Beyond Reference-Based Metrics: Analyzing Behaviors of Open LLMs on Data-to-Text Generation

Anonymous ACL submission

Abstract

We analyze the behaviors of open large language models (LLMs) on the task of data-to-text (D2T) generation, i.e., generating coherent and relevant text from structured data. To avoid the issue of LLM training data contamination with standard benchmarks, we design QUINTD – a tool for collecting novel structured data records from public APIs. Using a dataset collected with QUINTD and leveraging reference-free evaluation, we analyze model behaviors on five D2T generation tasks. We find that recent open LLMs (Llama2, Mistral, and Zephyr) can generate fluent and coherent text from standard data formats in zero-shot settings. However, we also show that the semantic accuracy of the outputs is a major issue: both according to our GPT-4-based metric and human annotators, more than 80% of the outputs of open LLMs contain a semantic error. We publicly release the code, data, and model outputs.[1]

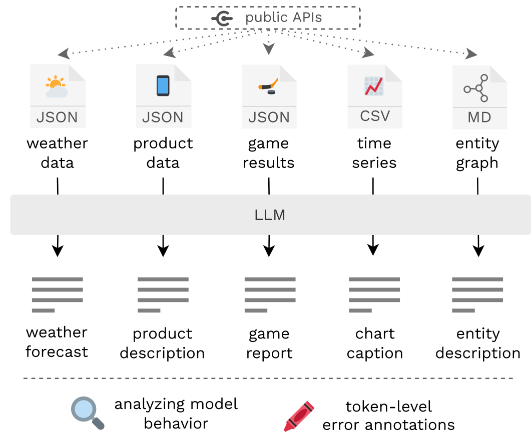

Figure 1: We experiment with using LLMs for generating text from structured data in various domains, analyzing model behavior and evaluating their output on token level.

1 Introduction

Large language models (LLMs; Ouyang et al., 2022; Touvron et al., 2023a,b; Jiang et al., 2023; Tunstall et al., 2023) have already left a mark in many areas of natural language processing (NLP). Surprisingly, their applicability to the task of data-to-text (D2T) generation (Reiter and Dale, 1997; Gatt and Krahmer, 2018) remains underexplored, with limited evaluation on a handful of well-established benchmarks only (Axelsson and Skantze, 2023; Yuan and Färber, 2023). Generating text from structured data is arguably challenging for LLMs, given the specifics of D2T generation, such as long inputs, complex non-linear structure, and strict requirements on semantic accuracy. However, a more significant issue is perhaps the lack of testing grounds. The current D2T generation benchmarks are not only getting saturated (Van Miltenburg et al., 2023), but also promote optimization towards traditional reference-based evaluation metrics, which were shown to correlate poorly with human judgment (Gehrmann et al., 2023; van der Lee et al., 2021; Novikova et al., 2017). When it comes to the models, using closed LLMs (OpenAI, 2023a,b) is increasingly considered a bad research practice due to its non-reproducibility (Rogers, 2023; Chen et al., 2023). On top of that, contamination of LLM training data with standard benchmarks further restricts the space for experiments (Golchin and Surdeanu, 2023; Aiyappa et al., 2023; Balloccu et al., 2024).

In this paper, we propose an approach that allows us to analyze model behavior in D2T generation on novel, real-world structured data records with reference-free evaluation metrics. We begin by realizing that *unlabeled data are plentiful*. To leverage the data for our experiments, we introduce QUINTD[2] – a tool for collecting structured data from five domains in standard formats: JSON,

[1] https://anonymous.4open.science/r/quintd/

[2] Quintet of Unlabeled Inputs for Natural Tasks in Data-to-text, pronounced as "quintet"

Task Id	Domain	Task Description	Source	Format
openweather	Weather	Generating a weather forecast from weather data.	OpenWeather	JSON
gsmarena	Technology	Describing a product based on its attributes.	GSM Arena	JSON
ice_hockey	Sport	Describing an outcome of an ice-hockey game.	Rapid API	JSON
owid	Health	Generating a caption for a time series.	Our World In Data	CSV
wikidata	World facts	Describing entities and relations in a knowledge graph.	Wikidata	Markdown

Table 1: The domains and tasks included in the QUINTD data collection tool we use for testing D2T generation with LLMs. In our experiments, we download 100 development and 100 test examples of input data for each domain.

CSV, and Markdown. We choose the domains so that the data can be directly used as input for five distinct D2T generation tasks. Specifically, our tasks include generating weather forecasts, sports reports, product descriptions, chart captions, and entity descriptions (see Table 1). Next, we collect a set of 1,000 inputs with QUINTD and use the inputs as an ad-hoc benchmark (called QUINTD-1) for testing the abilities of LLMs for D2T generation. We assume that the data formats in QUINTD-1 are common in the LLMs' pretraining corpora, so we specify the task using instructions instead of standard finetuning with human-written outputs, capitalizing on the zero-shot abilities of instruction-tuned LLMs (§2).

We push towards better reproducibility by *focusing on open LLMs*, which – apart from being more accessible – also achieve increasingly better results across tasks (Zheng et al., 2023; Beeching et al., 2023). For our experiments, we use three open LLMs with 7B parameters: Llama-2 (Touvron et al., 2023b; TogetherAI, 2023), Mistral (Jiang et al., 2023), and Zephyr (Tunstall et al., 2023). We also use GPT-3.5 (OpenAI, 2023b) as a closed model baseline for the final experiments. Given the behavioral nature of the experiments with LLMs (Holtzman et al., 2023), we put emphasis on reporting model behavior throughout the process (§3).

Another piece of the puzzle is *reference-free evaluation*: using the input data as a ground for comparison instead of human references (§4). For evaluation, we use manual annotations from human crowdworkers (van der Lee et al., 2021), along with a customized automatic metric based on GPT-4 (Liu et al., 2023; Chiang and Lee, 2023; Kocmi and Federmann, 2023a). To get a fine-grained picture of model errors, we annotate semantic accuracy errors on the level of individual tokens (Thomson and Reiter, 2020; Thomson et al., 2023).

Based on our results, we provide general recommendations about D2T generation with open LLMs across tasks and formats (§5). Our main findings are as follows:

- **Open LLMs can generate fluent outputs from structured data** in common formats under zero-shot settings.
- **Semantic accuracy is a major obstacle**: both human annotators and GPT-4-based metric report that over 80% of outputs of open LLMs on our data contain a semantic error.
- **Long data inputs cause practical issues**, including the need for long-context models, increased GPU memory requirements, and unavailability of few-shot approaches.
- **Outputs can be empirically improved by following several rules-of-thumb** for preprocessing the model input, such as including units, removing unnecessary fields, or prefixing the model answer.

2 Reference-Free D2T Generation

2.1 Data Collection Tool

We introduce a tool named QUINTD for collecting ad-hoc test sets using public APIs in five different domains, and we collect one such set (QUINTD-1) for our experiments. Our main reasons for departing from the traditional scheme of benchmarking on well-established datasets are:

1. Any published test sets may be potentially included in the training data of LLMs.
2. Public sources of structured data offer enough resources for creating ad-hoc test sets.
3. Without human references, our data collection scheme is lightweight and replicable.

Given the available public sources of data, we settled on the five tasks which are described in Table 1 (see Appendix A for more details). The tasks are based on structured data in common formats: JSON, CSV, and Markdown.

Prompt

```
Based on the given data:
```
{DATA}
```
Your task is to write a brief, fluent, and coherent single-paragraph {output_type} in natural language. The text should be balanced and neutral. Make sure that all the facts mentioned in the text can be derived from the input data, do *not* add any extra information.
```

Output prefix

```
Sure! Here is the {output_type}:
"
```

Figure 2: The prompt $\mathcal{P}$ and the model output prefix we used for the experiments in this paper. `DATA` is filled with the data record x and `output_type` is filled accordingly for each domain $\mathcal{D}$ (see Table 1).

2.2 QUINTD-1 Dataset

The benchmark we collected using QUINTD for our experiments in this paper (QUINTD-1) contains 500 examples in the development set and 500 examples in the test set (100 examples per domain for each split). We keep the size of the dataset moderate for a quick experimental turnaround.

We downloaded the data between November 2023 and January 2024. Note that the dataset contains only **unlabeled** data without any reference outputs (e.g., weather data, but not a textual weather forecast). New versions of the benchmark can be easily generated with the QUINTD tool we provide.

2.3 Task Definition

Each example in QUINTD-1 consists of a structured data record x from a domain $\mathcal{D} \in \{$`openweather`, `gsmarena`, `ice_hockey`, `owid`, `wikidata`$\}$. Given x and a prompt $\mathcal{P}$, the goal is to generate natural language output y faithful to the data x, according to the instructions in the prompt $\mathcal{P}$ (see Figure 2).

3 Experiments

3.1 Experimental Process

Our goal is to avoid extensive data preprocessing and prompt engineering since these steps could harm the reproducibility and generalizability of our experiments. With this goal in mind, we decided to use the same prompt template $\mathcal{P}$ for all the domains and models.

For a set of preliminary experiments, we first wrote down the initial version of the prompt and used the data without further preprocessing. We then iteratively improved our experimental setup by observing outputs on the development set. We describe all the observations and modifications we made before generating the final outputs on the test set in §3.3.

3.2 Models

For our experiments, we selected the following LLMs available under an open license:

- **Llama2** (Touvron et al., 2023a; TogetherAI, 2023), `togethercomputer/Llama-2-7B-32K-Instruct`
- **Mistral** (Jiang et al., 2023), `mistralai/Mistral-7B-Instruct-v0.1`
- **Zephyr** (Tunstall et al., 2023). `HuggingFaceH4/zephyr-7b-beta`

The models are instruction-tuned, operate with 32k context, and perform well on recent benchmarks. All the models have 7B parameters and thus fit on a single NVIDIA A40 (48G VRAM) in 16-bit precision. The models are available through HuggingFace (Wolf et al., 2020).

We accessed the models via an API provided by the `text-generation-webui` framework[3] running locally. For the final experiments, we also included GPT-3.5 (`gpt-3.5-turbo-1106`) accessed through the OpenAI API (OpenAI, 2023b).

3.3 Observations from Preliminary Experiments

During development, we made several observations which we took into account for our final experimental setup:

Any input field may appear in the output. The models do not always select the most relevant fields for the given output. For example, we observed that the models commonly mention identifiers, timestamps, files, and other metadata, leading to unnatural outputs. Due to this, we decided not to include these irrelevant fields in the input.

Units need to be specified explicitly. If the units are not specified in the data record, the models tend to resort to their best guess. This may go unnoticed if the unit is evident from the context (e.g.,

[3] `https://github.com/oobabooga/text-generation-webui`

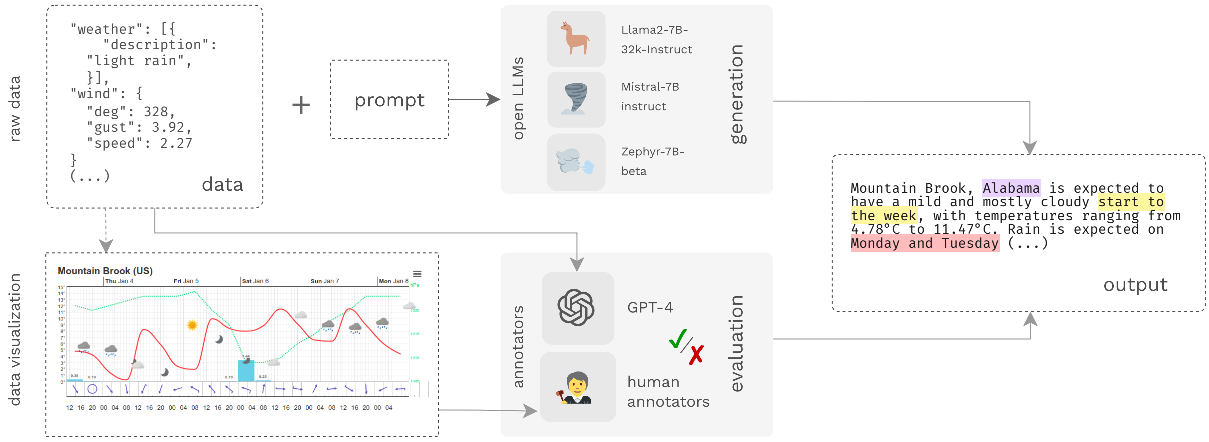

Figure 3: Our experimental setup. We first generate the outputs using LLMs that are given raw data and a task-specific prompt. We annotate the token-level semantic errors in the LLM outputs with (a) an automatic metric based on GPT-4 that matches the output to the raw data, and (b) human annotators, who annotate the errors in the output given the data visualization.

the model will usually not report the temperature in Fahrenheit instead of Celsius), but it may get problematic if the value is ambiguous (e.g., wind speed in km/h versus m/s). Therefore, we explicitly add units to all data records where appropriate.

Understandable labels are enough. On the flip side, we decided not to add extra descriptions to the keys if the key was understandable from the label (e.g., `homeTeam` or `dimensions`). As discussed by Kasner et al. (2023), pretrained models tend to interpret the fields correctly as long as the label is human-readable. We only decided to include chart metadata for the CSV files in the `owid` domain.

Long inputs can be troublesome. The inputs in some domains can easily get longer than 10-20k tokens. This issue is amplified by the fact that the evaluated LLMs tokenize numbers into individual digits. To accommodate for the long inputs, we picked models that accept up to 32k tokens.[4] However, with long inputs, the GPU memory consumption also gets considerably higher, so we needed to downsample the data in `owid` and `openweather` to keep their length under ~8k tokens.

Few-shot experiments are infeasible. Due to the context-length limitations described above, we were not able to conduct few-shot experiments since we could not robustly fit an additional $(x_{\text{example}}, y_{\text{example}})$ pair in the prompt. We attempted to include only y_{example} (making the setup "half-shot"), but we observed that the models then tended to use entities from the example (unrelated to the actual input) in their outputs. Therefore, we decided not to follow this line of experiments (see §5.3 for discussion).

Deterministic decoding and sampling are on par. In our preliminary experiments, we observed a roughly similar output quality for both deterministic decoding and sampling.[5] For the final experiments, we decided to use deterministic decoding, which is non-parametric and conceptually more suitable for D2T generation.

Prefixing the output makes parsing easier. Even with variations of a *"generate only the output"* instruction appended to the prompt, the models (especially Llama2) tended to first confirm the request. For that reason, we decided to prefix the input for all the models with *"Sure! Here is the {output_type}: ""*. The opening quote at the end of the prefix allowed us to robustly parse the text simply by stripping the closing quote from the model output.

The outputs are fluent but inaccurate. We observed that the vast majority of model outputs were grammatically and stylistically correct, capturing the output type specified in the prompt. However, we also noticed that the outputs contained many factual errors (even after emphasizing the focus on factual accuracy in the prompt, see Figure 2). This

[4] For this reason, we use Llama-2-7B-32k with 32k token context (TogetherAI, 2023) instead of the official Llama-2-7B-Instruct, which only supports 4k context (Touvron et al., 2023b).

[5] We used the `text-generation-webui` default decoding parameters: `temperature`=0.7, `top_p`=0.9, and `top_k`=20.

observation led us to evaluate the model outputs using token-level annotations focused on semantic accuracy errors (Reiter and Thomson, 2020).

3.4 Final Experiments

Taking the observations in §3.3 into account, we proceeded to generate the outputs on the test set of QUINTD-1 for token-level error analysis. We first preprocessed the data as mentioned: we stripped out unnecessary fields, added units, and downsampled the data to fit the context. For all the models mentioned in §3.2, we used the prompt in Figure 2 and deterministic decoding with a maximum length of 512 tokens.

For comparison, we also generated outputs for the same inputs and identical prompts with GPT-3.5.[6] Note that even though we fixed the temperature and seed to 0, the rest of the decoding parameters are inaccessible to us and may differ from the parameters we used for the open models.

4 Evaluation

For evaluation, we focus on *semantic accuracy* errors. We compare the generated texts to the input data, looking for parts of texts that are not faithful to the input data. We annotate the errors on the token level, considering all the tokens in the output text as potential sources of errors.

Regarding the error taxonomy, we settled on four error categories: `INCORRECT`, `NOT_CHECKABLE`, `MISLEADING`, and `OTHER`. The taxonomy is inspired by the methodology discussed in Thomson and Reiter (2020) and Thomson et al. (2023). To keep the annotation tractable, we decided not to distinguish between fine-grained categories (e.g., *incorrect name* vs. *incorrect number*). The descriptions of our error categories, as presented in the instructions for annotation, are included in Table 2.

We employ two complementary evaluation schemes:

- $\mathcal{E}_{\text{gpt}}$: **an automatic metric** based on GPT-4 (§4.1),
- $\mathcal{E}_{\text{hum}}$: **human evaluation** based on crowdsourcing (§4.2).

These two schemes are based on similar instructions and produce (nearly) equivalent outputs. The main idea in introducing multiple schemes is to compensate for the shortcomings of each approach and thus increase the replicability and robustness of our results.

4.1 GPT-4-based Evaluation

LLM-based metrics can be customized for a particular task without the need for training data. For our experiments, we employ a metric based on GPT-4 (`gpt-4-1106-preview`, OpenAI, 2023a), which was shown to be superior in following fine-grained instructions compared to other LLMs and to have high correlations with human judgment on evaluating generated texts (Zhao et al., 2023; Sottana et al., 2023; Kocmi and Federmann, 2023a,b).

$\mathcal{E}_{\text{gpt}}$ is instantiated using a prompt and a system message describing the task. We instruct the model to produce a JSON output with sequentially ordered errors using the following format:

```
{
  "errors": [{
    "reason": [REASON],
    "text": [TEXT_SPAN],
    "type": [ERROR_CATEGORY]
    },
    ...]
}.
```

Note that we require that the model first generates the free-form text *reason* for the error. Generating the reason incurs almost no extra cost, and our cursory observations suggest that requiring it leads to more precise outputs.

Concerning the alignment of the model outputs with the original text, we perform matching on `TEXT_SPAN`. We ensure that the model response is a valid JSON using OpenAI's `response_format` parameter. See Appendix B for more details about the metric, including the prompt and the system message.

4.2 Human-based Evaluation

An automatic metric based on a closed LLM makes the evaluation potentially non-reproducible and biased (Kocmi and Federmann, 2023a; Wang et al., 2023b), for which we compensate by obtaining annotations from human annotators.

For the human annotation metric $\mathcal{E}_{\text{hum}}$, we prepared a custom web interface where an annotator is able to annotate a text span with a selected error category. We created custom visualizations for each data format. Unlike with $\mathcal{E}_{\text{gpt}}$, we did not ask the crowdworkers for free-form reasoning about

[6] We did not include GPT-3.5 in our preliminary experiments since closed models are not our focus. We also did not use GPT-4 because we reserve the model for evaluation; see §4.1.

Error	Description
`INCORRECT`	The fact in the text contradicts the data.
`NOT_CHECKABLE`	The fact in the text cannot be checked given the data.
`MISLEADING`	The fact in the text is misleading in the given context.
`OTHER`	The text is problematic for another reason, e.g., grammatically or stylistically incorrect, irrelevant, or repetitive.
Example	
data	**Nokia 3310** | *color*: black, blue, grey | *display*: 320x240px
text	Nokia 3310 is produced in Finland and features a 320x320 display. It is available in black color. The data seem to provide only partial information about the phone.

Table 2: Categories of errors annotated in our evaluation and an example demonstrating the error types. See Appendix C for an explanation of individual errors in the example.

the errors since that would make the annotation more complex.

We hired annotators on the Prolific crowdsourcing platform.[7] In total, we hired 100 annotators, each annotating 20 examples (4 model outputs for each of the five domains). We selected annotators with at least 10 completed tasks and a 100% approval rate, having English as their primary language. We paid the annotators £9 per hour, according to the platform's recommendations. The median time for completing the annotations was 47 minutes. See Appendix C for the instructions for the annotators and the annotation interface and Appendix E for the data visualizations.

5 Results and Discussion

A summary of the token-level annotations is in Table 3 and 4, with detailed results per domain provided in Appendix F.

5.1 How Accurate Are the Model Outputs?

Depending on the model, between 74-85% of examples contain an error according to $\mathcal{E}_{\text{hum}}$, suggesting that open LLMs make semantic errors very often. According to $\mathcal{E}_{\text{gpt}}$, the number is as high as 88-93%.

The most common error type is `INCORRECT`. As shown in Table 3, all the open LLMs make more than **two statements contradicting the data per output on average**. The `NOT_CHECKABLE` errors are also relatively common: more than one per output on average according to $\mathcal{E}_{\text{hum}}$, and at least one being present in more than 26% of examples according to both metrics.

The results vary widely according to the domain (see Appendix F). For example, the outputs in `wikidata` contain much more `NOT_CHECKABLE` errors on average (1.54 per output according to $\mathcal{E}_{\text{hum}}$) than `INCORRECT` errors (0.12 per output according to $\mathcal{E}_{\text{hum}}$), suggesting that with simpler inputs, the models tend to introduce extra information. The `openweather` domain seems to be the most complex with the longest outputs (~164 tokens), more than eight errors in the output on average, and >90% of outputs containing an error.

The differences between the open LLMs are not major. Out of the open LLMs, Zephyr has the best results across categories and metrics, followed by Llama2. However, the outputs of Mistral are longer on average, leaving more space for errors. GPT-3.5 (which we consider separately) does generally better according to both $\mathcal{E}_{\text{gpt}}$ and $\mathcal{E}_{\text{hum}}$, although it still makes an error in 60-75% of examples (2 errors per example on average). In general, the results show that LLMs make too many semantic errors to be usable in practice for D2T generation in a zero-shot setting.

5.2 Do Evaluation Methods Agree?

To quantify the agreement of our evaluation metrics, we computed the Pearson correlation coefficient between the error counts on the level of tokens, examples, and domains (see Appendix D for details). The correlation on the level of tokens is weak ($r_{\text{token}} = 0.26$) but gets better on the example-level ($r_{\text{example}} = 0.55$) and even better on the domain-level ($r_{\text{domain}} = 0.92$). In Table 5, we show the percentage of tokens marked by individual metrics. The metrics agree on the specific tokens in less than 6%, although they both mark around 21% of tokens as erroneous.

We also measure inter-annotator agreement between human annotators. For that, we obtained annotations from two annotators for 100 model outputs. The results are similar: the annotators agree

[7] https://prolific.com

	Incorrect		Not Checkable		Misleading		Other		All categories		
	$\mathcal{E}_{\text{gpt}}$	$\mathcal{E}_{\text{hum}}$	$\mathcal{E}_{\text{gpt}}$	$\mathcal{E}_{\text{hum}}$	$\mathcal{E}_{\text{gpt}}$	$\mathcal{E}_{\text{hum}}$	$\mathcal{E}_{\text{gpt}}$	$\mathcal{E}_{\text{hum}}$	$\mathcal{E}_{\text{gpt}}$	$\mathcal{E}_{\text{hum}}$	**# Tok.**
Llama2	**2.46**	1.57	0.90	1.25	**0.20**	0.25	0.13	**0.10**	3.70	3.18	83.8
Mistral	2.80	2.03	0.52	1.12	0.37	0.44	0.11	0.25	3.80	3.85	114.9
Zephyr	2.50	**1.44**	**0.40**	**0.77**	0.39	**0.20**	**0.06**	0.16	**3.35**	**2.58**	98.0
GPT-3.5	1.57	0.65	0.32	0.49	0.42	0.18	0.02	0.07	2.32	1.39	84.9

Table 3: The average *numbers of errors per output* (lower is better) based on GPT-4 ($\mathcal{E}_{\text{gpt}}$) and human annotators ($\mathcal{E}_{\text{hum}}$). We also include the average number of tokens per output in the rightmost column. The results of the best open LLM are emphasized.

	Incorrect		Not Checkable		Misleading		Other		All categories	
	$\mathcal{E}_{\text{gpt}}$	$\mathcal{E}_{\text{hum}}$	$\mathcal{E}_{\text{gpt}}$	$\mathcal{E}_{\text{hum}}$	$\mathcal{E}_{\text{gpt}}$	$\mathcal{E}_{\text{hum}}$	$\mathcal{E}_{\text{gpt}}$	$\mathcal{E}_{\text{hum}}$	$\mathcal{E}_{\text{gpt}}$	$\mathcal{E}_{\text{hum}}$
Llama2	0.78	0.53	0.46	0.57	**0.15**	0.17	0.09	**0.08**	0.92	0.86
Mistral	0.78	0.54	0.32	0.50	0.23	0.21	0.08	0.14	0.93	0.81
Zephyr	**0.77**	**0.47**	**0.26**	**0.42**	0.27	**0.16**	**0.04**	0.12	**0.88**	**0.76**
GPT-3.5	0.64	0.38	0.21	0.29	0.26	0.13	0.02	0.06	0.75	0.61

Table 4: The ratio of *outputs containing at least one error* (lower is better) based on GPT-4 ($\mathcal{E}_{\text{gpt}}$) and human annotators ($\mathcal{E}_{\text{hum}}$). The results of the best open LLM are emphasized.

	$\mathcal{E}_{\text{gpt}}$	$\mathcal{E}_{\text{hum}}$	$\mathcal{E}_{\text{gpt}} + \mathcal{E}_{\text{hum}}$
Incorrect	0.135	0.099	0.040
Not checkable	0.039	0.076	0.016
Misleading	0.022	0.022	0.001
Other	0.008	0.018	0.001
All categories	0.204	0.214	0.059

Table 5: The ratio of *tokens marked as erroneous* by GPT-4 ($\mathcal{E}_{\text{gpt}}$), human annotators ($\mathcal{E}_{\text{hum}}$), and both metrics at the same time ($\mathcal{E}_{\text{gpt}} + \mathcal{E}_{\text{hum}}$).

weakly on the token level ($r_{\text{token}} = 0.36$), stronger on the example level ($r_{\text{example}} = 0.53$), and even stronger on the domain level ($r_{\text{domain}} = 0.85$). We conclude that while the details regarding error spans and categories may vary, the annotators as well as GPT-4 generally agree on the accuracy of model outputs for a given set of examples. In the future, the agreement could be improved by measuring errors on the phrase level (Vamvas and Sennrich, 2022).

5.3 Recommendations and Directions

Forget fluency, solve accuracy. The output of LLMs is satisfactory regarding the style, format, and purpose of the text. However, the amount of semantic errors remains very high. Improving the semantic accuracy of the models (Li et al., 2022), along with new model-based evaluation metrics (Liu et al., 2023; Xu et al., 2023), could thus help to bring improve LLM-based D2T generation systems where it is most needed.

Use efficient models. The memory issues with long context, making few-shot experiments infeasible, can potentially be solved by using more efficient long-context models equipped with Flash Attention (Dao et al., 2022) and fast inference libraries such as `llama.cpp`.[8] A potentially suitable open LLM in this respect is the long-context Llama2 (Xiong et al., 2023; model to be released).

Test the models in the wild. Except for using an ad-hoc dataset of real-world data as we did in our work, the ecological validity of D2T evaluation can also be ensured by continuous evaluation with human users (Zheng et al., 2023) and evaluating the real-world impact of the systems (Reiter, 2023).

Multilinguality is an opportunity. With the recent efforts in extending D2T generation to low-resource languages (Cripwell et al., 2023), multilingual D2T generation with open LLMs seems a promising direction. Although we did not go beyond English, initial steps were already done by works such as Lorandi and Belz (2023).

Be careful about subtle bugs. During our preliminary experiments, we uncovered subtle bugs in API calls such as incorrect instruction templates[9] or involuntary input truncation. With the apparent ease of API access and robustness of LLMs, such bugs could go unnoticed and artificially skew the

[8] https://github.com/ggerganov/llama.cpp

[9] https://huggingface.co/docs/transformers/chat_templating

model performance.

6 Related Work

6.1 D2T Generation Tasks

Weather Forecasts First attempts for generating weather forecasts include template-based and statistical approaches (Belz, 2005, 2008; Angeli et al., 2010) for the Sumtime-meteo and WeatherGov datasets (Sripada et al., 2002; Liang et al., 2009). More recently, Balakrishnan et al. (2019) introduced a weather forecast dataset with tree-structured meaning representations. Our weather forecasts are less structured and based on a 5-day weather outlook.

Product Descriptions Our phone specifications are closest to Wen et al. (2015, 2016), who introduced a dataset for generating descriptions of laptops and TVs. Their solution was based on recurrent neural networks, although templates remained a go-to approach for the task (Wang et al., 2017). Recently, Shao et al. (2021) and Koto et al. (2022) also proposed specialized architectures based on pretrained language models for the data from big e-commerce platforms.

Sport Reports All the D2T generation datasets from the Rotowire family (Wiseman et al., 2017; Wang, 2019), including SportSett:Basketball (Thomson et al., 2021), and ESPN-NBA (Nie et al., 2018) focus on generating basketball reports. Along with MLB (Puduppully et al., 2019b), these datasets belong among the most challenging D2T datasets, attracting various neural-based solutions (Puduppully et al., 2019a, 2022; Puduppully and Lapata, 2021; Rebuffel et al., 2020). We use instead simpler data covering ice hockey game summaries.

Chart Captions Following the early rule-based approaches (Demir et al., 2008, 2012), the approaches for chart captioning recently tackle large-scale datasets from data analytic institutions (Obeid and Hoque, 2020; Kantharaj et al., 2022). We focus on one of the tasks from Sharma et al. (2021), which is generating descriptions of time series in the health domain.

Entity Descriptions The task of generating descriptions for a knowledge graph has been covered extensively in D2T generation (Gardent et al., 2017; Ferreira et al., 2020; Agarwal et al., 2021; Chen et al., 2020; Ribeiro et al., 2020, *inter alia*). Our task is to describe an entity provided a list of its properties, which is closely related to generating entity descriptions from Wikipedia infotables (Lebret et al., 2016).

6.2 D2T Generation with LLMs

Recent works have focused on exploring the capabilities of closed LLMs on existing D2T generation datasets. Axelsson and Skantze (2023) evaluated GPT-3.5 (OpenAI, 2023b) on WebNLG, along with Yuan and Färber (2023), who also tested the model on the AGENDA dataset (Koncel-Kedziorski et al., 2019). Both works found that regardless of potential data contamination, the LLMs rank behind state-of-the-art finetuned models on automatic metrics. Zhao et al. (2023) tested closed models on modified table-to-text generation datasets and found out that in terms of faithfulness, GPT-4 can outperform state-of-the-art models.

6.3 Beyond Reference-Based Metrics

Many works have recently investigated the potential of using LLMs for automatic reference-free evaluation of a generated text, generally achieving high correlations with human judgment (Zhao et al., 2023; Sottana et al., 2023; Kocmi and Federmann, 2023a,b; Chiang and Lee, 2023; Wang et al., 2023a; Fu et al., 2023). However, they also voice concerns about its non-reproducibility (Kocmi and Federmann, 2023a) and potential bias of these models (Wang et al., 2023b).

Holtzman et al. (2023) suggest that the research on LLMs should move away from reporting benchmark scores, investigating model behaviors instead. In this vein, Upadhyay and Massie (2022) analyzed the ability of models to produce different types of content in D2T generation. Regarding human evaluation, Thomson and Reiter (2020) proposed a protocol for reference-free token-level annotation of complex D2T generation output.

7 Conclusion

We provided an exploratory study into D2T generation with open LLMs. We proposed new directions for D2T generation, including using ad-hoc test sets, data in common formats, and reference-free evaluation. By a combination of GPT-4-based metric and human evaluation, we evaluated the performance of LLMs on five domains, providing token-level annotations of model outputs across five domains and recommendations for future directions in D2T generation.

Limitations

In our work, we do not include a comparison to other D2T generation approaches. The main reason is that our benchmark is reference-free, while a large majority of prior approaches are based on models finetuned on reference outputs. However, we believe that our work still satisfies our main goal of providing insights into behaviors of open LLM models on D2T generation.

We acknowledge that reference-free metrics currently have various shortcomings, including reliance on closed models or specific human annotation protocols, leading to limited replicability and a high price of execution. None of the approaches also produces flawless outcomes and have only moderate correlations with each other. We believe that these shortcomings will be addressed in the future with open model-based metrics.

Our choice of models is limited to 7B-parameter open LLMs due to our limited computational resources. Also, unlike some other LLMs such as GPT-Neo (Black et al., 2022) or BLOOM (BigScience Workshop et al., 2022), the models we used do not disclose the data they were trained on. For this reason, we find it ever more important to test the models on benchmarks whose labels could have *not* been included in their training data.

The approaches based on LLMs may produce factually incorrect information. Any text produced by the LLMs therefore needs to be carefully examined, and no decisions should be based on the generated text alone.

Ethical Considerations

The human evaluation study was approved by the internal ethics committee of our institution. The annotators were hired over Prolific and paid the platform-recommended wage of 9 GBP/hour. The annotators were preselected based on their primary language (English) and their country of residence (US, UK, Ireland, Australia, New Zealand). All annotators were shown detailed instructions and explanation of the data types, data sources, and the purpose of the research (see Appendix C for details). The domains in QUINTD were selected so that they do not contain any sensitive or potentially offensive content. We do not collect any demographic data about the participants.

References

Oshin Agarwal, Heming Ge, Siamak Shakeri, and Rami Al-Rfou. 2021. Knowledge Graph Based Synthetic Corpus Generation for Knowledge-enhanced Language Model Pre-training. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021*, pages 3554–3565, Online.

Rachith Aiyappa, Jisun An, Haewoon Kwak, and Yong-yeol Ahn. 2023. Can we trust the evaluation on ChatGPT? In *Proceedings of the 3rd Workshop on Trustworthy Natural Language Processing (TrustNLP 2023)*, pages 47–54, Toronto, Canada.

Gabor Angeli, Percy Liang, and Dan Klein. 2010. A Simple Domain-independent Probabilistic Approach to Generation. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, EMNLP 2010, 9-11 October 2010, MIT Stata Center, Massachusetts, USA, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 502–512.

Agnes Axelsson and Gabriel Skantze. 2023. Using Large Language Models for Zero-shot Natural Language Generation from Knowledge Graphs. *CoRR*, abs/2307.07312.

Anusha Balakrishnan, Jinfeng Rao, Kartikeya Upasani, Michael White, and Rajen Subba. 2019. Constrained Decoding for Neural NLG from Compositional Representations in Task-oriented Dialogue. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Volume 1: Long Papers*, pages 831–844, Florence, Italy.

Simone Balloccu, Patrícia Schmidtová, Mateusz Lango, and Ondřej Dušek. 2024. Leak, cheat, repeat: Data contamination and evaluation malpractices in closed-source llms. *arXiv preprint arXiv:2402.03927*.

Edward Beeching, Clémentine Fourrier, Nathan Habib, Sheon Han, Nathan Lambert, Nazneen Rajani, Omar Sanseviero, Lewis Tunstall, and Thomas Wolf. 2023. Open LLM Leaderboard. https://huggingface.co/spaces/HuggingFaceH4/open_llm_leaderboard.

Anja Belz. 2005. Corpus-driven generation of weather forecasts. In *Proc. 3rd Corpus Linguistics Conference*.

Anja Belz. 2008. Automatic generation of weather forecast texts using comprehensive probabilistic generation-space models. *Nat. Lang. Eng.*, 14(4):431–455.

BigScience Workshop, Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, et al. 2022. Bloom: A 176b-parameter open-access multilingual language model. *arXiv preprint arXiv:2211.05100*.

Sidney Black, Stella Biderman, Eric Hallahan, Quentin Anthony, Leo Gao, Laurence Golding, Horace He, Connor Leahy, Kyle McDonell, Jason Phang, et al. 2022. Gpt-neox-20b: An open-source autoregressive language model. In *Proceedings of BigScience Episode# 5–Workshop on Challenges & Perspectives in Creating Large Language Models*, pages 95–136.

Armand Boschin and Thomas Bonald. 2019. Wikidatasets: standardized sub-graphs from wikidata. *arXiv preprint arXiv:1906.04536*.

Lingjiao Chen, Matei Zaharia, and James Zou. 2023. How is ChatGPT's behavior changing over time? *CoRR*, abs/2307.09009.

Wenhu Chen, Yu Su, Xifeng Yan, and William Yang Wang. 2020. KGPT: Knowledge-grounded Pre-training for Data-to-text Generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020*, pages 8635–8648, Online.

David Cheng-Han Chiang 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), ACL 2023*, pages 15607–15631, Toronto, Canada.

Liam Cripwell, Anya Belz, Claire Gardent, Albert Gatt, Claudia Borg, Marthese Borg, John Judge, Michela Lorandi, Anna Nikiforovskaya, and William Soto Martinez. 2023. The 2023 WebNLG Shared Task on Low Resource Languages. Overview and Evaluation Results (WebNLG 2023). In *Proceedings of the Workshop on Multimodal, Multilingual Natural Language Generation and Multilingual WebNLG Challenge (MM-NLG 2023)*, pages 55–66, Prague, Czech Republic.

Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. 2022. FlashAttention: Fast and Memory-efficient Exact Attention with IO-awareness. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022*, New Orleans, LA, USA.

Seniz Demir, Sandra Carberry, and Kathleen F. McCoy. 2008. Generating Textual Summaries of Bar Charts. In *INLG 2008 - Proceedings of the Fifth International Natural Language Generation Conference, June 12-14, 2008, Salt Fork*, Ohio, USA.

Seniz Demir, Sandra Carberry, and Kathleen F. McCoy. 2012. Summarizing Information Graphics Textually. *Comput. Linguistics*, 38(3):527–574.

Thiago Ferreira, Claire Gardent, Nikolai Ilinykh, Chris van der Lee, Simon Mille, Diego Moussallem, and Anastasia Shimorina. 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+)*.

Jinlan Fu, See-Kiong Ng, Zhengbao Jiang, and Pengfei Liu. 2023. GPTScore: Evaluate as You Desire. *CoRR*, abs/2302.04166.

Claire Gardent, Anastasia Shimorina, Shashi Narayan, and Laura Perez-Beltrachini. 2017. The WebNLG Challenge: Generating Text from RDF Data. In *Proceedings of the 10th International Conference on Natural Language Generation, INLG 2017, Santiago de Compostela*, pages 124–133, Spain.

Albert Gatt and Emiel Krahmer. 2018. Survey of the State of the Art in Natural Language Generation: Core tasks, applications and evaluation. *J. Artif. Intell. Res.*, 61:65–170.

Sebastian Gehrmann, Elizabeth Clark, and Thibault Sellam. 2023. Repairing the Cracked Foundation: A Survey of Obstacles in Evaluation Practices for Generated Text. *J. Artif. Intell. Res.*, 77:103–166.

Shahriar Golchin and Mihai Surdeanu. 2023. Time Travel in LLMs: Tracing Data Contamination in Large Language Models. *CoRR*, abs/2308.08493.

Ari Holtzman, Peter West, and Luke Zettlemoyer. 2023. Generative Models as a Complex Systems Science: How can we make sense of large language model behavior? *CoRR*, abs/2308.00189.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7B. *CoRR*, abs/2310.06825.

Shankar Kantharaj, Rixie Tiffany Ko Leong, Xiang Lin, Ahmed Masry, Megh Thakkar, Enamul Hoque, and Shafiq R. Joty. 2022. Chart-to-Text: A Large-scale Benchmark for Chart Summarization. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022*, pages 4005–4023, Dublin, Ireland.

Zdeněk Kasner, Ioannis Konstas, and Ondřej Dušek. 2023. Mind the Labels: Describing Relations in Knowledge Graphs With Pretrained Models. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2023, Dubrovnik*, pages 2390–2407, Croatia.

Tom Kocmi and Christian Federmann. 2023a. GEMBA-MQM: Detecting Translation Quality Error Spans with GPT-4. In *Proceedings of the Eighth Conference on Machine Translation, WMT 2023*, pages 768–775, Singapore.

Tom Kocmi and Christian Federmann. 2023b. Large Language Models Are State-of-the-art Evaluators of Translation Quality. In *Proceedings of the 24th Annual Conference of the European Association for Machine Translation, EAMT 2023*, pages 193–203, Tampere, Finland.

Rik Koncel-Kedziorski, Dhanush Bekal, Yi Luan, Mirella Lapata, and Hannaneh Hajishirzi. 2019. Text Generation from Knowledge Graphs with Graph Transformers. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, Volume 1 (Long and Short Papers)*, pages 2284–2293, USA.

Fajri Koto, Jey Han Lau, and Timothy Baldwin. 2022. Can Pretrained Language Models Generate Persuasive, Faithful, and Informative Ad Text for Product Descriptions? In *Proceedings of the Fifth Workshop on E-Commerce and NLP (ECNLP 5)*, pages 234–243, Dublin, Ireland.

Rémi Lebret, David Grangier, and Michael Auli. 2016. Neural Text Generation from Structured Data with Application to the Biography Domain. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016*, pages 1203–1213, Austin, Texas, USA.

Wei Li, Wenhao Wu, Moye Chen, Jiachen Liu, Xinyan Xiao, and Hua Wu. 2022. Faithfulness in Natural Language Generation: A Systematic Survey of Analysis, Evaluation and Optimization Methods. *CoRR*, abs/2203.05227.

Percy Liang, Michael I. Jordan, and Dan Klein. 2009. Learning Semantic Correspondences with Less Supervision. In *ACL 2009, Proceedings of the 47th Annual Meeting of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing of the AFNLP, 2-7 August 2009*, pages 91–99, Singapore.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 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, EMNLP 2023*, pages 2511–2522, Singapore.

Michela Lorandi and Anja Belz. 2023. Data-to-text Generation for Severely Under-resourced Languages with GPT-3.5: A Bit of Help Needed from Google Translate (WebNLG 2023). In *Proceedings of the Workshop on Multimodal, Multilingual Natural Language Generation and Multilingual WebNLG Challenge (MM-NLG 2023)*, pages 80–86.

Feng Nie, Jinpeng Wang, Jin-Ge Yao, Rong Pan, and Chin-Yew Lin. 2018. Operation-guided Neural Networks for High Fidelity Data-To-text Generation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3879–3889, Brussels, Belgium.

Jekaterina Novikova, Ondřej Dušek, Amanda Cercas Curry, and Verena Rieser. 2017. Why We Need New Evaluation Metrics for NLG. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017*, pages 2241–2252, Copenhagen, Denmark.

Jason Obeid and Enamul Hoque. 2020. Chart-to-Text: Generating Natural Language Descriptions for Charts by Adapting the Transformer Model. In *Proceedings of the 13th International Conference on Natural Language Generation, INLG 2020*, pages 138–147, Dublin, Ireland.

OpenAI. 2023a. GPT-4 Technical Report. *CoRR*, abs/2303.08774.

OpenAI. 2023b. Introducing ChatGPT. `https://openai.com/blog/chatgpt`. Accessed on January 9, 2024.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022*, New Orleans, LA, USA.

Ratish Puduppully, Li Dong, and Mirella Lapata. 2019a. Data-to-text Generation with Content Selection and Planning. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019*, pages 6908–6915, Honolulu, Hawaii, USA.

Ratish Puduppully, Li Dong, and Mirella Lapata. 2019b. Data-to-text Generation with Entity Modeling. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Volume 1: Long Papers*, pages 2023–2035, Florence, Italy.

Ratish Puduppully, Yao Fu, and Mirella Lapata. 2022. Data-to-text Generation with Variational Sequential Planning. *Trans. Assoc. Comput. Linguistics*, 10:697–715.

Ratish Puduppully and Mirella Lapata. 2021. Data-to-text Generation with Macro Planning. *Trans. Assoc. Comput. Linguistics*, 9:510–527.

Clément Rebuffel, Laure Soulier, Geoffrey Scoutheeten, and Patrick Gallinari. 2020. A Hierarchical Model for Data-to-text Generation. In *Advances in Information Retrieval - 42nd European Conference on IR Research, ECIR 2020, Lisbon, Portugal, April 14-17, 2020, Proceedings, Part I*, volume 12035 of *Lecture Notes in Computer Science*, pages 65–80.

Ehud Reiter. 2023. We should evaluate real-world impact! `https://ehudreiter.com/2023/11/13/evaluate-real-world-impact/`. Accessed on January 11, 2024.

Ehud Reiter and Robert Dale. 1997. Building applied natural language generation systems. *Nat. Lang. Eng.*, 3(1):57–87.

Ehud Reiter and Craig Thomson. 2020. Shared Task on Evaluating Accuracy. In *Proceedings of the 13th International Conference on Natural Language Generation, INLG 2020*, pages 227–231, Dublin, Ireland.

Leonardo F. R. Ribeiro, Martin Schmitt, Hinrich Schütze, and Iryna Gurevych. 2020. Investigating Pretrained Language Models for Graph-to-text Generation. *CoRR*, abs/2007.08426.

Anna Rogers. 2023. Closed AI Models Make Bad Baselines. `https://hackingsemantics.xyz/2023/closed-baselines/`. Accessed on January 11, 2024.

Huajie Shao, Jun Wang, Haohong Lin, Xuezhou Zhang, Aston Zhang, Heng Ji, and Tarek F. Abdelzaher. 2021. Controllable and Diverse Text Generation in E-commerce. In *WWW '21: The Web Conference 2021*, pages 2392–2401, Virtual Event / Ljubljana, Slovenia.

Mandar Sharma, John S. Brownstein, and Naren Ramakrishnan. 2021. TCube: Domain-agnostic Neural Time-series Narration. *CoRR*, abs/2110.05633.

Andrea Sottana, Bin Liang, Kai Zou, and Zheng Yuan. 2023. Evaluation Metrics in the Era of GPT-4: Reliably Evaluating Large Language Models on Sequence to Sequence Tasks. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023*, pages 8776–8788, Singapore.

Somayajulu Sripada, Ehud Reiter, Jim Hunter, and Jin Yu. 2002. Sumtime-meteo: Parallel corpus of naturally occurring forecast texts and weather data. *Computing Science Department, University of Aberdeen, Aberdeen, Scotland, Tech. Rep. AUCS/TR0201*.

Craig Thomson and Ehud Reiter. 2020. A Gold Standard Methodology for Evaluating Accuracy in Data-To-text Systems. In *Proceedings of the 13th International Conference on Natural Language Generation, INLG 2020*, pages 158–168, Dublin, Ireland.

Craig Thomson, Ehud Reiter, and Somayajulu Sripada. 2021. SportSett:Basketball - A Robust and Maintainable Dataset for Natural Language Generation. page 9.

Craig Thomson, Ehud Reiter, and Barkavi Sundararajan. 2023. Evaluating factual accuracy in complex data-to-text. *Computer Speech & Language*, 80:101482.

TogetherAI. 2023. Preparing for the era of 32K context: Early learnings and explorations. `https://www.together.ai/blog/llama-2-7b-32k`. Accessed on January 2, 2024.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurélien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023a. LLaMA: Open and Efficient Foundation Language Models. *CoRR*, abs/2302.13971.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurélien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023b. Llama 2: Open Foundation and Fine-tuned Chat Models. *CoRR*, abs/2307.09288.

Lewis Tunstall, Edward Beeching, Nathan Lambert, Nazneen Rajani, Kashif Rasul, Younes Belkada, Shengyi Huang, Leandro von Werra, Clémentine Fourrier, Nathan Habib, Nathan Sarrazin, Omar Sanseviero, Alexander M. Rush, and Thomas Wolf. 2023. Zephyr: Direct Distillation of LM Alignment. *CoRR*, abs/2310.16944.

Ashish Upadhyay and Stewart Massie. 2022. Content Type Profiling of Data-to-text Generation Datasets. In *Proceedings of the 29th International Conference on Computational Linguistics, COLING 2022*, pages 5770–5782, Gyeongju, Republic of Korea.

Jannis Vamvas and Rico Sennrich. 2022. As little as possible, as much as necessary: Detecting over-and undertranslations with contrastive conditioning. In *60th Annual Meeting of the Association for Computational Linguistics*, pages 490–500. Association for Computational Linguistics.

Chris van der Lee, Albert Gatt, Emiel van Miltenburg, and Emiel Krahmer. 2021. Human evaluation of automatically generated text: Current trends and best practice guidelines. *Comput. Speech Lang.*, 67:101151.

Emiel Van Miltenburg, Miruna Clinciu, Ondřej Dušek, Dimitra Gkatzia, Stephanie Inglis, Leo Leppänen, Saad Mahamood, Stephanie Schoch, Craig Thomson, and Luou Wen. 2023. Barriers and enabling factors for error analysis in NLG research. *Northern European Journal of Language Technology*, 9.

Hongmin Wang. 2019. Revisiting Challenges in Data-to-text Generation with Fact Grounding. In *Proceedings of the 12th International Conference on Natural Language Generation, INLG 2019*, pages 311–322, Tokyo, Japan.

Jiaan Wang, Yunlong Liang, Fandong Meng, Haoxiang Shi, Zhixu Li, Jinan Xu, Jianfeng Qu, and Jie Zhou. 2023a. Is ChatGPT a Good NLG Evaluator? A Preliminary Study. *CoRR*, abs/2303.04048.

Jinpeng Wang, Yutai Hou, Jing Liu, Yunbo Cao, and Chin-Yew Lin. 2017. A Statistical Framework for Product Description Generation. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing, IJCNLP 2017, Volume 2: Short Papers*, pages 187–192, Taipei, Taiwan.

Peiyi Wang, Lei Li, Liang Chen, Dawei Zhu, Binghuai Lin, Yunbo Cao, Qi Liu, Tianyu Liu, and Zhifang Sui. 2023b. Large Language Models are not Fair Evaluators. *CoRR*, abs/2305.17926.

Tsung-Hsien Wen, Milica Gašic, Nikola Mrkšic, Lina M Rojas-Barahona, Pei-Hao Su, David Vandyke, and Steve Young. 2015. Toward multi-domain language generation using recurrent neural networks. In *NIPS Workshop on Machine Learning for Spoken Language Understanding and Interaction*.

Tsung-Hsien Wen, Milica Gasic, Nikola Mrksic, Lina Maria Rojas-Barahona, Pei-Hao Su, David Vandyke, and Steve J. Young. 2016. Multi-domain Neural Network Language Generation for Spoken Dialogue Systems. In *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 120–129, San Diego California, USA.

Sam Wiseman, Stuart M. Shieber, and Alexander M. Rush. 2017. Challenges in Data-to-document Generation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017*, pages 2253–2263, Copenhagen, Denmark.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art Natural Language Processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, EMNLP 2020 - Demos*, pages 38–45, Online.

Wenhan Xiong, Jingyu Liu, Igor Molybog, Hejia Zhang, Prajjwal Bhargava, Rui Hou, Louis Martin, Rashi Rungta, Karthik Abinav Sankararaman, Barlas Oguz, Madian Khabsa, Han Fang, Yashar Mehdad, Sharan Narang, Kshitiz Malik, Angela Fan, Shruti Bhosale, Sergey Edunov, Mike Lewis, Sinong Wang, and Hao Ma. 2023. Effective Long-context Scaling of Foundation Models. *CoRR*, abs/2309.16039.

Wenda Xu, Danqing Wang, Liangming Pan, Zhenqiao Song, Markus Freitag, William Yang Wang, and Lei Li. 2023. INSTRUCTSCORE: Explainable Text Generation Evaluation with Finegrained Feedback.

Shuzhou Yuan and Michael Färber. 2023. Evaluating Generative Models for Graph-to-text Generation. In *Proceedings of the 14th International Conference on Recent Advances in Natural Language Processing, RANLP 2023*, pages 1256–1264, Varna, Bulgaria.

Yilun Zhao, Haowei Zhang, Shengyun Si, Linyong Nan, Xiangru Tang, and Arman Cohan. 2023. Investigating Table-to-Text Generation Capabilities of LLMs in Real-World Information Seeking Scenarios.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging LLM-as-a-judge with MT-bench and Chatbot Arena. *CoRR*, abs/2306.05685.

A QUINTD Data

Here, we describe the data sources we include in the QUINTD collection tool and the procedure of collecting the QUINTD-1 benchmark. To replicate the data collection, please refer to the scripts we provide.

A.1 Selection of Data Sources

When selecting the data sources, we had the following desiderata:

- Data needs to be publicly available.
- Data needs to represent a common data-to-text task.
- Data needs to be in a common format (or straightforwardly transformable to one).

We settled on the data sources described in Appendix A.2. All the sources can be accessed using an API. Note that some of the APIs have access limits, either for the requests made from a single account per day or for a number of requests from an IP address within a time window. However, these limits do not severely limit the data collection process on the scale we use here.

A.2 Data Collection

Table 6 summarizes the sources of data and output types for each domain.

A.2.1 OpenWeather

OpenWeather (OpenWeatherMap.org) is an online service that provides global weather data via web interface and API. The API responses are in the JSON format documented at the official website.

Domain Id	Source	Output type
openweather	OpenWeather	five-day weather forecast
gsmarena	GSMArena	product description
ice_hockey	RapidAPI	ice hockey game summary
owid	OurWorldInData	chart caption
wikidata	Wikidata	entity description

Table 6: The sources of data and output types for individual domains in QUINTD.

For our experiments, we used the `forecast5` API, which allows to download a 5-day forecast with 3-hour resolution for any location specified by its GPS coordinates.

The free tier is limited to 1,000 API calls per day, which is enough to download our whole test set in one bulk. However, at the time of experiments, the free API only allowed to download the data for the *time when the request was made*. At the time of writing, OpenWeather is pushing a new One Call API 3.0 which allows to download weather data for any timestamp, but only *4 days ahead* (instead of 5). These restrictions somehow limit the replicability of our QUINTD-1 dataset (at least with the free API) but do not limit downloading a new batch of data with a similar format.

For the QUINTD-1 dataset, we randomly sampled 100 cities for each split from the list of cities with a population over 1000 and used their coordinates in the queries to OpenWeather API. All the data forecasts were downloaded on Jan 3, 2024.

A.2.2 GSMArena

GSMArena is a website providing specifications and reviews for mobile devices. For downloading the data, we used the unofficial `gsmarena-api` tool, which returns the data in a JSON format. Note that GSMArena imposes limitations on the number of requests per IP address, which may induce delays when downloading a larger amount of data.

To create a balanced sample, we downloaded detailed specifications of 10 products from each available brand and randomly selected 100 products for each split from the downloaded set.

A.2.3 RapidAPI Ice Hockey

RapidAPI is a service that provides API access to data from multiple domains, including sport, finance, entertainment, and others. Most APIs are provided in a freemium mode, i.e., with a limited number of daily API calls.

For QUINTD, we selected the `IceHockeyAPI` (popularity 9.1 / 10), which provides access to ice hockey games from world top leagues. Our choice was influenced by our own personal preferences, combined with the desire to cover a sport that has not been covered previously in sports report generation.

We used the `matches` endpoint which returns high-level details about a game. Note that the API allows only 50 requests per day, but that does not limit the data collection since the endpoint returns *all the games* played on a particular day in a single request. We downloaded the games played on 27 November 2023 for the development set (184 games) and 29 November 2023 for the test set (216 games), taking a random sample of 100 for each split.

A.2.4 OurWorldInData

OurWorldInData is a public database and web interface for data about world developments in various domains and sources. We used the official API (currently experimental), which is accessible through the Python package `owid-catalog`. The package allows accessing individual CSV tables as Pandas dataframes.

For our data collection, we decided to limit ourselves to time series, i.e., a single column with values changing over time. Besides the simplicity of visualizing such a chart (which is used by human annotators for checking the correctness of the output), there is also a clear goal for the target chart description: describing the developments of a value over time. We also limited ourselves to the health domain. In particular, we selected the tables COVID data (columns `new_cases_smoothed_per_million`, `new_tests_smoothed_per_thousand`, `people_vaccinated_per_hundred`, `reproduction_rate`, and `positive_rate`) and Life expectancy data (column `life_expectancy_0`).

We downloaded the data for all countries with non-empty entries in the table, taking a random sample of 100 examples for each split. On model input, we formatted the data for each time series as a two-column CSV, including the title, the description, and the unit for each example as a comment (#) at the beginning of the input.

A.2.5 Wikidata

Wikidata is a large open-source knowledge graph containing factual information about entities and their properties. Wikidata provides access through an official API, but we instead decided to extract our data using the `wikidatasets` (Boschin and Bonald, 2019) Python library, which provides access to preprocessed properties of entities from particular domains. It allowed us to avoid crawling and filtering the knowledge graph, and its offline processing made the data collection faster.[10]

For our dataset, we selected the entities from the `companies`, `countries`, `films`, and `humans` domains. For each entity, we randomly extracted between 2 to 10 properties in the knowledge graph. We extracted up to 100 subgraphs for each domain and took a random sample of 100 subgraphs for each split. On model input, we formatted each subgraph as a simple Markdown-formatted text snippet, using the entity as a title and including a bullet point for each key-value pair.

B GPT-4 Evaluation

We used the prompt in Figure 4 for instantiating the GPT-4-based metric.[11] We ensured that the output is a valid JSON using a recently introduced parameter `response_format` in the OpenAI API. At the price of \$0.01 per 1k input tokens and \$0.03 per 1k generated tokens, the evaluation process costs approximately \$45 in total.

B.1 Aligning the Errors

For aligning the errors with the original text, we perform string matching on the text span decoded by GPT-4 in the `TEXT_SPAN` field. In our preliminary experiments, this method proved to be more robust than either asking for start and end indices of the error span (which would rely on the model's ability to count characters) or performing sequence tagging on the copy of the input (which would rely on the model's ability to perfectly copy the input).

We tried to respect the monotonic ordering of text spans but fell back to full-text search if the span is not found following the previous one. We consider this approach successful since matching completely failed only in a minority of cases (137 out of 6927). Based on our manual examination, these mostly include cases where GPT-4 tried to suggest a *missing* piece of text as an error or did not manage to copy the input text verbatim.

C Human Evaluation

As described in §4.2, we set up the human evaluation campaign on Prolific. To make the data more accessible to the annotators, we created custom data visualizations for each domain. For the data in `openweather` and `owid`, we used interactive graphs from Highcharts.com, and we manually created the tables for other domains. You can find the full instructions for human annotators in Figure 5 and the examples of data visualizations in Appendix E.

D Metric Correlation

The Pearson correlation coefficients (§5.2) were computed using two lists (for $\mathcal{E}_{\text{hum}}$ and $\mathcal{E}_{\text{gpt}}$) as follows (note that each error category was considered separately):

- For r_{domain}, we concatenated the average error counts per domain (see Table 12).
- For r_{example}, we concatenated the count of errors per example.
- For r_{token}, we concatenated the binary indicators marking an error per token.

E Examples

Here, we present an example of inputs and model outputs (along with annotations) for each domain:

- `openweather`: Figure 7 (in) and Table 7 (out),
- `gsmarena`: Figure 8 (in) and Table 7 (out),
- `ice_hockey`: Figure 9 (in) and Table 9 (out),
- `owid`: Figure 10 (in) and Table 10 (out),
- `wikidata`: Figure 11 (in) and Table 11 (out).

Note that the graphs for `openweather` and `owid` are interactive when accessed through the web interface.

[10] All the entities and properties are linked with an identifier to the Wikidata database, making the process also replicable through the official API.

[11] Note that the example in the prompt differs from the example used for human annotators (see Figure 5). We revised the example to be more instructive, but we were not able to re-run the GPT-4 evaluation due to our limited budget.

F Full Results

Here, we include the tables with results for individual domains:

- Table 12 presents the average *numbers of errors per output* separately for each domain (the aggregated results are in Table 3),
- Table 13 presents the ratio of *outputs containing at least one error* separately for each domain (the aggregated results are in Table 4).

System Message

````
You are an expert data-to-text error annotation system. You undestand structured data and you
can correcly operate with units and numerical values. You are designed to output token-level
annotations in JSON.
````

Prompt

````
Given the data:
```
data
```
Annotate all the errors in the following text:
```
text
```
Output the errors as a JSON list "errors" in which each object contains fields "reason", "text",
and "type". The value of "text" is the text of the error. The value of "reason" is the reason
for the error. The value of "type" is one of 0, 1, 2, 3 based on the following list:
- 0: Incorrect fact: The fact in the text contradicts the data.
- 1: Not checkable: The fact in the text cannot be checked in the data.
- 2: Misleading: The fact in the text is misleading in the given context.
- 3: Other: The text is problematic for another reason, e.g. grammatically or stylistically
incorrect, irrelevant, or repetitive.
The list should be sorted by the position of the error in the text.
*Example:*
data:
```
[ [ "Aditi Bhagwat", "occupation", "television actor" ], [ "Aditi Bhagwat", "date of birth", "18
January 1981" ] ]
```
text:
```
Aditi Bhagwat, born on January 18, 1991, used to be a popular Indian television actor. The data
comes from a knowledge graph.
```
output:
``` "errors": ["reason": "The data mentions that the actor was born on 1981", "text": "1991",
"type": 0, "reason": "Misleadingly suggests that the actor is not alive", "text": "used to
be", type: 2, "reason": "Popularity is not mentioned in the data", "text": "popular", type: 1,
"reason", "Nationality is not mentioned in the data", "text": "Indian", type: 1, "reason": "The
note is superfluous", "text": "The data comes from a knowledge graph.", type: 3]
```
Note that some details may not be mentioned in the text: do not count omissions as errors. Also
do not be too strict: some facts can be less specific than in the data (rounded values, shortened
or abbreviated text, etc.), do not count these as errors. If there are no errors in the text,
"errors" will be an empty list.
````

Figure 4: The prompt we used for the GPT-4 evaluation metric.

In this task, you will annotate **20** examples in total. For each example, you will see **data** on the left side and the corresponding generated **text** on the right side. Your task is to **annotate errors** in the text with respect to the data.
There are five types of errors that you can mark in the generated text:

1. **Incorrect fact**: The fact in the text contradicts the data.
2. **Not checkable** : The fact in the text cannot be checked given the data.
3. **Misleading**: The fact in the text is misleading in the given context.
4. **Other** : The text is problematic for another reason, e.g. grammatically or stylistically incorrect, irrelevant, or repetitive.

How to mark and submit the annotations?

Use your mouse to **highlight specific parts of the text** containing the errors. To switch between error categories, repeatedly click on the highlighted text (the last click removes the highlight). Note that highlighting from the right to left can work better for longer spans.
Once you think you have marked all the errors present in the text, click the **Mark example as complete** button (you can still update the annotation later). You will be able to submit the annotations once they are all are marked as complete.

How should I decide on the errors?

- Each error span should include all the words related to the error (but nothing else).
- If you think the fact is probably true, but cannot be derived from the data, mark it as not checkable.
- If you are not really sure if the fact should be marked as an error, leave it unmarked.

An example of the annotated output

An example of the data input and the corresponding text annotated with errors:

data
Nokia 3310

- **color:** black, blue, grey
- **display:** 320x240px

text (product description)
Nokia 3310 is produced in Finland and features a 320x320 display. It is available in black color. The data seem to provide only partial information about the phone.

explanation

- **produced in Finland**: The country where the phone is produced is not mentioned in the data.
- **320x320**: The data mentions that the display has resolution 320x240px.
- **available in black color**: Misleadingly suggests that the phone is not available in other colors.
- **The data seem to provide only partial information about the phone.**: The note is irrelevant for the phone description.

Note that for the sake of brevity, this particular example is based on a small data input and contains many errors, which may not always be the case.

What kinds of data and text can I encounter?

You can encounter the following kinds of texts:

- a 5-day weather forecast generated from weather data,
- a description of a product generated from product specifications
- an ice hockey game report generated from information about the game,
- a caption of a health-related chart,
- a description of an entity (human, country, film, or company) based on its properties.

What is the source of the data and the texts?

The data is downloaded from public sources (openweathermap.org, gsmarena.com, rapidapi.com, ourworldindata.com, wikidata.com). The texts are generated by large language models.

What is the purpose of the research?

The research is conducted by [REDACTED]The research is a part of the [REDACTED] and was approved by the ethical committee [REDACTED]

By continuing, you agree that the anonymous annotations will be made available to the research community to help evaluating and improving data-to-text generation systems based on large language models.

Figure 5: The instructions given to the human annotators.

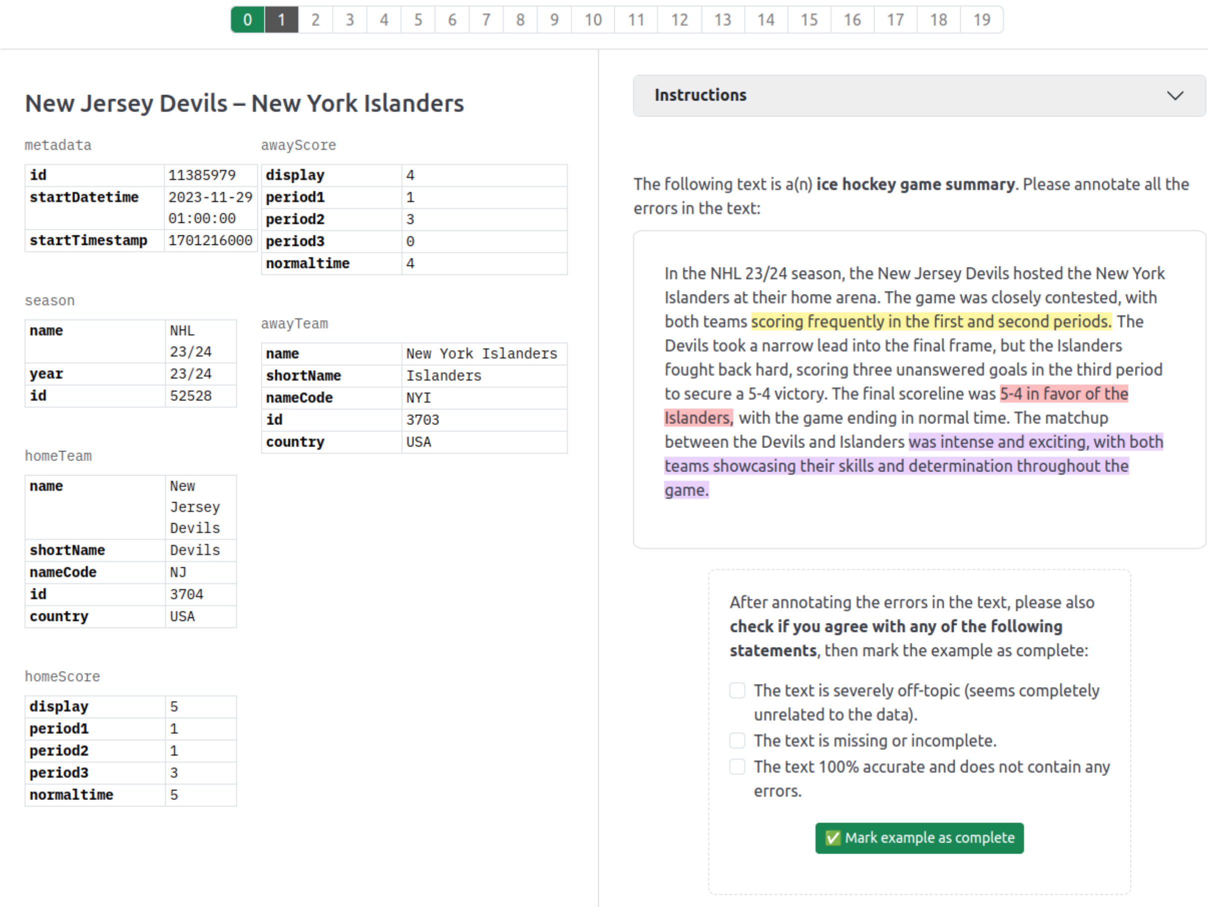

Figure 6: The annotation interface for human annotators.

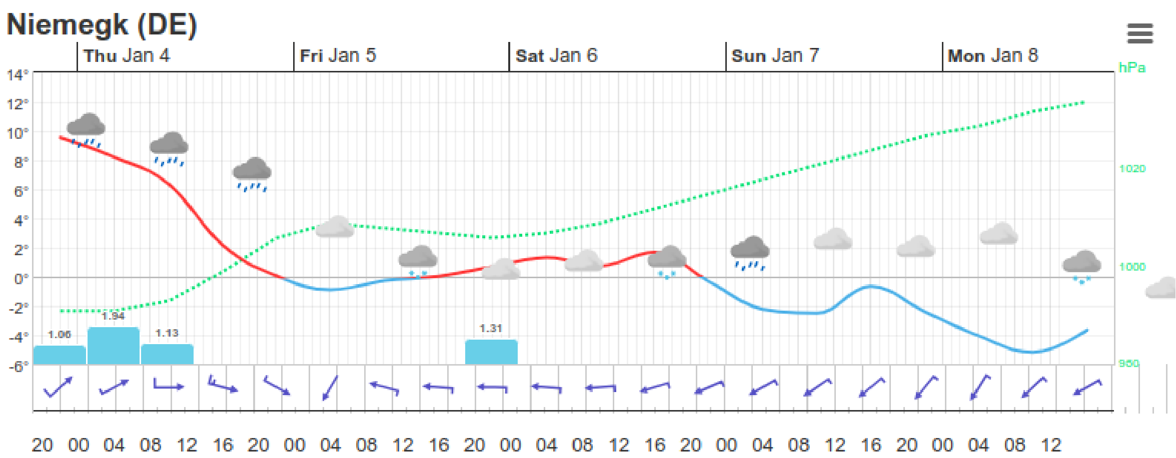

Figure 7: An example of an input from `openweather` (see the outputs in Table 7).

Model	Human annotations ($\mathcal{E}_{\text{hum}}$)	GPT-4 annotations ($\mathcal{E}_{\text{gpt}}$)
Llama2	The weather in Niemegk, Germany, will be cloudy and rainy for the next few days. On January 3rd, the temperature will be around 9.55°C, with a feels-like temperature of 6.59°C. The weather will be light rain, with a wind speed of 6.24 m/s and a gust of 12.1 m/s. The rain will last for 3 hours, and the pressure will be 991 hPa.	The weather in Niemegk, Germany, will be cloudy and rainy for the next few days. On January 3rd, the temperature will be around 9.55°C, with a feels-like temperature of 6.59°C. The weather will be light rain, with a wind speed of 6.24 m/s and a gust of 12.1 m/s. The rain will last for 3 hours, and the pressure will be 991 hPa.
Mistral	From January 3rd to January 7th, the city of Niemegk in Germany will experience a range of weather conditions. On January 3rd, the temperature will be 9.55°C with a feels like temperature of 6.59°C. There will be light rain with a 3-hour precipitation of 1.06mm. The wind speed will be 6.24m/s with a gust of 12.1m/s. On January 4th, the temperature will be 8.2°C with a feels like temperature of 5.64°C. There will be light rain with a 3-hour precipitation of 1.94mm. The wind speed will be 4.31m/s with a gust of 11.6m/s. On January 5th, the temperature will be 6.37°C with a feels like temperature of 2.25°C. There will be light rain with a 3-hour precipitation of 1.13mm. The wind speed will be 7.07m/s with a gust of 15.51m/s. On January 6th, the temperature will be 2.17°C with a feels like temperature of -3.55°C. There will be overcast clouds with a 3-hour precipitation of 0.31mm. The wind speed will be 3.54m/s with a gust of 7.11m/s. On January 7th, the temperature will be -0.89°C with a feels like temperature of -0.89°C. There will be overcast clouds with a 3-hour precipitation of 0.31mm. The wind speed will be 0.63m/s with a gust of 0.7m/s.	From January 3rd to January 7th, the city of Niemegk in Germany will experience a range of weather conditions. On January 3rd, the temperature will be 9.55°C with a feels like temperature of 6.59°C. There will be light rain with a 3-hour precipitation of 1.06mm. The wind speed will be 6.24m/s with a gust of 12.1m/s. On January 4th, the temperature will be 8.2°C with a feels like temperature of 5.64°C. There will be light rain with a 3-hour precipitation of 1.94mm. The wind speed will be 4.31m/s with a gust of 11.6m/s. On January 5th, the temperature will be 6.37°C with a feels like temperature of 2.25°C. There will be light rain with a 3-hour precipitation of 1.13mm. The wind speed will be 7.07m/s with a gust of 15.51m/s. On January 6th, the temperature will be 2.17°C with a feels like temperature of -3.55°C. There will be overcast clouds with a 3-hour precipitation of 0.31mm. The wind speed will be 3.54m/s with a gust of 7.11m/s. On January 7th, the temperature will be -0.89°C with a feels like temperature of -0.89°C. There will be overcast clouds with a 3-hour precipitation of 0.31mm. The wind speed will be 0.63m/s with a gust of 0.7m/s.
Zephyr	Over the next five days, the weather in Niemegk, Germany will be characterized by fluctuating temperatures and varying precipitation. On Thursday, January 3, light rain and a low temperature of 9.55°C can be expected, with a chance of rain continuing into Friday. Snowfall is possible on Friday, with a low temperature of -0.89°C. Saturday will be overcast with a low temperature of 0.03°C, and light rain is forecasted for Saturday evening. Sunday will be overcast as well, with a low temperature of -2.23°C. Monday will see a drop in temperatures, with a low of -4.07°C and scattered clouds. Temperatures will continue to fall throughout the week, with lows of -5.19°C and -9.51°C on Tuesday and Wednesday, respectively. Wind speeds will vary from 3.1 to 15.51 meters per second, with gusts reaching up to 15.51 meters per second on Monday.	Over the next five days, the weather in Niemegk, Germany will be characterized by fluctuating temperatures and varying precipitation. On Thursday, January 3, light rain and a low temperature of 9.55°C can be expected, with a chance of rain continuing into Friday. Snowfall is possible on Friday, with a low temperature of -0.89°C. Saturday will be overcast with a low temperature of 0.03°C, and light rain is forecasted for Saturday evening. Sunday will be overcast as well, with a low temperature of -2.23°C. Monday will see a drop in temperatures, with a low of -4.07°C and scattered clouds. Temperatures will continue to fall throughout the week, with lows of -5.19°C and -9.51°C on Tuesday and Wednesday, respectively. Wind speeds will vary from 3.1 to 15.51 meters per second, with gusts reaching up to 15.51 meters per second on Monday.
GPT-3.5	Over the next five days, the weather in Niemegk, Germany is expected to be quite varied. On January 4th, there will be light rain with temperatures ranging from 2.17°C to 9.55°C. The following day, January 5th, there will be light snow with temperatures dropping to as low as -0.89°C. On January 6th, the weather will be overcast with temperatures ranging from -0.32°C to 1.69°C. The trend continues on January 7th with overcast clouds and temperatures dropping to as low as -2.51°C. Finally, on January 8th, there will be scattered clouds with temperatures ranging from -5.19°C to -3.67°C. Throughout this period, wind speeds will vary between 0.63 m/s and 8.12 m/s, and the pressure will range from 991 hPa to 1034 hPa.	Over the next five days, the weather in Niemegk, Germany is expected to be quite varied. On January 4th, there will be light rain with temperatures ranging from 2.17°C to 9.55°C. The following day, January 5th, there will be light snow with temperatures dropping to as low as -0.89°C. On January 6th, the weather will be overcast with temperatures ranging from -0.32°C to 1.69°C. The trend continues on January 7th with overcast clouds and temperatures dropping to as low as -2.51°C. Finally, on January 8th, there will be scattered clouds with temperatures ranging from -5.19°C to -3.67°C. Throughout this period, wind speeds will vary between 0.63 m/s and 8.12 m/s, and the pressure will range from 991 hPa to 1034 hPa.

Table 7: An example of the model outputs on `openweather` (see the input in Figure 7).

Gigabyte GSmart Mika M3

name: GSmart Mika M3, id: gigabyte_gsmart_mika_m3-6797

quick specifications

Display size	5.0"
Display resolution	720x1280 pixels
Camera pixels	13MP
Video pixels	Video recorder
RAM size	1GB RAM
Chipset	MT6582
Battery size	1900mAh
Battery type	Li-Ion

detailed specifications

Network	**Technology**	GSM / HSPA
	2G bands	GSM 850 / 900 / 1800 / 1900 - SIM 1 & SIM 2
	3G bands	HSDPA 900 / 2100
	Speed	HSPA 21.1/5.76 Mbps
Launch	**Announced**	2014, October. Released 2014, November
	Status	Discontinued
Body	**Dimensions**	144 x 70.5 x 8.3 mm (5.67 x 2.78 x 0.33 in)
	Weight	146 g (5.15 oz)
	SIM	Dual SIM (Mini-SIM/ Micro-SIM)
Display	**Type**	IPS LCD
	Size	5.0 inches, 68.9 cm2 (~67.9% screen-to-body ratio)
	Resolution	720 x 1280 pixels, 16:9 ratio (~294 ppi density)
Platform	**OS**	Android 4.4.2 (KitKat)
	Chipset	Mediatek MT6582 (28 nm)
	CPU	Quad-core 1.3 GHz Cortex-A7
	GPU	Mali-400MP2
Memory	**Card slot**	microSDHC (dedicated slot)
	Internal	8GB 1GB RAM
Main Camera	**Single**	13 MP, AF
	Features	LED flash
	Video	Yes
Selfie camera	**Single**	8 MP
	Video	
Sound	**Loudspeaker**	Yes
	3.5mm jack	Yes
Comms	**WLAN**	Wi-Fi 802.11 b/g/n, hotspot
	Bluetooth	3.0
	Positioning	GPS
	NFC	No
	Radio	FM radio
	USB	microUSB 2.0
Features	**Sensors**	Accelerometer, proximity
Battery	**Type**	Li-Ion 1900 mAh, removable
	Stand-by	Up to 150 h
	Talk time	Up to 4 h 30 min
Misc	**Colors**	Navy Blue

Figure 8: An example of an input from gsmarena (see the outputs in Table 8).

Model	Human annotations ($\mathcal{E}_{\text{hum}}$)	GPT-4 annotations ($\mathcal{E}_{\text{gpt}}$)
Llama2	The Gigabyte GSmart Mika M3 is a smartphone that was released in 2014. It has a 5-inch IPS LCD display with a resolution of 720x1280 pixels and a 13MP main camera. The device also has an 8MP front camera and is powered by a Mediatek MT6582 chipset with 1GB of RAM and 8GB of internal storage. It has dual SIM support and is available in Navy Blue.	The Gigabyte GSmart Mika M3 is a smartphone that was released in 2014. It has a 5-inch IPS LCD display with a resolution of 720x1280 pixels and a 13MP main camera. The device also has an 8MP front camera and is powered by a Mediatek MT6582 chipset with 1GB of RAM and 8GB of internal storage. It has dual SIM support and is available in Navy Blue.
Mistral	The GSmart Mika M3 is a smartphone developed by Gigabyte. It was released in November 2014 and is powered by the Android 4.4.2 operating system. The device features a 5.0-inch IPS LCD display with a resolution of 720 x 1280 pixels, which translates to 294 pixels per inch. The smartphone is equipped with a 13-megapixel rear camera with autofocus and LED flash, as well as an 8-megapixel front-facing camera. It is powered by a Mediatek MT6582 chipset with a quad-core 1.3 GHz Cortex-A7 CPU and a Mali-400MP2 GPU. The device has 8GB of internal storage and 1GB of RAM, with the option to expand storage via a microSDHC card. The GSmart Mika M3 also features Wi-Fi, Bluetooth, GPS, and NFC connectivity, as well as a 3.5mm headphone jack and a microUSB 2.0 port for charging and data transfer. The device is available in Navy Blue and is powered by a 1900mAh Li-Ion battery, which provides up to 150 hours of stand-by time and up to 4 hours and 30 minutes of talk time.	The GSmart Mika M3 is a smartphone developed by Gigabyte. It was released in November 2014 and is powered by the Android 4.4.2 operating system. The device features a 5.0-inch IPS LCD display with a resolution of 720 x 1280 pixels, which translates to 294 pixels per inch. The smartphone is equipped with a 13-megapixel rear camera with autofocus and LED flash, as well as an 8-megapixel front-facing camera. It is powered by a Mediatek MT6582 chipset with a quad-core 1.3 GHz Cortex-A7 CPU and a Mali-400MP2 GPU. The device has 8GB of internal storage and 1GB of RAM, with the option to expand storage via a microSDHC card. The GSmart Mika M3 also features Wi-Fi, Bluetooth, GPS, and NFC connectivity, as well as a 3.5mm headphone jack and a microUSB 2.0 port for charging and data transfer. The device is available in Navy Blue and is powered by a 1900mAh Li-Ion battery, which provides up to 150 hours of stand-by time and up to 4 hours and 30 minutes of talk time.
Zephyr	The Gigabyte GSmart Mika M3 is a smartphone that boasts a 5.0-inch IPS LCD display with a resolution of 720 x 1280 pixels. It runs on Android 4.4.2 (KitKat) and is powered by a quad-core 1.3 GHz Cortex-A7 processor with a Mali-400MP2 GPU. The device comes with 8GB of internal storage and 1GB of RAM, which can be expanded via a dedicated microSDHC card slot. The primary camera is a 13-megapixel autofocus lens with an LED flash, while the secondary camera is an 8-megapixel lens. Connectivity options include Wi-Fi, Bluetooth, GPS, FM radio, and microUSB 2.0. The smartphone is equipped with a 1900mAh removable Li-Ion battery that provides up to 4 hours of talk time and up to 150 hours of standby time. The device is available in Navy Blue and supports dual SIM cards (Mini-SIM/ Micro-SIM).	The Gigabyte GSmart Mika M3 is a smartphone that boasts a 5.0-inch IPS LCD display with a resolution of 720 x 1280 pixels. It runs on Android 4.4.2 (KitKat) and is powered by a quad-core 1.3 GHz Cortex-A7 processor with a Mali-400MP2 GPU. The device comes with 8GB of internal storage and 1GB of RAM, which can be expanded via a dedicated microSDHC card slot. The primary camera is a 13-megapixel autofocus lens with an LED flash, while the secondary camera is an 8-megapixel lens. Connectivity options include Wi-Fi, Bluetooth, GPS, FM radio, and microUSB 2.0. The smartphone is equipped with a 1900mAh removable Li-Ion battery that provides up to 4 hours of talk time and up to 150 hours of standby time. The device is available in Navy Blue and supports dual SIM cards (Mini-SIM/ Micro-SIM).
GPT-3.5	The GSmart Mika M3, also known as the Gigabyte GSmart Mika M3, is a smartphone that was announced in October 2014 and released in November of the same year. It features a 5.0-inch IPS LCD display with a resolution of 720 x 1280 pixels and is powered by a Quad-core 1.3 GHz Cortex-A7 processor. The phone comes with 1GB of RAM and 8GB of internal storage, which can be expanded via a microSDHC card slot. The main camera is 13MP with LED flash, while the selfie camera is 8MP. The phone runs on Android 4.4.2 (KitKat) and is equipped with a 1900mAh removable battery. Other features include dual SIM capability, Wi-Fi, Bluetooth, GPS, FM radio, and microUSB 2.0. The GSmart Mika M3 is available in the color Navy Blue and has a sleek and compact design, measuring 144 x 70.5 x 8.3 mm and weighing 146g.	The GSmart Mika M3, also known as the Gigabyte GSmart Mika M3, is a smartphone that was announced in October 2014 and released in November of the same year. It features a 5.0-inch IPS LCD display with a resolution of 720 x 1280 pixels and is powered by a Quad-core 1.3 GHz Cortex-A7 processor. The phone comes with 1GB of RAM and 8GB of internal storage, which can be expanded via a microSDHC card slot. The main camera is 13MP with LED flash, while the selfie camera is 8MP. The phone runs on Android 4.4.2 (KitKat) and is equipped with a 1900mAh removable battery. Other features include dual SIM capability, Wi-Fi, Bluetooth, GPS, FM radio, and microUSB 2.0. The GSmart Mika M3 is available in the color Navy Blue and has a sleek and compact design, measuring 144 x 70.5 x 8.3 mm and weighing 146g.

Table 8: An example of the model outputs on `gsmarena` (see the input in Figure 8).

Kristall Saratov – HC Norilsk

metadata

id	11482971
startDatetime	2023-11-28 14:30:00
startTimestamp	1701181800

season

name	VHL 23/24
year	23/24
id	53511

homeTeam

name	Kristall Saratov
shortName	Kristall Saratov
nameCode	KRI
id	78185
country	Russia

homeScore

display	1
period1	0
period2	0
period3	1
normaltime	1

awayScore

display	5
period1	2
period2	3
period3	0
normaltime	5

awayTeam

name	HC Norilsk
shortName	HC Norilsk
nameCode	NOR
id	485576
country	Russia

Figure 9: An example of an input from `ice_hockey` (see the outputs in Table 9).

Model	Human annotations ($\mathcal{E}_{\text{hum}}$)	GPT-4 annotations ($\mathcal{E}_{\text{gpt}}$)
Llama2	In the VHL 23/24 season, Kristall Saratov hosted HC Norilsk at their home arena. Kristall Saratov won the game with a final score of 1-5. Kristall Saratov's home team scored one goal in the first period, while HC Norilsk scored two goals in the second and third periods. The game lasted for 60 minutes, with Kristall Saratov's home team leading the game for the entire duration.	In the VHL 23/24 season, Kristall Saratov hosted HC Norilsk at their home arena. Kristall Saratov won the game with a final score of 1-5. Kristall Saratov's home team scored one goal in the first period, while HC Norilsk scored two goals in the second and third periods. The game lasted for 60 minutes, with Kristall Saratov's home team leading the game for the entire duration.
Mistral	On November 28th, 2023, the VHL 23/24 season game between Kristall Saratov and HC Norilsk took place. The game was played at the home arena of Kristall Saratov, with the home team winning 1-0. The game was played in three periods, with the first period ending in a 0-0 tie, the second period ending in a 2-3 score in favor of HC Norilsk, and the third period ending in a 1-0 score in favor of Kristall Saratov. The game ended in normal time, with a total score of 3-1 in favor of HC Norilsk.	On November 28th, 2023, the VHL 23/24 season game between Kristall Saratov and HC Norilsk took place. The game was played at the home arena of Kristall Saratov, with the home team winning 1-0. The game was played in three periods, with the first period ending in a 0-0 tie, the second period ending in a 2-3 score in favor of HC Norilsk, and the third period ending in a 1-0 score in favor of Kristall Saratov. The game ended in normal time, with a total score of 3-1 in favor of HC Norilsk.
Zephyr	In the VHL 23/24 season, Kristall Saratov hosted HC Norilsk. The first period saw two quick goals from Norilsk, with one in the second and third periods as well, resulting in a dominant 5-0 victory. Kristall Saratov managed to score a single goal in the third period, but it was not enough to overcome Norilsk's lead. The final score was 1-5 in favor of HC Norilsk.	In the VHL 23/24 season, Kristall Saratov hosted HC Norilsk. The first period saw two quick goals from Norilsk, with one in the second and third periods as well, resulting in a dominant 5-0 victory. Kristall Saratov managed to score a single goal in the third period, but it was not enough to overcome Norilsk's lead. The final score was 1-5 in favor of HC Norilsk.
GPT-3.5	In a match from the VHL 23/24 season, Kristall Saratov faced off against HC Norilsk. The game took place on November 28, 2023, at 15:30. Despite a goal from Kristall Saratov in the third period, HC Norilsk emerged victorious with a final score of 5-1. The away team dominated the game, scoring 2 goals in the first period and 3 goals in the second period. Kristall Saratov struggled to keep up, resulting in a clear win for HC Norilsk.	In a match from the VHL 23/24 season, Kristall Saratov faced off against HC Norilsk. The game took place on November 28, 2023, at 15:30. Despite a goal from Kristall Saratov in the third period, HC Norilsk emerged victorious with a final score of 5-1. The away team dominated the game, scoring 2 goals in the first period and 3 goals in the second period. Kristall Saratov struggled to keep up, resulting in a clear win for HC Norilsk.

Table 9: An example of the model outputs on `ice_hockey` (see the input in Figure 9).

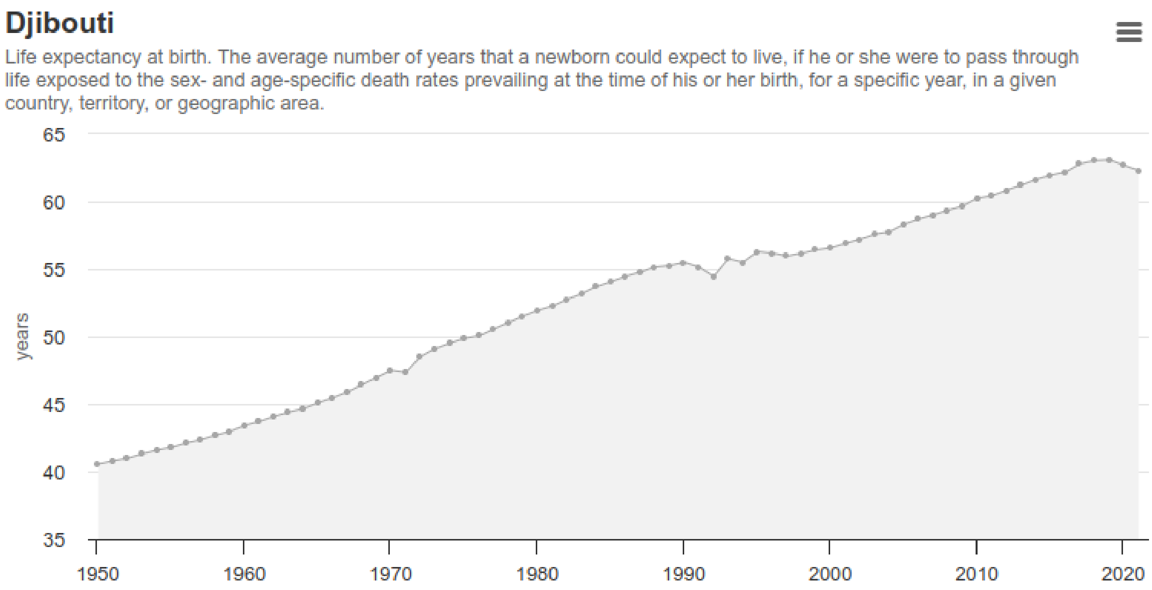

Figure 10: An example of an input from `owid` (see the outputs in Table 10).

Model	Human annotations ($\mathcal{E}_{\text{hum}}$)	GPT-4 annotations ($\mathcal{E}_{\text{gpt}}$)
Llama2	Life expectancy at birth in Djibouti has increased significantly over the past few decades, with a notable increase from 40.5665 years in 1950 to 62.3049 years in 2021.	Life expectancy at birth in Djibouti has increased significantly over the past few decades, with a notable increase from 40.5665 years in 1950 to 62.3049 years in 2021.
Mistral	This chart displays the life expectancy at birth in Djibouti from 1950 to 2021. The data shows a steady increase in life expectancy over the years, with a significant jump in the 1970s and 1980s. In 2021, the average life expectancy at birth in Djibouti was 62.3 years.	This chart displays the life expectancy at birth in Djibouti from 1950 to 2021. The data shows a steady increase in life expectancy over the years, with a significant jump in the 1970s and 1980s. In 2021, the average life expectancy at birth in Djibouti was 62.3 years.
Zephyr	This line graph illustrates the evolution of life expectancy at birth in Djibouti from 1950 to 2021. Initially, the average number of years that a newborn could expect to live was around 40.5 years in 1950, and it gradually increased over the following decades, reaching 62.3 years in 2021. The trend shows a steady improvement, with some fluctuations in the 1980s and 1990s. The data presented here is based on the sex- and age-specific death rates prevailing at the time of birth for a specific year in a given country, territory, or geographic area.	This line graph illustrates the evolution of life expectancy at birth in Djibouti from 1950 to 2021. Initially, the average number of years that a newborn could expect to live was around 40.5 years in 1950, and it gradually increased over the following decades, reaching 62.3 years in 2021. The trend shows a steady improvement, with some fluctuations in the 1980s and 1990s. The data presented here is based on the sex- and age-specific death rates prevailing at the time of birth for a specific year in a given country, territory, or geographic area.
GPT-3.5	The chart shows the life expectancy at birth in Djibouti from 1950 to 2021. The life expectancy has steadily increased over the years, from 40.57 years in 1950 to 62.30 years in 2021, with some fluctuations in between. The data reflects the average number of years a newborn could expect to live if exposed to the prevailing death rates at the time of birth.	The chart shows the life expectancy at birth in Djibouti from 1950 to 2021. The life expectancy has steadily increased over the years, from 40.57 years in 1950 to 62.30 years in 2021, with some fluctuations in between. The data reflects the average number of years a newborn could expect to live if exposed to the prevailing death rates at the time of birth.

Table 10: An example of the model outputs on `owid` (see the input in Figure 10).

Driven Brands

country	United States of America
industry	automotive services industry
instance of	business

Figure 11: An example of an input from `wikidata` (see the outputs in Table 11).

Model	Human annotations ($\mathcal{E}_{\text{hum}}$)	GPT-4 annotations ($\mathcal{E}_{\text{gpt}}$)
Llama2	Driven Brands is an American franchise holding company that owns, operates, and franchises automotive services businesses in the United States. The company was founded in 1996 and is headquartered in Charlotte, North Carolina. It operates several brands, including Maaco, Meineke, Econo Lube, and CARSTAR.	Driven Brands is an American franchise holding company that owns, operates, and franchises automotive services businesses in the United States. The company was founded in 1996 and is headquartered in Charlotte, North Carolina. It operates several brands, including Maaco, Meineke, Econo Lube, and CARSTAR.
Mistral	Driven Brands is a business operating in the automotive services industry in the United States of America.	Driven Brands is a business operating in the automotive services industry in the United States of America.
Zephyr	Driven Brands is a business operating in the automotive services industry based in the United States of America. As an instance of the broader category of businesses, Driven Brands provides goods or services to customers in exchange for financial remuneration.	Driven Brands is a business operating in the automotive services industry based in the United States of America. As an instance of the broader category of businesses, Driven Brands provides goods or services to customers in exchange for financial remuneration.
GPT-3.5	Driven Brands is a business based in the United States of America, operating within the automotive services industry.	Driven Brands is a business based in the United States of America, operating within the automotive services industry.

Table 11: An example of the model outputs on `wikidata` (see the input in Figure 11).

		Incorrect		Not Checkable		Misleading		Other		All categories		
		$\mathcal{E}_{\text{gpt}}$	$\mathcal{E}_{\text{hum}}$	$\mathcal{E}_{\text{gpt}}$	$\mathcal{E}_{\text{hum}}$	$\mathcal{E}_{\text{gpt}}$	$\mathcal{E}_{\text{hum}}$	$\mathcal{E}_{\text{gpt}}$	$\mathcal{E}_{\text{hum}}$	$\mathcal{E}_{\text{gpt}}$	$\mathcal{E}_{\text{hum}}$	**# Tok.**
Llama2	`openweather`	5.77	4.27	1.94	2.23	0.34	0.58	0.43	0.16	8.48	7.24	166.7
	`gsmarena`	1.88	0.67	0.37	0.87	0.33	0.20	0.11	0.09	2.69	1.83	90.3
	`ice_hockey`	1.82	1.47	0.85	0.99	0.04	0.09	0.06	0.12	2.77	2.67	63.3
	`owid`	2.18	1.30	0.22	0.58	0.22	0.32	0.01	0.07	2.63	2.27	60.3
	`wikidata`	0.65	0.15	1.13	1.59	0.08	0.08	0.06	0.07	1.92	1.89	38.5
Mistral	`openweather`	7.18	5.65	0.59	2.47	0.48	1.36	0.09	0.53	8.34	10.01	193.5
	`gsmarena`	1.86	0.51	0.80	1.10	0.54	0.30	0.05	0.14	3.25	2.05	146.3
	`ice_hockey`	1.91	1.47	0.72	0.92	0.13	0.14	0.05	0.13	2.81	2.66	92.4
	`owid`	2.67	2.42	0.18	0.40	0.41	0.29	0.04	0.07	3.30	3.18	91.1
	`wikidata`	0.40	0.10	0.32	0.72	0.27	0.13	0.32	0.38	1.31	1.33	51.0
Zephyr	`openweather`	6.46	4.22	0.42	1.01	0.38	0.34	0.01	0.16	7.27	5.73	130.9
	`gsmarena`	1.72	0.33	0.93	1.04	0.85	0.23	0.03	0.11	3.53	1.71	142.8
	`ice_hockey`	1.49	0.89	0.19	0.61	0.08	0.11	0.02	0.07	1.78	1.68	83.1
	`owid`	2.51	1.68	0.20	0.49	0.43	0.22	0.01	0.10	3.15	2.49	85.2
	`wikidata`	0.33	0.09	0.24	0.72	0.23	0.12	0.21	0.36	1.01	1.29	48.1
GPT-3.5	`openweather`	3.88	1.57	0.17	0.57	0.96	0.38	0.00	0.05	5.01	2.57	112.8
	`gsmarena`	1.30	0.20	1.13	0.80	0.63	0.21	0.03	0.17	3.09	1.38	129.5
	`ice_hockey`	0.72	0.81	0.12	0.46	0.05	0.07	0.00	0.09	0.89	1.43	84.4
	`owid`	1.81	0.64	0.03	0.25	0.35	0.17	0.00	0.01	2.19	1.07	62.2
	`wikidata`	0.14	0.05	0.13	0.36	0.11	0.06	0.06	0.05	0.44	0.52	35.7

Table 12: The average *numbers of errors per output* for each domain (lower is better). We also include the average number of tokens per output in the rightmost column. See Table 3 for aggregated results.

		Incorrect		Not Checkable		Misleading		Other		All categories	
		$\mathcal{E}_{\text{gpt}}$	$\mathcal{E}_{\text{hum}}$	$\mathcal{E}_{\text{gpt}}$	$\mathcal{E}_{\text{hum}}$	$\mathcal{E}_{\text{gpt}}$	$\mathcal{E}_{\text{hum}}$	$\mathcal{E}_{\text{gpt}}$	$\mathcal{E}_{\text{hum}}$	$\mathcal{E}_{\text{gpt}}$	$\mathcal{E}_{\text{hum}}$
Llama2	`openweather`	0.96	0.76	0.70	0.71	0.26	0.30	0.25	0.12	1.00	0.90
	`gsmarena`	0.83	0.38	0.28	0.49	0.21	0.18	0.06	0.05	0.90	0.74
	`ice_hockey`	0.77	0.70	0.49	0.51	0.04	0.07	0.05	0.10	0.93	0.89
	`owid`	0.92	0.70	0.18	0.37	0.19	0.24	0.01	0.06	0.93	0.88
	`wikidata`	0.42	0.12	0.63	0.79	0.07	0.08	0.06	0.05	0.83	0.87
Mistral	`openweather`	0.99	0.78	0.23	0.61	0.23	0.38	0.06	0.17	1.00	0.92
	`gsmarena`	0.89	0.33	0.45	0.54	0.33	0.19	0.05	0.12	0.99	0.73
	`ice_hockey`	0.83	0.74	0.52	0.62	0.13	0.12	0.04	0.09	0.96	0.88
	`owid`	0.91	0.75	0.17	0.27	0.29	0.24	0.04	0.06	0.98	0.86
	`wikidata`	0.29	0.08	0.25	0.44	0.19	0.10	0.20	0.24	0.70	0.67
Zephyr	`openweather`	0.99	0.82	0.27	0.45	0.30	0.23	0.01	0.14	1.00	0.90
	`gsmarena`	0.87	0.22	0.53	0.47	0.45	0.19	0.02	0.09	0.99	0.61
	`ice_hockey`	0.79	0.54	0.16	0.39	0.08	0.11	0.02	0.06	0.84	0.76
	`owid`	0.93	0.70	0.14	0.31	0.30	0.20	0.01	0.07	0.96	0.85
	`wikidata`	0.28	0.06	0.20	0.49	0.21	0.08	0.15	0.22	0.60	0.66
GPT-3.5	`openweather`	0.96	0.64	0.15	0.32	0.53	0.23	0.00	0.05	0.99	0.75
	`gsmarena`	0.69	0.15	0.68	0.42	0.34	0.13	0.03	0.12	0.93	0.57
	`ice_hockey`	0.52	0.64	0.09	0.30	0.05	0.07	0.00	0.08	0.56	0.76
	`owid`	0.90	0.43	0.02	0.13	0.30	0.14	0.00	0.01	0.92	0.57
	`wikidata`	0.12	0.04	0.11	0.27	0.10	0.06	0.06	0.05	0.33	0.38

Table 13: The ratio of *outputs containing at least one error* for each domain (lower is better). See Table 4 for aggregated results.