# factgenie: A Framework for Span-based Evaluation of Generated Texts

**Zdeněk Kasner Ondřej Plátek Patrícia Schmidtová**
**Simone Balloccu Ondřej Dušek**
Institute of Formal and Applied Linguistics
Faculty of Mathematics and Physics, Charles University
`${surname}@ufal.mff.cuni.cz`

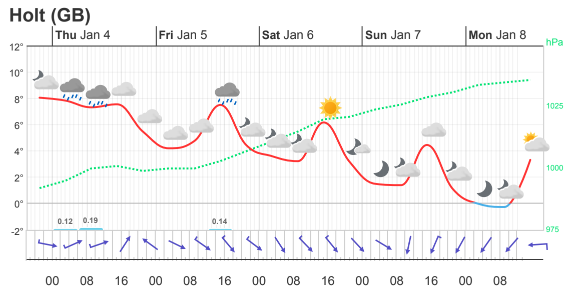
## Abstract

We present factgenie: a framework for annotating and visualizing word spans in textual model outputs. Annotations can capture various span-based phenomena such as semantic inaccuracies or irrelevant text. With factgenie, the annotations can be collected both from human crowdworkers and large language models. Our framework consists of a web interface for data visualization and gathering text annotations, powered by an easily extensible codebase.[1]

## 1 Introduction

The fluency of texts generated by large language models (LLMs) is reaching the level of human-written texts. However, the texts generated by LLMs still contain various types of errors such as incorrect claims, claims not grounded in the input, or irrelevant statements. For precise and fine-grained evaluation of model outputs, it is necessary to identify these errors on the level of word spans. There are two major ways to collect the span annotations: using either human (Thomson and Reiter, 2020) or LLM-based annotators (Kocmi and Federmann, 2023; Kasner and Dušek, 2024).

None of the existing NLP error annotation platforms are suitable for gathering and visualizing word-level annotations from both human and LLM-based annotators. Some platforms are limited to specific tasks like machine translation (Klejch et al., 2015) and retrieval-augmented generation (ES et al., 2024). Other platforms are more flexible but allow either only human (Federmann, 2018; Nakayama et al., 2018) or only LLM (Dalvi et al., 2024) annotations. Systems supporting both annotation modalities typically include humans as post-editors only (Kim et al., 2024) and existing evaluation or visualization platforms require externally pre-annotated data (Trebuňa and Dušek, 2023; Masson et al., 2024; Fittschen et al., 2024).




(a) Custom visualization of the input data.

llama2

The weather in Holt, England, will be mostly cloudy with scattered clouds for the next five days. The temperature will range from 4.2 to 7.4 degrees Celsius, with a high of 6.1 degrees Celsius on January 7th. The wind speed will be around 2.5 to 4.6 meters per second, with gusts of up to 9.32 meters per second on January 4th. There will be light rain on January 4th, with a total of 0.12 millimeters of precipitation. The humidity will be around 77% to 97%.

(b) Annotated model output.

Figure 1: Elements from the factgenie user interface: (a) custom visualization of the input data, (b) the corresponding LLM output with span annotations. The highlight colors correspond to custom annotation categories defined for the annotation process ( = incorrect fact, = fact not checkable, = misleading fact).

The lack of suitable tools for span-based error annotation motivated us to develop factgenie, a lightweight and customizable framework that enables collecting annotations from both humans and LLMs. Specifically, factgenie can be used both to (a) collect annotations from human workers through crowdsourcing services and (b) collect annotations by prompting an LLM through an API. Besides that, factgenie can be used for visualizing the input data and the corresponding model outputs.

The software design of factgenie targets researchers, who can easily self-host and customize

[1] Code is available at https://github.com/kasnerz/factgenie/. System demonstration video: https://youtu.be/CsVcCGvOzPY.



*Proceedings of the 17th International Natural Language Generation Conference: System Demonstrations*, pages 13–15
September 23–27, 2024. 

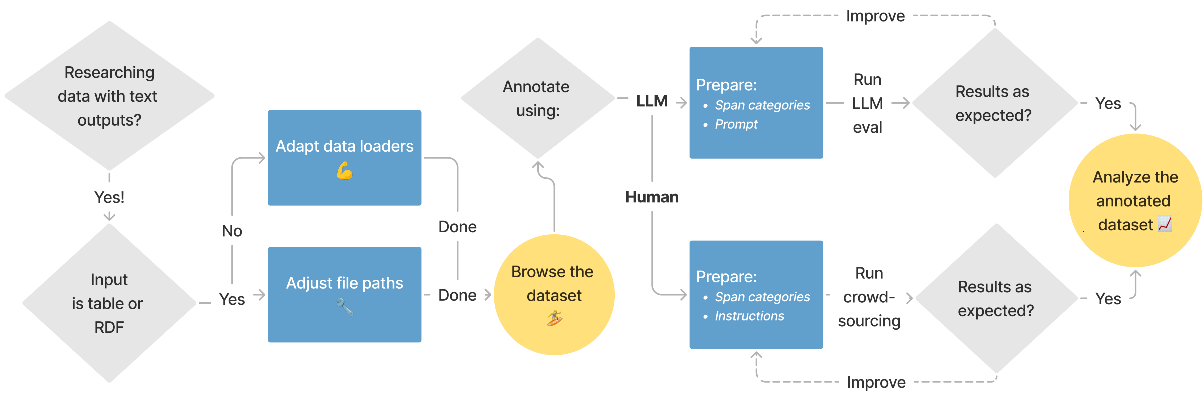



Figure 2: `factgenie` workflow. Actions needed for using `factgenie` for custom tasks are shown in blue rectangles.

it for individual experiments. The benefits of `factgenie` include:

- Visualization of input data and model outputs with a few lines of code,
- Ready-made web interface for collecting annotations from crowdsourcing services,
- Support for gathering model-based annotations from multiple LLM APIs,
- Tools for managing and visualizing collected annotations.

## 2 Framework

Software-wise, `factgenie` is a combination of a Flask backend and an HTML-based frontend. The frontend is powered by Boostrap 5.3 and jQuery, additionally using the YPet library for collecting span annotations. For visualizing the example input data, we use TinyHTML and Highcharts.JS.[2]

Figure 1 shows an example with weather data and the corresponding model-generated weather forecast. The model output was annotated for errors through `factgenie`. Note that the colors and labels of text span annotation categories can be customized for each set of annotations.

The framework can be used as-is or customized to cover a wide range of tasks and needs with minimal effort. To load and preview a new dataset, researchers first need to write a data loader class. Existing data loaders include various visualizations of tabular, RDF, and JSON data. As shown in Figure 2, loading a dataset in a supported format can be as easy as changing a path to the data on the file system. To add a custom dataset type, the researcher must extend the `Dataset` class. Once the dataset is loaded, `factgenie` allows data inspection and rapid prototyping of LLM annotations and crowdsourcing campaigns.

## 3 Human Annotations

To collect error annotation from human crowdworkers, researchers typically build custom web interfaces. With `factgenie`, researchers can easily build an annotation interface in four steps:

1. Define the campaign parameters (annotation span categories, number of examples per annotator, etc.),
2. Write instructions for the annotators,
3. Host `factgenie` on a public URL,
4. Redirect the annotators to the running `factgenie` instance.

The interface can be previewed for internal testing throughout the process. As shown in Figure 2, `factgenie` provides the feedback necessary for debugging and improving the evaluation campaign by an immediate visualization of the collected annotations.

## 4 LLM Annotations

It is useful to obtain annotations from LLMs in the same format as from human annotators. For that, `factgenie` provides a lightweight wrapper for model APIs.[3] The process of collecting annotations from LLMs consists of the following steps:

1. Define the campaign parameters (annotation span categories, model decoding parameters, API endpoint, etc.)

[2]In principle, `factgenie` can render datasets using any custom HTML code and JS libraries.

[3]We currently support the Ollama API for self-hosted LLMs and the OpenAI API for cloud LLMs.

2. Write the prompt and system message for the model,[4]
3. Run the LLM annotation inference.

Similarly to human annotations (Section 3), the evaluation progress can be monitored and immediately visualized.

## 5 Roadmap

The development of `factgenie` is ongoing and open to external developers. We are currently working on facilitating the management of evaluation campaigns by adding an option to set-up the evaluation campaign from the web interface in addition to configuration files. In the future, we plan to add more ready-made classes for data loaders, model APIs, and crowdsourcing services.

## Acknowledgements

This work was funded by the European Union (ERC, NG-NLG, 101039303) and Charles University projects GAUK 40222 and SVV 260 698. It used resources of the LINDAT/CLARIAH-CZ Research Infrastructure (Czech Ministry of Education, Youth, and Sports project No. LM2018101).

## References

Fahim Dalvi, Maram Hasanain, Sabri Boughorbel, Basel Mousi, Samir Abdaljalil, Nizi Nazar, Ahmed Abdelali, Shammur Absar Chowdhury, Hamdy Mubarak, and Ahmed Ali. 2024. LLMeBench: A flexible framework for accelerating LLMs benchmarking. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 214–222, St. Julians, Malta.

Shahul ES, Jithin James, Luis Espinosa Anke, and Steven Schockaert. 2024. RAGAs: Automated evaluation of retrieval augmented generation. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2024 - System Demonstrations, St. Julians*, pages 150–158, Malta.

Christian Federmann. 2018. Appraise evaluation framework for machine translation. In *COLING 2018, The 27th International Conference on Computational Linguistics: System Demonstrations*, pages 86–88, Santa Fe, New Mexico.

Elisabeth Fittschen, Tim Fischer, Daniel Brühl, Julia Spahr, Yuliia Lysa, and Phuoc Thang Le. 2024. AnnoPlot: Interactive visualizations of text annotations. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 106–114, St. Julians, Malta.

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

Hannah Kim, Kushan Mitra, Rafael Li Chen, Sajjadur Rahman, and Dan Zhang. 2024. MEGAnno+: A Human-LLM collaborative annotation system. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 168–176, St. Julians, Malta.

Ondrej Klejch, Eleftherios Avramidis, Aljoscha Burchardt, and Martin Popel. 2015. MT-ComparEval: Graphical evaluation interface for machine translation development. *Prague Bull. Math. Linguistics*, 104:63–74.

Tom Kocmi and Christian Federmann. 2023. 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.

Maxime Masson, Christian Sallaberry, Marie-Noelle Bessagnet, Annig Le Parc Lacayrelle, Philippe Roose, and Rodrigo Agerri. 2024. TextBI: An interactive dashboard for visualizing multidimensional NLP annotations in social media data. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 1–9, St. Julians, Malta.

Hiroki Nakayama, Takahiro Kubo, Junya Kamura, Yasufumi Taniguchi, and Xu Liang. 2018. doccano: Text annotation tool for human. Software available from `https://github.com/doccano/doccano`.

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.

František Trebuňa and Ondřej Dušek. 2023. VisuaLLM: Easy web-based visualization for neural language generation. In *Proceedings of the 16th International Natural Language Generation Conference: System Demonstrations*, pages 6–8, Prague, Czechia.

[4]The prompt needs to instruct the model to produce JSON with a specific structure. Note that the APIs we support can ensure decoding JSON output, see, e.g., `https://platform.openai.com/docs/guides/json-mode`.