Abstract

Evaluations in machine learning rarely use the latest metrics, datasets, or human evaluation in favor of remaining compatible with prior work. The compatibility, often facilitated through leaderboards, thus leads to outdated and standardized evaluation practices. We posit that the standardization is taking place in the wrong spot. Evaluation infrastructure should enable researchers to use the latest methods and what should be standardized instead is how to incorporate these new evaluation advances. We introduce GEMv2, the new version of the Generation, Evaluation, and Metrics Benchmark which uses a modular infrastructure for dataset, model, and metric developers to benefit from each other’s work. GEMv2 supports 40 documented datasets in 51 languages, ongoing online evaluation for all datasets, and our interactive tools make it easier to add new datasets to the living benchmark.

1 Introduction

The standard evaluation process in natural language processing involves comparisons to prior results in a fixed environment, often facilitated through benchmarks and leaderboards. This process, if executed correctly, can advance reproducibility (Belz et al., 2021) and standardize evaluation choices that lead to better dataset diversity. But static benchmarks also prevent the adoption of new datasets or metrics (Raji et al., 2021), and many evaluation advancements are thus put aside. That means that the focus on surpassing the best prior reported scores reinforces outdated evaluation designs. Furthermore, this process ignores properties that do not match the leaderboard metric (Ethayarajh and Jurafsky, 2020; Bowman and Dahl, 2021; Dehghani et al., 2021). This issue is particularly pertinent in natural language generation (NLG) since the model quality cannot be estimated using accuracy and instead, NLG relies on automatic and human evaluation approaches that constantly improve (Gehrmann et al., 2022; Kasai et al., 2022).

To bridge the gap between advantages of leaderboards and in-depth and evolving evaluations, the Generation, Evaluation, and Metrics benchmark (GEM, Gehrmann et al., 2021) proposed a “living” benchmark. As such, GEM is participatory in that contributors propose new datasets and expand the selection of metrics. Model developers using GEM retain full agency over the evaluation process but are able to choose from a wider range of tasks and metrics. GEM further introduced evaluation suites (Mille et al., 2021; Dhole et al., 2021) that are compatible with its datasets and test various robustness and fairness aspects of models.
We uncovered several shortcomings in GEMv1 that hindered its scaling and adoption: (1) Centralized data management made adding new datasets too complex. (2) Computing all metrics in a single framework led to dependency issues and was challenging for those with limited compute resources. (3) Participants needed more guidance in our dataset documentation process (McMillan-Major et al., 2021) to guarantee data card quality.

We introduce GEMv2, a modular and extendable NLG evaluation infrastructure which allows for continuous integration of newly developed datasets. We release a data card collection and rendering tool that makes it easier to follow for both card creators and readers. These improvements led to an expansion of GEM from 13 to 40 tasks and from 18 to 51 supported languages. We also introduce an online evaluation process that collects model outputs and computes metrics for all datasets.

2 Features and Functionality

Since evaluation practices evolve, we focus on modularity and maintainability to ensure that new dataset and metrics are compatible with all other features. Model developers are able to use new datasets and metrics without any changes to their existing setup. In this section, we describe the supported user journeys for various stakeholders in generation research.

J1 - Document a Dataset

Every GEM dataset is documented using the data card template by McMillan-Major et al. (2021), which we revised using the Data Card Playbook (Pushkarna et al., 2022). A new card can be filled out or an existing summary in a foreign language).

J2 - Choose a Dataset

The data card viewer

https://huggingface.co/spaces/GEM/DatasetCardForm

Figure 1: One of the data cards for GEM datasets. (a) shows the header which has the name, a summary, and example code to load it. (b) links to relevant papers and websites, alongside an author list. (c) is the Quick-Use section which summarizes the most important aspect of a dataset, including language(s), PII, and licensing information. (d) is the detailed view which has multiple sections. Each section provides a glance at categories of included questions on hover, and expands to full details on click.
presents information at multiple detail levels in separate columns. Anyone can quickly get a high-level overview of a dataset or read extended information on a documentation category (see Figure 1).

J3 - Create a Data Loader Each dataset has a separate repository at huggingface.co/GEM, with a loader using the Datasets library (Lhoest et al., 2021). Through this, all supported datasets can be loaded via the same code,

```
from datasets import load_dataset
data = load_dataset('GEM/$dataset_name', '$config_name')
```

where $config_name is the (optional) specification of the dataset configuration to use. To stratify how datasets are accessed, they are implemented according to the following conventions:

- **linearized_input**: Linearization processes convert structured input to a string. For reproducibility, we implement linearization schemes from prior work (e.g., Saleh et al., 2019; Kale and Rastogi, 2020).
- **target** and **references**: String targets and List[string] references ensure compatibility with existing training and eval scripts.
- **gem_id**: A unique example ID is used to track data points regardless of shuffling.

J4 - Evaluate a Model Model outputs can be evaluated locally using the gem-metrics library or online which will add the outputs to our result overview (J6). Both methods require a standardized input format that specifies the dataset and split and which allows us to evaluate all 100+ data splits via the call `gem_metrics outputs.json`.

J5 - Add a new Metric In gem-metrics, each metric implements a `compute()` function and our library handles caching, parallelism, tokenization, etc. To avoid dependency conflicts, a metric can optionally specify a docker environment, as suggested by Deutsch and Roth (2022).

```python
from .texts import Predictions
from .texts import References
from .metric import ReferencedMetric

class NewMetric(ReferencedMetric):
    def _initialize(self):
        """Load models and artifacts."""
        pass
```

J6 - Use Prior Results Comparisons to prior work often only copy reported numbers which could be computed using different evaluation parameters, and a lack of released model outputs frequently prevents a fair side-by-side comparison outside of leaderboards (Gehrmann et al., 2022). To improve comparability, we add every online submission to a growing corpus of model outputs which evaluation researchers can use to develop better metrics or to conduct analyses.

3 Dataset Selection and Loading

To identify candidate datasets, continued to follow the SuperGLUE process (Wang et al., 2019) by soliciting tasks to be included from the research community. Our request to suggest multilingual, challenging, and/or interesting NLG tasks led to 40 submissions. To avoid quality judgments, we imposed only three requirements for selection: (1) consent from dataset authors, (2) availability under a permissive license, (3) the task needs to be able to be cast as a text-to-text problem. 27 tasks were selected in addition to the 13 existing ones (Gehrmann et al., 2021). Three datasets are simplification evaluation sets added to the WikiAuto loader (Jiang et al., 2020), while all others have independent data loaders. All data loaders and cards were produced as part of a month-long hackathon, and we invited the dataset authors and GEM participants to contribute to one or more of the datasets. Afterwards, the organizers managed the ongoing maintenance. New datasets can be added on an ongoing basis, subject to the three requirements. GEMv2 currently supports 40 datasets, listed in Appendix A and described in this section.

Figure 2 shows the distributions of training example count, task types, and their input and target lengths. Data-to-text and summarization are most common, followed by response generation. While data-to-text tasks are spread across resource availability categories, summarization datasets tend to

---

3 Documentation on how to add new datasets can be found at gem-benchmark.com/tutorials.
4 Marie (2022) discusses how this practice leads to harmful claims using a translation example (Costa-jussà et al., 2022).
5 Changes to datasets are documented in the appendix.
Figure 2: An overview of the properties of the currently supported datasets in GEM. (Top left) A histogram of the supported task types. The most represented tasks are Data-to-Text, followed by Summarization, Response Generation, and Simplification. (Bottom left) The frequency of different training corpus sizes for dataset configurations, broken down by their task types. While some task types are represented across all resource availability levels, some are concentrated on high resource. (Right) An overview of input and target lengths of different dataset configurations according to the mT5 tokenizer (Xue et al., 2021). Summarization tasks have input lengths of over 1,000 while all other tasks remain under 1,000 tokens. There is a lot more between-task variance in output length. Four dataset configurations are hidden due to the axis truncation.

be larger. While datasets vary in target length, the median input length tends to remain under 500 tokens, likely motivated by modeling limitations. Exceptions to this are summarization, with input lengths beyond what is supported by most models (e.g., WikiCatSum (Perez-Beltrachini et al., 2019) and XLSum (Hasan et al., 2021)), and a class of data-to-text datasets with the communicative goal to generate game summaries from large sports statistic tables (e.g., Hayashi et al., 2019; Thomson et al., 2020; Puduppully et al., 2019a).

We put an emphasis on language diversity, as prior work has found that fewer than 30% of NLG publications (even counting evaluations on machine translation) evaluate on non-English tasks (Gehrmann et al., 2022). While a lot of this focus on English can be traced to a lack of multilingual resources, many non-English NLG datasets have been released in recent years (e.g., Hasan et al., 2021; Ladhak et al., 2020; Mille et al., 2020; Cahyawijaya et al., 2021). As shown in Table 2, we support languages across all resource classes in the taxonomy by Joshi et al. (2020). However, the focus on English is still apparent in the number of datasets supporting a particular language, shown in Table 1, where English is far above all other languages. Moreover, most of the language diversity stems from the three highly multilingual datasets XLSum (Hasan et al., 2021), WikiLingua (Ladhak et al., 2020), and data from the surface realization shared task ‘20 (Mille et al., 2020). Excluding those, there are 13 datasets supporting non-English languages, 9 of which are exclusively non-English.

Of the 40 datasets, 14 have multiple configurations which can differ in task setup, languages, their encoding in romanized or original script, or domain. For example, we modified WikiLingua (Ladhak et al., 2020) to have splits from and to any of the 18 supported languages, enabling better cross-lingual evaluations. Seventeen datasets have challenge splits, many of which were created for GEM. For example, the challenge set for the conversational weather dataset (Balakrishnan et al., 2019) selects examples from the original test split with complex discourse relations.

4 Data Cards

Each dataset is accompanied by documentation about how it was created, who created it, how it should be used, and the risks in using it (Bender and Friedman, 2018; Gebru et al., 2018). Our original data documentation process (McMillan-Major et al., 2021) required filling out a markdown template following instructions in a separate guide. We
Table 1: The languages supported in GEMv2 and in how many of its datasets they appear.

| Count | Languages |
|-------|-----------|
| 1     | Amharic, Azerbaijani, Bengali, Burmese, Dutch, Gujarati, Hausa, Igbo, Javanese, Kirundi, Kyrgyz, Marathi, Nepali, Oromo, Pashto, Persian, Punjabi, Scottish Gaelic, Ser- bian, Sinhala, Somali, Sundanese, Swahili, Swedish, Tamil, Telugu, Tigrinya, Ukrainian, Urdu, Uzbek, Welsh, Yoruba |
| 2     | Czech, Italian, Thai, Turkish, Vietnamese |
| 3     | Arabic, Finnish, Hindi, Japanese, Korean, Portuguese |
| 4     | Indonesian |
| 6     | Chinese, German, Russian, Spanish |
| 8     | French |
| 28    | English |

Table 2: Supported languages categorized into the resource taxonomy by Joshi et al. (2020).

| Tax. | Languages |
|------|-----------|
| 0    | West African Pidgin English, Sinhala |
| 1    | Azerbaijani, Burmese, Gujarati, Igbo, Javanese, Kirundi, Kyrgyz, Nepali, Oromo, Pashto, Scottish Gaelic, Somali, Sundanese, Telugu, Welsh |
| 2    | Amharic, Hausa, Marathi, Punjabi, Swahili, Tigrinya, Yoruba |
| 3    | Bengali, Indonesian, Tamil, Thai, Ukrainian, Urdu, Uzbek |
| 4    | Czech, Dutch, Finnish, Hindi, Italian, Korean, Persian, Portuguese, Russian, Serbian, Swedish, Turkish, Vietnamese |
| 5    | Arabic, Chinese, English, French, German, Japanese, Spanish |

We analyzed the existing template and the resulting data cards under the dimensions provided in the data card playbook (Pushkarna et al., 2022) and identified the following improvements:

- **Accountability:** It needs to be clear who will maintain and extend the data cards when a dataset changes, when limitations of a dataset are found, or when it is deprecated (Corry et al., 2021).
- **Utility:** The recommended evaluation process for a dataset should be prominently shown.
- **Quality:** We need a process to validate data card completeness and quality.
- **Impact & Consequences:** It needs to be clear that we are curators, not editors, and that critiques reflect on the data, not the creators.
- **Risk & Recommendations I:** We need to expand the documentation of potential PII issues.
- **Risk & Recommendations II:** To help decide whether to use a dataset, the card needs to discuss differences from other datasets with similar communicative goals.

We modified our template following these insights and to be in line with the playbook approach of dividing between telescope, periscope, and microscope questions based on the length of the expected answer. We implemented this template in an interactive collection tool that can create new cards or load and update existing ones. The tool shows progress bars for the overall answer status and a breakdown for each of the subsections to indicate where more content should be added. The tool further improves the user experience by conditionally rendering questions based on prior answers, e.g., *Is there a risk of PII? → What kind of PII?*

The output of the tool is a structured json file that we convert into a simple markdown file for the data loader and an optimized web viewer and embedded in our website (Figure 1). The viewer presents important information at the top and splits the detailed rendering into three columns, corresponding to the telescope, periscope, and microscope split. This enables an easy navigation since high-level information can be found by focusing on the left column, moving toward the right for additional details.

The structured format enables us to study trends in dataset construction practices beyond those shown in Section 3.6 For example, 66% of the data cards report that PII is unlikely or definitely not included, while it is likely or definitely included in 33%. In the free-text explanations, we find four types of justifications for absent PII: The majority (7) stated that the data format or domain was restricted to avoid PII. Two stated that the data is in the public domain (e.g., Wikipedia) and another two used fully simulated data. One response described that crowd raters were instructed to avoid mentioning PII. We found that multiple of the PII-likely datasets only use public domain data, indicating that there is confusion about PII definitions.

Another typically hidden aspect is the data sourcing. Our datasets present an almost even split between automatically-, crowdworker-, and expert-created datasets, with crowdworker-created ones being slightly more common, possibly confounded if experts were hired through crowdfunding platforms, as was done for SQuality (Wang et al., 2022). It may thus also possible to compare which of these collection methods leads to more insightful modeling results. We follow up by asking

---

6We encourage others to use the publicly available files for additional investigations.
which crowdworking platform was used and unsurprisingly, Amazon Mechanical Turk was the most frequent answer, followed by participatory experiments and other non-specified platforms.

5 System Design

To support the automatic evaluation of outputs, we use the Hugging Face Hub to integrate datasets, metrics, and user interfaces for GEM users to submit their outputs. The system architecture is shown in Figure 3, and consists of five main components: Spaces We host Streamlit applications on Spaces for the submission of predictions, downloading of results, and visualization of model performance. Datasets Dataset repositories are used to host the datasets, submissions, evaluations, and results. AutoTrain We use AutoTrain, Hugging Face’s AutoML platform, to run all evaluation jobs using Hugging Face Benchmarks, a library that defines how metrics are computed within AutoTrain. Metrics We use GEM-metrics to perform the metric computations. In addition to supporting common metrics like BLEU (Papineni et al., 2002) and ROUGE (Lin, 2004), the Docker integration simplifies the calculation of multiple model-based metrics like BLEURT (Sellam et al., 2020).

On submission, a dataset repository with the model outputs is created under the GEM-submissions organisation on the Hugging Face Hub. In parallel, an evaluation job is triggered in AutoTrain which downloads the submission from the Hub, along with all the reference splits of the GEM datasets. These references are used to compute a wide variety of NLG metrics via GEM-metrics. The resulting metrics are then pushed to a dataset repository on the Hub, and used to source the visualization of results on the GEM website and Space.

6 Conclusion

We introduce GEMv2 which unifies infrastructure for generation research. We propose a consistent workflow from documenting and choosing datasets to loading and evaluating on them while keeping all supported datasets and metrics compatible with each other. We demonstrate the scalability by releasing the initial version with support for 40 datasets in 51 languages. Of the supported datasets, 23 are improved through configurations, filtering, and re-splitting processes and 17 datasets have challenge sets. We release a submission tool that computes metrics and makes model outputs available to download for evaluation researchers. Researchers who are interested in integrating their dataset are welcome to contact us for support.
7 Broader Impact

As discussed in the main part of the paper, GEMv2 aims to avoid any explicit curation decisions about inclusion and exclusion of datasets beyond licensing and consent. This is a change from the originally set out strict inclusion criteria based on dataset quality. The reason for this is that the entire research community should be the authority to decide whether a dataset is useful and what it is useful for. For example, a dataset with noisy outputs may still be useful to study hallucination avoidance methods. However, this change has implications on how dataset deprecation needs to be handled, in particular for datasets with newly found issues or datasets with better alternatives. Documenting issues and alternatives using the data cards is thus becoming more important in GEMv2 and we encourage researchers to update data cards. Another side effect of positioning GEMv2 as infrastructure that support dataset creators is a decreased risk of erasure. All our documentation and dataset loaders center the work of the creators to encourage users to cite the datasets they use.

Another open issue that we have been working on is the interplay between multilingualism and metrics. We now support multiple languages for which no NLG metrics have been tested, and for which our tokenization schemes may be inappropriate. The freedom to combine every dataset with every metric may lead to more flawed evaluations in those cases. In addition, some datasets were released with specific metrics that we do not support yet.

A final issue we want to point out is the lack of discussion of human evaluation in this overview paper which we omitted for brevity. Human evaluation does not scale and every task requires its own evaluation approach, especially when the goal is to deploy a system to real users. We have thus taken the approach to develop better human evaluation for only a subset of tasks, solving issues pointed out by Tang et al. (2022), Howcroft et al. (2020), and van der Lee et al. (2019), and we will release detailed instructions separately. However, these instructions will not replace a better understanding of the users of deployed systems.

References

Fernando Alva-Manchego, Louis Martin, Antoine Bordes, Carolina Scarton, Benoît Sagot, and Lucia Specchia. 2020. ASSET: A dataset for tuning and evaluation of sentence simplification models with multiple rewriting transformations. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 4668–4679, Online. Association for Computational Linguistics.

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 Annual Meeting of the Association for Computational Linguistics, pages 831–844, Florence, Italy. Association for Computational Linguistics.

Anya Belz, Shubham Agarwal, Anastasia Shimorina, and Ehud Reiter. 2021. A systematic review of reproducibility research in natural language processing. In Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, pages 381–393, Online. Association for Computational Linguistics.

Emily M. Bender and Batya Friedman. 2018. Data statements for natural language processing: Toward mitigating system bias and enabling better science. Transactions of the Association for Computational Linguistics, 6:587–604.

Chandra Bhagavatula, Ronan Le Bras, Chaitanya Malaviya, Keisuke Sakaguchi, Ari Holtzman, Hannah Rashkin, Doug Downey, Wen tau Yih, and Yejin Choi. 2020. Abductive commonsense reasoning. In International Conference on Learning Representations.

Samuel R. Bowman and George Dahl. 2021. What will it take to fix benchmarking in natural language understanding? In Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 4843–4855, Online. Association for Computational Linguistics.

Bill Byrne, Kartik Krishnamoorthi, Saravanan Ganesh, and Mihir Kale. 2021. TicketTalk: Toward human-level performance with end-to-end, transaction-based dialog systems. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pages 671–680, Online. Association for Computational Linguistics.

Bill Byrne, Kartik Krishnamoorthi, Chinnadurai Sankar, Arvind Neelakantan, Ben Goodrich, Daniel Duckworth, Semih Yavuz, Amit Dubey, Kyu-Young Kim, and Andy Cedilnik. 2019. Taskmaster-1: Toward a realistic and diverse dialog dataset. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pages 4516–4525, Hong Kong, China. Association for Computational Linguistics.
Rodriguez, Sashank Santhanam, João Sedoc, Thibault Sellam, Samira Shaikh, Anastasia Shimo-rina, Marco Antonio Sobrevilla Cabezudo, Hendrik Strobelt, Nishant Subramani, Wei Xu, Dyi Yang, Akhila Yerukola, and Jiawei Zhou. 2021. The GEM benchmark: Natural language generation, its evaluation and metrics. In Proceedings of the 1st Workshop on Natural Language Generation, Evaluation, and Metrics (GEM 2021), pages 96–120, Online. Association for Computational Linguistics.

Sebastian Gehrmann, Elizabeth Clark, and Thibault Sellam. 2022. Repairing the cracked foundation: A survey of obstacles in evaluation practices for generated text. CoRR, abs/2202.06935.

Tahmid Hasan, Abhik Bhattacharjee, Md. Saiful Islam, Kaz Mubasshir, Yuan-Fang Li, Yong-Bin Kang, M. Sohel Rahman, and Rifat Shahriyar. 2021. XL-sun: Large-scale multilingual and standardised summarization for 44 languages. In Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, pages 4693–4703, Online. Association for Computational Linguistics.

Hiroaki Hayashi, Yusuke Oda, Alexandra Birch, Ioannis Konstas, Andrew Finch, Minh-Thang Luong, Graham Neubig, and Katsuhito Sudo. 2019. Findings of the third workshop on neural generation and translation. In Proceedings of the 3rd Workshop on Neural Generation and Translation, pages 1–14, Hong Kong. Association for Computational Linguistics.

David M. Howcroft, Anya Belz, Miruna-Adriana Clinciu, Dimitra Gkatzia, Sadid A. Hasan, Saad Mahamood, Simon Mille, Emiel van Millingen, Sashank Santhanam, and Verena Rieser. 2020. Twenty years of confusion in human evaluation: NLG needs evaluation sheets and standardised definitions. In Proceedings of the 13th International Conference on Natural Language Generation, pages 169–182, Dublin, Ireland. Association for Computational Linguistics.

Chao Jiang, Mounica Maddela, Wuwei Lan, Yang Zhong, and Wei Xu. 2020. Neural CRF model for sentence alignment in text simplification. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 7943–7960, Online. Association for Computational Linguistics.

Pratik Joshi, Sebastin Santy, Amar Budhiraja, Kalika Bali, and Monojit Choudhury. 2020. The state and fate of linguistic diversity and inclusion in the NLP world. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 6282–6293, Online. Association for Computational Linguistics.

Juraj Juraska, Kevin Bowden, and Marilyn Walker. 2019. ViGGO: A video game corpus for data-to-text generation in open-domain conversation. In Proceedings of the 12th International Conference on Natural Language Generation, pages 164–172, Tokyo, Japan. Association for Computational Linguistics.

Mihir Kale and Abhinav Rastogi. 2020. Text-to-text pre-training for data-to-text tasks. In Proceedings of the 13th International Conference on Natural Language Generation, pages 97–102, Dublin, Ireland. Association for Computational Linguistics.

Moussa Kamal Eddine, Antoine Tixier, and Michalis Vazigiannis. 2021. BARThez: a skilled pretrained French sequence-to-sequence model. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 9369–9390, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Jenna Kanerva, Filip Ginter, Li-Hsin Chang, Iiro Rastas, Valtteri Skantsi, Jemina Kilpeläinen, Hanna-Mari Kupari, Jenna Saarni, Maija Sevön, and Otto Tarkka. 2021. Finnish paraphrase corpus. In Proceedings of the 23rd Nordic Conference on Computational Linguistics (NoDaLiDa), pages 288–298, Reykjavik, Iceland (Online). Linköping University Electronic Press, Sweden.

Jenna Kanerva, Filip Ginter, and Sampo Pyysalo. 2020. Turku enhanced parser pipeline: From raw text to enhanced graphs in the IWPT 2020 shared task. In Proceedings of the 16th International Conference on Parsing Technologies and the IWPT 2020 Shared Task on Parsing into Enhanced Universal Dependencies, pages 162–173, Online. Association for Computational Linguistics.

Jenna Kanerva, Samuel Rönnqvist, Riina Kekki, Tapio Salakoski, and Filip Ginter. 2019. Template-free data-to-text generation of Finnish sports news. In Proceedings of the 22nd Nordic Conference on Computational Linguistics, pages 242–252, Turku, Finland. Linköping University Electronic Press.

Jungo Kasai, Keisuke Sakaguchi, Ronan Le Bras, Lavinia Dunagan, Jacob Morrison, Alexander R. Fabbri, Yejin Choi, and Noah A. Smith. 2022. Bidimensional leaderboards: Generate and evaluate language hand in hand. In Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics, Seattle, Washington. Association for Computational Linguistics.

Joongwon Kim, Mounica Maddela, Reno Kriz, Wei Xu, and Chris Callison-Burch. 2021a. BiSECT: Learning to split and rephrase sentences with bitexts. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 6193–6209, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Seokhwan Kim, Yang Liu, Di Jin, Alexandros Papangelis, Karthik Gopalakrishnan, Behnam Hedayatnia, and Dilek Z. Hakkani-Tür. 2021b. “How robust r u?”: Evaluating task-oriented dialogue systems on spoken conversations. 2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU), pages 1147–1154.
Faisal Ladhak, Esin Durmus, Claire Cardie, and Kathleen McKeown. 2020. WikiLingua: A new benchmark dataset for cross-lingual abstractive summarization. In Findings of the Association for Computational Linguistics: EMNLP 2020, pages 4034–4048, Online. Association for Computational Linguistics.

Quentin Lhoest, Albert Villanova del Moral, Yacine Jerinite, Abhishek Thakur, Patrick von Platen, Suraj Patil, Julien Chaumond, Mariama Drame, Julien Plu, Lewis Tunstall, Joe Davison, Mario Šaško, Gunjan Chhablani, Bhavitya Malik, Simon Brandeis, Teven Le Scao, Victor Sanh, Canwen Xu, Nicolas Patry, Angelina McMillan-Major, Philipp Schmid, Sylvain Gugger, Clément Delangue, Théo Matusièrê, Lysandre Debut, Stan Bekman, Pierrick Cistic, Thibault Goehringer, Victor Mustar, François Lagunas, Alexander Rush, and Thomas Wolf. 2021. Datasets: A community library for natural language processing. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, pages 175–184, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Bill Yuchen Lin, Wangchunshu Zhou, Ming Shen, Pei Zhou, Chandra Bhagavatula, Yejin Choi, and Xiang Ren. 2020. CommonGen: A constrained text generation challenge for generative commonsense reasoning. In Findings of the Association for Computational Linguistics: EMNLP 2020, pages 1823–1840, Online. Association for Computational Linguistics.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In Text Summarization Branches Out, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Benjamin Marie. 2022. Science left behind.

Angelina McMillan-Major, Salomey Osei, Juan Diego Rodriguez, Pawan Sásanka Ammanananchi, Sebastian Gehrmann, and Yacine Jerinite. 2021. Reusable templates and guides for documenting datasets and models for natural language processing and generation: A case study of the HuggingFace and GEM data and model cards. In Proceedings of the 1st Workshop on Natural Language Generation, Evaluation, and Metrics (GEM 2021), pages 121–135, Online. Association for Computational Linguistics.

Simon Mille, Anya Belz, Bernd Bohnet, Thiago Castro Ferreira, Yvette Graham, and Leo Wanner. 2020. The third multilingual surface realisation shared task (SR’20): Overview and evaluation results. In Proceedings of the Third Workshop on Multilingual Surface Realisation, pages 1–20, Barcelona, Spain (Online). Association for Computational Linguistics.

Simon Mille, Kaustubh Dhole, Saad Mahamood, Laura Perez-Beltrachini, Varun Gangal, Mihir Kale, Emiel van Miltenburg, and Sebastian Gehrmann. 2021. Automatic construction of evaluation suites for natural language generation datasets. In Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1).

Linyong Nan, Dragomir Radev, Rui Zhang, Amrit Rau, Abhinand Sivaprasad, Chiachun Hsieh, Xianguang Tang, Aadit Vyas, Neha Verma, Pranav Krishna, Yangxiaokang Liu, Nadia Irwanto, Jessica Pan, Faiyaz Rahman, Ahmad Zaidi, Mutethia Mutumba, Yasin Tarabar, Ankit Gupta, Tao Yu, Yi Chern Tan, Xi Victoria Lin, Caiming Xiong, Richard Socher, and Nazneen Fatema Rajani. 2021. DART: Open-domain structured data record to text generation. In Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 432–447, Online. Association for Computational Linguistics.

Shashi Narayankutty, Shuyi Zhang, and Mirella Lapata. 2018. Don’t give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 1797–1807, Brussels, Belgium. Association for Computational Linguistics.

Jekaterina Novikova, Ondřej Dušek, and Verena Rieser. 2017. The E2E dataset: New challenges for end-to-end generation. In Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue, pages 201–206, Saarbrücken, Germany. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Ankur Parikh, Xuezhi Wang, Sebastian Gehrmann, Manaal Faruqui, Bhuwan Dhingra, Diyi Yang, and Dipanjan Das. 2020. ToTTo: A controlled table-to-text generation dataset. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 1173–1186, Online. Association for Computational Linguistics.

Laura Perez-Beltrachini and Mirella Lapata. 2021. Models and datasets for cross-lingual summarisation. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 9408–9423, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Laura Perez-Beltrachini, Yang Liu, and Mirella Lapata. 2019. Generating summaries with topic templates and structured convolutional decoders. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 5107–5116, Florence, Italy. Association for Computational Linguistics.

Ratish Pushpunotty, Li Dong, and Mirella Lapata. 2019a. Data-to-text generation with entity modeling. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 2023–
Ratish Puduppully and Mirella Lapata. 2021. **Data-to-text generation with macro planning.** *Transactions of the Association for Computational Linguistics*, 9:510–527.

Ratish Puduppully, Jonathan Mallinson, and Mirella Lapata. 2019b. University of Edinburgh’s submission to the document-level generation and translation shared task. In *Proceedings of the 3rd Workshop on Neural Generation and Translation*, pages 268–272, Hong Kong. Association for Computational Linguistics.

Mahima Pushkarna, Andrew Zaldivar, and Oddur Kjartansson. 2022. *Data cards: Purposeful and transparent dataset documentation for responsible ai*.

Jun Quan, Shian Zhang, Qian Cao, Zizhong Li, and Deyi Xiong. 2020. *RiSAWOZ: A large-scale multi-domain Wizard-of-Oz dataset with rich semantic annotations for task-oriented dialogue modeling*. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 930–940, Online. Association for Computational Linguistics.

Inioluwa Deborah Raji, Emily Denton, Emily M. Bender, Alex Hanna, and Amandalynne Paullada. 2021. AI and the everything in the whole world benchmark. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks I*, *NeurIPS Datasets and Benchmarks 2021*, December 2021, virtual.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. *SQuAD: 100,000+ Questions for Machine Comprehension of Text*. arXiv e-prints, page arXiv:1606.05250.

Abhinav Rastogi, Xiaoxue Zang, Srinivas Sunkara, Raghav Gupta, and Pranav Khaitan. 2020. Towards scalable multi-domain conversational agents: The schema-guided dialogue dataset. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8689–8696.

Fahimeh Saleh, Alexandre Berard, Ioan Calapodescu, and Laurent Besacier. 2019. *Naver labs Europe’s systems for the document-level generation and translation task at WNMT 2019*. In *Proceedings of the 3rd Workshop on Neural Generation and Translation*, pages 273–279, Hong Kong. Association for Computational Linguistics.

Thomas Scialom, Paul-Alexis Dray, Sylvain Lamprier, Benjamin Piwowski, and Jacopo Staiano. 2020. Mlsum: The multilingual summarization corpus. *arXiv preprint arXiv:2004.14900*.

Thibault Sellam, Dipanjan Das, and Ankur Parikh. 2020. **BLEURT: Learning robust metrics for text generation.** In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7881–7892, Online. Association for Computational Linguistics.
A Dataset Overviews

We provide a detailed overview of all the supported datasets in Table 3. Input and output lengths are reported in number of tokens according to the mT5 tokenizer (Xue et al., 2021). When multiple configurations for a dataset are available, we report the median of the sizes and lengths.

B Changes to Datasets

B.1 BiSECT

The original released BiSECT (Kim et al., 2021a) training, validation, and test splits are maintained to ensure a fair comparison. Note that the original BiSECT test set was created by manually selecting 583 high-quality Split and Rephrase instances from 1000 random source-target pairs sampled from the EMEA and JRC-Acquis corpora from the OPUS parallel corpus (Tiedemann and Nygaard, 2004).

As the first challenge set, we include the HSPLIT-Wiki test set, containing 359 pairs (Sulem et al., 2018). For each complex sentence, there are four reference splits; To ensure replicability, as reference splits, we again follow the original BiSECT paper and present only the references from HSPLIT2-full. In addition to the two evaluation sets used in the original BiSECT paper, we also introduce a second challenge set. For this, we initially consider all 7,293 pairs from the EMEA and JRC-Acquis corpora. From there, we classify each pair using the classification algorithm from Section 4.2 of the original BiSECT paper. The three classes are as follows:

1. Direct Insertion: when a long sentence \( l \) contains two independent clauses and requires only minor changes in order to make a fluent and meaning-preserving split \( s \).

2. Changes near Split, when \( l \) contains one independent and one dependent clause, but modifications are restricted to the region where \( l \) is split.

3. Changes across Sentences, where major changes are required throughout \( l \) in order to create a fluent split \( s \).

We keep only pairs labeled as Type 3, and after filtering out pairs with significant length differences (signaling potential content addition/deletion), we present a second challenge set of 1,798 pairs.
Table 3: Detailed information about all the datasets currently supported in GEM. We present the name of the dataset, the paper(s) in which the dataset was introduced, the NLG task it performs, the languages the dataset caters to and their resourcedness taxonomy class, the size of the training set (rounded), and the lengths of input and output.

| Dataset | Citation | Task | Language(s) | Taxonomy | Size | Input Length | Output Length |
|---------|----------|------|-------------|----------|------|--------------|---------------|
| ART | (Bhagavatula et al., 2020) | Reasoning | en | 5 | 50k | 59k | 138 | 41 |
| i2b2 | (Kim et al., 2021a) | Simplification | en, de, es, fr | 5 | 200k–1M | 266–434 | 224–387 |
| CoCite | (Devaraj et al., 2021) | Simplification | en | 5 | 3.5k | 3k | 0 |
| Conversational Weather | (Balakrishnan et al., 2019) | Response Generation | en | 5 | 25k | 417 | 315 |
| CrossWOWZ | (Zhu et al., 2020) | Response Generation | zh | 5 | 5k | 4.5k | 5 |
| CS Restaurants | (Dalek and Jurčíček, 2019) | Response Generation | cs | 4 | 3.5k | 70 | 58 |
| DART | (Nan et al., 2021) | Data-to-Text | en | 5 | 60k | 60k | 60 |
| DSTC10 | (Kim et al., 2021b) | Data-to-Text | en | 5 | 20k | 1337 | 95 |
| E2E NLG | (Novikova et al., 2017; Dalek et al., 2020; Dalek et al., 2019) | Data-to-Text | en, de | 5 | 35k | 146 | 135 |
| FairytaleQA | (Xu et al., 2022) | Question Generation | en | 5 | 5.5k | 335 | 15.9 |
| IndoNLG | (Charyavijaya et al., 2021) | Summarization | id, ja, sv, zh | 1–3 | 14k–20k | 2021 | 456 |
| MLB | (Poduppuully et al., 2019a) | Data-to-Text | en | 5 | 23k | 24665 | 2580 |
| MLSum | (Scialom et al., 2020) | Summarization | es, de | 5 | 220k–250k | 4152 | 147 |
| Opusparcus | (Creutz, 2018) | Paraphrasing | de, en, es, fi, fr, ru, sv | 4–5 | 6–35M | 6–35M | 6–35M |
| OrangeSum | (Kanerva et al., 2021) | Summarization | fr | 5 | 20k | 1984 | 138 |
| R2SAWOC | (Quan et al., 2020) | Response Generation | zh | 5 | 10k | 50k | 50 |
| RotoWrite En-De | (Wixman et al., 2017; Hayashi et al., 2019) | Data-to-Text | en, de | 5 | 242 | 242 | 242 |
| Schema-Guided Dialog | (Rastogi et al., 2020) | Response Generation | en | 5 | 160k | 188 | 51 |
| SciDiety | (Sun et al., 2021) | Slide-Generation | en | 5 | 2k | 2k | 2k |
| SIMPTIKI | (Tonelli et al., 2016) | Simplification | it | 4 | 815 | 815 | 815 |
| SportSenti | (Thomson et al., 2020) | Data-to-Text | en | 5 | 3.7k | 5990 | 1620 |
| Squad V2 | (Rajpurkar et al., 2020) | Question Generation | en | 5 | 120k | 768 | 55 |
| Squad V2 | (Wang et al., 2022) | Summarization | zh | 5 | 250k | 2500 | 227 |
| Surface Realization ST 2020 | (Müller et al., 2020) | Data-to-Text | en, es, fr, hi, hi, ja, pt, ru, zh | 3–5 | 250k | 892 | 126 |
| TaskMaster | (Byrne et al., 2019) | Response Generation | en | 5 | 190k | 972 | 55 |
| TextTo | (Panik et al., 2020) | Data-to-Text | en | 5 | 120k | 357 | 357 |
| Turku Hockey | (Kanerva et al., 2019) | Data-to-Text | fi | 4 | 27k–51.1k | 158 | 58 |
| Turku Paraphrase | (Kanerva et al., 2021) | Paraphrasing | en | 5 | 81k–130k | 87 | 47 |
| ViGo | (Jänsch et al., 2019) | Data-to-Text | en | 5 | 5k | 120 | 109 |
| WebNLG | (Gardent et al., 2017a,b) | Data-to-Text | en, ru | 4–5 | 14k–35k | 169.5 | 157 |
| WikiAuto | +ASSSET/TURK| Split&Rephrase | (Jiang et al., 2020; Alva-Manchego et al., 2020; Xu et al., 2016; Zhang et al., 2020) | Simplification | en | 5 | 480k | 480k |
| WikiCarSum | (Perez-Beltrachini et al., 2019) | Summarization | en, cs, de, en, es, fr, hi, id, it, ja, ko, nl, pt, ru, th, tr, vi, zh | 3–5 | 48k | 43527 | 256 |
| WidiLog | (Ludhak et al., 2020) | Summarization | aa, az, ba, ca, cy, en, es, gd, fa, gu, ha, hi, id, ja, ko, ky, mr, my, ne, om, sh, sr, si, so, sw, ta, te, th, ti, uk, ur, ur, vi, yo, zh-CN, zh-TW | 0–5 | 1.3M–300k | 470–9924 | 137–614 |
| XLSum | (Hasan et al., 2021) | Summarization | en | 5 | 23k | 1845 | 153 |
| XWiki | (Perez-Beltrachini and Lapata, 2021) | Summarization | en, de, fr | 4–5 | 44k–461k | 1743 | 102 |

B.2 FairytaleQA

The original release of FairytaleQA (Xu et al., 2022) used separate files to store the fairytale story content and experts-labeled QA-pairs. It provided baseline benchmarks on both Question Answering and Question Generation tasks. In GEMv2, we re-organize the data to be specifically prepared for the Question Generation task. The original dataset contains 2 answers created by different annotators in the evaluation and test splits, but we only take the first answer into consideration for the Question Generation task. The input for this task would be the concatenation of each answer labeled by human experts and the related story section(s), and the output target would be the corresponding question labeled by human experts.

B.3 MLB Data to Text

We follow the serialization format introduced in (Puduppully and Lapata, 2021) for the linearized_input field. Specifically, we serialize the home team records, the visiting team records, and the player records. We next serialize the records of the innings in chronological order.

B.4 Opusparcus

Compared to the original release of Opusparcus (Creutz, 2018), available through the Language Bank of Finland, the GEMv2 release contains a few additions to facilitate the use of this resource:

- The validation and test sets now come in two versions, the so-called regular validation and test sets and the full sets. The regular sets only contain
sentence pairs that qualify as paraphrases. The full sets are the original sets from the original release, which contain all sentence pairs successfully annotated by the annotators, including the sentence pairs that were rejected as paraphrases. The validation sets were called development sets in the original release.

The training sets are orders of magnitudes larger than the validation and test sets. Therefore the training sets have not been annotated manually and the true paraphrase status of each entry is unknown. In the original release, each training set entry is accompanied by an automatically calculated ranking score, which reflects how likely that entry contains a true paraphrase pair. The entries are ordered in the data, best first, worst last. If you use the original release, you need to control yourself how large and how clean a portion of the training data you will use.

In the GEMv2 release, the training sets come in predefined subsets. Using the so-called quality parameter, the user can control for the estimated proportion (in percent) of true paraphrases in the retrieved training subset. Allowed quality values range between 60 and 100, in increments of 5 (60, 65, 70, ..., 100). A value of 60 means that 60% of the sentence pairs in the training set are estimated to be true paraphrases (and the remaining 40% are not). A higher value produces a smaller but cleaner set. The smaller sets are subsets of the larger sets, such that the quality=95 set is a subset of quality=90, which is a subset of quality=85, and so on. Depending on this parameter, the dataset can fall into all resourcedness categories in Figure 2.

B.5 ROTO_WIRE_English-German
We introduce a field linearized_input, which serializes the input table into a string. We follow a serialization format similar to that of Saleh et al. (2019). More specifically, we serialize all the records of the home team followed by that of the visiting team. We next serialize the records of the players of the home team followed by that of the visiting team. We rank the players by points in descending order. In addition, we add information about the relative rank of a player within a team following Puduppully et al. (2019b).

B.6 SciDuet
The original released SciDuet (Sun et al., 2021) uses two json files to store paper information and slide information, respectively. In GEMv2, we merge these two files and reorganize the structure so that each data instance contains the complete input (i.e., paper title/abstract/section headers/section content, as well as slide title) and output (i.e., slide text content). In addition, we introduce a new challenging dataset in GEMv2 by removing slides if their titles match any section headers from the corresponding paper.

B.7 SIMPITIKI
The original release of SIMPITIKI (Tonelli et al., 2016) includes two xml files, corresponding to the version 1 and version 2 respectively. The second version has better sentence boundaries. However, no training, validation and test splits were officially proposed for both release. In GEM, we randomly and independently split both xml files into training, validation and test sets. Note that version 1 and version 2 have different splits. We also generated challenge sets were some simplification transformations in the test set are not part of the training set and thus unseen in the training phase. Then, as SIMPITIKI leverages data from Wikipedia and the Municipality of Trento corpora, we further propose splits based on the respective data source.

B.8 SportSett Basketball
Similar to MLB Data-to-Text, SportSett also follows the serialization format introduced in (Puduppully and Lapata, 2021) for the linearized_input field. The serialization starts with current game’s information such as date and venue of the game. This is followed with both team’s information (line-scores) including their next game’s information as well. Finally, the players’ information (box-scores) is serialised, starting with home team’s players and then visiting team’s players.

B.9 squad_v2
SQuAD2.0 (Rajpurkar et al., 2016) combines the 100,000 questions in SQuAD1.1 with over 50,000 unanswerable questions written by crowdsworkers to look similar to answerable ones. The original SQuAD2.0 dataset has only training and dev (validation) splits. A new test split is created from the train split and added as part of the squad_v2 dataset.

B.10 Taskmaster-3
According to Byrne et al. (2021), the Taskmaster-3 (also called TicketTalk) dataset consists of 23,789 movie ticketing dialogs, where the customer’s goal
is to purchase tickets after deciding on theater, time, movie name, number of tickets, and date, or opt out of the transaction. This collection was created using the "self-dialog" method, i.e., a single, crowdsourced worker is paid to create a conversation writing turns for both speakers - the customer and the ticketing agent.

B.11 Turku Hockey
To ease the use of the data, in addition to the game-level structuring as used in the original Turku Hockey data release (Kanerva et al., 2019), we provide a simplified event-level structuring. In the event-level generation, the structured input data is linearized to string representation separately for each game event, and the task objective is thus to generate the description separately for each game event directly using the linearized input representation. In comparison, the objective of the game-level generation is to process the structured data for the entire game at once, and generate descriptions for all relevant events. The linearized event inputs are produced using similar approach as described in the original paper.

B.12 Turku Paraphrase
In GEMv2, the Turku Paraphrase data can be loaded with three different configurations, plain, classification, and generation. While the plain configuration models the data similarly to the original release, the two other options directly applies several transformations beneficial for the named task. In classification each example is provided using both (text1, text2, label) and (text2, text1, label) ordering, as paraphrase classification does not depend on the order of the given statements. In cases with a directionality annotation in the paraphrase pair, the label is flipped accordingly when creating the additional examples. In generation, on the other hand, the data is pre-processed to include only examples suitable for the paraphrase generation task, therefore discarding, e.g., negative and highly context dependent examples, which does not fit the generation task as such. In addition, the examples with annotated directionality (one statements being more detailed than the other, for instance one mentioning a woman while the other a person), the example is always provided using ordering where the input is more detailed and the output more general in order to prevent model hallucination (model learning to generate facts not present in the input). For more details about the annotated labels and the directionality, see Kanerva et al. (2020).

B.13 WikiLingua
The original release of WikiLingua (Ladhak et al., 2020) released a dataset of article-summary pairs in 18 languages, but had only created train/val/test splits for 4 language pairs (es-en, tr-en, ru-en, vi-en), for the purposes of crosslingual evaluation. As part of GEMv1, we created train/val/test splits for all 18 languages. To further facilitate building multilingual and crosslingual models for all 18 languages, the GEMv2 release contains the following changes to the GEMv1 release:

In the original WikiLingua release, each document-summary pair in any of the 17 non-English languages has a corresponding parallel document-summary pair in English. A given English document-summary pair can have parallel document-summary pairs in multiple languages. In order to facilitate crosslingual experiments across all language pairs, for the GEMv2 release, we align document-summary pairs across the other 17 languages via English. For example, if a given document-summary pair in English has corresponding parallel pairs in Turkish and Vietnamese, we can then align these to get Turkish-Vietnamese parallel pairs. As a result, in addition to supporting all the functionality in GEMv1, the v2 loader allows the user to specify and load crosslingual data for any language pair in the dataset.

In addition to the original evaluation sets (val and test), we also have sub-sampled versions in order to facilitate faster development cycles. To create the sub-sampled versions, for each evaluation set, we randomly sample 3,000 instances. We further clean the dataset by removing payloads for thumbnails that were scraped into the document and summary texts and we filter out all instances with a summary length longer than 60% of the input document length. This removes around 5% of the data.

C Contribution Statements
Organizing GEM would not be possible without community contributions and the mutual goal of improving NLG and its evaluation. To give proper credit to all contributors, this section lists the involvements of all co-authors. Besides the detailed list, everyone contributed to discussion sessions,

---

13Evaluation sets that have fewer than 3,000 instances were not sub-sampled.
made dataset suggestions, and participated in proofreading the final paper.

**Dataset Loaders**  The new data loaders and associated data cards were created by the following people:

- **ART**: Chandra Bhagavatula, Nico Daheim, Aman Madaan
- **BiSect**: Jenny Chim, Reno Kriz
- **Conversational Weather**: Vipul Raheja, Michael White
- **CrossWOZ**: Qi Zhu
- **DSTC10**: Nico Daheim, Di Jin, Alexandros Papangelis
- **FairyTaleQA**: Bingsheng Yao
- **IndoNLG**: Bryan Wilie, Samuel Cahyawijaya, Genta Indra Winata
- **MLB**: Ratish Puduppully
- **Opusparcus**: Mathias Creutz
- **OrangeSum**: Moussa Kamal Eddine
- **RiSAWOZ**: Tianhao Shen, Deyi Xiong, Chaobin You
- **RotoWire En-De**: Hiroaki Hayashi, Ratish Puduppully
- **SciDuet**: Yufang Hou, Dakuo Wang
- **SIMPITIKI**: Sebastien Montella, Vipul Raheja
- **Split and Rephrase**: Cristina Garbacea, Reno Kriz, Li Zhang
- **SportSett**: Craig Thomson, Ashish Upadhyay
- **Squad V2**: Abinaya Mahendiran
- **SQuALITY**: Alex Wang
- **Surface Realisation ST**: Bernd Bohnet, Simon Mille
- **TaskMaster**: Tosin Adewumi
- **ToTTo (port)**: Abinaya Mahendiran
- **Turku Hockey**: Filip Ginter, Jenna Kanerva
- **Turku Paraphrase**: Filip Ginter, Jenna Kanerva
- **ViGGo**: Juraj Juraska, Aman Madaan
- **WikiCatSum**: Ronald Cardenas Acosta, Laura Perez-Beltrachini
- **WikiLingua (port)**: Jenny Chim, Faisal Ladhak
- **XLSum**: Abhik Bhattacharjee, Tahmid Hasan, Rifat Shahriyar
- **XSum (port)**: Abinaya Mahendiran
- **XWikis**: Ronald Cardenas Acosta, Laura Perez-Beltrachini

Lewis Tunstall designed and implemented the infrastructure to host GEMv2 on the Hugging Face Hub. Sebastian Gehrmann addressed the remaining loader issues and ported the remaining GEMv1 datasets. Anna Shvets developed dataset-agnostic bias detection filters. Simon Mille coordinated progress during the hackathon.

**Documentation**  The updated tutorials for using GEM and adding new data loaders were developed and tested by Jenny Chim, Paul Pu Liang, and Anna Shvets.

**Data Cards**  The questions in the revised data card template were created during sessions led by Mahima Pushkarna with the help of Yacine Jernite, Angelina McMillan-Major, Nishant Subramani, Pawan SASanka Ammanamanchi, and Sebastian Gehrmann. The collection tool was implemented by Yacine Jernite and Sebastian Gehrmann. The data card rendering tool was developed by Vivan Tsai and Mahima Pushkarna.

**Human Evaluation**  The human evaluation working group is led by João Sedoc. Its members include Jenny Chim, Elizabeth Clark, Daniel Deutsch, Kaustubh Dhole, Khyathi Raghavi Chandu, Sebastian Gehrmann, Yufang Hou, Yixin Liu, Saad Mahamood, Simon Mille, Vitaly Nikolaev, Salomey Osei, Dragomir Radev, Yisi Sang, and Alex Wang.

**Metrics**  The metrics library, originally developed for GEMv1, was extended by Jordan Clive, Nico Daheim, Daniel Deutsch, Ondrej Dusek, Sebastian Gehrmann, Aman Madaan, Joshua Maynez, Vikas Raunak, Leonardo F. R. Ribeiro, and Anna Shvets.

**Paper Writing and Analyses**  Sebastian Gehrmann led the writing of the paper. Abinaya Mahendiran and Jekaterina Novikova contributed analyses that were used to create Figure 2 and Table 3.

**Submission Infrastructure**  Lewis Tunstall led the development of the submission infrastructure. Hendrik Strobelt led the extension of the result visualization tool to ensure compatibility with the new submission system.

**Baselines**  Additional baseline results were provided by Tosin Adewumi, Mihir Sanjay Kale, Joshua Maynez, and Leonardo F. R. Ribeiro.