasks follow the same structure and differ only in minor details, such as task description. In the following, we describe the structure of the prompt template in detail.

Lines 1–4 outline the primary instructions for the evaluation process, including the description of the evaluated task. Lines 6–7 specify the aspect name and its definition. The rules presented in lines 9–12 provide additional instructions designed to narrow the scope of issues the annotator LLM should address. These rules are informed by preliminary observations of various failure modes. Since LLMs are known to confuse different evaluation aspects (Hu, Gao, et al., 2024), the rule in line 11 ensures that they remain focused on the specific evaluation aspect. Additionally, the rule in line 12 requires the annotator to justify any score lower than the maximum by at least one identified error. This ensures that the interpretability of the evaluation is not reduced by annotations with lower scores but no errors. This instruction also simplifies the parsing by explicitly defining the output format when no errors are detected.

Lines 14-18 present detailed steps for error identification, inspired by (Liu, Iter, et al., 2023). Line 18 introduces the scoring scale, including an explanation of lowest and highest scores. The scale is presented as categorical, which is based on the intuition that adjectival categorical scales could be easier for language models to interpret than numeric scales. Initially, a categorical scale was also used for error severity. However, we observed that LLMs often confused these two scales, leading to a change to a numeric scale for error severity.

Lines 20–21 describe the input that was used to generate the evaluated output. Depending on the task, different headers are applied. For example, in news article summarization, "Article" and "Summary" are used for input and output. Some tasks may involve multiple inputs; for instance, evaluating knowledge-grounded dialogue requires both dialogue history and context knowledge. Similarly, question-answering tasks require both a context and a question as inputs. In such cases, additional context is presented under separate headers. Lines 23–24 contain the output to be evaluated. Finally, lines 26–39 describe the required output format. Although structured formats like JSON might make parsing easier, prompting LLMs to reason within strict structured outputs has been shown to impair their performance (Tam et al., 2024). Therefore, the models are instructed to produce textual outputs, which are then parsed using regular expressions. These regular expressions extract individual error annotations and overall scores from the outputs.

**Supervisor Prompt**

The prompt template used by the supervisor LLM to unify error analyses from multiple annotators is presented in Figure A.17. This template instructs the LLM to merge the provided annotation sets into a unified annotation set based on the guidelines outlined in the prompt. Specifically, it instructs the LLM to merge all error annotations that refer to the same issue at approximately the same location in the text. Therefore, if two error annotations generated by different LLMs refer to the same issue but choose slightly different span boundaries in the output text, the supervisor combines these into a single annotation. At the same time, the supervisor is instructed to maintain the granularity of error annotations, ensuring that distinct locations in the text with the same type of error are not merged, and that different types of errors at the same location are treated as separate annotations.

Despite the detailed instruction provided in the evaluation prompts, each LLM in the ensemble generates error analyses, particularly the error spans, in slightly different formats. For example, annotator models may occasionally include two non-adjacent spans under a single annotation or insert comments about the span directly into the location line. Therefore, the supervisor prompt also functions as a normalization step to standardize the annotations into a consistent format. Specifically, the LLM is instructed to avoid grouping multiple error locations into lists under a single error annotation and to exclude any extra text that is not part of the output from the location line. It is also directed to remove any markdown formatting that might be present in the annotations. Since the tasks with longer outputs might involve a higher number of error annotations when combining evaluations from multiple models, we restrict the number of errors in the result to a maximum of eight.

## 3.3 Dataset

The ensemble approach presented in Section 3.2 is applied to generate synthetic evaluation data, which we then use to train a specialized LLM-based evaluator. The dataset consists of five task categories: data-to-text, summarization, dialogue

| Category | Task | Inputs | Output |
|---|---|---|---|
| Data-to-text | RDF-to-text | RDF triples | textual description |
| | attribute-value-to-text | attribute-value pairs | textual description |
| | table-to-text | table | textual summary |
| | logical NLG | table | observations/insights |
| Summarization | article summarization | article | summary |
| | forum post summarization | forum post | summary |
| | dialogue summarization | dialogue | summary |
| Dialogue generation | dialogue generation | dialogue, (context) | response |
| Question answering | table question answering | table, question | answer |
| | narrative question answering | story, question | answer |
| Story generation | story generation | prompt | story |

**Table 3.1** Overview of tasks in each category with corresponding input and outputs. Optional inputs are enclosed in parentheses.

response generation, story generation and question answering. These categories were chosen to represent a diverse range of tasks, each with unique objectives, input-output relationships and evaluation aspects. Data-to-text represents tasks where an input is *transformed* from one representation to another. Summarization focuses on *compressing* information from a longer text into a shorter version. Question answering requires *extracting* relevant information from a text in response to a specific query. Finally, both dialogue response generation and story generation involve *creating* outputs with novel content.

Each task category comprises one or more subtasks. A task category refers to a group of related tasks, such as summarization or data-to-text, whereas subtasks are specific instances typically associated with a dataset and a distinct input-output format. For example, table-to-text and RDF-to-text are subtasks within our data-to-text category, while the summarization category includes tasks like dialogue summarization and news article summarization. Some tasks may belong to more than one category. For example, FeTaQA (Nan, Hsieh, et al., 2022), a table question answering task, can be categorized as both a QA and a data-to-text subtask. Overall, our dataset includes 13 subtasks drawn from 15 source datasets.

To obtain outputs with varying quality and diverse types of errors, we sample outputs from multiple systems for each input. These range from rule-based approaches to the latest state-of-the-art LLMs. For older systems, we primarily use pre-generated outputs from existing datasets, while outputs from more recent systems, including LLMs, are newly generated. An overview of the evaluated systems is presented in Appendix A.1.

For each subtask, we select a set of relevant evaluation aspects with corresponding definitions. We consider two aspects distinct if they are associated with different tasks, even if they share the same name. For instance, *coherence* in dialogue response generation refers to the coherence of the response with the dialogue history, while in summarization, it refers to the coherence of a summary alone.

The inputs for the evaluator models are sampled as follows: we determine the number of systems and aspects per input for each source dataset. For each input, we randomly sample $n$ systems and their corresponding outputs. Then, $m$ aspects are sampled for each input-output pair. This results in $n \times m$ (input, output, aspect) triples for each input, which are then used to create the prompts

| Task | Datasets | Systems | Aspects | Examples |
|---|---|---|---|---|
| Summarization | 5 | 39 | 6 | 12,070 |
| Data-to-text | 4 | 29 | 7 | 7,894 |
| Dialogue | 3 | 36 | 9 | 10,074 |
| Story Generation | 1 | 9 | 6 | 3,200 |
| Question Answering | 2 | 15 | 11 | 4,849 |

**Table 3.2** Dataset Statistics

for synthetic data generation. For most tasks, we set $n = 4$ and $m = 3$. This sampling strategy aims for a balanced distribution of inputs, system outputs and evaluation aspects. Additionally, it helps to ensure that the model is exposed to different outputs for the same input, and does not learn to evaluate based solely on the patterns in the inputs. Finally, by presenting multiple aspects for the same input-output pair, the model is encouraged to learn differences in output quality across various evaluation aspects. Table 3.2 shows the basic statistics of our dataset.

Since the individual can LLMs often differ in their assessment of the same output, we merge their evaluations in a way that minimizes inconsistencies between identified errors and overall scores. Specifically, we begin by removing outliers from the scores for each evaluated output. Outlier is defined as a score that differs from the other scores for the same output by more than two standard deviations. However, the difference must be at least one to ensure that scores from sets with more uniform scores are not unnecessarily excluded. The remaining scores are then averaged to obtain the final aggregated score for the output. The details are presented in Algorithm 1.

Table 3.3 shows the percentage of instances where a particular model's score was excluded for a given task category. While the percentages are generally low, the models disagree dramatically more in their assessment of data-to-text outputs. Command R+ has the highest relative frequency of outlier scores for both data-to-text and question answering outputs, followed by Gemma in the same task categories.

The aggregation method produces more granular floating-point scores, enabling the fine-tuned model to learn more fine-grained scoring. To restrict the space of output scores to a reasonable number, we converted the averaged floating-point scores to integers between 0 and 100, and additionally binned the values to multiples of five. Figure 3.1 shows the distribution of scores in the training dataset obtained by this procedure. With a few exceptions, the scores are reasonably distributed across the scale, except for the value 100, which represents nearly one-third of all scores. However, given that LLMs tend to over-annotate the errors (Fernandes et al., 2023), we believe that having substantial proportion of examples evaluated at the highest score could be beneficial for the fine-tuning process.

The rest of this section outlines the tasks, source datasets, systems and evaluation aspects used to generate the dataset.

### 3.3.1 Summarization

As source data for summarization, we utilize five datasets that include three distinct tasks: news article summarization, forum post summarization and dialogue

| Model | D2T | Summ | Dialogue | Story | QA |
|---|---|---|---|---|---|
| Command R+ | 28.88% | 0.13% | 0.73% | 0.94% | 9.61% |
| Gemma | 10.32% | 1.96% | 1.52% | 1.03% | 5.07% |
| Mistral | 4.89% | 0.93% | 0.63% | 0.25% | 1.22% |
| Nemotron | 4.27% | 1.48% | 4.02% | 1.03% | 0.85% |
| Qwen | 1.39% | 1.36% | 1.43% | 1.12% | 0.47% |

**Table 3.3** Percentage of outliers for each model per task category.

**Algorithm 1** Score aggregation from multiple LLMs

**Input:** List of sets of scores $S = \{s_1, s_2, \ldots, s_n\}$
**Output:** Averaged list of scores.
**for** each set of scores $s_i \in S$ **do**
    Compute the mean $\mu_i$ and standard deviation $\sigma_i$ of $s_i$.
    Remove outliers from $s_i$ based on the following conditions:
    **for** each score $x \in s_i$ **do**
        **if** $|x - \mu_i| > 2\sigma_i$ **and** $|x - \mu_i| > 1$ **then**
            Remove $x$ from $s_i$.
        **end if**
    **end for**
    Compute the average $\bar{s}_i$ of the remaining scores in $s_i$.
**end for**
**return** averaged scores

summarization. Our evaluation dataset includes six commonly used aspects: four related to meaning of the evaluated text, one to its form and one to both. Some of these aspects overlap significantly in their definitions. This is intentional, as our goal is to make the evaluator model robust to small variations in aspect definitions, and enable generalization to new aspects. Table A.1 provides an overview of the evaluated systems.

### Datasets

- **CNN/DailyMail** (Hermann et al., 2015) is a popular summarization dataset that consists of news articles from CNN and DailyMail, paired with corresponding bullet-point summaries. Originally developed for question answering, it was later adapted for summarization (Nallapati et al., 2016). For this dataset, we use pre-generated outputs from Stiennon et al. (2020), which include results from 11 different systems, including a human reference and an extractive baseline. Since our meta-evaluation datasets contain inputs from CNN/DailyMail, we ensure there is no overlap in the source texts between these datasets when sampling the inputs.

- **Newsroom** (Grusky et al., 2018) is a large-scale dataset of news articles and their summaries, collected from diverse sources, domains, authors and time range. We use outputs of the systems evaluated in the original paper, which include three summarization systems and two extractive baselines.

- **SAMSum** dataset (Gliwa et al., 2019) addresses the dialogue summarization task, containing dialogues with short summaries created by linguists. We

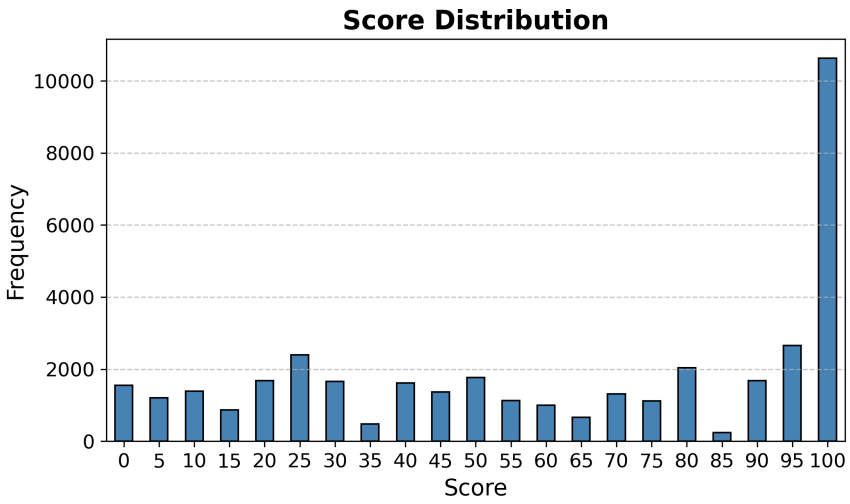


**Figure 3.1** Distribution of overall scores in the dataset.

use pre-generated outputs of six systems from (Gao and Wan, 2022).

- **TL;DR** (Völske et al., 2017) is a collection of Reddit posts with user-created summaries. Unlike many popular datasets that focus on news articles, TL;DR contains informal and less structured texts spanning diverse topics. Similarly to CNN/DailyMail, we use outputs from Stiennon et al. (2020).

- **XSum** (Narayan et al., 2018) is a dataset for *extreme summarization* task, designed for abstractive approaches. Unlike datasets such as Newsroom or CNN/DailyMail, which favor extractive summarization, XSum contains BBC articles paired with concise, single-sentence summaries. We utilize pre-generated outputs from five systems and baselines evaluated in the original paper.

### Aspects

- **Consistency** evaluates whether the summary is factually aligned with the source text. This involves determining if the facts in the summary can be entailed by the source. Consistency is closely tied to hallucinations, which may be categorized as factual or non-factual, as discussed in Section 2.5.3. In our approach, all information is required to be supported by the source text, making both types of hallucinations inconsistent with the source. The definition of consistency used in the prompts is: *Extent to which the facts in the summary are consistent with the source text. Factually consistent summary should not contain facts that are not supported by the source text.*

- **Accuracy** is largely synonymous with consistency. It evaluates whether the factual information from the source is accurately represented in the summary. Following Stiennon et al. (2020), we define accuracy as: *Extent to which the factual information in the summary accurately matches the post. An accurate summary should not contain information that is not present in the post, should not contradict the post, and generally should not be misleading.*

| Format | Example | Datasets |
|---|---|---|
| JSON | `{"name": "The Phoenix", "eatType": "pub", "food": "Indian", ...}` | E2E NLG |
| attribute-value (1) | `name: The Phoenix\neatType: restaurant\nfood: Indian\n ...` | E2E NLG |
| attribute-value (2) | `name[The Phoenix], eatType[restaurant], food[Indian], ...` | E2E NLG |
| RDF (1) | `"The_Velvet_Underground \| genre \| Proto-punk" ...` | WebNLG 2020 |
| RDF (2) | `(The_Velvet_Underground, genre, Proto-punk) ...` | WebNLG 2020 |
| CSV | `united states,32,1,31,12\naustralia,5,0,5,3\n ...` | LogicNLG |
| linearized table | `<page_title> List of Norwegian fjords </page_title> <section_title> ...` | ToTTo |

**Table 3.4** Input data formats for data-to-text tasks.

- **Relevance** of a summary is concerned with content selection. A relevant summary should include important points from the source text while omitting unimportant details. Compared to consistency, relevance is more subjective, as determining what information should or should not be included in a summary can sometimes be ambiguous. The definition used in our prompts is: *Extent to which the summary captures important information of the source text. A relevant summary should include all and only important information from the source text.*

- **Coverage** evaluates how much of the important information from the source text is covered by the summary. In this sense, it is closely related to relevance, however, it does not include non-redundancy as a criterion. We use a definition adapted from Stiennon et al. (2020): *Extent to which the summary covers the important information in the source text. A summary has good coverage if it mentions the main information from the source text that is important to understand the events described in the text. A summary has poor coverage if someone reading only the summary would be missing several important pieces of information about the event in the source text.*

- **Coherence** refers to the structural quality of the summary and involves attributes such as cohesion, consistency and relevance (Reinhart, 1980). It evaluates both the semantic and formal structure of the text. It is determined both by semantic and formal structure of the text. We define coherence as: *Extent to which the summary is well-structured and organized, presenting information in a logical order that flows naturally from sentence to sentence. Coherent summary forms a unified body of information and makes it easy to understand the main ideas.*

- **Fluency** focuses on the formal quality of the text, including grammaticality and naturalness. Unlike coherence, fluency is concerned with sentence–level quality rather than the overall structure of the text. We define fluency as: *Formal quality of individual sentences of the summary. A fluent sentence should be grammatical, natural and easy to understand.*

### 3.3.2 Data-to-text

For data-to-text category, we collected inputs from four datasets that represent four distinct tasks: table-to-text, RDF-to-text, attribute-value list to text and logical NLG. Since it is important for an NLG evaluation method to reliably evaluate outputs with respect to structured data in different formats, we include

a number of different input formats in our training data, including JSON, CSV and linearized tables with markup. Table 3.4 provides an overview of the input formats. The list of evaluated systems is shown in A.2.

**Datasets**

- **E2E NLG** dataset (Dušek et al., 2020; Novikova et al., 2017) was chosen as a representative of simple data-to-text tasks. In this dataset, models are tasked with generating descriptions of restaurant venues based on attribute-value-based meaning representations (MRs). Target descriptions were crowdsourced using textual and pictorial representations of MRs as stimuli. Compared to other datasets we use, E2E NLG has a limited set of domains and lower diversity. Therefore, we created only 1,000 training examples based on its inputs. To ensure diversity, we selected pre-generated outputs from five systems described by (Dušek et al., 2020), considering their model architectures and evaluation results across various metrics and aspects. The inputs and outputs were sampled from the test set. Three different input formats are used in the training data: MR, JSON and text.

- **WebNLG 2020** (Ferreira et al., 2020) is an RDF-to-text dataset designed for generating natural language text from RDF triples collected from DBPedia knowledge base (Mendes et al., 2012). Each input consists of one to seven triples, where each triple represents a binary relation in the form (subject, property, object). Similarly to E2E NLG, we selected a diverse set of output systems based on model architectures and evaluation results from the WebNLG+ 2020 Challenge.

- **ToTTo** (Parikh et al., 2020) is an open-domain table-to-text generation dataset. It consists of Wikipedia tables with highlighted cells, and the task is to generate single–sentence descriptions of the data in these highlighted cells. Inputs are provided in two formats: full tables, which include indices to highlighted cells, and linearized tables, where only the highlighted data is presented in a linear order, while the structure is annotated with markup tags. As we observed that even medium-sized open-source LLMs often struggle with hallucinations when using full tables, we restricted our evaluation to linearized tables.

- **LogicNLG** (Chen, Chen, et al., 2020) introduced the task of logical NLG, where models generate statements that can be logically entailed from the data in a table. This task involves various aggregations and comparisons, which makes it more difficult than simply transforming structured data to free-form text. While the task explicitly requires generating five logical statements per input, we sampled between one to five statements from each generated output to increase the diversity of output lengths. Inputs were formatted as CSV with "|" as a separator.

**Aspects**

- **Faithfulness** measures whether all information in the generated text is supported by the data, making it equivalent to precision. Similarly to factual

consistency in summarization, we consider both factual and non–factual hallucinations as errors. In our evaluations, we defined faithfulness as: *Extent to which the information in the text is supported by the data.*

- **Correctness** evaluates whether the information from the data is accurately presented in the generated text. The output is maximally correct if it does not contain any incorrect statements with respect to the input. Correctness overlaps significantly with faithfulness, but its definition varies based on the specific task. For example, we define correctness for LogicNLG as: *Extent to which the statements are logically and factually correct with respect to the provided data.*

- **Coverage** refers to the degree to which the generated text covers the information in the data. The output has maximum coverage when all information from the data is included in the text, which makes it analogous to recall. For example, in WebNLG, it evaluates whether all predicates and their arguments are mentioned, while in ToTTo, it determines whether all highlighted table cells are described. We apply coverage to evaluate all datasets except LogicNLG, where the task is to infer interesting observations, rather than fully cover the source data. We define coverage as follows, with slight variations depending on the task: *The extent to which the text includes description of all information presented in the data.*

- **Informativeness** is closely related to coverage, as it evaluates how much of the information the generated text provides. However, it does not require complete coverage of the data, therefore it is applicable to tasks such as LogicNLG, where full coverage of a table is not necessary. The definition used in our evaluation depends on the particular dataset. For example, the definition used for LogicNLG is: *The extent to which the statements provide interesting or useful information about the data.*

- **Fluency** refers to the formal quality of the generated text, and includes grammaticality, naturalness and readability. Some definitions also include coherence (Ferreira et al., 2020), although coherence is usually treated as a separate aspect. In our prompts, we define fluency as: *Extent to which the text is grammatical, natural and easy to understand.*

- **Grammaticality** focuses on the correctness of grammar and spelling in the generated text. A text is fully grammatical if it contains no grammatical or spelling errors. While grammaticality is often included as a sub-aspect of fluency, both aspects are commonly used in practice. Therefore, we include it to help the model learn differences between evaluation aspects on different levels of hierarchy. In our dataset, grammaticality is defined as: *The extent to which the text is grammatical (free of grammar and spelling errors).*

- **Naturalness** refers either to the human-likeness of the text or the likelihood that it was produced by a native speaker. Like grammaticality, naturalness is often treated as an attribute of fluency. Additionally, its evaluation often includes assessment of grammaticality, as this can be an indicator of whether the text was produced by a native speaker. This illustrates how evaluation

aspects often overlap or have hierarchical relationships. For our purposes, naturalness is defined as: *The extent to which the text is likely to have been produced by a native speaker.*

### 3.3.3 Dialogue Response Generation

For dialogue response generation, we source the inputs from three dialogue datasets, focusing on open-domain non-task-oriented dialogue response generation. Table A.3 provides an overview of the evaluated systems.

#### Datasets

- **Wizard of Wikipedia** (Dinan et al., 2018) consists of conversations grounded in one of 1365 topics and corresponding knowledge retrieved from Wikipedia. In these conversations, either participant may select the topic and initiate the discussion, although they have asymmetric roles. One participant takes on the role of the wizard, an expert with access to a topic-relevant knowledge, on which they can base their responses. The other participant acts as an apprentice, a curious learner that is eager to discuss the chosen topic. To create inputs of varying length, we randomly select a dialogue history length between two turns and the full conversation, and truncate the dialogue to this length. The last utterance is replaced by a system output, except when the reference is used as evaluated output.

- **EmpatheticDialogues** (Rashkin, 2018) is a dataset of dialogues grounded in emotional situations, designed to train and evaluate dialogue models on empathetic response generation. Each conversation is associated with an emotional label, where one of the participants describes a situation in which they experienced a given emotion. We sample up to five turns from each dialogue and replace the last utterance with a generated response. The emotion label is used as additional context for an annotator LLM to evaluate the appropriateness and empathy of the responses but is excluded from the prompts used for system output generation.

- **DailyDialog** (Li, Su, et al., 2017) includes conversations on various daily life topics, annotated with emotion labels and communicative intents. We include data from DailyDialog to represent diverse topics and scenarios in the training set. Alongside newly generated outputs, we also collect pre-generated outputs from three sources (Gupta et al., 2019; Huang et al., 2020; Zhao, Lala, et al., 2020) to represent older dialogue systems.

#### Aspects

- **Coherence** in dialogue is a concept slightly different from coherence in tasks that involve generation of standalone texts, such as summarization or story generation. In dialogue, coherence measures how meaningful and logically consistent the response is with the preceding conversation. This includes not only alignment of the response with the last utterance, but also consistency with the dialogue participant's earlier responses in terms of

logic and style. In our evaluation, coherence is broadly defined as: *Extent to which the response is a meaningful continuation of previous dialogue.*

- **Relevance** evaluates how closely a response aligns with the topic of conversation. In this sense, this aspect overlaps with coherence. We define relevance as: *Extent to which the response is relevant and on-topic given the dialogue history.*

- **Appropriateness** addresses whether the response is semantically and pragmatically appropriate in the given context. Depending on the definitions, appropriateness might overlap to a large extent with coherence and relevance. For our evaluation, we define appropriateness as: *Extent to which the response is semantically and pragmatically appropriate given the conversation history.*

- **Empathy** is evaluated specifically on responses from the EmpatheticDialogues dataset, where the goal is to determine whether the response acknowledges and reflects the emotions of the other participant. We define empathy as: *Extent to which the response shows understanding of the feelings of the person talking about their experience.*

- **Interestingness** is concerned with the informational value of the response, specifically whether it presents stimulating ideas, facts or opinions. Although it is one of the more subjective evaluation aspects, we define it vaguely and let the annotator models determine the criteria for interestingness. Therefore, in our dataset, we define interestingness as: *Extent to which the response is interesting given the dialogue history.*

- **Engagingness** is closely related to interestingness but is sometimes treated as a distinct aspect (e.g., Mehri and Eskenazi, 2020a; See, Roller, et al., 2019). While interestingness focuses on the context itself, engagingness emphasizes maintaining the user's attention and encouraging them to continue with the conversation. For our purposes, engagingness is defined as: *Extent to which the response captures and maintains the user's interest, encouraging further interaction. Engaging responses contain opinions, preferences, thoughts or interesting facts.*

- **Fluency** in dialogue response generation has a similar meaning to its use in other tasks and refers to the formal quality of the response. We define fluency as: *The extent to which the response is grammatically correct, natural and fluent.*

- **Understandability** evaluates both the content and form of a response, focusing on its clarity and ease of comprehension. The definition we use for our evaluation is: *Extent to which the response is easy to understand and comprehend given the dialogue history.*

### 3.3.4 Story Generation

The NLG tasks discussed so far generally contain inputs that are relatively longer compared to the outputs. This pattern is especially common in tasks like summarization and dialogue generation, although certain data-to-text tasks

also share this characteristic. To represent scenarios with short inputs and long outputs, we include story generation in the training data. Table A.4 lists the evaluated systems.

### Datasets

As the source of inputs, we use **WritingPrompts** (Fan et al., 2018), a story generation dataset derived from Reddit's WritingPrompts subreddit, where users submit prompts that can inspire other users to write stories. The dataset consists of a diverse range of topics, story lengths and writing styles. We reuse existing outputs from the OpenMEVA dataset (Guan, Zhang, et al., 2021) and generate additional outputs by four LLMs. To increase the diversity of generated stories in terms of their length, we generate the outputs with two different prompt versions, each requiring a different length of the story (see A.2 for details). The outputs are then randomly sampled from either the shorter or the longer set. As we observed a tendency of LLMs to generate a long list of errors for longer inputs, our evaluator models are instructed to limit the number of identified errors to a maximum of eight, and to prioritize the most severe ones if necessary.

### Aspects

While relevance and coherence are two commonly used aspects for story generation, there is no consensus on which other evaluation aspects are the most relevant. Inspired by social sciences, Chhun et al. (2022) propose four additional aspects, aimed at providing a complete and non-redundant set of criteria. Following their work, we adopt the aspects defined in the HANNA benchmark, using our own definitions for most of them:

- **Relevance** measures the degree to which a story aligns with the given prompt (Chhun et al., 2022; Chiang and Lee, 2023), title (Jhamtani and Berg-Kirkpatrick, 2020; Xie et al., 2023; Yao, Peng, et al., 2019) or story beginning (Li, Wang). In some cases, relevance also evaluates whether the story remains on-topic for its duration (e.g., Goldfarb-Tarrant et al., 2020. Since our inputs are prompts, we define relevance as: *Extent to which the story is relevant to the writing prompt.*

- **Coherence** in story generation typically refers to logical consistency and narrative flow (e.g., Jhamtani and Berg-Kirkpatrick, 2020; Li, Cui, et al., 2023; Yao, Peng, et al., 2019). Other works define coherence more vaguely, such as how much the story "makes sense" (Chhun et al., 2022) or how well its sentences "fit together" (Xie et al., 2023). For our purposes, coherence is defined as: *Extent to which the story is logically consistent and coherent.*

- **Engagement** is a subjective and often vaguely defined aspect that evaluates how engaging the story is to the reader (e.g., Chhun et al., 2022; Li, Cui, et al., 2023). Due to its inherent subjectivity, we apply a simple definition and leave its interpretation to the evaluator models: *Extent to which the story is engaging and interesting.*

- **Empathy** is related to emotional commentary and empathy, and refers to how well the story conveys character's emotions. We define empathy as: *The clarity and depth with which the character's emotions are conveyed in the story.*
- **Surprise** is concerned with the story's ending, and evaluates its unexpectedness and originality. We define surprise as: *How surprising the end of the story was.*
- **Complexity** measures how intricate and elaborate the story is. Complexity is not necessarily an aspect of quality, but rather a feature of the text. Whether greater complexity is desired or not depends on the audience. Our definition of complexity is: *How elaborate the story is.*

### 3.3.5 Question Answering

The question answering subset of the data contains two distinct tasks: narrative question answering and table question answering. The inputs consist of a question and the context data in which the answer should be grounded. Evaluated systems are listed in Table A.5.

#### Datasets

- **NarrativeQA** (Kočiskỳ et al., 2018) consists of human-written question and free-form answers based on stories. The stories include books and movie scripts and are provided either as full texts or as human-written summaries. Since full stories do not fit into the context window of many LLMs, we use only summaries for generating the outputs. A.2 contains all variants of prompts we use for output generation.
- **FeTaQA** (Nan, Hsieh, et al., 2022) is a question answering dataset based on Wikipedia tables that requires a model to aggregate and reason about the entities in the table and their relations. This places the task at the intersection of data-to-text and question answering task categories.

#### Aspects

- **Correctness** evaluates if the answer to a question is correct with respect to the input. Since our models are instructed to assess the quality on an ordinal scale, we evaluate a *degree* of correctness - the answer should receive the maximum score if it is fully correct, while lower scores should reflect the number and severity of correctness issues. We define correctness as: *Extent to which the answer to the question is correct with respect to the input.*
- **Informativeness** addresses whether all information required by the question is provided in the answer. We define informativeness as: *Extent to which the answer provides all information that the question asked for.*
- **Completeness** evaluates comprehensiveness of the answer and the degree to which all aspects of the question are covered. The meaning is slightly different from informativeness, which is concerned about the information

that the question explicitly asks for. In our dataset, completeness is define as: *extent to which the answer is comprehensive and ensures all question aspects are addressed.*

- **Conciseness** measures the degree to which an answer is focused and directly answers the question without unnecessary details and elaboration. While the goal might often be to generate both complete and concise answers, these two aspects may correlate negatively. Conciseness is defined as: *Extent to which the answer is concise and to the point.*

- **Relevance** is concerned with specificity of the answer and measures the degree to which the answer addresses the particular question asked. Although it is related to conciseness, relevance is not that much concerned with the amount of detail in the answer. We define relevance as: *Extent to which the answer is specific and meaningful with respect to the question.*

- **Factuality** evaluates factual consistency of the answer with the provided context. In our dataset, this aspect is used in the narrative question answering task, and we use a similar definition as in summarization: *Extent to which the answer is supported by the summary.*

- **Faithfulness** is used for the table question answering task and is synonymous with factuality. We define faithfulness as: *Extent to which the information presented in the answer is supported by the input.*

- **Fluency** in question answering refers to the formal quality of the answer and is defined similarly as in the other tasks presented so far: *The extent to which the response is grammatical, natural and easy to understand.*

- **Naturalness** is interpreted in the same way as in data-to-text and dialogue response generation tasks, and the definition we apply is: *Extent to which the answer is likely to have been produced by a native speaker.*

- **Grammaticality** measures the grammatical quality of the answer and is defined as: *Extent to which the answer is grammatical (free of grammar and spelling errors).*

## 3.4 Fine-tuning

We use the dataset described in Section 3.3 for supervised fine-tuning of a specialized evaluator LLM, with instruction-tuned version of Llama 3.1 8B as a backbone. By aggregating scores from multiple models during dataset construction, we extend the original ordinal scale to a continuous scale, enabling the training of a model with more discriminative scoring capabilities. As outlined in Section 3.2.1, the averaged floating-point scores are transformed into integers in range 0–100 and then binned to the nearest multiples of five. This extends the output space from five to twenty values (instead of 100), which is a tradeoff between higher granularity in the predictions and manageable task complexity.

### 3.4.1 Prompts

Because the model is expected to learn the task from training data, and the structure of its output is different, we use a special prompt template for fine-tuning. A prompt template for dialogue response generation is shown in Figure A.18. Lines 1–2 briefly describe the task, with an option to provide additional task information, which might be needed to further specify the instructions for the evaluated model, as discussed in 3.2.2. Lines 4–5 specify the aspect name and its definition. Inputs and the output to evaluate are presented in lines 7–14. Finally, lines 16–17 contain more detailed instructions.

### 3.4.2 Models

The ensemble consists of six open-weight LLMs: five annotators and one supervisor. Each LLM is distributed under a license that permits at least non-commercial use and allows the model's outputs to be used as training data. At the time of writing, these models ranked among the top-performing open-weight LLMs on the Chatbot Area Leaderboard (Chiang, Zheng, et al., 2024). The annotator models include the following:

- Llama 3.1 Nemotron 70B (Wang, Bukharin, et al., 2024)
- Qwen 2.5 70B (Yang et al., 2024)
- Gemma 2 27B (Gemma Team et al., 2024)
- Command R+ 104B (Cohere For AI, 2024)
- Mistral Large 2 123B[1]

We apply Llama 3.3 70B (Dubey et al., 2024) as the supervisor model. To address computational constraints, we use quantized versions of these models (see Section 2.3.4), such that each model can fit onto two 48G VRAM GPUs. Inference is run through the Ollama platform[2]. Table 3.5 shows the quantization levels and tags of the Ollama models. In all cases, instruction-tuned versions of the models are applied (see Section 2.3.2 for a discussion of differences between base and instruction-tuned model variants). For synthetic data generation, we sample with top-p value of 1.0, and we set the temperature to zero to obtain consistent results.

| Model | Quantization | Tag |
|---|---|---|
| Command R+ 104B | 5-bit | `command-r-plus:104b-08-2024-q5_K_M` |
| Gemma 2 27B | 8-bit | `gemma2:27b-instruct-q8_0` |
| Llama 3.1 Nemotron 70B | 8-bit | `nemotron:70b-instruct-q8_0` |
| Mistral Large 2 123B | 4-bit | `mistral-large:123b-instruct-2407-q4_K_M` |
| Qwen 2.5 70B | 8-bit | `qwen2.5:72b-instruct-q8_0` |

**Table 3.5** Quantization levels and Ollama tags used for the models.

[1] `https://mistral.ai/news/mistral-large-2407/`
[2] `https://ollama.com/`

# 4 Experiments

This section present the details of our experiments. Our goal is to answer the following research questions, outlined in Chapter 1:

- **RQ1**: What level of performance in NLG evaluation can be achieved through prompt engineering with current open-weight LLMs in their quantized versions?
- **RQ2**: Can we develop an efficient LLM-based NLG evaluator through distillation from larger open-weight models while maintaining competitive performance?
- **RQ3**: What would be the performance of such evaluator when applied to unseen domains?
- **RQ4**: How would the model perform across unseen tasks?
- **RQ5**: How would the model perform on unseen evaluation aspects?

In Section 4.1, we introduce the datasets and benchmarks used for meta-evaluation of our methods. Section 4.2 presents the baseline metrics used for comparison with our approach. The experiments for the prompt based approach are described in Section 4.3, while the details of the fine-tuning experiments are provided in Section 4.4.

## 4.1 Meta-evaluation

We evaluate both prompt-based and fine-tuning approaches on multiple benchmarks, representing a diverse range of NLG tasks (refer to Section 3.3 for details about the tasks and evaluation aspects). Pearson ($r$), Spearman ($\rho$) and Kendall ($\tau$) coefficients described in Section 2.5.6 are used for meta-evaluation. We follow the previous work in selecting specific correlation coefficients for different meta-evaluation datasets.

**Summarization** SummEval (Fabbri, Kryściński, et al., 2021), is a standard meta-evaluation dataset for summarization. It consists of summaries generated from CNN/DailyMail articles, with 100 input articles and 16 different system outputs for each article. Human evaluations address four aspects: *factual consistency*, *relevance*, *coherence* and *fluency*. Each output is scored by three expert annotators and five crowdworkers. For our purposes, we use only expert annotations, averaging their scores before computing correlation coefficients. QAGS (Wang, Cho, et al., 2020) is used to measure the performance of systems in evaluating *factual consistency* of summaries from CNN/DailyMail and XSum datasets. It includes 235 CNN/DailyMail summaries and 239 XSum summaries, each annotated by three evaluators. Annotators assign a binary factual consistency score (yes/no) for each sentence of the summary. We follow (Wang, Cho, et al., 2020) and apply majority vote for each sentence annotation, followed by averaging sentence-level scores to obtain the overall score for the summary.

**Dialogue Response Generation** For meta-evaluation on dialogue tasks, we use TopicalChat (Gopalakrishnan et al., 2023) annotations from the USR dataset (Mehri and Eskenazi, 2020b). This dataset includes human evaluations for five aspects: *groundedness*, *coherence*, *interestingness*, *naturalness* and *understandability*. As the source data differ from our training data, TopicalChat serves as out-of-domain evaluation dataset. Since *groundedness* is not present in the training data, we evaluate it as an unseen aspect. Groundedness evaluates how effectively a response uses a given context, in this case external knowledge. We select groundedness as an unseen aspect because it differs significantly from the other aspects in the training set and the USR evaluation aspects. Since the provided context is often too large for a response to include all the information, our evaluation focuses on whether any relevant parts of the context, rather than all of it, have been used. We define groundedness as: *Given the fact that the response is conditioned on, groundedness indicates how well does the response use that fact.*

**Story Generation** HANNA (Chhun et al., 2022) is a benchmark for story generation based on the WritingPrompts dataset. It contains human annotations for 1056 stories across six aspects (detailed in section 3.3.4): *relevance*, *coherence*, *engagement*, *empathy*, *surprise* and *complexity*.

**Data-to-text** Following Zhong et al. (2022), Fu et al. (2024) and Liu and Shen, et al. (2024a), we use two standard datasets for data-to-text meta-evaluation: SFRES and SFHOT (Wen, Gasic, Mrkšić, et al., 2015). These datasets consist of dialogue acts (DAs) in structured format with generated responses providing information about restaurants and hotels in San Francisco. Human judgments are provided for *informativeness* and *naturalness*.

**Text Simplification** To evaluate the fine-tuned model on unseen tasks, we measure its performance on Wiki-DA (Alva-Manchego et al., 2021), a dataset of DA human ratings for text simplification. Along with scores for *fluency*, this dataset also includes two unseen aspects: *meaning preservation* and *simplicity*.

## 4.2 Baselines

We compare our methods with a number of evaluation metrics, including overlap-based and distance-based approaches presented in 2.5.4. Overlap-based metrics include ROUGE (Lin, 2004), BLEU (Papineni et al., 2002) and METEOR (Agarwal and Lavie, 2008). Distance-based metrics are represented by MoverScore (Zhao, Peyrard, et al., 2019) and BERTScore (Zhang, Kishore, et al., 2019), while BARTScore (Yuan et al., 2021) and UniEval (Zhong et al., 2022) represent trained metrics. We also include GPTScore (Fu et al., 2024) and G-Eval (Liu, Iter, et al., 2023) as LLM-based metrics. The specific metrics used differ slightly depending on the dataset.

Additionally, our comparisons include some task-specific and aspect-specific metrics. QAGS (Wang, Cho, et al., 2020) and FactCC (Kryscinski et al., 2020) are trained metrics designed for evaluating factual consistency in summarization.

USR (Mehri and Eskenazi, 2020b) is a trained reference-free metric designed to evaluate dialogue response generation tasks. The metric has several variants; we use the variant with the best Pearson correlation for each aspect in our comparison. We also include two text simplification metrics to compare the performance of the fine-tuned evaluator on unseen task: SARI (Xu, Napoles, et al., 2016) and LENS (Maddela et al., 2022).

To measure the improvement of the fine-tuned model relative to the base model, we include instruction-tuned Llama 3.1 8B as an additional baseline.

## 4.3 Prompt-based Evaluation

To answer **RQ1**, we assess prompting of the ensemble with open-weight LLMs, introduced in Section 3.4.2, as an independent NLG evaluation approach. The prompts described in Section 3.2.2 are applied with minor adjustments for specific tasks. For all experiments, we use the same parameters as during synthetic data generation, described in Section 3.4.2. However, in this case, the ensemble scores are obtained by simple averaging, without removing outliers.

## 4.4 Fine-tuning

To address **RQ2**, we use the instruction-tuned version of Llama 3.1 8B as the backbone for training the distilled NLG evaluator (see Section 3.4). To increase efficiency and explore parameter-efficient fine-tuning methods, we apply LoRA (introduced in Section 2.3.5) to update only the low-rank matrices while keeping the pre-trained weights frozen. The dataset presented in 3.3 is split into training and validation subsets, with 5% of the examples allocated for validation. Since the dataset contains repeated instances of the same inputs and system outputs[1], we sample the validation set to ensure no overlap in inputs between the training and validation sets. This guarantees that during evaluation, the model does not encounter any input it has seen during training. We use the validation set to search for suitable hyperparameters.

We use a LoRA rank of 16, set the LoRA alpha to 32, and apply fine-tuning to all modules of the architecture. The model is trained for one epoch with a batch size of 16 and a learning rate 2e-4. We apply the AdamW optimizer (Loshchilov and Hutter, 2018) with a weight decay of 0.01. A linear learning rate schedule is used, with a warm-up period corresponding to the first 5% of training steps, during which the learning rate increases linearly to its target value. All experiments were performed with the Unsloth[2] library. This leads to a resource-efficient training that requires approximately 20GB of VRAM and completes one training epoch in just six hours on a single A40 GPU.

[1]Here, "inputs" means the original inputs based on which the evaluated system outputs were generated, not the inputs for the evaluator model.

[2]`https://github.com/unslothai/unsloth`

# 5 Results and Analysis

This section presents the results of meta-evaluation on the benchmarks described in Section 4.1. We compare our approach with the baseline metrics presented in Section 4.2. For QAGS, SummEval, and TopicalChat, we use the correlation coefficients reported in Liu, Iter, et al., 2023 for all baseline metrics except Llama 3.1 8B. For SFRES and SFHOT, baseline results are adopted from Liu, Shen, et al., 2024b. We have computed the correlations for the baseline metrics on HANNA based on scores from Chhun et al., 2022. Finally, for text simplification, we use the results for baselines reported in Maddela et al., 2022.

| **Metric** | **QAGS-CNN** | | | **QAGS-XSUM** | | | **Average** | | |
|---|---|---|---|---|---|---|---|---|---|
| | $r$ | $\rho$ | $\tau$ | $r$ | $\rho$ | $\tau$ | $r$ | $\rho$ | $\tau$ |
| ROUGE-1 | 0.338 | 0.318 | 0.248 | -0.008 | -0.049 | -0.040 | 0.165 | 0.134 | 0.104 |
| ROUGE-2 | 0.459 | 0.418 | 0.333 | 0.097 | 0.083 | 0.068 | 0.278 | 0.250 | 0.200 |
| ROUGE-L | 0.357 | 0.324 | 0.254 | 0.024 | -0.011 | -0.009 | 0.190 | 0.156 | 0.122 |
| BERTScore | 0.576 | 0.505 | 0.399 | 0.024 | 0.008 | 0.006 | 0.300 | 0.256 | 0.202 |
| MoverScore | 0.414 | 0.347 | 0.271 | 0.054 | 0.044 | 0.036 | 0.234 | 0.195 | 0.153 |
| FactCC | 0.416 | 0.484 | 0.376 | 0.297 | 0.259 | 0.212 | 0.356 | 0.371 | 0.294 |
| QAGS | 0.545 | - | - | 0.175 | - | - | 0.375 | - | - |
| BARTScore | 0.735 | 0.680 | 0.557 | 0.184 | 0.159 | 0.130 | 0.459 | 0.420 | 0.343 |
| UniEval | 0.682 | 0.662 | 0.532 | 0.461 | 0.488 | 0.399 | 0.571 | 0.575 | 0.465 |
| G-Eval | 0.631 | 0.685 | 0.591 | 0.558 | 0.537 | 0.472 | 0.599 | 0.611 | 0.525 |
| Llama 3.1 8B | 0.275 | 0.242 | 0.219 | 0.218 | 0.230 | 0.218 | 0.247 | 0.236 | 0.219 |
| Gemma | 0.579 | 0.646 | 0.579 | 0.592 | 0.614 | **0.563** | 0.585 | 0.630 | 0.571 |
| Qwen | 0.678 | 0.720 | 0.635 | 0.568 | 0.569 | 0.526 | 0.623 | 0.644 | 0.581 |
| Nemotron | 0.705 | 0.733 | **0.650** | 0.587 | 0.586 | 0.540 | 0.646 | 0.659 | **0.595** |
| Mistral | 0.658 | 0.704 | 0.635 | 0.577 | 0.570 | 0.541 | 0.617 | 0.637 | 0.588 |
| Command R+ | 0.676 | 0.675 | 0.617 | 0.540 | 0.541 | 0.515 | 0.608 | 0.608 | 0.566 |
| Ensemble | **0.738** | **0.753** | 0.627 | **0.630** | **0.624** | 0.531 | **0.684** | **0.689** | 0.579 |
| Llama-FT | 0.606 | 0.640 | 0.532 | 0.604 | 0.604 | 0.521 | 0.605 | 0.622 | 0.527 |

**Table 5.1** Segment-level Pearson ($r$), Spearman ($\rho$) and Kendall ($\tau$) correlations of different metrics for factual consistency on QAGS. The highest values are highlighted in bold.

**Summarization** Meta-evaluation results for QAGS are presented in Table 5.1. Among all compared metrics, our ensemble approach achieves the highest average $r$ and $\rho$ correlations in both the CNN and XSum subsets. While the ensemble slightly lags behind Nemotron or Gemma on $\tau$, this should be interpreted with caution. Unlike the ensemble's averaged floating point scores, individual models provide integer scores between 1 and 5. For this reason, the calculation of $\tau$ might involve a larger number of ties, which are excluded from being counted as concordant or discordant (see Section 2.5.6 for details). As a result, $\rho$ serves as a more reliable indicator of the relative evaluation capability of the ensemble compared to the individual models.

Results for the SummEval dataset are shown in Table 5.2. Here, the ensemble achieves the highest average $\rho$ correlation among all metrics. Examining specific aspects, the ensemble has highest correlations in coherence, where it substantially outperforms all baselines, as well as all of the individual models. In factual consistency, Command R+ and Mistral achieve the best correlations, although the ensemble still remains competitive with both UniEval and G-Eval. The only

| Metric | Consistency | | Coherence | | Relevance | | Fluency | | Average | |
|---|---|---|---|---|---|---|---|---|---|---|
| | $\rho$ | $\tau$ | $\rho$ | $\tau$ | $\rho$ | $\tau$ | $\rho$ | $\tau$ | $\rho$ | $\tau$ |
| ROUGE-1 | 0.167 | 0.126 | 0.160 | 0.130 | 0.115 | 0.094 | 0.326 | 0.252 | 0.192 | 0.150 |
| ROUGE-2 | 0.184 | 0.139 | 0.187 | 0.155 | 0.159 | 0.128 | 0.290 | 0.219 | 0.205 | 0.161 |
| ROUGE-L | 0.128 | 0.099 | 0.115 | 0.092 | 0.105 | 0.084 | 0.311 | 0.237 | 0.165 | 0.128 |
| BERTScore | 0.284 | 0.211 | 0.110 | 0.090 | 0.193 | 0.158 | 0.312 | 0.243 | 0.225 | 0.175 |
| MOVERSscore | 0.159 | 0.118 | 0.157 | 0.127 | 0.129 | 0.105 | 0.318 | 0.244 | 0.191 | 0.148 |
| BARTScore | 0.448 | 0.342 | 0.382 | 0.315 | 0.356 | 0.292 | 0.356 | 0.273 | 0.385 | 0.305 |
| UniEval | 0.575 | 0.442 | 0.446 | 0.371 | 0.449 | 0.371 | 0.426 | 0.325 | 0.474 | 0.377 |
| G-Eval | 0.582 | 0.457 | 0.507 | 0.425 | 0.506 | **0.455** | **0.547** | **0.433** | 0.514 | 0.418 |
| Llama 3.1 8B | 0.179 | 0.163 | 0.176 | 0.150 | 0.167 | 0.136 | 0.225 | 0.211 | 0.187 | 0.165 |
| Nemotron | 0.559 | 0.517 | 0.469 | 0.391 | 0.419 | 0.356 | 0.358 | 0.325 | 0.451 | 0.397 |
| Mistral | 0.627 | 0.590 | 0.528 | 0.434 | 0.456 | 0.375 | 0.398 | 0.359 | 0.502 | **0.439** |
| Qwen | 0.567 | 0.521 | 0.525 | 0.433 | 0.388 | 0.317 | 0.433 | 0.389 | 0.478 | 0.415 |
| Command R+ | **0.633** | **0.603** | 0.239 | 0.203 | 0.360 | 0.302 | 0.347 | 0.320 | 0.395 | 0.357 |
| Gemma | 0.459 | 0.421 | 0.455 | 0.386 | 0.428 | 0.358 | 0.427 | 0.386 | 0.442 | 0.388 |
| Ensemble | 0.548 | 0.470 | **0.604** | **0.462** | **0.513** | 0.389 | 0.470 | 0.389 | **0.534** | 0.427 |
| Llama-FT | 0.503 | 0.433 | 0.518 | 0.412 | 0.448 | 0.360 | 0.358 | 0.311 | 0.457 | 0.379 |

**Table 5.2** Segment-level Spearman ($\rho$) and Kendall ($\tau$) correlations of different metrics on SummEval.

evaluation aspect where G-Eval clearly outperforms our approach is fluency. We hypothesize that this might be caused by the tokenized format of system outputs in SummEval, which might lead some models to over-annotate fluency errors and assign lower scores to these outputs. Although the models are explicitly instructed to ignore such issues when evaluating tokenized outputs, these errors are still often annotated.

The results for Command R+ illustrate the complementary strengths of different LLMs. While this model performs poorly in evaluating the coherence of summaries, it achieves the highest correlations in factual consistency.

Llama-FT, our fine-tuned evaluation model, surpasses all baselines and shows competitive performance compared to individual large prompted LLMs on QAGS. Notably, unlike in most of the other metrics, there is only a minor gap between the results for CNN and XSum subsets. On SummEval, the fine-tuned model is able to outperform almost all baselines and even some larger LLMs, although it ranks below the ensemble and G-Eval. For comparison, we include the results of prompted instruction-tuned Llama 3.1 8B as another baseline. The model performs at a level comparable to the weakest overlap-based metrics, which suggests that LoRA fine-tuning on relatively small synthetic data can significantly reduce the performance gap between smaller LLMs and state-of-the-art open-weight LLMs.

Both ROUGE and the two embedding-based metrics show relatively low correlations with human scores on both benchmarks, which is consistent with our discussion in Section 2.5.4. Trained metrics, particularly UniEval, bridge the gap between traditional metrics and LLM-based approaches.

**Dialogue response generation** The results for the TopicalChat dataset are presented in Table 5.3. The ensemble leads with $r$ and $\rho$ substantially higher than any of the baseline metrics. However, it lags behind Nemotron on groundedness when applied individually, which is probably caused by the weaker performance of certain ensemble models on this aspect, particularly Command R+ and Gemma. This is probably the cause of slightly lower overall correlation with human judg-

| Metric | Groundedness | | Coherence | | Interestingness | | Naturalness | | Average | |
|---|---|---|---|---|---|---|---|---|---|---|
| | $r$ | $\rho$ | $r$ | $\rho$ | $r$ | $\rho$ | $r$ | $\rho$ | $r$ | $\rho$ |
| ROUGE-L | 0.193 | 0.203 | 0.176 | 0.146 | 0.295 | 0.300 | 0.310 | 0.327 | 0.243 | 0.244 |
| BLEU-4 | 0.131 | 0.235 | 0.180 | 0.175 | 0.232 | 0.316 | 0.213 | 0.310 | 0.189 | 0.259 |
| METEOR | 0.250 | 0.302 | 0.212 | 0.191 | 0.367 | 0.439 | 0.333 | 0.391 | 0.290 | 0.331 |
| BERTScore | 0.214 | 0.233 | 0.226 | 0.209 | 0.317 | 0.335 | 0.291 | 0.317 | 0.262 | 0.273 |
| USR | 0.416 | 0.377 | 0.337 | 0.325 | 0.456 | 0.465 | 0.222 | 0.447 | 0.358 | 0.403 |
| UniEval | 0.602 | 0.455 | 0.455 | 0.330 | 0.573 | 0.430 | 0.577 | 0.453 | 0.552 | 0.417 |
| G-Eval | 0.594 | 0.605 | 0.549 | 0.565 | 0.627 | 0.631 | 0.531 | 0.551 | 0.575 | 0.588 |
| Llama 3.1 8B | 0.373 | 0.365 | 0.233 | 0.245 | 0.427 | 0.421 | 0.220 | 0.264 | 0.313 | 0.324 |
| Nemotron | **0.781** | **0.791** | 0.600 | **0.630** | 0.655 | 0.686 | 0.506 | 0.532 | 0.621 | 0.645 |
| Mistral | 0.658 | 0.648 | 0.541 | 0.554 | 0.645 | 0.659 | 0.509 | 0.529 | 0.586 | 0.596 |
| Qwen | 0.467 | 0.460 | 0.480 | 0.521 | 0.500 | 0.516 | 0.468 | 0.488 | 0.496 | 0.514 |
| Command R+ | 0.383 | 0.368 | 0.463 | 0.453 | 0.262 | 0.259 | 0.421 | 0.398 | 0.386 | 0.374 |
| Gemma | 0.332 | 0.366 | 0.465 | 0.481 | 0.549 | 0.562 | 0.489 | 0.515 | 0.459 | 0.484 |
| Ensemble | 0.704 | 0.697 | **0.622** | 0.621 | **0.675** | **0.692** | **0.599** | **0.604** | **0.649** | **0.653** |
| Llama-FT | 0.464 | 0.508 | 0.516 | 0.549 | 0.604 | 0.623 | 0.496 | 0.503 | 0.520 | 0.546 |

**Table 5.3** Segment-level Pearson ($r$) and Spearman ($\rho$) correlations of different metrics on TopicalChat.

| Metric | SFRES | | SFHOT | | Average |
|---|---|---|---|---|---|
| | Inf. | Nat. | Inf. | Nat. | |
| ROUGE-1 | 0.115 | 0.170 | 0.118 | 0.196 | 0.150 |
| ROUGE-L | 0.103 | 0.169 | 0.110 | 0.186 | 0.142 |
| BERTScore | 0.156 | 0.219 | 0.135 | 0.178 | 0.172 |
| MOVERScore | 0.153 | 0.190 | 0.172 | 0.242 | 0.189 |
| BARTScore | 0.238 | 0.289 | 0.235 | 0.288 | 0.263 |
| UniEval | 0.225 | 0.333 | 0.249 | 0.320 | 0.282 |
| GPTScore | 0.232 | 0.190 | 0.184 | 0.036 | 0.161 |
| G-Eval | 0.189 | 0.351 | 0.198 | 0.338 | 0.269 |
| Llama 3.1 8B | 0.087 | 0.032 | 0.090 | 0.090 | 0.060 |
| Gemma | 0.254 | 0.334 | **0.275** | 0.317 | 0.295 |
| Qwen | 0.226 | 0.347 | 0.245 | 0.315 | 0.283 |
| Nemotron | 0.178 | 0.284 | 0.226 | 0.281 | 0.242 |
| Mistral | 0.129 | 0.359 | 0.215 | 0.311 | 0.254 |
| Command R+ | 0.236 | 0.299 | 0.047 | 0.210 | 0.198 |
| Ensemble | 0.234 | **0.415** | 0.205 | **0.341** | **0.299** |
| Llama-FT | **0.279** | 0.345 | 0.224 | 0.311 | 0.290 |

**Table 5.4** Segment-level Spearman ($r$) correlations of different metrics on SFRES and SFHOT. **Inf.** = informativeness, **Nat.** = naturalness.

ments on this aspect.

Llama-FT shows strong performance on interestingness and naturalness, and even outperforms three individual LLMs. On groundedness, which is evaluated as an unseen aspect, Llama-FT shows considerable improvement over the base model and achieves better correlation with human scores than Gemma, Command R+ and Qwen. These results indicate that including a diverse range of evaluation aspects in the training data can improve the model's generalization to new aspects. Additionally, compared to the larger prompted LLMs, the fine-tuned model shows competitive performance on coherence, interestingness, and naturalness on TopicalChat.

**Data-to-text** Table 5.4 shows the results for data-to-task evaluation on an unseen domain. Our approach achieves the highest averaged $\rho$ correlation on SFHOT and SFRES, although it performs better on evaluating naturalness of

| Metric | Coh. | Rel. | Eng. | Emp. | Sur. | Com. | Avg. |
|---|---|---|---|---|---|---|---|
| BLEU | 0.539 | 0.514 | 0.483 | 0.410 | 0.471 | 0.516 | 0.489 |
| ROUGE-1 | **0.567** | 0.518 | 0.529 | **0.450** | **0.490** | **0.591** | **0.524** |
| METEOR | 0.560 | 0.522 | 0.510 | 0.435 | 0.488 | 0.555 | 0.512 |
| MoverScore | 0.551 | 0.523 | 0.495 | 0.418 | 0.478 | 0.530 | 0.499 |
| BERTScore | 0.566 | 0.531 | 0.520 | 0.441 | 0.488 | 0.563 | 0.518 |
| BARTScore | 0.501 | 0.467 | 0.465 | 0.416 | 0.436 | 0.488 | 0.462 |
| Llama 3.1 8B | 0.119 | 0.344 | 0.154 | 0.094 | 0.080 | 0.212 | 0.167 |
| Nemotron | 0.500 | 0.508 | 0.497 | 0.380 | 0.332 | 0.565 | 0.464 |
| Mistral | 0.372 | 0.490 | 0.425 | 0.373 | 0.334 | 0.507 | 0.417 |
| Qwen | 0.407 | 0.484 | 0.427 | 0.277 | -0.012 | 0.507 | 0.348 |
| Command R+ | 0.412 | 0.383 | 0.420 | 0.330 | 0.227 | 0.344 | 0.353 |
| Gemma | 0.453 | 0.445 | 0.461 | 0.356 | 0.323 | 0.521 | 0.427 |
| Ensemble | 0.528 | **0.559** | **0.538** | 0.434 | 0.343 | **0.591** | 0.499 |
| Llama-FT | 0.286 | 0.468 | 0.420 | 0.348 | 0.329 | 0.523 | 0.396 |

**Table 5.5** Segment-level Pearson ($r$) correlations of various metrics on HANNA. **Coh.** = coherence, **Rel.** = relevance, **Eng.** = engagement, **Emp.** = empathy, **Sur.** = surprise, **Com.** = complexity, **Avg.** = average.

| Metric | Coh. | Rel. | Eng. | Emp. | Sur. | Com. | Avg. |
|---|---|---|---|---|---|---|---|
| BLEU | 0.339 | 0.292 | 0.356 | 0.315 | 0.299 | 0.414 | 0.336 |
| ROUGE-1 | 0.389 | 0.330 | 0.416 | 0.354 | **0.355** | 0.503 | 0.391 |
| METEOR | 0.378 | 0.310 | 0.412 | 0.366 | 0.354 | **0.505** | 0.387 |
| MoverScore | 0.392 | 0.385 | 0.420 | 0.331 | 0.321 | 0.473 | 0.387 |
| BERTScore | 0.372 | 0.355 | 0.415 | 0.356 | 0.320 | 0.469 | 0.381 |
| BARTScore | 0.259 | 0.249 | 0.291 | 0.287 | 0.227 | 0.294 | 0.268 |
| Llama 3.1 8B | 0.093 | 0.334 | 0.140 | 0.086 | 0.064 | 0.184 | 0.150 |
| Nemotron | **0.429** | 0.445 | 0.427 | 0.328 | 0.263 | 0.464 | 0.393 |
| Mistral | 0.294 | 0.415 | 0.389 | 0.349 | 0.337 | 0.474 | 0.376 |
| Qwen | 0.344 | 0.423 | 0.364 | 0.262 | -0.006 | 0.416 | 0.301 |
| Command R+ | 0.325 | 0.362 | 0.372 | 0.320 | 0.215 | 0.348 | 0.324 |
| Gemma | 0.381 | 0.383 | 0.400 | 0.310 | 0.278 | 0.445 | 0.366 |
| Ensemble | 0.393 | **0.474** | **0.452** | **0.367** | 0.276 | 0.489 | **0.409** |
| Llama-FT | 0.337 | 0.467 | 0.412 | 0.335 | 0.281 | 0.430 | 0.377 |

**Table 5.6** Segment-level Spearman correlations ($\rho$) of various metrics on HANNA. **Coh.** = coherence, **Rel.** = relevance, **Eng.** = engagement, **Emp.** = empathy, **Sur.** = surprise, **Com.** = complexity, **Avg.** = average.

generated texts than informativeness. Llama-FT closely follows the ensemble in average $\rho$, showing a substantial improvement over the base model. It even ranks first in informativeness on the SFRES dataset. With the exception of BARTScore and G-Eval, the baseline metrics rank considerably lower. Overall, the correlations for both datasets are relatively low across all metrics, which may point to potential noise in the human annotations.

**Story generation** Tables 5.5, 5.6 and 5.7 show the results for $r$, $\rho$ and $\tau$ on the HANNA benchmark, respectively. Our ensemble approach achieves the highest average $\rho$ and surpasses all baselines on four of the six aspects. Llama-FT closely approaches the ensemble in terms of $\rho$ and $\tau$. It also outperforms the older metrics in $\tau$, but not in the other two coefficients. However, there is a considerable improvement over a base prompted Llama 8B, which ranks the lowest among all compared metrics.

Interestingly, ROUGE shows the highest average Pearson correlation, and

| Metric | Coh. | Rel. | Eng. | Emp. | Sur. | Com. | Avg. |
|---|---|---|---|---|---|---|---|
| BLEU | 0.248 | 0.209 | 0.260 | 0.230 | 0.220 | 0.305 | 0.245 |
| ROUGE-1 | 0.287 | 0.237 | 0.306 | 0.260 | 0.262 | 0.376 | 0.288 |
| METEOR | 0.278 | 0.224 | 0.303 | 0.269 | 0.261 | 0.377 | 0.285 |
| MoverScore | 0.289 | 0.280 | 0.308 | 0.242 | 0.236 | 0.353 | 0.285 |
| BERTScore | 0.273 | 0.257 | 0.304 | 0.260 | 0.234 | 0.348 | 0.279 |
| BARTScore | 0.185 | 0.177 | 0.209 | 0.206 | 0.164 | 0.212 | 0.192 |
| Llama 3.1 8B | 0.079 | 0.285 | 0.116 | 0.073 | 0.055 | 0.152 | 0.127 |
| Nemotron | **0.369** | **0.377** | **0.364** | 0.272 | 0.221 | 0.388 | **0.332** |
| Mistral | 0.253 | 0.354 | 0.334 | 0.301 | **0.284** | **0.405** | 0.322 |
| Qwen | 0.294 | 0.360 | 0.307 | 0.222 | -0.003 | 0.353 | 0.255 |
| Command R+ | 0.270 | 0.297 | 0.300 | 0.253 | 0.171 | 0.277 | 0.261 |
| Gemma | 0.329 | 0.328 | 0.343 | 0.267 | 0.241 | 0.384 | 0.315 |
| Ensemble | 0.307 | 0.367 | 0.350 | **0.280** | 0.208 | 0.378 | 0.315 |
| Llama-FT | 0.279 | 0.374 | 0.337 | 0.276 | 0.228 | 0.352 | 0.308 |

**Table 5.7** Segment-level Kendall ($\tau$) correlations of various metrics on HANNA. **Coh.** = coherence, **Rel.** = relevance, **Eng.** = engagement, **Emp.** = empathy, **Sur.** = surprise, **Com.** = complexity, **Avg.** = average.

| Metric | Fluency | Meaning | Simplicity | Average |
|---|---|---|---|---|
| BLEU | 0.460 | 0.622 | 0.438 | 0.507 |
| SARI | 0.335 | 0.534 | 0.366 | 0.412 |
| BERTScore | 0.636 | 0.682 | 0.614 | 0.644 |
| LENS | 0.816 | 0.662 | 0.733 | 0.737 |
| Llama 3.1 8B | 0.337 | 0.450 | 0.295 | 0.361 |
| Gemma | 0.755 | 0.769 | 0.688 | 0.737 |
| Qwen | 0.771 | 0.829 | 0.730 | 0.776 |
| Nemotron | 0.778 | 0.822 | 0.660 | 0.753 |
| Mistral | 0.705 | 0.744 | 0.735 | 0.728 |
| Command R+ | 0.704 | 0.787 | 0.601 | 0.697 |
| Ensemble | **0.840** | **0.864** | **0.770** | **0.825** |
| Llama-FT | 0.781 | 0.823 | 0.712 | 0.772 |

**Table 5.8** Segment-level Pearson ($r$) correlations of different metrics on Wiki-DA.

the highest Spearman correlation on surprise and complexity. Other overlap-based and distance-based metrics achieve comparably high scores. This result is unusual, given that these metrics substantially lag behind both trained and LLM-based metrics in other tasks. However, it should be emphasized that our metrics evaluated the stories without any reference texts.

**Text Simplification** The results for text simplification are presented in Table 5.8. Our ensemble shows superior performance across all aspects, with an average $r$ of 0.825. It outperforms LENS, a strong baseline metric, on all evaluated aspects. Although this is not the case for all individual LLMs—LENS outperforms each individual LLM on fluency, and four of them on simplicity—their combination in the ensemble achieves the best results both on average and for each aspect separately. Llama-FT, although not originally trained on the task, outperforms all baseline metrics in average correlation, and achieves a particularly high score on meaning preservation.

# 6 Conclusion

In this thesis, we have developed an LLM-based approach to explainable NLG evaluation, introducing two methods: an ensemble approach using large open-weight LLMs and a specialized evaluator trained by distillation from larger models. Through the ensemble, we have constructed a balanced synthetic dataset consisting of fine-grained evaluations across five task categories, including a number of subtasks, evaluation aspects, and systems. The evaluations in the dataset provide detailed error analyses that enhance the interpretability of the evaluation process, and serve as a chain-of-thought mechanism that helps to ground the model's decisions. Using this dataset, we applied LoRA to train an evaluator based on the Llama 3.1 8B model. This chapter concludes the thesis by summarizing our findings in Section 6.1 and outlining directions for future work in Section 6.2.

## 6.1 Discussion

Our meta-evaluation results demonstrate that the ensemble approach achieves strong correlations with human judgments, outperforming both traditional metrics, and trained or LLM-based baseline metrics across all evaluated tasks. This indicates that open-weight LLMs, even in their quantized versions, can be applied to build a robust, transparent and reproducible alternative to proprietary models for NLG evaluation. One limitation of this approach is the need to run several different LLMs simultaneously. However, this could be sidestepped by running the models in sequence at the cost of increased latency.

Our fine-tuned evaluator shows competitive performance, substantially improving upon the backbone model. It generally surpasses all traditional metrics, and for some tasks, also outperforms the trained metrics. While it exceeds the performance of some individual prompted LLMs in each task, it does not match the performance of the ensemble. Given the large scope of the task, we hypothesize that the performance could be further improved by training on larger dataset. Our manual inspection of the model's output reveals a tendency to over-annotate the errors for some tasks, though this does not dramatically affect the scores.

Nevertheless, the results demonstrate the fine-tuned model's ability to generalize across unseen tasks, domains and evaluation aspects. Notably, it outperforms the state-of-the-art LENS metric on text simplification, and approaches the ensemble's average correlation on this task. Compared to baselines, prompted large LLMs, and the ensemble, the model achieves strong correlations on unseen domains in the data-to-text task, although the correlations are generally low for the two datasets. Additionally, it shows the ability to generalize to unseen aspects in the dialogue response generation task, and particularly on text simplification task. These findings emphasize the importance of curated and balanced datasets that represent diverse evaluation scenarios. The results also suggest that LoRA is a promising technique for fine-tuning LLM-based NLG evaluation models, enabling training on 37,000 training examples in just six hours using a single GPU.

## 6.2 Future Work

While our ensemble approach demonstrates strong correlations with human judgments in evaluating generated text, this thesis explores only one ensemble variant: ensemble after inference. Future work could investigate alternative methods for combining multiple LLMs in the context of NLG evaluation, including different cooperation strategies (cf. Lu, Pang, et al., 2024).

This thesis focused on the error analysis primarily as a means to enhance interpretability, and as a reasoning process to ground the overall assessment of the evaluated text. Future research could direct more attention on the precision of error spans, including systematic comparisons with human-annotated error analyses.

Although the prompt design used in our approach builds on existing empirically tested methodologies, more ablation experiments should be carried out to better understand specific contributions of different prompt components.

Our fine-tuned model showed promising performance in terms of its correlation with human scores, though we have observed a tendency to over-annotate the errors in the outputs. While the cause of this is not clear, it could be attributed to the structure of the ensemble outputs after unification. Detailed data exploration and further refinement of the annotation merging process could address this issue. Additionally, other merging strategies could be explored, including more advanced rule-based preprocessing before LLM inference, or multi-step prompting techniques.

Since LLMs are applicable to a diverse range of tasks—including coding, mathematical problem solving, and instruction following—expanding our approach to more NLG tasks would be a logical direction for future work.

# Bibliography


Achiam, Josh et al. (2023). "Gpt-4 technical report". In: *arXiv preprint arXiv:2303.08774*.

Agarwal, Abhaya and Alon Lavie (2008). "Meteor, m-bleu and m-ter: Evaluation metrics for high-correlation with human rankings of machine translation output". In: *Proceedings of the Third Workshop on Statistical Machine Translation*, pp. 115–118.

Alva-Manchego, Fernando, Carolina Scarton, and Lucia Specia (2021). "The (un) suitability of automatic evaluation metrics for text simplification". In: *Computational Linguistics* 47.4, pp. 861–889.

Amidei, Jacopo, Paul Piwek, and Alistair Willis (2019). "Agreement is overrated: A plea for correlation to assess human evaluation reliability". In: *Proceedings of the 12th International Conference on Natural Language Generation*, pp. 344–354.

Ba, Jimmy Lei, Jamie Ryan Kiros, and Geoffrey E. Hinton (2016). "Layer normalization". In: *ArXiv e-prints*, arXiv–1607.

Banerjee, Satanjeev and Alon Lavie (2005). "METEOR: An automatic metric for MT evaluation with improved correlation with human judgments". In: *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pp. 65–72.

Barrios, Federico et al. (2016). "Variations of the similarity function of textrank for automated summarization". In: *arXiv preprint arXiv:1602.03606*.

Belz, Anja, Simon Mille, and David M Howcroft (2020). "Disentangling the Properties of Human Evaluation Methods: A Classification System to Support Comparability, Meta-Evaluation and Reproducibility Testing". In: *Proceedings of the 13th International Conference on Natural Language Generation*, pp. 183–194.

Belz, Anja and Ehud Reiter (2006). "Comparing automatic and human evaluation of NLG systems". In: *11th conference of the european chapter of the association for computational linguistics*, pp. 313–320.

Brown, Tom et al. (2020). "Language models are few-shot learners". In: *Advances in neural information processing systems* 33, pp. 1877–1901.

Cao, Meng, Yue Dong, and Jackie Chi Kit Cheung (2021). "Hallucinated but factual! inspecting the factuality of hallucinations in abstractive summarization". In: *arXiv preprint arXiv:2109.09784*.

Celikyilmaz, Asli, Elizabeth Clark, and Jianfeng Gao (2020). "Evaluation of text generation: A survey". In: *arXiv preprint arXiv:2006.14799*.

Chen, Lingjiao, Matei Zaharia, and James Zou (2023). "How is ChatGPT's behavior changing over time?" In: *arXiv preprint arXiv:2307.09009*.

Chen, Mingje, Gerasimos Lampouras, and Andreas Vlachos (2018). "Sheffield at e2e: structured prediction approaches to end-to-end language generation". In: *E2E NLG Challenge System Descriptions* 85.

Chen, Wenhu, Jianshu Chen, et al. (2020). "Logical natural language generation from open-domain tables". In: *arXiv preprint arXiv:2004.10404*.

Chhun, Cyril et al. (2022). "Of human criteria and automatic metrics: A benchmark of the evaluation of story generation". In: *arXiv preprint arXiv:2208.11646*.

Chiang, Cheng-Han and Hung-Yi Lee (2023). "Can Large Language Models Be an Alternative to Human Evaluations?" In: *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15607–15631.

Chiang, Wei-Lin, Lianmin Zheng, et al. (2024). "Chatbot arena: An open platform for evaluating llms by human preference". In: *arXiv preprint arXiv:2403.04132*.

Cho, Kyunghyun et al. (2014). "Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation". In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1724–1734.

Chung, Hyung Won et al. (2024). "Scaling instruction-finetuned language models". In: *Journal of Machine Learning Research* 25.70, pp. 1–53.

Cicchetti, Domenic V, Donald Shoinralter, and Peter J Tyrer (1985). "The effect of number of rating scale categories on levels of interrater reliability: A Monte Carlo investigation". In: *Applied Psychological Measurement* 9.1, pp. 31–36.

Cohen, Jacob (1960). "A coefficient of agreement for nominal scales". In: *Educational and psychological measurement* 20.1, pp. 37–46.

Cohere For AI (2024). *c4ai-command-r-plus-08-2024 (Revision dfda5ab)*. DOI: 10.57967/hf/3136. URL: https://huggingface.co/CohereForAI/c4ai-command-r-plus-08-2024.

Dettmers, Tim, Mike Lewis, et al. (2022). "Gpt3. int8 (): 8-bit matrix multiplication for transformers at scale". In: *Advances in Neural Information Processing Systems* 35, pp. 30318–30332.

Dettmers, Tim, Artidoro Pagnoni, et al. (2024). "Qlora: Efficient finetuning of quantized llms". In: *Advances in Neural Information Processing Systems* 36.

Dettmers, Tim and Luke Zettlemoyer (2023). "The case for 4-bit precision: k-bit inference scaling laws". In: *International Conference on Machine Learning*. PMLR, pp. 7750–7774.

Deutsch, Daniel, George Foster, and Markus Freitag (2023). "Ties matter: Meta-evaluating modern metrics with pairwise accuracy and tie calibration". In: *arXiv preprint arXiv:2305.14324*.

Devlin, Jacob (2018). "Bert: Pre-training of deep bidirectional transformers for language understanding". In: *arXiv preprint arXiv:1810.04805*.

Dinan, Emily et al. (2018). "Wizard of wikipedia: Knowledge-powered conversational agents". In: *arXiv preprint arXiv:1811.01241*.

Dong, Li et al. (2019). "Unified language model pre-training for natural language understanding and generation". In: *Advances in neural information processing systems* 32.

Doshi-Velez, Finale and Been Kim (2017). "Towards a rigorous science of interpretable machine learning". In: *arXiv preprint arXiv:1702.08608*.

Dou, Yao et al. (2021). "Is GPT-3 text indistinguishable from human text? scarecrow: A framework for scrutinizing machine text". In: *arXiv preprint arXiv:2107.01294*.

Duan, Jinhao et al. (2023). "Shifting attention to relevance: Towards the uncertainty estimation of large language models". In: *arXiv preprint arXiv:2307.01379*.

Dubey, Abhimanyu et al. (2024). "The llama 3 herd of models". In: *arXiv preprint arXiv:2407.21783*.

Dušek, Ondřej, Jekaterina Novikova, and Verena Rieser (2020). "Evaluating the state-of-the-art of end-to-end natural language generation: The e2e nlg challenge". In: *Computer Speech & Language* 59, pp. 123–156.

Elder, Henry et al. (Nov. 2018). "E2E NLG Challenge Submission: Towards Controllable Generation of Diverse Natural Language". In: *Proceedings of the 11th International Conference on Natural Language Generation*. Ed. by Krahmer, Emiel, Gatt, Albert, and Goudbeek, Martijn. Tilburg University, The Netherlands: Association for Computational Linguistics, pp. 457–462. DOI: 10.18653/v1/W18-6556. URL: https://aclanthology.org/W18-6556/.

Fabbri, Alexander, Faiaz Rahman, et al. (Aug. 2021). "ConvoSumm: Conversation Summarization Benchmark and Improved Abstractive Summarization with Argument Mining". In: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Ed. by Zong, Chengqing et al. Online: Association for Computational Linguistics, pp. 6866–6880. DOI: 10.18653/v1/2021.acl-long.535. URL: https://aclanthology.org/2021.acl-long.535/.

Fabbri, Alexander R, Wojciech Kryściński, et al. (2021). "Summeval: Re-evaluating summarization evaluation". In: *Transactions of the Association for Computational Linguistics* 9, pp. 391–409.

Fan, Angela, Mike Lewis, and Yann Dauphin (2018). "Hierarchical neural story generation". In: *arXiv preprint arXiv:1805.04833*.

Fernandes, Patrick et al. (2023). "The devil is in the errors: Leveraging large language models for fine-grained machine translation evaluation". In: *arXiv preprint arXiv:2308.07286*.

Ferreira, Thiago Castro et al. (2020). "The 2020 Bilingual, Bi-Directional WebNLG+ Shared Task: Overview and Evaluation Results (WebNLG+ 2020)". In: *Proceedings of the 3rd International Workshop on Natural Language Generation from the Semantic Web (WebNLG+)*, pp. 55–76.

Fleiss, Joseph L (1971). "Measuring nominal scale agreement among many raters." In: *Psychological bulletin* 76.5, p. 378.

Freitag, Markus, George Foster, et al. (2021). "Experts, errors, and context: A large-scale study of human evaluation for machine translation". In: *Transactions of the Association for Computational Linguistics* 9, pp. 1460–1474.

Freitag, Markus, Ricardo Rei, et al. (2022). "Results of WMT22 metrics shared task: Stop using BLEU–neural metrics are better and more robust". In: *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pp. 46–68.

Fu, Jinlan et al. (2024). "GPTScore: Evaluate as You Desire". In: *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 6556–6576.

Gao, Leo, Stella Biderman, et al. (2020). "The Pile: An 800GB Dataset of Diverse Text for Language Modeling". In: *arXiv preprint arXiv:2101.00027*.

Gao, Mingqi, Xinyu Hu, et al. (2024). "Analyzing and Evaluating Correlation Measures in NLG Meta-Evaluation". In: *arXiv preprint arXiv:2410.16834*.

Gao, Mingqi and Xiaojun Wan (2022). "DialSummEval: Revisiting summarization evaluation for dialogues". In: *Proceedings of the 2022 Conference of the North*

*American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 5693–5709.

Gatt, Albert and Emiel Krahmer (2018). "Survey of the state of the art in natural language generation: Core tasks, applications and evaluation". In: *Journal of Artificial Intelligence Research* 61, pp. 65–170.

Gehring, Jonas et al. (2017). "Convolutional sequence to sequence learning". In: *International conference on machine learning*. PMLR, pp. 1243–1252.

Gehrmann, Sebastian, Elizabeth Clark, and Thibault Sellam (2023). "Repairing the cracked foundation: A survey of obstacles in evaluation practices for generated text". In: *Journal of Artificial Intelligence Research* 77, pp. 103–166.

Gemma Team, Gemma Team et al. (2024). "Gemma 2: Improving open language models at a practical size". In: *arXiv preprint arXiv:2408.00118*.

Gliwa, Bogdan et al. (2019). "SAMSum Corpus: A Human-annotated Dialogue Dataset for Abstractive Summarization". In: *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, pp. 70–79.

Goldfarb-Tarrant, Seraphina et al. (2020). "Content Planning for Neural Story Generation with Aristotelian Rescoring". In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 4319–4338.

Goodfellow, Ian, Yoshua Bengio, and Aaron Courville (2016). *Deep Learning*. `http://www.deeplearningbook.org`. MIT Press.

Gopalakrishnan, Karthik et al. (2023). "Topical-chat: Towards knowledge-grounded open-domain conversations". In: *arXiv preprint arXiv:2308.11995*.

Goyal, Tanya, Junyi Jessy Li, and Greg Durrett (2022). "Snac: Coherence error detection for narrative summarization". In: *arXiv preprint arXiv:2205.09641*.

Graham, Yvette et al. (2013). "Continuous measurement scales in human evaluation of machine translation". In: *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pp. 33–41.

Grusky, Max, Mor Naaman, and Yoav Artzi (2018). "Newsroom: A Dataset of 1.3 Million Summaries with Diverse Extractive Strategies". In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. Association for Computational Linguistics.

Gu, Albert and Tri Dao (2023). "Mamba: Linear-time sequence modeling with selective state spaces". In: *arXiv preprint arXiv:2312.00752*.

Guan, Jian, Fei Huang, et al. (2020). "A Knowledge-Enhanced Pretraining Model for Commonsense Story Generation". In: *Transactions of the Association for Computational Linguistics* 8, pp. 93–108.

Guan, Jian, Zhexin Zhang, et al. (2021). "OpenMEVA: A benchmark for evaluating open-ended story generation metrics". In: *arXiv preprint arXiv:2105.08920*.

Gulçehre, Çağlar et al. (2016). "Pointing the Unknown Words". In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 140–149.

Guo, Qipeng et al. (Dec. 2020). "$\mathcal{P}^2$: A Plan-and-Pretrain Approach for Knowledge Graph-to-Text Generation". In: *Proceedings of the 3rd International Workshop on Natural Language Generation from the Semantic Web (WebNLG+)*. Ed. by Castro Ferreira, Thiago et al. Dublin, Ireland (Virtual): Association for

Computational Linguistics, pp. 100–106. URL: https://aclanthology.org/2020.webnlg-1.10/.

Gupta, Prakhar et al. (2019). "Investigating Evaluation of Open-Domain Dialogue Systems With Human Generated Multiple References". In: *Proceedings of the 20th Annual SIGdial Meeting on Discourse and Dialogue*, pp. 379–391.

Hastie, Helen and Anja Belz (2014). "A comparative evaluation methodology for NLG in interactive systems". In: *Proceedings of LREC'14*. European Language Resources Association, pp. 4004–4011.

He, Kaiming et al. (2016). "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778.

Hendrycks, Dan and Kevin Gimpel (2016). "Gaussian error linear units (gelus)". In: *arXiv preprint arXiv:1606.08415*.

Hermann, Karl Moritz et al. (2015). "Teaching machines to read and comprehend". In: *Advances in neural information processing systems* 28.

Houlsby, Neil et al. (2019). "Parameter-efficient transfer learning for NLP". In: *International conference on machine learning*. PMLR, pp. 2790–2799.

Howcroft, David M et al. (2020). "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, pp. 169–182.

Hu, Edward J, Yelong Shen, et al. (2021). "Lora: Low-rank adaptation of large language models". In: *arXiv preprint arXiv:2106.09685*.

Hu, Xinyu, Mingqi Gao, et al. (2024). "Are LLM-based Evaluators Confusing NLG Quality Criteria?" In: *arXiv preprint arXiv:2402.12055*.

Huang, Lishan et al. (2020). "GRADE: Automatic Graph-Enhanced Coherence Metric for Evaluating Open-Domain Dialogue Systems". In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 9230–9240.

Jacob, Benoit et al. (2018). "Quantization and training of neural networks for efficient integer-arithmetic-only inference". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2704–2713.

Jhamtani, Harsh and Taylor Berg-Kirkpatrick (2020). "Narrative Text Generation with a Latent Discrete Plan". In: *Findings of the Association for Computational Linguistics: EMNLP 2020*, pp. 3637–3650.

Jiang, Dongfu, Yishan Li, et al. (2023). "Tigerscore: Towards building explainable metric for all text generation tasks". In: *Transactions on Machine Learning Research*.

Jiang, Zhengbao, Yi Mao, et al. (July 2022). "OmniTab: Pretraining with Natural and Synthetic Data for Few-shot Table-based Question Answering". In: *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Ed. by Carpuat, Marine, Marneffe, Marie-Catherine de, and Meza Ruiz, Ivan Vladimir. Seattle, United States: Association for Computational Linguistics, pp. 932–942. DOI: 10.18653/v1/2022.naacl-main.68. URL: https://aclanthology.org/2022.naacl-main.68/.

Juraska, Juraj et al. (June 2018). "A Deep Ensemble Model with Slot Alignment for Sequence-to-Sequence Natural Language Generation". In: *Proceedings of*

*the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. Ed. by Walker, Marilyn, Ji, Heng, and Stent, Amanda. New Orleans, Louisiana: Association for Computational Linguistics, pp. 152–162. DOI: 10.18653/v1/N18-1014. URL: https://aclanthology.org/N18-1014/.

Kadavath, Saurav et al. (2022). "Language models (mostly) know what they know". In: *arXiv preprint arXiv:2207.05221*.

Karpinska, Marzena, Nader Akoury, and Mohit Iyyer (2021). "The perils of using Mechanical Turk to evaluate open-ended text generation". In: *arXiv preprint arXiv:2109.06835*.

Kasner, Zdeněk and Ondřej Dušek (2024). "Beyond Traditional Benchmarks: Analyzing Behaviors of Open LLMs on Data-to-Text Generation". In: *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12045–12072.

Kim, Sanghoon, Dahyun Kim, et al. (June 2024). "SOLAR 10.7B: Scaling Large Language Models with Simple yet Effective Depth Up-Scaling". In: *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 6: Industry Track)*. Ed. by Yang, Yi et al. Mexico City, Mexico: Association for Computational Linguistics, pp. 23–35. DOI: 10.18653/v1/2024.naacl-industry.3. URL: https://aclanthology.org/2024.naacl-industry.3/.

Kim, Seungone, Jamin Shin, et al. (2024). "Prometheus: Inducing fine-grained evaluation capability in language models". In: *The Twelfth International Conference on Learning Representations*.

Kim, Seungone, Juyoung Suk, et al. (2024). "Prometheus 2: An open source language model specialized in evaluating other language models". In: *arXiv preprint arXiv:2405.01535*.

Kingma, Diederik P. and Jimmy Ba (2014). "Adam: A method for stochastic optimization". In: *arXiv preprint arXiv:1412.6980*.

Knowles, Rebecca and Chi-kiu Lo (2024). "Calibration and context in human evaluation of machine translation". In: *Natural Language Processing*, pp. 1–25.

Kočiskỳ, Tomáš et al. (2018). "The NarrativeQA Reading Comprehension Challenge". In: *Transactions of the Association for Computational Linguistics* 6, pp. 317–328.

Kocmi, Tom and Christian Federmann (2023a). "GEMBA-MQM: Detecting translation quality error spans with GPT-4". In: *arXiv preprint arXiv:2310.13988*.

— (2023b). "Large language models are state-of-the-art evaluators of translation quality". In: *arXiv preprint arXiv:2302.14520*.

Kocmi, Tom, Vilém Zouhar, et al. (2024). "Error Span Annotation: A Balanced Approach for Human Evaluation of Machine Translation". In: *arXiv preprint arXiv:2406.11580*.

Kojima, Takeshi et al. (2022). "Large language models are zero-shot reasoners". In: *Advances in neural information processing systems* 35, pp. 22199–22213.

Kreutzer, Julia, Joshua Uyheng, and Stefan Riezler (2018). "Reliability and learnability of human bandit feedback for sequence-to-sequence reinforcement learning". In: *arXiv preprint arXiv:1805.10627*.

Kryscinski, Wojciech et al. (Nov. 2020). "Evaluating the Factual Consistency of Abstractive Text Summarization". In: *Proceedings of the 2020 Conference*

*on Empirical Methods in Natural Language Processing (EMNLP)*. Ed. by Webber, Bonnie et al. Online: Association for Computational Linguistics, pp. 9332–9346. DOI: 10.18653/v1/2020.emnlp-main.750. URL: https://aclanthology.org/2020.emnlp-main.750/.

Kryściński, Wojciech et al. (2019). "Neural text summarization: A critical evaluation". In: *arXiv preprint arXiv:1908.08960*.

Kusner, Matt et al. (July 2015). "From Word Embeddings To Document Distances". In: *Proceedings of the 32nd International Conference on Machine Learning*. Ed. by Bach, Francis and Blei, David. Vol. 37. Proceedings of Machine Learning Research. Lille, France: PMLR, pp. 957–966. URL: https://proceedings.mlr.press/v37/kusnerb15.html.

Lambert, Nathan et al. (2024). "T\" ULU 3: Pushing Frontiers in Open Language Model Post-Training". In: *arXiv preprint arXiv:2411.15124*.

Lapalme, Guy (Dec. 2020). "RDFjsRealB: a Symbolic Approach for Generating Text from RDF Triples". In: *Proceedings of the 3rd International Workshop on Natural Language Generation from the Semantic Web (WebNLG+)*. Ed. by Castro Ferreira, Thiago et al. Dublin, Ireland (Virtual): Association for Computational Linguistics, pp. 144–153. URL: https://aclanthology.org/2020.webnlg-1.16/.

Lei, Tao, Regina Barzilay, and Tommi Jaakkola (2016). "Rationalizing Neural Predictions". In: *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 107–117.

Leiter, Christoph et al. (2022). "Towards explainable evaluation metrics for natural language generation". In: *arXiv preprint arXiv:2203.11131*.

Lester, Brian, Rami Al-Rfou, and Noah Constant (2021). "The Power of Scale for Parameter-Efficient Prompt Tuning". In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 3045–3059.

Lewis, Mike (2019). "Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension". In: *arXiv preprint arXiv:1910.13461*.

Li, Qintong, Leyang Cui, et al. (2023). "Collaborative Evaluation: Exploring the Synergy of Large Language Models and Humans for Open-ended Generation Evaluation". In: *arXiv preprint arXiv:2310.19740*.

Li, Shiyao, Xuefei Ning, et al. (2024). "Evaluating quantized large language models". In: *arXiv preprint arXiv:2402.18158*.

Li, Xiang Lisa and Percy Liang (2021). "Prefix-Tuning: Optimizing Continuous Prompts for Generation". In: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 4582–4597.

Li, Yanran, Hui Su, et al. (2017). "DailyDialog: A Manually Labelled Multi-turn Dialogue Dataset". In: *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 986–995.

Lin, Chin-Yew (2004). "Rouge: A package for automatic evaluation of summaries". In: *Text summarization branches out*, pp. 74–81.

Liu, Ao, Haoyu Dong, et al. (Dec. 2022). "PLOG: Table-to-Logic Pretraining for Logical Table-to-Text Generation". In: *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. Ed. by Goldberg, Yoav,

Kozareva, Zornitsa, and Zhang, Yue. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, pp. 5531–5546. DOI: 10.18653/v1/2022.emnlp-main.373. URL: https://aclanthology.org/2022.emnlp-main.373/.

Liu, Chia-Wei, Ryan Lowe, et al. (2016). "How not to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation". In: *arXiv preprint arXiv:1603.08023*.

Liu, Minqian, Ying Shen, et al. (2024a). "X-Eval: Generalizable Multi-aspect Text Evaluation via Augmented Instruction Tuning with Auxiliary Evaluation Aspects". In: *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 8552–8571.

— (June 2024b). "X-Eval: Generalizable Multi-aspect Text Evaluation via Augmented Instruction Tuning with Auxiliary Evaluation Aspects". In: *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*. Ed. by Duh, Kevin, Gomez, Helena, and Bethard, Steven. Mexico City, Mexico: Association for Computational Linguistics, pp. 8560–8579. DOI: 10.18653/v1/2024.naacl-long.473. URL: https://aclanthology.org/2024.naacl-long.473/.

Liu, Qian, Bei Chen, et al. (2022). "TAPEX: Table Pre-training via Learning a Neural SQL Executor". In: *International Conference on Learning Representations*.

Liu, Yang, Dan Iter, et al. (Dec. 2023). "G-Eval: NLG Evaluation using Gpt-4 with Better Human Alignment". In: *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. Ed. by Bouamor, Houda, Pino, Juan, and Bali, Kalika. Singapore: Association for Computational Linguistics, pp. 2511–2522. DOI: 10.18653/v1/2023.emnlp-main.153. URL: https://aclanthology.org/2023.emnlp-main.153.

Liu, Yinhan (2019). "Roberta: A robustly optimized bert pretraining approach". In: *arXiv preprint arXiv:1907.11692* 364.

Liu, Zhengyuan and Nancy Chen (Nov. 2021). "Controllable Neural Dialogue Summarization with Personal Named Entity Planning". In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Ed. by Moens, Marie-Francine et al. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, pp. 92–106. DOI: 10.18653/v1/2021.emnlp-main.8. URL: https://aclanthology.org/2021.emnlp-main.8/.

Lommel, Arle et al. (2014). "Using a new analytic measure for the annotation and analysis of MT errors on real data". In: *Proceedings of the 17th Annual conference of the European Association for Machine Translation*, pp. 165–172.

Long, Jieyi (2023). "Large language model guided tree-of-thought". In: *arXiv preprint arXiv:2305.08291*.

Loshchilov, Ilya and Frank Hutter (2018). "Fixing Weight Decay Regularization in Adam". In.

Lowe, Ryan, Michael Noseworthy, et al. (2017). "Towards an automatic turing test: Learning to evaluate dialogue responses". In: *arXiv preprint arXiv:1708.07149*.

Lowe, Ryan, Nissan Pow, et al. (2015). "The Ubuntu Dialogue Corpus: A Large Dataset for Research in Unstructured Multi-Turn Dialogue Systems". In: *Proceedings of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pp. 285–294.

Lu, Jinliang, Ziliang Pang, et al. (2024). "Merge, ensemble, and cooperate! a survey on collaborative strategies in the era of large language models". In: *arXiv preprint arXiv:2407.06089*.

Lu, Qingyu, Baopu Qiu, et al. (2023). "Error Analysis Prompting Enables Human-Like Translation Evaluation in Large Language Models". In: *arXiv preprint arXiv:2303.13809*.

Maddela, Mounica et al. (2022). "LENS: A learnable evaluation metric for text simplification". In: *arXiv preprint arXiv:2212.09739*.

Manishina, Elena et al. (2016). "Automatic corpus extension for data-driven natural language generation". In: *10th International Conference on Language Resources and Evaluation (LREC)*, pp. 3624–3631.

Martinez, Hector P, Georgios N Yannakakis, and John Hallam (2014). "Don't classify ratings of affect; rank them!" In: *IEEE transactions on affective computing* 5.3, pp. 314–326.

Maynez, Joshua et al. (2020). "On faithfulness and factuality in abstractive summarization". In: *arXiv preprint arXiv:2005.00661*.

Mehri, Shikib and Maxine Eskenazi (2020a). "Unsupervised Evaluation of Interactive Dialog with DialoGPT". In: *Proceedings of the 21th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pp. 225–235.

— (2020b). "USR: An unsupervised and reference free evaluation metric for dialog generation". In: *arXiv preprint arXiv:2005.00456*.

Mendes, Pablo N, Max Jakob, and Christian Bizer (2012). *DBpedia: A multilingual cross-domain knowledge base*. European Language Resources Association (ELRA).

Mille, Simon and Stamatia Dasiopoulou (2018). "FORGe at E2E 2017". In: *Proceedings of the E2E NLG Challenge System Descriptions (2018)*.

Miller, George A (1956). "The magical number seven, plus or minus two: Some limits on our capacity for processing information." In: *Psychological review* 63.2, p. 81.

Montella, Sebastien et al. (2020). "Denoising Pre-Training and Data Augmentation Strategies for Enhanced RDF Verbalization with Transformers". In: *Proceedings of the 3rd International Workshop on Natural Language Generation from the Semantic Web (WebNLG+)*, pp. 89–99.

Nallapati, Ramesh et al. (2016). "Abstractive Text Summarization using Sequence-to-sequence RNNs and Beyond". In: *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning*, pp. 280–290.

Nan, Linyong, Lorenzo Jaime Flores, et al. (2022). "R2D2: Robust Data-to-Text with Replacement Detection". In: *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 6903–6917.

Nan, Linyong, Chiachun Hsieh, et al. (2022). "FeTaQA: Free-form table question answering". In: *Transactions of the Association for Computational Linguistics* 10, pp. 35–49.

Narayan, Shashi, Shay B Cohen, and Mirella Lapata (2018). "Don't Give Me the Details, Just the Summary! Topic-Aware Convolutional Neural Networks for

Extreme Summarization". In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Novikova, Jekaterina, Ondřej Dušek, and Verena Rieser (2017). "The E2E Dataset: New Challenges For End-to-End Generation". In: *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pp. 201–206.

— (2018). "RankME: Reliable human ratings for natural language generation". In: *arXiv preprint arXiv:1803.05928*.

Ouyang, Long et al. (2022). "Training language models to follow instructions with human feedback". In: *Advances in neural information processing systems* 35, pp. 27730–27744.

Papineni, Kishore et al. (2002). "Bleu: a method for automatic evaluation of machine translation". In: *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pp. 311–318.

Parikh, Ankur P et al. (2020). "ToTTo: A controlled table-to-text generation dataset". In: *arXiv preprint arXiv:2004.14373*.

Peng, Bo et al. (2023). "RWKV: Reinventing RNNs for the Transformer Era". In: *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 14048–14077.

Petroni, Fabio et al. (Nov. 2019). "Language Models as Knowledge Bases?" In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Ed. by Inui, Kentaro et al. Hong Kong, China: Association for Computational Linguistics, pp. 2463–2473. DOI: 10.18653/v1/D19-1250. URL: https://aclanthology.org/D19-1250/.

Preston, Carolyn C and Andrew M Colman (2000). "Optimal number of response categories in rating scales: reliability, validity, discriminating power, and respondent preferences". In: *Acta psychologica* 104.1, pp. 1–15.

Radford, Alec (2018). "Improving language understanding by generative pre-training". In.

Radford, Alec et al. (2019). "Language models are unsupervised multitask learners". In: *OpenAI blog* 1.8, p. 9.

Rafailov, Rafael et al. (2024). "Direct preference optimization: Your language model is secretly a reward model". In: *Advances in Neural Information Processing Systems* 36.

Raffel, Colin et al. (2020). "Exploring the limits of transfer learning with a unified text-to-text transformer". In: *Journal of machine learning research* 21.140, pp. 1–67.

Rajani, Nazneen Fatema et al. (2019). "Explain Yourself! Leveraging Language Models for Commonsense Reasoning". In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 4932–4942.

Rashkin, Hannah (2018). "Towards empathetic open-domain conversation models: A new benchmark and dataset". In: *arXiv preprint arXiv:1811.00207*.

Rebuffi, Sylvestre-Alvise, Hakan Bilen, and Andrea Vedaldi (2017). "Learning multiple visual domains with residual adapters". In: *Advances in neural information processing systems* 30.

Reinhart, Tanya (1980). "Conditions for text coherence". In: *Poetics today* 1.4, pp. 161–180.

Reiter, Ehud (2018). "A structured review of the validity of BLEU". In: *Computational Linguistics* 44.3, pp. 393–401.

Reiter, Ehud and Anja Belz (2009). "An Investigation into the Validity of Some Metrics for Automatically Evaluating Natural Language Generation Systems". In: *Computational Linguistics* 35.4, pp. 529–558.

Reiter, Ehud and Robert Dale (1997). "Building applied natural language generation systems". In: *Natural Language Engineering* 3.1, pp. 57–87.

Roller, Stephen et al. (Apr. 2021). "Recipes for Building an Open-Domain Chatbot". In: *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*. Ed. by Merlo, Paola, Tiedemann, Jorg, and Tsarfaty, Reut. Online: Association for Computational Linguistics, pp. 300–325. DOI: 10.18653/v1/2021.eacl-main.24. URL: https://aclanthology.org/2021.eacl-main.24/.

Ruder, Sebastian (2016). "An overview of gradient descent optimization algorithms". In: *arXiv preprint arXiv:1609.04747*.

Sachdeva, Rachneet et al. (2024). "Localizing and Mitigating Errors in Long-form Question Answering". In: *arXiv preprint arXiv:2407.11930*.

Schluter, Natalie (2017). "The limits of automatic summarisation according to rouge". In: *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, pp. 41–45.

Schmidtová, Patrícia et al. (2024). "Automatic Metrics in Natural Language Generation: A Survey of Current Evaluation Practices". In: *arXiv preprint arXiv:2408.09169*.

See, Abigail, Peter J Liu, and Christopher D Manning (2017). "Get To The Point: Summarization with Pointer-Generator Networks". In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1073–1083.

See, Abigail, Stephen Roller, et al. (June 2019). "What makes a good conversation? How controllable attributes affect human judgments". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Ed. by Burstein, Jill, Doran, Christy, and Solorio, Thamar. Minneapolis, Minnesota: Association for Computational Linguistics, pp. 1702–1723. DOI: 10.18653/v1/N19-1170. URL: https://aclanthology.org/N19-1170/.

Serban, Iulian, Alessandro Sordoni, Yoshua Bengio, et al. (2016). "Building end-to-end dialogue systems using generative hierarchical neural network models". In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 30. 1.

Serban, Iulian, Alessandro Sordoni, Ryan Lowe, et al. (2017). "A hierarchical latent variable encoder-decoder model for generating dialogues". In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 31. 1.

Shaw, Peter, Jakob Uszkoreit, and Ashish Vaswani (2018). "Self-Attention with Relative Position Representations". In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*. Association for Computational Linguistics.

Shazeer, Noam (2020). "Glu variants improve transformer". In: *arXiv preprint arXiv:2002.05202*.

Shinn, Noah et al. (2024). "Reflexion: Language agents with verbal reinforcement learning". In: *Advances in Neural Information Processing Systems* 36.

Shuttleworth, Reece et al. (2024). "LoRA vs Full Fine-tuning: An Illusion of Equivalence". In: *arXiv preprint arXiv:2410.21228*.

Smiley, Charese et al. (Nov. 2018). "The E2E NLG Challenge: A Tale of Two Systems". In: *Proceedings of the 11th International Conference on Natural Language Generation*. Ed. by Krahmer, Emiel, Gatt, Albert, and Goudbeek, Martijn. Tilburg University, The Netherlands: Association for Computational Linguistics, pp. 472–477. DOI: 10.18653/v1/W18-6558. URL: https://aclanthology.org/W18-6558/.

Sobrevilla Cabezudo, Marco Antonio and Thiago A. S. Pardo (Dec. 2020). "NILC at WebNLG+: Pretrained Sequence-to-Sequence Models on RDF-to-Text Generation". In: *Proceedings of the 3rd International Workshop on Natural Language Generation from the Semantic Web (WebNLG+)*. Ed. by Castro Ferreira, Thiago et al. Dublin, Ireland (Virtual): Association for Computational Linguistics, pp. 131–136. URL: https://aclanthology.org/2020.webnlg-1.14/.

Stiennon, Nisan et al. (2020). "Learning to summarize with human feedback". In: *Advances in Neural Information Processing Systems* 33, pp. 3008–3021.

Su, Jianlin et al. (2024). "Roformer: Enhanced transformer with rotary position embedding". In: *Neurocomputing* 568, p. 127063.

Sutskever, Ilya, Oriol Vinyals, and Quoc V. Le (2014). "Sequence to Sequence Learning with Neural Networks". In: *Neural Information Processing Systems*.

Tam, Zhi Rui et al. (2024). "Let me speak freely? a study on the impact of format restrictions on performance of large language models". In: *arXiv preprint arXiv:2408.02442*.

Taori, Rohan et al. (2023). "Alpaca: A strong, replicable instruction-following model". In: *Stanford Center for Research on Foundation Models. https://crfm. stanford. edu/2023/03/13/alpaca. html* 3.6, p. 7.

Thomson, Craig and Ehud Reiter (2020). "A gold standard methodology for evaluating accuracy in data-to-text systems". In: *arXiv preprint arXiv:2011.03992*.

Touvron, Hugo et al. (2023). "Llama: Open and efficient foundation language models". In: *arXiv preprint arXiv:2302.13971*.

Tran, Trung and Dang Tuan Nguyen (Dec. 2020). "WebNLG 2020 Challenge: Semantic Template Mining for Generating References from RDF". In: *Proceedings of the 3rd International Workshop on Natural Language Generation from the Semantic Web (WebNLG+)*. Ed. by Castro Ferreira, Thiago et al. Dublin, Ireland (Virtual): Association for Computational Linguistics, pp. 177–185. URL: https://aclanthology.org/2020.webnlg-1.21/.

Urbanek, Jack et al. (Nov. 2019). "Learning to Speak and Act in a Fantasy Text Adventure Game". In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Ed. by Inui, Kentaro et al. Hong Kong, China: Association for Computational Linguistics, pp. 673–683. DOI: 10.18653/v1/D19-1062. URL: https://aclanthology.org/D19-1062/.

Van der Lee, Chris et al. (2021). "Human evaluation of automatically generated text: Current trends and best practice guidelines". In: *Computer Speech & Language* 67, p. 101151.

Van Miltenburg, Emiel, Miruna Clinciu, Ondřej Dušek, Dimitra Gkatzia, Stephanie Inglis, Leo Leppänen, Saad Mahamood, Stephanie Schoch, et al. (2023). "Barriers and enabling factors for error analysis in NLG research". In: *Northern European Journal of Language Technology* 9.1.

Van Miltenburg, Emiel, Miruna-Adriana Clinciu, Ondřej Dušek, Dimitra Gkatzia, Stephanie Inglis, Leo Leppänen, Saad Mahamood, Emma Manning, et al. (2021). "Underreporting of errors in NLG output, and what to do about it". In: *arXiv preprint arXiv:2108.01182*.

Vaswani, Ashish et al. (2017). "Attention is All you Need". In: *Advances in Neural Information Processing Systems*.

Vinyals, Oriol, Meire Fortunato, and Navdeep Jaitly (2015). "Pointer networks". In: *Advances in neural information processing systems* 28.

Völske, Michael et al. (2017). "Tl; dr: Mining reddit to learn automatic summarization". In: *Proceedings of the Workshop on New Frontiers in Summarization*, pp. 59–63.

Wang, Alex, Kyunghyun Cho, and Mike Lewis (2020). "Asking and Answering Questions to Evaluate the Factual Consistency of Summaries". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 5008–5020.

Wang, Xuezhi, Jason Wei, et al. (2022). "Self-consistency improves chain of thought reasoning in language models". In: *arXiv preprint arXiv:2203.11171*.

Wang, Zhilin, Alexander Bukharin, et al. (2024). "Helpsteer2-preference: Complementing ratings with preferences". In: *arXiv preprint arXiv:2410.01257*.

Wei, Jason et al. (2022). "Chain-of-thought prompting elicits reasoning in large language models". In: *Advances in neural information processing systems* 35, pp. 24824–24837.

Wen, Tsung-Hsien, Milica Gasic, Dongho Kim, et al. (2015). "Stochastic language generation in dialogue using recurrent neural networks with convolutional sentence reranking". In: *arXiv preprint arXiv:1508.01755*.

Wen, Tsung-Hsien, Milica Gasic, Nikola Mrkšić, et al. (2015). "Semantically Conditioned LSTM-based Natural Language Generation for Spoken Dialogue Systems". In: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 1711–1721.

Wolf, Thomas et al. (2019). "Transfertransfo: A transfer learning approach for neural network based conversational agents". In: *arXiv preprint arXiv:1901.08149*.

Wu, Chien-Sheng et al. (Aug. 2021). "Controllable Abstractive Dialogue Summarization with Sketch Supervision". In: *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*. Ed. by Zong, Chengqing et al. Online: Association for Computational Linguistics, pp. 5108–5122. DOI: 10.18653/v1/2021.findings-acl.454. URL: https://aclanthology.org/2021.findings-acl.454/.

Xie, Zhuohan, Trevor Cohn, and Jey Han Lau (2023). "The Next Chapter: A Study of Large Language Models in Storytelling". In: *Proceedings of the 16th International Natural Language Generation Conference*, pp. 323–351.

Xiong, Ruibin et al. (2020). "On layer normalization in the transformer architecture". In: *International Conference on Machine Learning*. PMLR, pp. 10524–10533.

Xu, Wei, Courtney Napoles, et al. (2016). "Optimizing Statistical Machine Translation for Text Simplification". In: *Transactions of the Association for Computational Linguistics* 4. Ed. by Lee, Lillian, Johnson, Mark, and Toutanova, Kristina, pp. 401–415. DOI: 10.1162/tacl_a_00107. URL: https://aclanthology.org/Q16-1029/.

Xu, Wenda, Danqing Wang, et al. (2023). "INSTRUCTSCORE: Explainable Text Generation Evaluation with Finegrained Feedback". In: *arXiv preprint arXiv:2305.14282*.

Yang, An et al. (2024). "Qwen2. 5 Technical Report". In: *arXiv preprint arXiv:2412.15115*.

Yannakakis, Georgios N and John Hallam (2011). "Ranking vs. preference: a comparative study of self-reporting". In: *Affective Computing and Intelligent Interaction: 4th International Conference, ACII 2011, Memphis, TN, USA, October 9–12, 2011, Proceedings, Part I 4*. Springer, pp. 437–446.

Yao, Lili, Nanyun Peng, et al. (2019). "Plan-and-write: Towards better automatic storytelling". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 33. 01, pp. 7378–7385.

Yao, Shunyu, Dian Yu, et al. (2024). "Tree of thoughts: Deliberate problem solving with large language models". In: *Advances in Neural Information Processing Systems* 36.

Yao, Shunyu, Jeffrey Zhao, et al. (2023). "ReAct: Synergizing Reasoning and Acting in Language Models". In: *International Conference on Learning Representations (ICLR)*.

Yuan, Weizhe, Graham Neubig, and Pengfei Liu (2021). "Bartscore: Evaluating generated text as text generation". In: *Advances in Neural Information Processing Systems* 34, pp. 27263–27277.

Zhang, Tianyi, Varsha Kishore, et al. (2019). "Bertscore: Evaluating text generation with bert". In: *arXiv preprint arXiv:1904.09675*.

Zhang, Xiang, Senyu Li, et al. (Dec. 2023). "Don't Trust ChatGPT when your Question is not in English: A Study of Multilingual Abilities and Types of LLMs". In: *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. Ed. by Bouamor, Houda, Pino, Juan, and Bali, Kalika. Singapore: Association for Computational Linguistics, pp. 7915–7927. DOI: 10.18653/v1/2023.emnlp-main.491. URL: https://aclanthology.org/2023.emnlp-main.491/.

Zhang, Yizhe, Siqi Sun, et al. (July 2020). "DIALOGPT : Large-Scale Generative Pre-training for Conversational Response Generation". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*. Ed. by Celikyilmaz, Asli and Wen, Tsung-Hsien. Online: Association for Computational Linguistics, pp. 270–278. DOI: 10.18653/v1/2020.acl-demos.30. URL: https://aclanthology.org/2020.acl-demos.30/.

Zhao, Haiyan, Hanjie Chen, et al. (2024). "Explainability for large language models: A survey". In: *ACM Transactions on Intelligent Systems and Technology* 15.2, pp. 1–38.

Zhao, Tiancheng, Ran Zhao, and Maxine Eskenazi (2017). "Learning Discourse-level Diversity for Neural Dialog Models using Conditional Variational Au-

toencoders". In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 654–664.

Zhao, Tianyu, Divesh Lala, and Tatsuya Kawahara (2020). "Designing precise and robust dialogue response evaluators". In: *arXiv preprint arXiv:2004.04908*.

Zhao, Wei, Maxime Peyrard, et al. (Nov. 2019). "MoverScore: Text Generation Evaluating with Contextualized Embeddings and Earth Mover Distance". In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Ed. by Inui, Kentaro et al. Hong Kong, China: Association for Computational Linguistics, pp. 563–578. DOI: 10.18653/v1/D19-1053. URL: https://aclanthology.org/D19-1053/.

Zhao, Yilun, Linyong Nan, et al. (Dec. 2022). "ReasTAP: Injecting Table Reasoning Skills During Pre-training via Synthetic Reasoning Examples". In: *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. Ed. by Goldberg, Yoav, Kozareva, Zornitsa, and Zhang, Yue. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, pp. 9006–9018. DOI: 10.18653/v1/2022.emnlp-main.615. URL: https://aclanthology.org/2022.emnlp-main.615/.

Zhao, Yilun, Zhenting Qi, et al. (2023). "LoFT: Enhancing Faithfulness and Diversity for Table-to-Text Generation via Logic Form Control". In: *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 554–561.

Zhao, Yilun, Haowei Zhang, et al. (2023). "Investigating Table-to-Text Generation Capabilities of LLMs in Real-World Information Seeking Scenarios". In: *arXiv preprint arXiv:2305.14987*.

Zheng, Lianmin et al. (2023). "Judging llm-as-a-judge with mt-bench and chatbot arena". In: *Advances in Neural Information Processing Systems* 36, pp. 46595–46623.

Zhong, Ming et al. (2022). "Towards a Unified Multi-Dimensional Evaluator for Text Generation". In: *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 2023–2038.

Zhu, Qihao, Daya Guo, et al. (2024). "DeepSeek-Coder-V2: Breaking the Barrier of Closed-Source Models in Code Intelligence". In: *arXiv preprint arXiv:2406.11931*.

Zhu, Xunyu, Jian Li, et al. (2024). "A survey on model compression for large language models". In: *Transactions of the Association for Computational Linguistics* 12, pp. 1556–1577.

# List of Figures

2.1 Transformer architecture (source: Vaswani et al. (2017) . . . . . 11
2.2 Weight update matrix decomposition used in LoRA (source: Hu and Shen, et al. (2021) . . . . . 15

3.1 Distribution of overall scores in the dataset. . . . . 36

# List of Tables


3.1 Overview of tasks in each category with corresponding input and outputs. Optional inputs are enclosed in parentheses. . . . . 33
3.2 Dataset Statistics . . . . . 34
3.3 Percentage of outliers for each model per task category. . . . . 35
3.4 Input data formats for data-to-text tasks. . . . . 37
3.5 Quantization levels and Ollama tags used for the models. . . . . 45

5.1 Segment-level Pearson ($r$), Spearman ($\rho$) and Kendall ($\tau$) correlations of different metrics for factual consistency on QAGS. The highest values are highlighted in bold. . . . . 49
5.2 Segment-level Spearman ($\rho$) and Kendall ($\tau$) correlations of different metrics on SummEval. . . . . 50
5.3 Segment-level Pearson ($r$) and Spearman ($\rho$) correlations of different metrics on TopicalChat. . . . . 51
5.4 Segment-level Spearman ($r$) correlations of different metrics on SFRES and SFHOT. **Inf.** = informativeness, **Nat.** = naturalness. 51
5.5 Segment-level Pearson ($r$) correlations of various metrics on HANNA. **Coh.** = coherence, **Rel.** = relevance, **Eng.** = engagement, **Emp.** = empathy, **Sur.** = surprise, **Com.** = complexity, **Avg.** = average. . . . . 52
5.6 Segment-level Spearman correlations ($\rho$) of various metrics on HANNA. **Coh.** = coherence, **Rel.** = relevance, **Eng.** = engagement, **Emp.** = empathy, **Sur.** = surprise, **Com.** = complexity, **Avg.** = average. . . . . 52
5.7 Segment-level Kendall ($\tau$) correlations of various metrics on HANNA. **Coh.** = coherence, **Rel.** = relevance, **Eng.** = engagement, **Emp.** = empathy, **Sur.** = surprise, **Com.** = complexity, **Avg.** = average. . . . . 53
5.8 Segment-level Pearson ($r$) correlations of different metrics on WikiDA. . . . . 53

A.1 Overview of the evaluated systems in the dataset for the summarization task. Sources: GW = Gao and Wan (2022), GR = Grusky et al. (2018), NA = Narayan et al. (2018), ST = Stiennon et al. (2020), new = newly generated. . . . . 74
A.2 Overview of the evaluated systems in the dataset for the data-to-text task. **Sources**: DU = Dušek et al. (2020), FE = Ferreira et al. (2020), ZH = Zhao and Zhang, et al. (2023), new = newly generated. 75
A.3 Overview of the evaluated systems in the dataset for the dialogue response generation task. Sources: GU = Gupta et al. (2019), HU = Huang et al. (2020), ZH = Zhao and Lala, et al. (2020), new = newly generated. . . . . 75
A.4 Overview of the evaluated systems in the dataset for the story generation task. Sources: GU = Guan and Zhang, et al. (2021), new = newly generated. . . . . 76

A.5 Overview of the evaluated systems in the dataset for the question answering task. Sources: KO = Kočiskỳ et al. (2018), ZH = Zhao and Zhang, et al. (2023), new = newly generated. . . . . 76

# A Appendix

## A.1 System Outputs

Tables A.1–A.5 list the evaluated systems for each task category in our dataset.

| System | Type | Sources |
|---|---|---|
| Qwen 2.5 0.5B (Yang et al., 2024) | Instruction-tuned LLM | new |
| Llama 2 7B Chat (Touvron et al., 2023) | Instruction-tuned LLM | new |
| Gemma 2 2B (Gemma Team et al., 2024) | Instruction-tuned LLM | new |
| Nous Hermes 2 Mixtral 8x7B DPO[a] | Instruction-tuned LLM | new |
| GPT-4o[b] | Instruction-tuned LLM | new |
| OpenAI summarization (pre-trained) | pre-trained LMs | ST |
| OpenAI summarization (supervised) | LMs trained with SFT | ST |
| OpenAI summarization (RLHF) | LMs trained with SFT+PPO | ST |
| T5 (Raffel et al., 2020) | pre-trained LM | ST |
| UniLM (Dong et al., 2019) | pre-trained LM | ST |
| CODS (Wu et al., 2021) | BART-based hybrid model | GW |
| ConvoSumm (Fabbri, Rahman, et al., 2021) | BART-based model | GW |
| Ctrl-DiaSumm (Liu and Chen, 2021) | BART-based model | GW |
| ConvS2S (Gehring et al., 2017) | Convolutional seq-to-seq | NA |
| Topic-ConvS2S (Narayan et al., 2018) | Topic-conditioned ConvS2S | NA |
| S2S (Cho et al., 2014; Sutskever et al., 2014) | RNN-based seq-to-seq with attention | GR |
| PNG (Gulçehre et al., 2016; Vinyals et al., 2015) | Pointer-generator network | GR, GW, NA |
| TextRank (Barrios et al., 2016) | Ranking-based extractive summarization | GR |
| Reference | Human-written reference | ST |
| Title | Extractive baseline | ST |
| LEAD | Extractive baseline | NA |
| LEAD-2 | Extractive baseline | ST |
| LEAD-3 (See, Liu, et al., 2017) | Extractive baseline | GR, GW, ST |
| Ext-Oracle | Extractive oracle | NA |
| Fragments | Extractive oracle | NA |

**Table A.1** Overview of the evaluated systems in the dataset for the summarization task. Sources: GW = Gao and Wan (2022), GR = Grusky et al. (2018), NA = Narayan et al. (2018), ST = Stiennon et al. (2020), new = newly generated.

[a] `https://huggingface.co/NousResearch/Nous-Hermes-2-Mixtral-8x7B-DPO`
[b] `https://openai.com/index/hello-gpt-4o/`

## A.2 Prompts for Output Generation

We present prompt templates used to generate new LLM outputs for our dataset. Double curly brackets represent placeholders, which are filled with actual data when the prompt is applied to a specific example. Note that these prompts are intentionally not optimized for quality, as we are interested in generating outputs with errors.

Prompt templates for summarization are shown in Figures A.1–A.3, each corresponding to a different dataset. Templates for the data-to-text datasets are shown in Figures A.4–A.7. Figure A.8 shows the prompt template used to generate responses for DailyDialog and EmphateticDialogues datasets. Prompt template for Wizard of Wikipedia, which involves grounded response generation is shown in A.9. In addition to a dialogue history, it also contains knowledge as additional context. As described in 3.3.4, we apply two different prompts to

| System | Type | Sources |
|---|---|---|
| Qwen 2.5 Coder 1.5B (Yang et al., 2024) | Instruction-tuned LLM | new |
| Gemma 2 2B (Gemma Team et al., 2024) | Instruction-tuned LLM | new |
| Llama 2 7B Chat (Touvron et al., 2023) | Instruction-tuned LLM | new |
| Solar 10.7B (Kim, Kim, et al., 2024) | Instruction-tuned LLM | new |
| DeepSeek Coder v2 16B (Zhu, Guo, et al., 2024) | Instruction-tuned LLM | new |
| Nous Hermes 2 Mixtral 8x7B DPO[a] | Instruction-tuned LLM | new |
| Claude 3.5 Sonnet[b] | Instruction-tuned LLM | new |
| GPT-4o[c] | Instruction-tuned LLM | new |
| NILC (Sobrevilla Cabezudo and Pardo, 2020) | Fine-tuned BART | FE |
| Orange-NLG (Montella et al., 2020) | Fine-tuned BART | FE |
| Amazaon AI (Shanghai) (Guo et al., 2020) | Graph CNN + T5 | FE |
| GPT2-C2F (Chen, Chen, et al., 2020) | Fine-tuned GPT-2 | ZH |
| LoFT (Zhao, Qi, et al., 2023) | Fine-tuned BART | ZH |
| PLOG (Liu, Dong, et al., 2022) | Fine-tuned T5 | ZH |
| R2D2 (Nan, Flores, et al., 2022) | Fine-tuned T5 | ZH |
| Flan-T5 (Chung et al., 2024) | Instruction-tuned LM | ZH |
| Adapt (Elder et al., 2018) | RNN seq-to-seq | DU |
| Sheff2 (Chen, Lampouras, et al., 2018) | RNN seq-to-seq | DU |
| Slug (Juraska et al., 2018) | RNN + convolutional seq-to-seq | DU |
| Forge1 (Mille and Dasiopoulou, 2018) | Rule-based | DU |
| TR2 (Smiley et al., 2018) | Template-based | DU |
| DANGNT-SGU (Tran and Nguyen, 2020) | Template-based | FE |
| RALI-Université de Montréal (Lapalme, 2020) | Template-based | FE |

**Table A.2** Overview of the evaluated systems in the dataset for the data-to-text task. **Sources**: DU = Dušek et al. (2020), FE = Ferreira et al. (2020), ZH = Zhao and Zhang, et al. (2023), new = newly generated.

[a] `https://huggingface.co/NousResearch/Nous-Hermes-2-Mixtral-8x7B-DPO`
[b] `https://www.anthropic.com/news/claude-3-5-sonnet`
[c] `https://openai.com/index/hello-gpt-4o/`

| System | Type | Sources |
|---|---|---|
| Claude 3.5 Sonnet[a] | Instruction-tuned LLM | new |
| GPT-4o[b] | Instruction-tuned LLM | new |
| Tülu 3 (Lambert et al., 2024) | Instruction-tuned LLM | new |
| Dolphin 2.9 Llama 3 8B[c] | Instruction-tuned LLM | new |
| Vicuna 7B (Zheng et al., 2023) | Instruction-tuned LLM | new |
| BlenderBot-small (Roller et al., 2021) | Dialogue LM | new |
| DialoGPT-small (Zhang, Sun, et al., 2020) | Dialogue LM | new |
| GPT-Neo 125M (Gao, Biderman, et al., 2020) | Pre-trained LM | new |
| GPT-2 (Wolf et al., 2019) | Pre-trained LM | ZH |
| CVAE (Zhao, Zhao, et al., 2017) | Conditional variational autoencoder | GU |
| HRED (Serban, Sordoni, Bengio, et al., 2016) | Hierarchical recurrent encoder-decoder | GU, ZH |
| Transformer-generator (Dinan et al., 2018) | Transformer-based generative model | HU |
| Transformer-ranker (Urbanek et al., 2019) | Transformer-based ranking model | HU |
| DualEncoder (Lowe, Pow, et al., 2015) | LSTM dual encoder | GU |
| VHRED (Serban, Sordoni, Lowe, et al., 2017) | Latent variable HRED | ZH |
| S2S (Cho et al., 2014; Sutskever et al., 2014) | RNN-based seq-to-seq with attention | GU, ZH |

**Table A.3** Overview of the evaluated systems in the dataset for the dialogue response generation task. Sources: GU = Gupta et al. (2019), HU = Huang et al. (2020), ZH = Zhao and Lala, et al. (2020), new = newly generated.

[a] `https://www.anthropic.com/news/claude-3-5-sonnet`
[b] `https://openai.com/index/hello-gpt-4o/`
[c] `https://huggingface.co/cognitivecomputations/dolphin-2.9-llama3-8b`

| System | Type | Sources |
|---|---|---|
| Gemma 2 2B (Gemma Team et al., 2024) | Instruction-tuned LLM | new |
| Dolphin 2.9 Llama 3 8B[a] | Instruction-tuned LLM | new |
| Nous Hermes 2 Mixtral 8x7B DPO[b] | Instruction-tuned LLM | new |
| GPT-4o[c] | Instruction-tuned LLM | new |
| GPT-2 (Radford et al., 2019) | Pre-trained LM | GU |
| GPT-KG (Guan, Huang, et al., 2020) | Knowledge-enhanced GPT-2 | GU |
| Fusion (Fan et al., 2018) | Convolutional seq-to-seq with attention | GU |
| Plan&Write (Yao, Peng, et al., 2019) | Hierarchical RNN-based model | GU |
| S2S (Cho et al., 2014; Sutskever et al., 2014) | RNN-based seq-to-seq | GU |

**Table A.4** Overview of the evaluated systems in the dataset for the story generation task. Sources: GU = Guan and Zhang, et al. (2021), new = newly generated.

[a] `https://huggingface.co/cognitivecomputations/dolphin-2.9-llama3-8b`
[b] `https://huggingface.co/NousResearch/Nous-Hermes-2-Mixtral-8x7B-DPO`
[c] `https://openai.com/index/hello-gpt-4o/`

| System | Type | Sources |
|---|---|---|
| Claude 3.5 Sonnet[a] | Instruction-tuned LLM | new |
| GPT-4o[b] | Instruction-tuned LLM | new |
| Nous Hermes 2 Mixtral 8x7B DPO[c] | Instruction-tuned LLM | new |
| DeepSeek Coder v2 16B (Zhu, Guo, et al., 2024) | Instruction-tuned LLM | new |
| Llama 2 7B Chat (Touvron et al., 2023) | Instruction-tuned LLM | new |
| Qwen 2.5 Coder 1.5B (Yang et al., 2024) | Instruction-tuned LLM | new |
| Qwen 2.5 0.5B (Yang et al., 2024) | Instruction-tuned LLM | new |
| GPT-Neo 125M (Gao, Biderman, et al., 2020) | Pre-trained LM | new |
| BART (Lewis, 2019) | Pre-trained LM | ZH |
| Flan-T5 (Chung et al., 2024) | Instruction-tuned LM | ZH |
| OmniTab (Jiang, Mao, et al., 2022) | Table pre-trained LM | ZH |
| ReasTAP (Zhao, Nan, et al., 2022) | Table pre-trained LM | ZH |
| TAPEX (Liu, Chen, et al., 2022) | Table pre-trained LM | ZH |
| Reference | Human-written reference | KO |

**Table A.5** Overview of the evaluated systems in the dataset for the question answering task. Sources: KO = Kočiskỳ et al. (2018), ZH = Zhao and Zhang, et al. (2023), new = newly generated.

[a] `https://www.anthropic.com/news/claude-3-5-sonnet`
[b] `https://openai.com/index/hello-gpt-4o/`
[c] `https://huggingface.co/NousResearch/Nous-Hermes-2-Mixtral-8x7B-DPO`

sample stories of different lengths, shown in Figures A.10 and A.11. To generate answers based on the NarrativeQA dataset, we applied three different prompt templates. A.12 serves as the default prompt, while A.13 was used to obtain very short answers from the LLMs. Finally, A.14 is needed for GPT-Neo, which is a text completion model without instruction-tuning.

```
Your task is to create very a short, single-sentence summary of the given
    article.

Article:
```
{{ article }}
```
```

**Figure A.1** Prompt template for the XSum dataset

```
Your task is to create a short, single-paragraph summary of the given article.

Article:
```
{{ article }}
```
```

**Figure A.2** Prompt template for the Newsroom dataset

```
Your task is to create a short and concise summary of the following dialogue:
```
{{ dialogue }}
```
```

**Figure A.3** Prompt template for the SAMSum dataset

```
You are given a meaning representation for a food service venue, which
    consists of a set of attribute-value pairs in the form of "attribute:
    value".
Based on the meaning representation:

{{ data }}

Your task is to write a brief, fluent, and coherent single-sentence text in
    natural language. The text should be balanced and neutral. Make sure that
    the text covers all information in the input data, and that all the facts
    mentioned in the text can be derived from the input data. Do not add any
    extra information.
```

**Figure A.4** Prompt template for the E2E NLG dataset (adapted from Kasner and Dušek, 2024)

```
You are given a set RDF triples as input data. Your task is to write a brief, fluent, and coherent single-paragraph text in natural language based on the input data. The text should be balanced and neutral. Make sure that the text covers all information in the data, and that all the facts mentioned in the text can be derived from the data. Do not add any extra information.

Input data:
```
{{ data }}
```

Your output:
```

**Figure A.5** Prompt template for the WebNLG2020 dataset (adapted from Kasner and Dušek, 2024)

```
Your task is to summarize the information from the given input table data into a single coherent sentence.

Input data:
```
{{ data }}
```

Your output:
```

**Figure A.6** Prompt template for the ToTTo dataset.

```
Given the table below, generate 5 logically sound statements that involve comparison or aggregation of data points. The sentence should go beyond simply describing individual rows or cells and should capture a relationship or inference based on the table data. Each statements should describe a different fact.

Generate your output in exactly this format:
```
1. <Statement 1>
2. <Statement 2>
...
5. <Statement 5>
```
Do not generate any reasoning, only generate a list of five statements.

Input data:
```
Title: {{ title }}
Table:
{{ table }}
```

Your output:
```

**Figure A.7** Prompt template for the LogicNLG dataset

```
Generate a single response based on the dialogue history between participants
    A and B. Generate only the response on one line. Do not include any other
    text.
Dialogue history:
{{ context }}
```

**Figure A.8** Prompt template for dialogue response generation

```
Generate a single response based on the dialogue history between participants
    A and B. You are participants B. Your response should use the provided
    knowledge. Generate only the response on one line. Do not include any
    other text.

Dialogue history:
{{ context }}

Knowledge:
{{ knowledge }}
```

**Figure A.9** Prompt template for grounded dialogue response generation

```
Write a short story (around 500 words) for the following prompt:
{{ prompt }}
```

**Figure A.10** Prompt template for story generation.

```
Write a short story (100-200 words) for the following prompt:
{{ prompt }}
```

**Figure A.11** Prompt template for story generation (short outputs)

```
Based on the provided summary:
{{ summary }}

Answer the following question:
{{ question }}
```

**Figure A.12** Prompt template for the NarrativeQA dataset.

```
Based on the provided summary:
{{ summary }}

Answer the following question:
{{ question }}

Make your answer very concise (just a few words), but complete at the same
    time.
```

**Figure A.13** Prompt template for generating concise answers on NarrativeQA.

```
Summary:
{{ summary }}

Question:
{{ question }}

Answer:
```

**Figure A.14** Prompt template for generating answers with text completion models on NarrativeQA.

```
Based on the provided table:
```
Title: {{ title }}
Subtitle: {{ subtitle }}
Table:
{{ table }}
```
Answer the following question:
{{ question }}
```

**Figure A.15** Prompt template for generating answers for the FeTaQA dataset.

## A.3 Evaluation Prompts

Prompt templates for the annotator and supervisor LLMs, introduced in Section 3.2.2 are presented in Figures A.16–A.17. Prompt template used for fine-tuning and inference of the distilled model is shown in Figure A.18.

```
### Instructions
Your task is to evaluate an output of data-to-text task, where the model was instructed to write a single-paragraph description of a venue based on the given data. The data consist of a set of attribute-value pairs in the form 'attribute: value'.
Based on the given data and the generated text, identify errors in the text with respect to {{ aspect_name }} (described below).
For each error, determine its severity on a scale from 1 to 5, where 1 is the least severe and 5 is the most severe.

Definition of {{ aspect_name }}:
{{ aspect_definition }}

Rules:
Do not make assumptions and do not bring in external knowledge not present in the provided context.
Identify only the errors related to the {{ aspect_name }} of the text. Do not consider other aspects like {{ negative_aspects }}!
If there are no errors related to {{ aspect_name }} in the text, you should output 'No Error' and provide 'Excellent' score.

Steps:
1. Carefully read the data and identify the main attributes and their values.
2. Read the generated text and compare it with the source data with respect to {{ aspect_name }}.
3. If the text contains any error that negatively affects its {{ aspect_name }}, identify its exact location (specific word or phrase), explain why it is considered an error, and determine the severity of the error.
4. Finally, provide an overall score for the {{ aspect_name }} of the text. The score should be a label on the following scale (lowest to highest): 'Unacceptable', 'Poor', 'Fair', 'Good', 'Excellent'. The score 'Unacceptable' indicates that the text is {{ min_score_desc }}, while 'Excellent' indicates that the text is {{ max_score_desc }}.

### Data
{{ input }}

### Generated Text
{{ output }}

### Output format:
Generate your output exactly in this format:
'''
Error 1:
Location: <location of the error - the exact word or phrase in the response>
Explanation: <explanation for the error, including the reason why it is considered {{ aspect_name }} issue>
Severity: <integer from 1 to 5>

Error 2:
...

Overall score: <one of: Unacceptable, Poor, Fair, Good, Excellent>
Explanation of the score: <explanation of the score>
'''
```

**Figure A.16** Annotator prompt template for data-to-text task

```
### Instructions
You are given multiple error annotation sets for an AI model output. Your task is to merge the annotation sets to a single final annotation set.
The result shouldn't contain any duplicates. If there are multiple error annotations for approximately the same location that describe the same issue, you should merge them into single location. Otherwise, the error annotations should be as granular as possible. If there are multiple different locations with the same issue, each should have its own error annotation. Likewise, if there are multiple issues with respect to the same location, each should have its own error annotation. Use the following guidelines:
* Each error annotation should describe a single issue.
* Merge only annotations where the locations have significant overlap.
* When merging multiple locations, choose a single span from the output text that covers the locations from merged annotations.
* Never include multiple spans from the annotations under the same "Location" line.
* Do not include any other text in "Location" line than the text that is actually in the output, except for annotations that mention omissions or similar issues.
* When merging explanations, combine the most relevant information from the merged annotations.
* Severity levels range from 1 to 5 from least severe to most severe. Use the most severe level from the merged annotations.
* Final annotation set should not include more than 8 error annotations. If there are more than that, use only the most severe ones.
* Make the final annotation set as concise as possible in terms of number of error annotations.

Don't use any markdown formatting. Generate merged error annotations in this format, without any additional text:
‘‘‘
Error 1:
Location: <span of text from the output, or None if not applicable>
Explanation: <explanation>
Severity: <severity level>
‘‘‘

### Model output
{{ output }}

### Error annotations
{{ annotations }}
```

**Figure A.17** Supervisor prompt template for merging of annotation

```
### Task
Your task is to evaluate a model output for {{ task_name }} task with respect
    to {{ aspect_name }}. {{ extra_task_info }}

### Aspect Definition
{{ aspect_name }} - {{ aspect_definition }}

### Dialogue history
{{ input }}
{% if context %}
### Knowledge
{{ context }}
{% endif %}
### Response
{{ output }}

### Instructions
For any error in the output, identify its location, assign a severity level
    and provide an explanation. Report at most 8 errors. If there are more
    errors, report only the most severe ones. Finally, provide an overall
    score between 0 and 100 for {{ aspect_name }} of the output.
```

**Figure A.18** Prompt template for the fine-tuned LLM