ReX-GG: a LLM Ensemble Pipeline for Relation-extraction and Graph Generation

Giacomo Magnifico, Eduard Barbu
Institute of Computer Science, University of Tartu
Tartu, Estonia
{giacomo.magnifico, eduard.barbu}@ut.ee

Abstract

Current LLM ensemble frameworks focus on multi-step setups with additional modules for answer ranking, often opting for token and span analysis rather than structured outputs, leading to heavyweight architectures with potential fail states along the pipeline. Faster, lighter solutions are more vulnerable to hallucination propagation and can lack output control in more complex pipelines. This paper proposes a customisable, lightweight ensemble workflow of coordinated Large Language Models that leverages JSON-structured outputs and anonymous peer-review ranking to mitigate hallucinatory outputs and single-model failure points. The pipeline is demonstrated on a relation extraction task applied to popular science articles in English, targeting four ontologically-grounded relation types (strong causation, weak causation, contrastive, and compositional), with semantic node canonicalisation and interactive, colour-coded HTML causal graphs as the final output. Performance is evaluated through an anonymous user study, achieving an average perceived accuracy of 0.778 against a human-annotated gold standard. The modular architecture supports flexible deployment across both API-based and in-house LLM setups, and the full framework is released under an open license to foster reproducibility and collaborative research.

Keywords: graph, human survey, lightweight framework, LLM ensemble, relation extraction, structured output

1. Introduction

Since the advent of OpenAI and the permeating use of Large Language Models (LLMs) in everyday life and academia, the ease of access to state-of-the-art models and diffusion has led to new avenues of research; from testing the limits of these models to mitigating their shortcomings, exposing their faults and tuning their performance. As expected, the use of individual models has led to the development of LLM ensembles, tailored to achieve more robust performance and to leverage perceived differences among these models (Jiang et al., 2023). Other ways to enhance the reliability and robustness of their output have been found at the intersection of formal structure and natural language processing (NLP), in the form of information-based graphs, with examples of knowledge-graph-informed architectures that leverage the power of ontologies and formal structure (Zhou et al., 2025; Zhu et al., 2025). Within this specific niche, it is possible to find widely different research avenues - from graph-based LLM ensemble performance evaluations (He et al., 2025) to analyses of LLM inner workings via graph representation of their chain of thought (Zhiqiang et al., 2025). The production of knowledge graph outputs seems to be less explored, with Pham Hoang Le et al. (2025) and Li et al. (2025) as examples, although only the former explores ensembles and structured outputs, while the latter focuses on LLM tuning.

To foster further research in this niche and to provide an accessible tool to the research community, this paper presents a **lightweight example of an information-extraction ensemble workflow, tailored for relation extraction and interactive graphs**. The proposed workflow can be deployed with a customisable selection of coordinated models, delivering robust performance and a reduced risk of hallucinatory or malformed output thanks to the peer-reviewed ranking of schema-enforced JSONs. The practical application of the pipeline targets English popular science articles, with performance tested via an anonymous survey to provide public-informed evaluation of the relation-extraction framework; anecdotal evidence shows a preference for colour-coded graph edges representative of the evaluated bonds. The secondary advantages of this work can be identified as the following:

- simple overarching structure with flexible end use;
- built-in mitigation of hallucination propagation;
- reduced risk of malformed outputs failure-chains;
- fast deployment as a modular piece in larger architectures.

Details regarding the structure and behaviour of the ensemble are provided in Section 3.1, along with the ontology and details regarding the nature of the relations chosen for extraction; further information related to the canonicalization of the extracted pairwise relation triplets and development of the graph structure are given in Sections 3.2 and

Proceedings of the Knowledge Graphs and Large Language Models Workshop (KG-LLM) @ LREC26, pages 93–101
May 16, 2026.

3.3. In Sections 5 and 6, we discuss the results of our user study presented in Section 4, expand on the key findings of this paper, and outline potential future work. We reserve Sections 7 and 8 to discuss the shortcomings and limitations of our scope and to address potential concerns about the ethical standards of this work.

2. Related Works

Multiple works deploy different sets of LLMs as ensembles with variable cohesion for a multitude of purposes: from Xu et al. (2025), which evaluates a 7B selection of models and enhances decision-making through token span analysis, to the more complex database-oriented tasks focusing on tabular data QA seen in Bujnowski et al. (2025), to striving for the optimization of the ensemble with additional framework refinement as proposed by Tekin et al. (2024).

Related to our work are the concepts and implementation of the "LLM forest" found in He et al. (2025), which consists of a graph-*informed* ensemble of models to achieve higher performance in data inputation for subsequent downstream tasks; the performance of graph-informed processes is further investigated in Zhiqiang et al. (2025), alongside the Explainable Graph Language Model framework. Graph-*represented* decisions, which are closer to the scope of this paper, are explored in Xiong et al. (2025) as a means to clarify the reasoning process of LLMs, converting large freeform text chain-of-thoughts (CoTs) into a readable and measurable graph to explain the causal bonds; the difference in scope lies within model-reasoning applications compared to extractive tools for information transmission. While the output of LLMs is converted into graph form, and the enforced structure within responses is a shared property, the work presented in Wu et al. (2025) makes use of external repositories and ultimately outputs natural language text based on knowledge graphs.

The two papers that share most of the background with the ensemble of our proposed work are Parfenova and Pfeffer (2025) and Pham Hoang Le et al. (2025); while the former proposes a two-staged ensemble with the aid of a moderator to be deployed for inductive coding, with an emphasis on the use of LoRA finetuned architectures, the latter provides a two-stage pipeline with a smaller zero-shot ensemble. Despite the structural similarities, our work differs both in scope and use case: the focus on causal relations and the use of few-shot prompts distances it from Pham Hoang Le et al. (2025), and the preferred deployment as an intermediary between freeform text and HTML graphs falls further away from Parfenova and Pfeffer (2025)'s scope.

Lastly, what could be considered a precursor for this work is presented in Li et al. (2025) as a single-model pipeline that specifically produces causal graphs from narrative text; although both papers share similar goals, the core difference lies in the use of finetuned BERT architectures and summarisers to overcome the performance of base LLMs, turning the pipeline into a robust but heavy-weight solution.

The reviewed literature reveals that while LLM ensembles have grown increasingly capable, their complexity has scaled accordingly, often requiring additional ranking modules, finetuned architectures, or external knowledge repositories. Works that produce graph-structured outputs tend to focus on model reasoning transparency rather than information transmission, and those that do target relation extraction either rely on heavyweight finetuned pipelines or limit themselves to single-model setups. No existing work combines a lightweight heterogeneous ensemble with schema-enforced structured outputs, ontologically-grounded relation types, and interactive graph visualisation in a single deployable framework. This paper directly addresses that gap, proposing a *modular, robust yet lightweight solution for everyday information extraction that prioritises accessibility and ease of deployment without sacrificing output reliability*.

3. Pipeline Structure

3.1. Model Ensemble

Relation Extraction. After receiving input from the user with the name and extension of the file to be processed, the read information is stored and sent to each model in the ensemble for the preliminary relation extraction. Our chosen focus is on narratively-driven direct pairwise relations, examples of which can be found both in scientific-literature-based tasks (Pham Hoang Le et al., 2025) and narrative text (Li et al., 2025); the relation types we focus on are based on the works by Ross (2025) and Magnifico (2025), among others, and are defined below.

- **Strong Causation**. Explains a phenomenon by specifying explicit causal mechanisms or multiple steps in a causal chain. The underlying process is made clear. *"Your lack of sleep is making you clumsy. That's why you spilled the milk."*

- **Weak Causation**. Explains through correlational or probabilistic relationships between variables without specifying the underlying mechanism. Often involves associations, tendencies, or indirect influences. *"Diseases with*

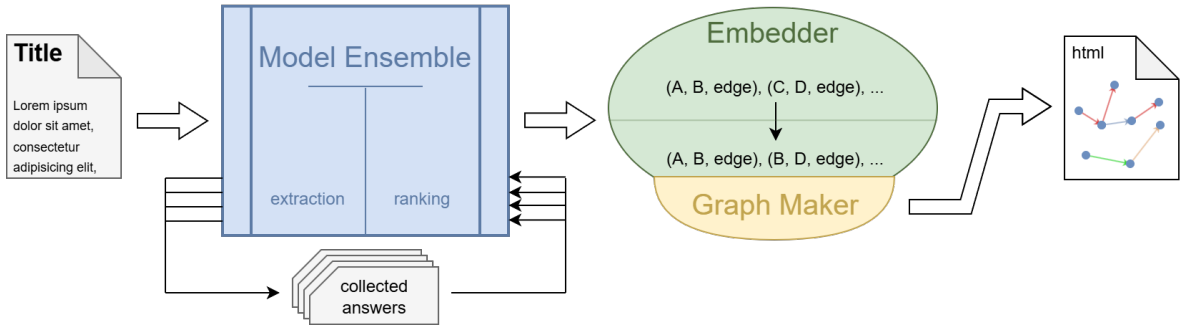

Figure 1: Schematic representation of the pipeline presented in the paper: a text file is fed to the LLM ensemble, where relation extraction is performed, and the best output is decided through anonymous peer-ranking; the extracted relation triplets iterate through an embedding model to improve graph structure and assimilate nodes with semantic similarity. The final result is a causal graph in HTML format.

uncertain causes were about 50% more likely to attract religious or magical treatments."

- **Contrastive**. Explains a phenomenon by highlighting differences between alternative conditions or cases (A vs B). *"In the treatment group, 70% recovered; in the placebo group, only 30% recovered."*
- **Compositional**. States what something is made of, contains, consists of, or is defined as (whole/entity → part/definition). *"Protons and neutrons are combinations of even tinier particles, called quarks."*

To maximise the potential of the extraction pipeline, further rules for the two extremes of the pairwise relations are introduced. First, each leftmost extreme (the future *head* node) is defined as a *[concept, cause, outcome A, entity]*, while the rightmost extreme (the future *tail* node) is defined as a *[concept, effect, outcome B, constituent]*. Second, each extracted extreme must be a *minimal span of words*, preferably a *nominal phrase* - as it is the minimal identifier for a concept, be it a subject or an object in a sentence - and with each pronoun/anaphora resolved to its *explicit referent*. Third, rewriting the original text is preferable to quoting long sentences directly. A possible example of the extracted relations for the paragraph

"Of course your hands are all jittery, you had a big gulp of coffee earlier! Maybe that's why you're dropping stuff?"

would be *["big gulp of coffee", "jittery hands", strong causation]*, and *["jittery hands", "dropped stuff", weak causation]*. After lengthy tuning, the final set of instructions follows the best practices for prompt engineering and can be summarised with this outline:

- task definition;
- output schema example;
- rules for schema completion;
  - rules for node extraction;
  - rules for edge extraction;
- set of input-output examples

As a reliable method of enforcing the schema structure given with the prompt, the choice of `structured_output` settings is enabled; the resulting output is in JSON format as a single-entry Python dictionary, with the word "document" as the *key* and a list of smaller dictionaries as the *value*. The list formatting allows for the iterative use of dictionaries with the same shared keys ("bond", "span 1", "span 2") associated with different values, corresponding to the extracted pairwise relations and relation extremes. In this section of the pipeline, there is no interaction among the models, and each output is produced independently. All the JSON-structured outputs from this first pass through the ensemble, representative of the relation bonds extracted from the input document, are then passed along in the pipeline.

Model Peer Ranking. As the outputs of the first ensemble pass have been stored separately for each individual model, they are anonymised to avoid both potential self-preference bias from any architecture and tuning-induced preferences present in the training data. To facilitate easy post-processing de-anonymisation, each model is assigned a corresponding letter from the English alphabet, starting with A. The anonymised outputs are then given to each model in the ensemble, along with the original document and a reminder of the classification rules, with the explicit task of assigning a 10-point grade to each.

Once more, these outputs are forced into a JSON structure and must provide the following dictionary entries:

- *answer :* string, label of the ranked answer (e.g. A, B, C);

- *points :* integer, points assigned to the ranked answer (0 to 10);
- *thinking :* string, providing a reasoning behind the score of the ranked answer.

.

The points for each answer label are stored and summed as each model in the ensemble goes through this process, forming a "leaderboard" with the comprehensive rating of each anonymous output. Afterwards, the answers are de-anonymised, and the user is informed of *how high each model's output scored, along with the one chosen as the most accurate*. The best-voted output is then sent along the pipeline.

Of relevant mention is the use, through the entire ensemble pipeline, of continuously updated `query_logs` and `error_logs`; not only is the entirety of the content generated by the ensemble safely stored in a separate text file, but each potential break in the sequence due to unstructured outputs is recorded in a dedicated log file. From the input given to the ensemble to the final structured output, everything is stored for maximum clarity of use.

The models used for the setup presented in this paper have been deployed through API calls via the OpenRouter platform[1], with an approximate cost of £10 for the entirety of the multi-stage process. One model for each major corporation has been chosen based on popularity and performance ratings on aggregator websites as of February 2026. The final ensemble of LLMs is comprised of GPT 5.2 (2025), Claude Sonnet 4.5 (2025), Gemini 3 Flash (2025), Deepseek 3.2 (2024), Llama 3.3 (70B-instruct) (2024), Nova 2 Lite V1 (2024), Grok 4.1 Fast (2025); all of the outputs are set to `structured`, with a 4000 token limit and the same fixed seed for the entire process to keep consistency as high as possible.

3.2. Embedding Model

After the LLM ensemble, the second block of our pipeline prepares the best-voted answers (as lists of dictionary-encoded lists of semantic triples) for later graph representation. As the semantic triples are stored with a high likelihood of non-identical text strings for similar concepts (e.g. “a coffee”, “coffee”, “nice cup of coffee”), a graph derived by the raw list would be scattered and represent only two-node relations; with no changes, the compositional properties of causal bonds would be lost, as well as any other defined relation type that spans multiple concepts. To achieve a cohesive structure and minimise fragmentation, we turn to semantic embedding as a lightweight yet effective form of control. Each list of dictionary-encoded relation pairs is processed as detailed in Algorithm 1 before moving forward in the pipeline.

Algorithm 1 Semantic Triples Node Assimilation

$inputs \leftarrow$ list of dictionaries
$relations = [\,]$
for i in $inputs$ **do**
  if $i[relation] \neq$ none **then**
    $pair \leftarrow (i[A], i[B], i[relation])$
    $relations \leftarrow relations + pair$
  end if
end for
for $i = 1$ to $i = \#relations$ **do**
  $x \leftarrow relations[i-1][1]$
  $y \leftarrow relations[i][0]$
  $X, Y = embedder.encode(x, y)$
  if $embedder.similar(X, Y) \geq value$ **then**
    $y \leftarrow x$
  end if
end for

The first half of the process shows how the inputs received from the LLM ensemble are converted from dictionary encoding into *tuple* object types, all while keeping the RDF structure

$$A \rightarrow relation \rightarrow B$$

with possible examples being

$$coffee \rightarrow strong\ causation \rightarrow\ jittery\ hands$$

$$jittery\ hands \rightarrow weak\ causation \rightarrow drop\ stuff.$$

An immediate change in the conversion is the substitution of the relation types mentioned in Section 3.1 with the colour associated with them. Since the final output of the pipeline has to be intuitively understandable and easy to read, we opt for the colour-coding *[red, orange, cyan, green]* for *[strong causation, weak causation, contrastive, compositional]*. Thus, the two examples above would be converted into

$$coffee \rightarrow red \rightarrow jittery\ hands$$

$$jittery\ hands \rightarrow orange \rightarrow drop\ stuff.$$

Worth noting is that each pairwise relation is stored in a *A-B-relation* order, as a popular format for graph information storage is the *(head, tail, directed edge)* notation.

Presented in the second half of Algorithm 1 is our solution for graph sparcity: an iteration over the list of stored semantic triples using an embedding model to achieve a rudimentary but efficient form of multi-node relation canonicalization. As subsequent triples are iterated through, the *tail* of

[1] https://openrouter.ai/

the previous one is compared with the *head* of the following through vector embedding and similarity measure between the embeds; if the semantic similarity is higher than a chosen threshold, the two concepts are assumed to be syntactic variations of the same semantic term. If that is the case, the two nodes become one and the same, transforming a pair of arcs

$$A \rightarrow relation \rightarrow B$$

$$C \rightarrow relation \rightarrow D$$

into

$$A \rightarrow relation \rightarrow B$$

$$B \rightarrow relation \rightarrow D$$

which represent the assimilated triplet composition

$$A \rightarrow relation \rightarrow (B \rightarrow relation \rightarrow D).$$

We deploy `Jasper-Token-Compression-600M` (Zhang et al., 2025) as a semantic embedding model, given its superior performance relative to near-state-of-the-art language models.

3.3. Interactive Graph Maker

Given the input from the embedding algorithm, we populate a `NerworkX`[2] directed graph with given dimensions of 1080p $\times$ 1080p. The directed graph is then converted into a `PyVis`[3] object to allow for better interactivity, and it is exported as an HTML file - akin to the sample provided in Figure 2 as an image. We chose this specific format because it enables a more hands-on approach for the end user, allowing them to move nodes around, enable/disable graph physics, highlight nodes and vertices, and so on.

4. Human Evaluation

To avoid authorship bias in the effectiveness and correctness of the graphs generated by the pipeline, an anonymous survey was distributed to paid volunteers recruited via the *Prolific* web app[4]. Two graphs were annotated by the authors to serve as a human-annotated gold standard baseline, and then evaluated by two dedicated groups of 3 anonymous testers. The remaining 18 articles in the set were processed by the pipeline and each was evaluated by 6 testers. The scoring method was as follows: a four-point evaluation of the usefulness of the graph; a five-point evaluation of the accuracy of the information provided in the graph; and a final four-point score for the tester's confidence in their scores. Although the perceived usefulness of the graphs was

Figure 2: A full graph output from our pipeline, derived from the article "Snow fleas use their tails to jump around the ice", available at this link.

not considered in the current evaluation, the testers' confidence was used to compute a weighted average across annotator groups for each article. The scores for the accuracy of author-annotated graphs served as a baseline for comparing the accuracy of the remaining data, as they represented the "ideal accuracy" for manual annotators.

	Accuracy	**Length**	$\times\sigma$
A1	0.807	2342	0.2
A13	0.939	2374	1.1
A14	0.903	2444	0.9
A7	1.000	2603	1.6
A10	0.745	3062	0.2
A12	0.880	3245	0.7
A5	0.550	3270	1.6
A18	0.521	3709	1.8
A3	0.880	3895	0.7
A15	0.953	4253	1.2
A6	0.623	4426	1.1
A4	0.790	4764	0.1
A8	0.835	4967	0.4
A16	0.831	5215	0.4
A11	0.636	5950	1.0
A17	0.815	9078	0.3
A2	0.632	9542	1.0
A9	0.670	10321	0.8
Avg.	*0.778*	*4748*	*1*

Table 1: Scores for the perceived accuracy of each graph, normalised against the human-annotated gold standard, with the distance from the mean measured in standard deviations.

Divided by original document and in increasing

[2] https://networkx.org/en/

[3] https://pyvis.readthedocs.io/en/latest/index.html

[4] https://www.prolific.com/

length order, the scores for each graph's perceived accuracy and standard deviation steps from the average are shown in Table 4. A visual representation is available in Figure 3, along with the projected mean of the perceived accuracy and a $\pm\sigma$ representation. The furthest outliers in the data (A5, A7, A18) are still within 2 σ from the mean, and the majority of values are within one σ.

To provide a visual representation of the robustness of our proposed solution, Figure 4 presents the average grading for each individual model (dotted lines) across the selection of articles, with the model ensemble pipeline as a comparison (full line).

5. Performance Analysis

The processed results from the anonymous survey can be helpful in defining the key properties of the deployed pipeline, ranging from the relation between performance and document length and the perceived accuracy of the graphs to the robustness of the ensemble.

Ensemble vs Single-model Rank. As shown in Figure 4, the variability in the performance of each single LLM presents a wide margin of potential error that can lead to inaccurate graphs and poorly-formed outputs. Even the best-performing model is shown to have instances of rankings lower than 30 points, and the less-performing models at times rise to higher scores. Overall, the ensemble ranking leads to more stable, robust performance.

Document Length vs Accuracy. Based on the results shown in Table 4 and the distribution presented in Figure 3, we can conclude that there is no direct relation between document length (measured in characters) and perceived accuracy of the relative graph. Both the Table and the Figure show that the highest accuracy and the lowest are both within the initial range (2000 to 5000 characters); it could be speculated that the accuracy scores show a diminishing trend as the documents become longer, but a more balanced evaluation would be that the variance between results lowers with longer documents.

Human-rated Accuracy. Taking into consideration the mixed nature of the Accuracy score, indicative both of completeness compared to the original text (e.g. number of core relations reported in graph) and accuracy of the information presented (e.g. two elements put in correlation rather than contrast), our pipeline produces high-quality outputs with near-human annotation peaks and an average of *0.778*. All of the human ratings for the interactive graph outputs are within 2 standard deviations from the mean, with the majority (66%) being within 1 standard deviation. Even without the removal of the largest outliers in both directions, the limited span of distribution for the ratings supports the assumption of an effective pipeline.

Robustness. Due to the nature of the ensemble architecture, each of the models is forced to produce a JSON output, and each part of the content-generation and recording sequence carefully looks for formatting errors; as each output is stored independently and given anonymously for peer-review ranking to each model, the pipeline avoids the pitfalls of hallucination propagation and unusable outputs. The anonymous nature of the ensemble rankings allows for personalised use with a *single LLM ensemble setups* for even lighter-weight scenarios, and it is adaptable to both in-house and API LLM usage.

6. Conclusions

In this paper, we have presented a lightweight pipeline for information extraction that leverages an ensemble workflow of seven independent models, coordinated to reliably mitigate hallucinatory outputs and malformed graph generation. Our framework achieves robust performance by leveraging JSON-structured outputs and independent model-peer-review ranking strategies. When the deployed pipeline is tested on a selection of English popular science articles, the reported performance is close to the human-annotated gold standard, with an average accuracy of the provided information of 0.778 - measured by aggregated scoring from anonymous annotators. The outputs are presented to end users as interactive, colour-coded HTML files that facilitate immediate, intuitive understanding.

Anecdotal evidence supports the ease of use and deployment of the proposed pipeline even by less-specialised users, and the modular nature of the ensemble allows for highly customisable settings; possible implementations that benefit from the anonymous peer-review model-ranking include free and open source solutions, such as in-house single-LLM ensembles with multiple-persona structures, or heterogeneous ensembles that can benefit from an array of different-scale models.

Future work is necessary to reach higher performance and more interpretable results, as reports from anonymous users in the study suggest that a central topic node and a specific top-down or left-right directionality would be beneficial for graph understanding. Further testing is also necessary to verify the robustness of our pipeline across different settings and contexts, aiming for higher generalisation and a wider selection of relation types; a potential addendum could be investigating input multimodality to determine the best format for maximal accuracy. Finally, developing a simpler software solution that allows non-command-line execution of

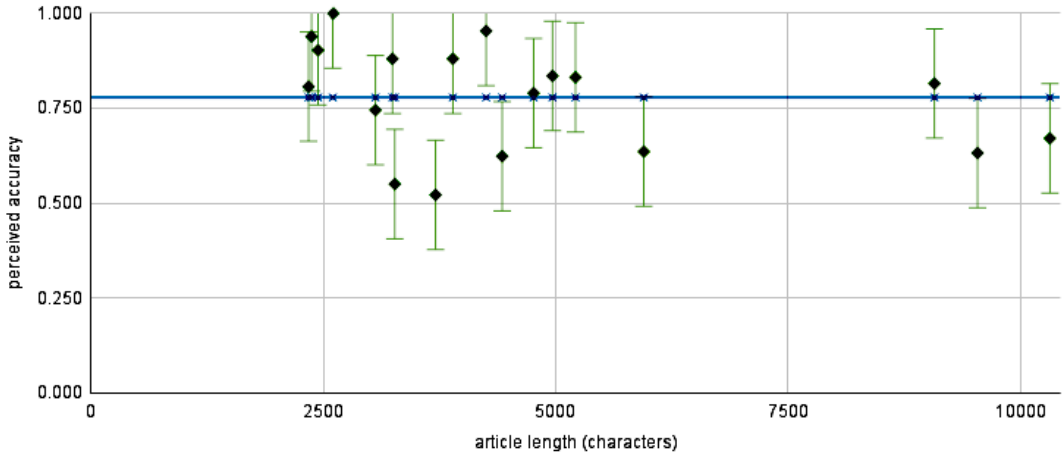

Figure 3: Performance of the graph produced by our pipeline according to the anonymous survey. The averaged accuracy scores for each article are represented as filled ⋄, along with the line marking the mean value.

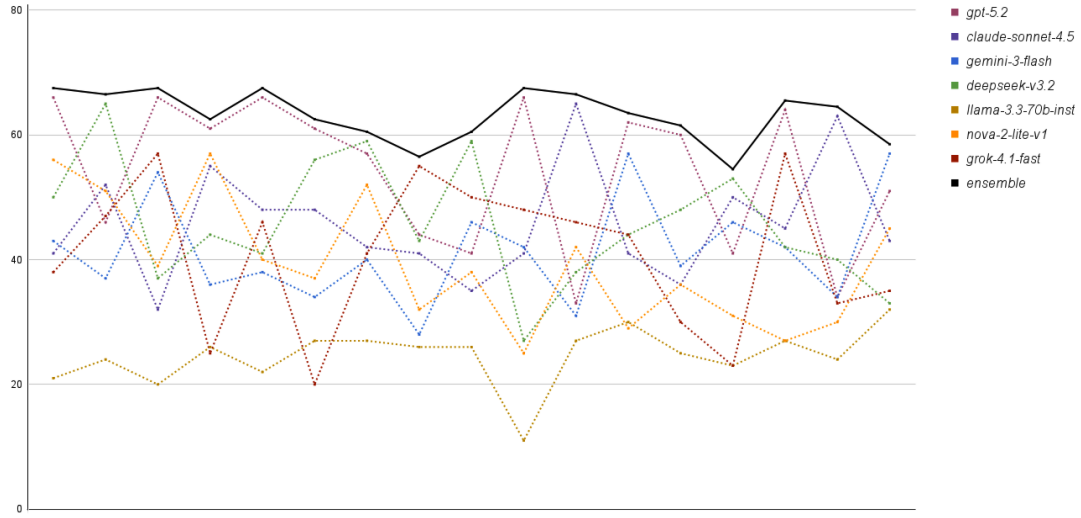

Figure 4: Ranking of our ensemble scores compared to the performance of the individual model, ranging from a minimum of 0 to a maximum of 70 points, over the articles evaluated.

the pipeline would improve accessibility and ease of use for the general public.

All the software presented in this paper is freely available at our GitHub repository[5] to foster cooperative research and an open-science mindset in the academic community.

7. Limitations

Since our paper focuses on presenting a lightweight alternative to heavier information-extraction pipelines, some limitations regarding performance and size are bound to the nature of the work. The current workflow has been tested with shorter documents, which are easier to read and evaluate for the anonymous testers; consequently, proper testing with sizeable documents (e.g. entire scientific research papers) falls outside of the current scope despite it being, perhaps, the more interesting application for our pipeline. For the sake of simplicity and customisation, aside from the processed data and the log information, all the information flow is strictly online, as the models are called through API, and a true analysis of the reasoning process for LLMs is not available. Anecdotal evidence also suggests that the output graphs would benefit from a central node with the general topic of the input document as the origin, with radial orientation to improve comprehension. Lastly, the human evaluation is more qualitative than quantitative, as the number of evaluated documents is on the lower end of the spectrum.

[5] https://github.com/gima9552/ReX-GG

8. Ethical Considerations

The reliance on LLMs to extract information from texts and present it to the public can inherit data-induced bias and carry the harm of potential misinformation. Our framework mitigates the propagation of hallucinatory outputs by leveraging the ensemble-based ranking, but it is not guaranteed to have a completely faithful output in every single case; therefore, the recommendation is to always introduce human supervision in the loop, and generally defer judgment to the user. The evaluation done with anonymous testers was conducted with no risk of harm and within safe work regimes, and the payment was above the platform's minimum wage standards. To the best of our knowledge, the work presented in this paper does not pose any significant risk of harm in itself.

9. Acknowledgements

This work was supported by the project "Terminology-Aware Machine Translation for Accessible Science" (TaMTAS, MOB3ERA10), funded by the Estonian Research Council under the Mobilitas 3.0 ERA-NET programme and co-funded by the European Union under the CHIST-ERA Call 2025 "Science in Your Own Language."

10. Bibliographical References

Anthropic. 2025. Introducing claude sonnet 4.5.

Amazon AWS. 2024. Amazon nova foundation models.

Pawel Bujnowski, Tomasz Dryjanski, Christian Goltz, Bartosz Swiderski, Natalia Paszkiewicz, Bartlomiej Kuzma, Jacek Rutkowski, Jakub Stepka, Milosz Dudek, Wojciech Siemiatkowski, Weronika Plichta, Bartłomiej Paziewski, Maciej Grabowski, Katarzyna Beksa, Zuzanna Bordzicka, Filip Ostrowski, and Grzegorz Sochacki. 2025. Samsung research Poland at SemEval-2025 task 8: LLM ensemble methods for QA over tabular data. In *Proceedings of the 19th International Workshop on Semantic Evaluation (SemEval-2025)*, pages 1223–1232, Vienna, Austria. Association for Computational Linguistics.

DeepSeek-AI, :, Xiao Bi, Deli Chen, Guanting Chen, Shanhuang Chen, Damai Dai, Chengqi Deng, Honghui Ding, Kai Dong, Qiushi Du, Zhe Fu, Huazuo Gao, Kaige Gao, Wenjun Gao, Ruiqi Ge, Kang Guan, Daya Guo, Jianzhong Guo, Guangbo Hao, Zhewen Hao, Ying He, Wenjie Hu, Panpan Huang, Erhang Li, Guowei Li, Jiashi Li, Yao Li, Y. K. Li, Wenfeng Liang, Fangyun Lin, A. X. Liu, Bo Liu, Wen Liu, Xiaodong Liu, Xin Liu, Yiyuan Liu, Haoyu Lu, Shanghao Lu, Fuli Luo, Shirong Ma, Xiaotao Nie, Tian Pei, Yishi Piao, Junjie Qiu, Hui Qu, Tongzheng Ren, Zehui Ren, Chong Ruan, Zhangli Sha, Zhihong Shao, Junxiao Song, Xuecheng Su, Jingxiang Sun, Yaofeng Sun, Minghui Tang, Bingxuan Wang, Peiyi Wang, Shiyu Wang, Yaohui Wang, Yongji Wang, Tong Wu, Y. Wu, Xin Xie, Zhenda Xie, Ziwei Xie, Yiliang Xiong, Hanwei Xu, R. X. Xu, Yanhong Xu, Dejian Yang, Yuxiang You, Shuiping Yu, Xingkai Yu, B. Zhang, Haowei Zhang, Lecong Zhang, Liyue Zhang, Mingchuan Zhang, Minghua Zhang, Wentao Zhang, Yichao Zhang, Chenggang Zhao, Yao Zhao, Shangyan Zhou, Shunfeng Zhou, Qihao Zhu, and Yuheng Zou. 2024. Deepseek llm: Scaling open-source language models with longtermism.

Google. 2025. Gemini 3 flash documentation.

Xinrui He, Yikun Ban, Jiaru Zou, Tianxin Wei, Curtiss Cook, and Jingrui He. 2025. LLM-forest: Ensemble learning of LLMs with graph-augmented prompts for data imputation. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 6921–6936, Vienna, Austria. Association for Computational Linguistics.

Dongfu Jiang, Xiang Ren, and Bill Yuchen Lin. 2023. LLM-blender: Ensembling large language models with pairwise ranking and generative fusion. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14165–14178, Toronto, Canada. Association for Computational Linguistics.

Zehan Li, Ruhua Pan, and Xinyu Pi. 2025. Beyond LLMs a linguistic approach to causal graph generation from narrative texts. In *Proceedings of the The 7th Workshop on Narrative Understanding*, pages 36–51, Albuquerque, New Mexico. Association for Computational Linguistics.

Giacomo Magnifico. 2025. Automated classification of causal relations. evaluating different LLM performances. In *Proceedings of the 9th Student Research Workshop associated with the International Conference Recent Advances in Natural Language Processing*, pages 27–36, Varna, Bulgaria. INCOMA Ltd., Shoumen, Bulgaria.

Meta. 2024. Llama 3.3: Model cards & prompt formats.

OpenAI. 2025. Introducing gpt 5.2.

Angelina Parfenova and Jürgen Pfeffer. 2025. Measuring what matters: Evaluating ensemble LLMs with label refinement in inductive coding. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 10803–10816, Vienna, Austria. Association for Computational Linguistics.

Nguyen Pham Hoang Le, An Dinh Thien, Son T. Luu, and Kiet Van Nguyen. 2025. DocIE@XLLM25: ZeroSemble - robust and efficient zero-shot document information extraction with heterogeneous large language model ensembles. In *Proceedings of the 1st Joint Workshop on Large Language Models and Structure Modeling (XLLM 2025)*, pages 288–297, Vienna, Austria. Association for Computational Linguistics.

Lauren N. Ross. 2025. *Explanation in Biology*. Elements in the Philosophy of Biology. Cambridge University Press.

Selim Furkan Tekin, Fatih Ilhan, Tiansheng Huang, Sihao Hu, and Ling Liu. 2024. LLM-TOPLA: Efficient LLM ensemble by maximising diversity. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 11951–11966, Miami, Florida, USA. Association for Computational Linguistics.

Junde Wu, Jiayuan Zhu, Yunli Qi, Jingkun Chen, Min Xu, Filippo Menolascina, Yueming Jin, and Vicente Grau. 2025. Medical graph RAG: Evidence-based medical large language model via graph retrieval-augmented generation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 28443–28467, Vienna, Austria. Association for Computational Linguistics.

xAI. 2025. Grok 4.1.

Zhen Xiong, Yujun Cai, Zhecheng Li, and Yiwei Wang. 2025. Mapping the minds of LLMs: A graph-based analysis of reasoning LLMs. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 17751–17763, Suzhou, China. Association for Computational Linguistics.

Yangyifan Xu, Jianghao Chen, Junhong Wu, and Jiajun Zhang. 2025. Hit the sweet spot! span-level ensemble for large language models. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 8314–8325, Abu Dhabi, UAE. Association for Computational Linguistics.

Dun Zhang, Ziyang Zeng, Yudong Zhou, and Shuyang Lu. 2025. Jasper-token-compression-600m technical report.

Mo Zhiqiang, Yang Hua, Jiahui Li, Yuan Liu, Shawn Wong, and Jianmin Huang. 2025. Judge and improve: Towards a better reasoning of knowledge graphs with large language models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 5303–5320, Suzhou, China. Association for Computational Linguistics.

Yigeng Zhou, Wu Li, Yifan Lu, Jing Li, Fangming Liu, Meishan Zhang, Yequan Wang, Daojing He, Honghai Liu, and Min Zhang. 2025. Reflection on knowledge graph for large language models reasoning. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 23840–23857, Vienna, Austria. Association for Computational Linguistics.

Xiangrong Zhu, Yuexiang Xie, Yi Liu, Yaliang Li, and Wei Hu. 2025. Knowledge graph-guided retrieval augmented generation. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 8912–8924, Albuquerque, New Mexico. Association for Computational Linguistics.