Constrained Linked Entity ANnotation using RAG (CLEANR)

Task entry of the Graz University of Technology, for the BioASQ Task GutBrainIE on Gut-Brain Interplay Information Extraction at CLEF 2025 in the team ToGS

Benedikt Kantz[1,*], Stefan Lengauer[1], Peter Waldert[1] and Tobias Schreck[1]

[1]Graz University of Technology, Rechbauerstrasse 12, Graz, Austria

Abstract

Structured information extraction from text relies heavily on natural language processing tools and a robust understanding of the structure. Language Models (LMs) provide the text understanding for long and unstructured input, even in domain-specific data. The generative aspect of these systems, however, can be unstructured and quickly return data that does not conform to the intended structural constraints. Our system, Constrained Linked Entity ANnotation using RAG (CLEANR), introduces structured output based on the ontological constraint placed through a grammar to the LM. This addition enables us to reliably utilize relatively small and inexpensive models in our pipeline to process domain-specific data for information extraction in the CLEF GutBrainIE task, resulting in good precision in the Relation Extraction (RE) tasks and improving the Graphwise solution by taking the union.

Keywords

RAG, LM, Semantic retrieval, Structured Output

1. Introduction

Information Extraction from natural language can provide a way to structure the knowledge present in scientific texts for literature searches, search engines, or provide knowledge to LM for fact-based QA-tasks. These Knowledge Graphs (KGs) can be created by mining the relations from texts and creating an evidence-based KG from the facts and relations present in the texts. A highly reliable and yet broad source of text is required for the creation of such graphs [1, 2]. PubMed is one such provider, collecting papers in the medical domain that can be searched and downloaded [3]. The GutBrainIE Task within the BioASQ 2025 Laboratory provides abstracts on the topic of the gut-brain axis and related microbiome information to extract relevant links from and challenges the participant to create accurate KGs from this data.

Our entry to this task, CLEANR tackles this task by extracting the relations through a generative approach by a LM and combines Retrieval-Augmented Generation (RAG) and structured output to arrive at the correct outputs for these domain-specific situations. We employ a similar structure to the Retrieval-Augmented Generation-based Relation Extraction (RAG4RE) system [4], which utilizes RAG to provide the model with similar texts and relations to the query, thereby resulting in a few-shot system. Our contribution to this system is the use of structured output, which not only enforces a consistent JSON schema but also adheres to the possible relations provided by the tasks. This enables even non-fine-tuned models to perform well in this new domain, as demonstrated in our results. We test our approach on a variety of open and closed models, with and without finetuning, emphasizing the strengths of our novel RE strategy.

CLEF 2025 Working Notes, 9 – 12 September 2025, Madrid, Spain

*Corresponding author.

benedikt.kantz@tugraz.at (B. Kantz); s.lengauer@tugraz.at (S. Lengauer); pwaldert@tugraz.at (P. Waldert); tobias.schreck@tugraz.at (T. Schreck)

https://dakantz.at/ (B. Kantz); https://stefan-lengauer.at/ (S. Lengauer); https://github.com/MrP01 (P. Waldert)

0000-0003-3294-8421 (B. Kantz); 0000-0001-5136-4320 (S. Lengauer); 0009-0004-8459-7381 (P. Waldert); 0000-0003-0778-8665 (T. Schreck)

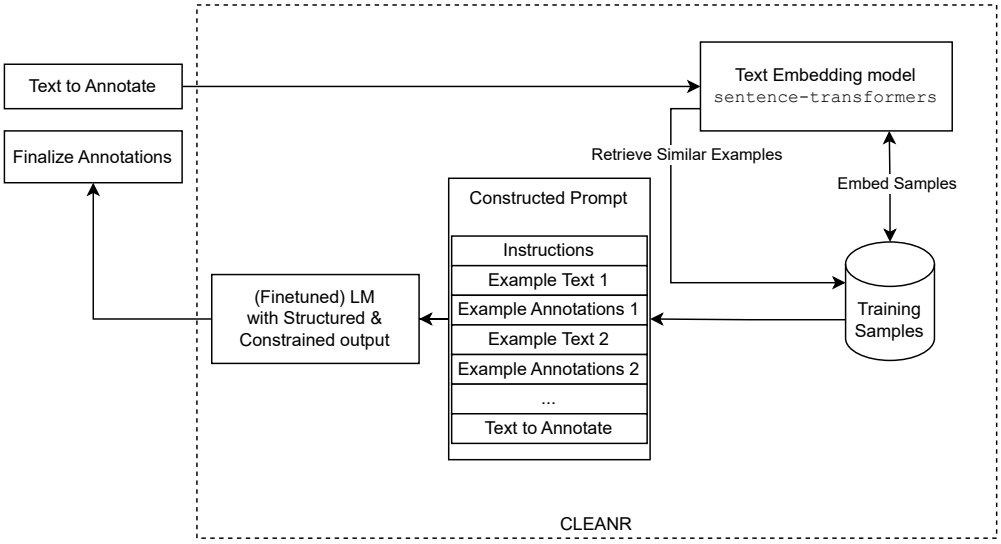

Figure 1: Overview of the CLEANR architecture.

2. Related Work

CLEANR is inspired by existing Relationship Extraction systems, such as RAG4RE [4], which utilizes RAG as its approach to incorporate detailed training data through semantic retrieval processes in the prompt for the LM. This few-shot approach, combined with dynamic retrieval, enables the system to be extended or "retrained" by simply adding or re-weighting the training samples, allowing test-time adaptation and generalization of the system with just a few new examples online without redeploying or retraining the model.

Prior systems, such as REBEL [5], train a supervised model to perform RE using special output tokens and fine-tune it for hours, as the REBEL 2021 model was trained for 9 hours. Our system aims to reduce the effort and time required for training.

3. Task Description

We investigate Subtasks 6.2.1, 6.2.2, and 6.2.3 within the GutBrainIE Task [6] of the BioASQ Laboratory [7] in this paper. These tasks focus on the RE from titles and abstracts within the PubMed database on the topic of gut-brain interplay. The subtasks we explore require three levels of expression detail – just the entities, entities and relation type, and, finally, the entities, relation, and location within the text. The task provides a labeled dataset, split into four tiers of annotated samples - platinum, gold, silver, and bronze. Human annotators annotate the first three tiers with a varying degree of expertise in the field. At the same time, the last one is automated using a "[...] distantly supervised [approach] [...] comprising automatically generated annotations." [6]

4. Methodology

CLEANR extends the approach to use RAG for RE using two key contributions. The first novelty of our methodology is the addition of constrained LM generation for RE. The second addition of our approach is the introduction of a re-weighting of the samples in the retrieval process to prefer samples with a higher degree of confidence (i.e., prefer the Gold annotations over the Bronze annotations in our setting). We use the `sentence-transformer` system [8] to embed the given training samples and store them in a Postgres database using the pgvector extension.

We, furthermore, utilize `llama-cpp`[1] and `llama-cpp-agent`[2] for both efficient inference of pre-trained models and constrained generation from a provided grammar. The grammar is generated using dynamically created Python types from the provided schema, as shown Section A.1. The necessary entities and links are taken from the provided schema from the GutBrainIE Task [6]. The schema can be constructed by taking the set of relations between head entities, tail entities, and predicates and converting these into allowed outputs for the LM, e.g. `Bacteria | Interact | Drug`. These entities and links could be exchanged for any other domain or setting, making our system very straightforward to adapt. The generated types are then automatically transformed into the GGML Backus-Naur Form (GBNF) syntax using the `llama-cpp-agent` package, which is then used to constrain the LM output to the exact schema provided by the task description. We extend the existing grammar features of `llama-cpp-agent` to include enumerable and literal support, to properly constrain the LM to only allow correct relations, including directions within the relations (i.e., the object and subject may not be switched). The contribution is already present as a pull request on GitHub for the original project[3]. We also repair any JSONs that may not be complete due to output sequence length limitations.

We also fine-tune a small 3B parameter model from Hermes 3-family of models [9] to the dataset and generative use case with few-shot prompts to illustrate the strength of our method compared to a finetune system. This is achieved using the `torchtune` framework to apply a Low-Rank Adaption (LoRA) [10] on the network.

Our RE utilizing the constrained and finetuned model is then used within the architecture illustrated in Figure 1, where we use a classic few-shot approach with RAG [11] to perform the RE[4]. This architecture utilizes the `sentence-transformer` to retrieve semantically similar samples from the database based on the text to be annotated. These are then used to build the prompt for the constrained LM, which are then parsed into the final annotation format required by the task.

4.1. Combination of Results

We also collaborated with the Graphwise team [12] to combine the strengths of our Test-Time method in the precision P with their strong method. We took the set union and intersection between the CLEANR results and theirs based on the Subject-Predicate-Object triplets predicted by our approaches. The results are presented in Section A.3.

4.2. Evaluation Methodology

CLEANR was initially evaluated using our implementation of the $F_{1,micro}$ metric, which yielded promising results when using the evaluation script that counted each duplicate entry. The results presented in this report paper, however, were all generated using the latest version of the final evaluation script of the task [6].

5. Experimental Setup

5.1. Training setup

We utilize the `torchtune` system to fine-tune the `Hermes-3-Llama-3.2-3B` model[5] on the provided training data, aiming to develop a multi-turn query-response system. The finetuned model is used to compare with our few-shot RAG system. Our training parameters can be found in Table 1.

We used a single RTX 8000 to fine-tune the model using LoRA, taking about 12 hours.

[1] https://github.com/ggml-org/llama.cpp
[2] https://github.com/Maximilian-Winter/llama-cpp-agent
[3] https://github.com/Maximilian-Winter/llama-cpp-agent/pull/89.
[4] The Named Entity Recognition (NER) results from Section A.2 are obtained using the same methodology
[5] https://huggingface.co/NousResearch/Hermes-3-Llama-3.2-3B-GGUF

Table 1
Parameters for the training setup.

Parameter Category	Name	Value
Generative Paramaters	Sequence Tokens	4096
LoRA parameters	Finetuned modules	Q, V and K
	Rank	32
	Alpha	64
	Dropout	0.0
Learning	Weight Decay	0.01
	Learning Rate	$3 \cdot 10^{-4}$
	Warmup steps	100
	Epochs	1
	Batch Size	2
	Gradient accumulation steps	8

Table 2
Models used in our approach.

Developers	Model	Open-Weight	Applied LoRA
OpenAI	GPT 4o-mini	×	×
Nous Research	Hermes 3 Llama 3.1 8B	✓	×
Nous Research	Hermes 3 Llama 3.2 3B	✓	✓

Table 3
Parameters for the annotation setup.

Parameter Category	Name	Value
RAG	top k	5
Generative	Context	8196
	Temperature	0.1
	Generated Tokens t	2048
	Quantization	Q8

Table 4
Reweigh coefficients.

Collection	Reweight
Platinum	1.0
Gold	0.9
Silver	0.8
Bronze	0.7

5.2. RE Process

Our approach is focused on test time retrievak and relies mainly on fixed-weight models – we therefore show them in Table 2. As CLEANR uses a RAG-approach, we show the generative parameters in Table 3.

For the reweighting for the RAG based on the classes, we first retrieve the top k matching documents (by cosine similarity) from the collections. The embeddings are generated using a `sentence-transformer` model [8][6], then reweigh them slightly by mutiplying the distances using the coefficients in Table 4 and reranking them again and taking the resulting top k results.

Our system uses a Postgres Database with version 17 with the pgvector extension as documented by our Docker Compose file for storage and efficient and fast retrieval for the RAG, with a RTX 4090 used

[6]Using the all-MiniLM-L6-v2 model

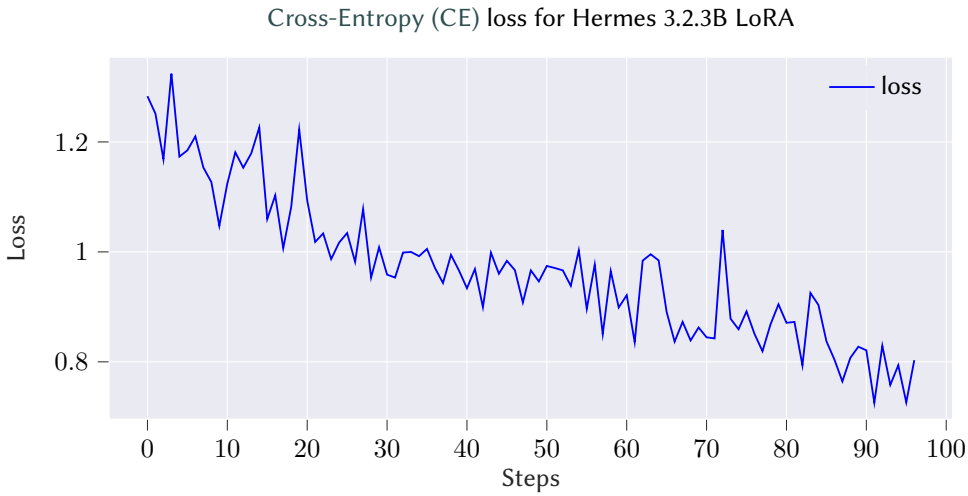

Figure 2: Loss over the learning steps.

for inference of the Open-Weight models.

5.3. Reproducibility

Our code is available on https://github.com/Dakantz/CLEANR and includes all necessary details to reproduce our results, such as dependency versions, training setups, and annotation system.

6. Results

We perform our evaluation on the dev-set provided within the GutbrainIE tasks using the latest evaluation script. The results are below the baseline posted by the task. Nevertheless, our system combines RAG and structured generation to retrieve data without the need for fine-tuning or adaptation to the model even with comparatively small LMs, and still achieves a comparatively good precision. We perform additional finetuning on the LM (the Cross-Entropy (CE) loss is plotted in Figure 2), where the $F_{1,micro}$ score increases only for the last task. The P_{micro}, however, does benefit significantly from the finetune.

The strength of our system is evident in its very competitive precision, which indicates that the system retrieves the correct results, reaching up to 0.8, outperforming the baseline and many other submitted systems for Subtasks 6.2.1 and 6.2.2. The system, however, retrieves too few results, resulting in a very weak recall R, which significantly drops our $F_{1,micro}$ result.

Our results show that the addition of retrieved data significantly improves the output, as almost all methods that utilize it experience a notable performance increase. We also observe a small impact of fine-tuning on the $F_{1,micro}$ score for the first two tasks, similar to our reordering approach. The best model using our methodology is the OpenAI 4o-mini model, primarily due to the high recall using our RAG approach. There appears to be some merit to our method, as it slightly improves the solution of Graphwise, most likely due to the higher precision shown in Section A.3.

6.1. Test set results

We additionally compare our results to the test set results to set them into context. The Tables 8 to 10 contain the test results for our CLEANR. These results align quite well with our dev set evaluation, with only a minor difference resulting in the best $F_{1,micro}$ by the Hermes 8B model, which applies both our RAG and Reorder approaches. The micro precision is not as high as on the dev set, but still higher than the best results in this category on the leaderboard. This indicates that our efficient method has merit

Table 5
Dev Set Result for Subtask 6.2.1 for various models and approaches.

Model	RAG	LoRA	Reorder	P	R	F_1	P_{micro}	R_{micro}	$F_{1,micro}$
Hermes 3B	×	×	×	0.03	0.02	0.02	0.15	0.02	0.03
			✓	0.04	0.01	0.02	0.14	0.02	0.03
		✓	×	0.17	0.06	0.09	**0.80**	0.16	0.27
			✓	0.17	0.06	0.09	**0.80**	0.16	0.27
	✓	×	×	0.20	0.09	0.11	0.53	0.19	0.28
			✓	0.11	0.06	0.08	0.42	0.15	0.23
		✓	×	0.13	0.05	0.07	0.64	0.13	0.22
			✓	0.15	0.06	0.08	0.68	0.15	0.25
Hermes 8B	×	×	×	0.04	0.01	0.02	0.31	0.05	0.09
			✓	0.04	0.01	0.02	0.31	0.05	0.09
	✓	×	×	**0.26**	0.12	**0.15**	0.55	0.25	0.34
			✓	0.12	0.08	0.09	0.47	0.23	0.31
Openai 4-1	×	×	×	0.08	0.11	0.07	0.22	0.16	0.19
			✓	0.04	0.05	0.04	0.19	0.14	0.16
	✓	×	×	0.20	0.16	0.15	0.37	0.26	0.31
			✓	0.20	0.16	0.15	0.31	0.25	0.28
Openai 4o-mini	×	×	×	0.07	0.09	0.07	0.19	0.16	0.17
			✓	0.09	0.11	0.08	0.21	0.19	0.20
	✓	×	×	0.15	**0.18**	0.15	0.45	**0.30**	**0.36**
			✓	0.11	0.10	0.10	0.34	0.24	0.28

in situations where high micro precision is important, particularly when only a few good relations are required. The worse scores on Subtask 6.2.3, however, indicate that our system is still unable to pinpoint the correct entities from which the relations originate properly.

Our combined results in Tables 11 to 13 tell a similar story to our observations on the test set, where the union performs very well, and the intersection has a very high micro precision.

7. Conclusions and Future Work

In this paper, we present CLEANR, a resource-efficient test-time system that combines existing systems to perform information extraction efficiently. Our system benefits from structured output and RAG approaches, demonstrating that fine-tuning may not be necessary when a strong enough model is available. The evaluated performance of CLEANR, however, indicates that we need to further improve the retrieval approach – especially the recall R_{micro}. The system, nevertheless, appears to have some merit, as its precision is high compared to other systems on the leaderboard.

We, however, identify a few possible improvements for our model, namely:

- Add more information to the system prompt, i.e., describe the task better and add the schema to the input such that it is not only constrained by the output, but can better decide on the results,
- use more domain-specific models (like a BERT model trained specifically on PubMed data) for the retrieval,
- constrain the returned data - either manually using a heuristic afterwards, or parse the response during generation and eliminate results that may not fit, e.g., by semantic search. A straightforward approach could be to limit or extend the generated output sequence length, as we repair any "broken" JSON anyway, or even extend the result by running the prompts multiple times or with a higher temperature,
- increase the model output to force the model to return more relations to improve the recall.
- CLEANR, additionally, does not implement any NER functionality as the LM does not build upon any prior entities. The NER task, however, could be solved using a very similar approach.

Table 6
Further Dev Set Result for Subtask 6.2.2 for various models and approaches.

Model	RAG	LoRA	Reorder	P	R	F_1	P_{micro}	R_{micro}	$F_{1,micro}$
Hermes 3B	×	×	×	0.03	0.02	0.02	0.16	0.02	0.03
			✓	0.02	0.01	0.01	0.12	0.01	0.02
		✓	×	0.15	0.05	0.07	**0.76**	0.14	0.24
			✓	0.15	0.05	0.07	**0.76**	0.14	0.24
	✓	×	×	0.17	0.08	0.10	0.49	0.17	0.25
			✓	0.08	0.05	0.06	0.40	0.13	0.20
		✓	×	0.13	0.04	0.06	0.63	0.12	0.20
			✓	0.11	0.05	0.07	0.65	0.13	0.22
Hermes 8B	×	×	×	0.05	0.01	0.02	0.24	0.03	0.06
			✓	0.05	0.01	0.02	0.24	0.03	0.06
	✓	×	×	**0.26**	0.12	0.15	0.53	0.23	0.32
			✓	0.10	0.08	0.08	0.44	0.20	0.27
Openai 4-1	×	×	×	0.08	0.10	0.07	0.20	0.14	0.16
			✓	0.04	0.05	0.04	0.15	0.11	0.13
	✓	×	×	0.20	0.16	0.15	0.36	0.25	0.30
			✓	0.16	0.14	0.13	0.29	0.22	0.25
Openai 4o-mini	×	×	×	0.06	0.07	0.05	0.16	0.13	0.14
			✓	0.09	0.11	0.08	0.18	0.17	0.17
	✓	×	×	0.16	**0.18**	**0.15**	0.43	**0.29**	**0.35**
			✓	0.12	0.10	0.10	0.33	0.22	0.26

These improvements can be implemented through minor adjustments to the system, which could slightly enhance performance. We explore some of these suggestions in Section A.2, discussing them and possible reasons why they might fail or have some merit. A significant improvement could come from improved model performance, i.e., through a reasoning step allowing the model to "contemplate" the relations or using more recent agentic approaches. However, little improvement can be made in Subtask 6.2.3, as the task requires the model to accurately pinpoint the text segment from which the result was obtained. A possible remedy for this issue could be further improving the structured output by only allowing valid pairs from the text, which might even be preselected using a different NER model.

Acknowledgments

This work is partially supported by the HEREDITARY Project, as part of the European Union's Horizon Europe research and innovation programme under grant agreement No GA 101137074.

Declaration on Generative AI

During the preparation of this work, the authors used Grammarly in order to: Grammar and spelling check. After using these tool, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

References

[1] S. Ji, S. Pan, E. Cambria, P. Marttinen, P. S. Yu, A survey on knowledge graphs: Representation, acquisition, and applications, IEEE Transactions on Neural Networks and Learning Systems 33 (2022) 494–514. doi:10.1109/TNNLS.2021.3070843.

Table 7
Dev Set Result for Subtask 6.2.3 for various models and approaches.

Model	RAG	LoRA	Reorder	P	R	F_1	P_{micro}	R_{micro}	$F_{1,micro}$
Hermes 3B	×	×	×	0.00	0.00	0.00	0.00	0.00	0.00
			✓	0.00	0.00	0.00	0.00	0.00	0.00
		✓	×	0.03	0.03	0.03	**0.19**	**0.08**	**0.11**
			✓	0.03	0.03	0.03	**0.19**	**0.08**	**0.11**
	✓	×	×	**0.06**	0.02	0.03	0.06	0.04	0.05
			✓	0.00	0.01	0.00	0.03	0.02	0.03
		✓	×	0.03	0.02	0.02	0.14	0.05	0.07
			✓	0.04	0.02	0.02	0.18	0.06	0.09
Hermes 8B	×	×	×	0.00	0.00	0.00	0.00	0.00	0.00
			✓	0.00	0.00	0.00	0.00	0.00	0.00
	✓	×	×	0.05	**0.03**	**0.03**	0.10	0.07	0.08
			✓	0.01	0.01	0.00	0.04	0.03	0.03
Openai 4-1	×	×	×	0.00	0.00	0.00	0.00	0.00	0.00
			✓	0.00	0.01	0.00	0.01	0.00	0.01
	✓	×	×	0.00	0.00	0.00	0.01	0.01	0.01
			✓	0.00	0.00	0.00	0.00	0.00	0.00
Openai 4o-mini	×	×	×	0.00	0.00	0.00	0.00	0.00	0.00
			✓	0.00	0.00	0.00	0.00	0.00	0.00
	✓	×	×	0.00	0.00	0.00	0.02	0.01	0.01
			✓	0.00	0.00	0.00	0.01	0.01	0.01

Table 8
Test Set Result for Subtask 6.2.1 for various models and approaches.

Model	RAG	LoRA	Reorder	P	R	F_1	P_{micro}	R_{micro}	$F_{1,micro}$
Hermes 3B	×	×	×	0.04	0.01	0.01	0.12	0.02	0.03
			✓	0.04	0.01	0.01	0.12	0.02	0.03
		✓	×	0.14	0.05	0.07	**0.74**	0.12	0.21
			✓	0.14	0.05	0.07	**0.74**	0.12	0.21
	✓	×	✓	0.18	0.09	0.11	0.57	0.17	0.26
		✓	✓	0.16	0.09	0.11	0.73	0.16	0.26
Hermes 8B	×	×	×	0.10	0.04	0.05	0.50	0.09	0.15
			✓	0.10	0.04	0.05	0.50	0.09	0.15
	✓	×	✓	**0.22**	0.13	0.15	0.57	0.26	**0.36**
Openai 4	×	×	×	0.10	0.10	0.08	0.23	0.19	0.20
			✓	0.09	0.08	0.07	0.24	0.17	0.20
	✓	×	×	0.17	0.14	0.15	0.37	0.28	0.32
			✓	0.20	**0.14**	**0.15**	0.39	**0.30**	0.34

[2] F. Gao, Y. Yang, P. Gao, M. Gu, S. Zhao, Y. Chen, H. Yuan, M. Lan, A. Zhou, L. He, Self-supervised bgp-graph reasoning enhanced complex kbqa via sparql generation, Information Processing & Management 61 (2024) 103802. URL: https://www.sciencedirect.com/science/article/pii/S0306457324001614. doi:https://doi.org/10.1016/j.ipm.2024.103802.

[3] N. Milošević, W. Thielemann, Comparison of biomedical relationship extraction methods and models for knowledge graph creation, Journal of Web Semantics 75 (2023) 100756.

[4] S. Efeoglu, A. Paschke, Retrieval-augmented generation-based relation extraction, 2024. URL: https://arxiv.org/abs/2404.13397. arXiv:2404.13397.

[5] P.-L. Huguet Cabot, R. Navigli, REBEL: Relation extraction by end-to-end language generation, in: Findings of the Association for Computational Linguistics: EMNLP 2021, Association for

Table 9
Test Set Result for Subtask 6.2.2 for various models and approaches.

Model	RAG	LoRA	Reorder	P	R	F_1	P_{micro}	R_{micro}	$F_{1,micro}$
Hermes 3B	×	×	×	0.03	0.01	0.01	0.12	0.02	0.03
			✓	0.03	0.01	0.01	0.12	0.02	0.03
		✓	×	0.13	0.04	0.06	0.68	0.09	0.17
			✓	0.13	0.04	0.06	0.68	0.09	0.17
	✓	×	✓	0.17	0.08	0.10	0.55	0.16	0.24
		✓	✓	0.15	0.08	0.10	**0.71**	0.15	0.24
Hermes 8B	×	×	×	0.09	0.04	0.05	0.36	0.06	0.10
			✓	0.09	0.04	0.05	0.36	0.06	0.10
	✓	×	✓	**0.23**	0.13	0.14	0.56	0.25	**0.34**
Openai 4	×	×	×	0.10	0.09	0.08	0.20	0.16	0.18
			✓	0.09	0.07	0.06	0.21	0.15	0.17
	✓	×	×	0.19	0.14	0.14	0.37	0.27	0.31
			✓	0.21	**0.15**	**0.15**	0.39	**0.28**	0.33

Table 10
Test Set Result for Subtask 6.2.3 for various models and approaches.

Model	RAG	LoRA	Reorder	P	R	F_1	P_{micro}	R_{micro}	$F_{1,micro}$
Hermes 3B	×	×	×	0.00	0.00	0.00	0.00	0.00	0.00
			✓	0.00	0.00	0.00	0.00	0.00	0.00
		✓	×	0.03	0.02	0.02	**0.18**	0.04	0.07
			✓	0.03	0.02	0.02	**0.18**	0.04	0.07
	✓	×	✓	**0.04**	**0.03**	0.02	0.04	0.02	0.03
		✓	✓	0.02	0.02	0.02	0.17	**0.05**	**0.08**
Hermes 8B	×	×	×	0.00	0.01	0.01	0.02	0.01	0.01
			✓	0.00	0.01	0.01	0.02	0.01	0.01
	✓	×	✓	0.03	0.03	**0.02**	0.06	0.04	0.05
Openai 4	×	×	×	0.00	0.00	0.00	0.00	0.00	0.00
			✓	0.00	0.00	0.00	0.00	0.00	0.00
	✓	×	×	0.00	0.00	0.00	0.01	0.00	0.01
			✓	0.00	0.00	0.00	0.00	0.00	0.00

Table 11
Test Set Result for Subtask 6.2.1 for the Graphwise collaboration.

Set	Model	RAG	LoRA	Reorder	P	R	F_1	P_{micro}	R_{micro}	$F_{1,micro}$
∩	Hermes 3B + Graphwise	×	×	×	0.17	0.06	0.08	**0.89**	0.13	0.23
∪	Hermes 3B + Graphwise	×	×	×	**0.42**	**0.41**	**0.41**	0.71	**0.61**	**0.66**

Computational Linguistics, Punta Cana, Dominican Republic, 2021, pp. 2370–2381. URL: https://aclanthology.org/2021.findings-emnlp.204.

[6] M. Martinelli, G. Silvello, V. Bonato, G. M. Di Nunzio, N. Ferro, O. Irrera, S. Marchesin, L. Menotti, F. Vezzani, Overview of GutBrainIE@CLEF 2025: Gut-Brain Interplay Information Extraction, in: G. Faggioli, N. Ferro, P. Rosso, D. Spina (Eds.), CLEF 2025 Working Notes, 2025.

[7] A. Nentidis, G. Katsimpras, A. Krithara, M. Krallinger, M. Rodríguez-Ortega, E. Rodriguez-López, N. Loukachevitch, A. Sakhovskiy, E. Tutubalina, D. Dimitriadis, G. Tsoumakas, G. Giannakoulas, A. Bekiaridou, A. Samaras, G. M. Di Nunzio, N. Ferro, S. Marchesin, M. Martinelli, G. Silvello,

Table 12
Test Set Result for Subtask 6.2.2 for the Graphwise collaboration.

Set	Model	RAG	LoRA	Reorder	P	R	F_1	P_{micro}	R_{micro}	$F_{1,micro}$
∪	Hermes 3B + Graphwise	×	×	×	**0.43**	**0.40**	**0.41**	**0.71**	**0.59**	**0.64**

Table 13
Test Set Result for Subtask 6.2.3 for the Graphwise collaboration.

Set	Model	RAG	LoRA	Reorder	P	R	F_1	P_{micro}	R_{micro}	$F_{1,micro}$
∪	Hermes 3B + Graphwise	×	×	×	**0.26**	**0.23**	**0.23**	**0.35**	**0.32**	**0.34**

G. Paliouras, Overview of BioASQ 2025: The thirteenth BioASQ challenge on large-scale biomedical semantic indexing and question answering, Lecture Notes in Computer Science, Springer, 2025.
[8] N. Reimers, I. Gurevych, Sentence-bert: Sentence embeddings using siamese bert-networks, in: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, 2019. URL: https://arxiv.org/abs/1908.10084.
[9] R. Teknium, J. Quesnelle, C. Guang, Hermes 3 technical report, 2024. URL: https://arxiv.org/abs/2408.11857. arXiv:2408.11857.
[10] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen, Lora: Low-rank adaptation of large language models, 2021. URL: https://arxiv.org/abs/2106.09685. arXiv:2106.09685.
[11] W. Fan, Y. Ding, L. Ning, S. Wang, H. Li, D. Yin, T.-S. Chua, Q. Li, A survey on rag meeting llms: Towards retrieval-augmented large language models, 2024. URL: https://arxiv.org/abs/2405.06211. arXiv:2405.06211.
[12] A. Datseris, M. Kuzmanov, I. Nikolova-Koleva, D. Taskov, S. Boytcheva, Graphwise @ clef-2025 gutbrainie: Towards automated discovery of gut-brain interactions: Deep learning for ner and relation extraction from pubmed abstracts, in: G. Faggioli, N. Ferro, P. Rosso, D. Spina (Eds.), Working Notes of CLEF 2025 – Conference and Labs of the Evaluation Forum, CEUR Workshop Proceedings, CEUR-WS, 2025.

A. Appendix

A.1. Model Constraints

Our CLEANR system relies on, at its core, dynamically generated types from the GutBrainIE schema. This enables our system to perform two tasks at the same time:

- validate the input data to check whether it fits the schema,
- constraint the LM to the correct relations.

We therefore provide the code in Listing 1 to build our schema here. The function requires the relations as a list of allowed combinations, enumerates all possibilities and combines it in a single `Enum` type that is set as field in the dynamic Pydantic[7] type.

Listing 1: Dynamic types generated from the relations.

```
def build_model(relations=relations):
    possible_links = {}
    for relation in relations:
        heads = [clean_label(head) for head in relation["heads"]]
```

[7] https://docs.pydantic.dev/latest/

```
        tails = [clean_label(tail) for tail in relation["tails"]]
        predicates = [clean_label(pred) for pred in relation["predicate"]]

        for head in heads:
            enum_head = enumize_label(head)
            for tail in tails:
                enum_tail = enumize_label(tail)
                for pred in predicates:
                    enum_pred = enumize_label(pred)
                    possible_links["_".join([enum_head, enum_pred, enum_tail])] = (
                        "␣|␣".join([head, pred, tail])
                    )
    link_type = Enum("LinkType", possible_links)
    relation_type = create_model(
        "Relation",
        link_type=(link_type, ...),
        subject_text_span=(str, ...),
        subject_location=(LabelLocation, ...),
        object_text_span=(str, ...),
        object_location=(LabelLocation, ...),
    )
    relation_union = create_model("Relations", relations=(list[relation_type], ...))
    return relation_union
```

A.2. Further Experiments

We also conduct additional experiments with our approach using the small Hermes 3B model to investigate some of the possible improvements we suggest in Section 7 to address the weaknesses in our approach. We present them in Tables 14 to 16. These results indicate that our variations do not improve the scores, suggesting that we have either reached the limits of our small models or require some further research and adjustments to our methodology. The additional, longer training for the model (indicated by LoRA+) did help the model achieve performance similar to that of the OpenAI models, beating it by only a margin. This fin-etune of 3 epochs, however, took significantly longer than using the base model directly, using our constrained output, and imposed a significant reduction in precision. The output loss is shown Figure 3. We also employ a new embedding model, the `NeuML/pubmedbert-base-embeddings`[8] for the RAG embeddings, showing only minor improvements compared to our initial results. We also experimented with variations in output token lengths, including fewer allowed tokens, which resulted in slightly lower overall performance. Adding the possible entities and descriptions to the prompts also slightly reduced performance.

These experiments suggest that our approach, in combination with our small models, can not beat the specifically trained baseline. We did not attempt larger models, which could still offer improved performance, as the RAG4RE approach has been shown to do [4].

We additionally explore the NER task in a limited setting in Table 17. These experiments yield similarly poor performance, most likely due to the approach's inability to accurately pinpoint the correct locations of the entities in the input texts, and thus failing to extract the proper indices required for validation. We address this shortcoming by extracting the indices from the text based on the predicted text spans, with little apparent performance impact.

A.2.1. Experimenting with the output lengths

Further experiments include a study of the scores for capped outputs and ground truths, effectively calculating the micro averages for different k's in Figure 4. These evaluations suggest that our method may initially return the best-effort results and does not generate too many relations at once, indicating

[8] https://huggingface.co/NeuML/pubmedbert-base-embeddings

Table 14
Dev Set Result for Subtask 6.2.1 for further experiments on various models and approaches. LoRA+ denotes longer finetuning, Low t less generated tokens, Entities added information to the system prompt regarding the possible entities.

Model	RAG	LoRA+	Reorder	Low t	Entities	P	R	F_1	P_{micro}	R_{micro}	$F_{1,micro}$
Hermes 3B	×	×	×	×	✓	0.06	0.09	0.06	0.15	0.08	0.11
				✓	×	0.04	0.01	0.02	0.10	0.01	0.02
					✓	0.07	0.07	0.05	0.14	0.06	0.09
			✓	×	✓	0.06	0.09	0.06	0.15	0.08	0.11
				✓	✓	0.07	0.07	0.05	0.14	0.06	0.09
		✓	×	×	×	0.21	**0.12**	**0.14**	0.60	**0.25**	0.35
				✓	×	0.22	0.12	0.14	0.60	0.24	0.34
					✓	0.17	0.10	0.11	0.58	0.19	0.29
			✓	✓	✓	0.17	0.10	0.11	0.58	0.19	0.29
	✓	×	×	✓	×	**0.24**	0.09	0.12	0.65	0.16	0.26
					✓	0.16	0.07	0.09	0.47	0.15	0.23
			✓	✓	✓	0.16	0.07	0.09	0.47	0.15	0.23
		✓	×	×	×	0.20	0.12	0.14	**0.66**	**0.25**	**0.36**
				✓	×	0.18	0.10	0.12	0.65	0.21	0.32
					✓	0.17	0.07	0.09	0.55	0.15	0.24
			✓	✓	✓	0.17	0.07	0.09	0.55	0.15	0.24

Table 15
Further Dev Set Result for Subtask 6.2.2 for further experiments on various models and approaches. LoRA+ denotes longer finetuning, Low t less generated tokens, Entities added information to the system prompt regarding the possible entities.

Model	RAG	LoRA+	Reorder	Low t	Entities	P	R	F_1	P_{micro}	R_{micro}	$F_{1,micro}$
Hermes 3B	×	×	×	×	✓	0.07	0.08	0.06	0.14	0.07	0.09
				✓	×	0.04	0.01	0.02	0.11	0.01	0.02
					✓	0.07	0.07	0.05	0.12	0.05	0.07
			✓	×	✓	0.07	0.08	0.06	0.14	0.07	0.09
				✓	✓	0.07	0.07	0.05	0.12	0.05	0.07
		✓	×	×	×	0.21	**0.12**	0.14	0.57	0.23	0.32
				✓	×	**0.22**	0.11	0.14	0.57	0.22	0.31
					✓	0.17	0.09	0.11	0.54	0.17	0.26
			✓	✓	✓	0.17	0.09	0.11	0.54	0.17	0.26
	✓	×	×	✓	×	0.21	0.08	0.11	0.64	0.15	0.25
					✓	0.15	0.07	0.08	0.47	0.15	0.22
			✓	✓	✓	0.15	0.07	0.08	0.47	0.15	0.22
		✓	×	×	×	0.20	0.11	**0.14**	**0.65**	**0.23**	**0.35**
				✓	×	0.19	0.10	0.12	0.63	0.20	0.30
					✓	0.17	0.07	0.09	0.55	0.15	0.23
			✓	✓	✓	0.17	0.07	0.09	0.55	0.15	0.23

that the model's performance is at fault here, or that the model should output more results, also supported by the improved performance of our extended fine-tuning.

A.3. Graphwise collaboration

We also collaborated with the Graphwise team to combine our results, taking both the intersection and the union between our results. The results of this collaboration can be found in Tables 18 to 20, matching our test results quite well. These results indicate that the LoRA fine-tune models perform best

Table 16
Dev Set Result for Subtask 6.2.3 for further experiments on various models and approaches. LoRA+ denotes longer finetuning, Low t less generated tokens, Entities added information to the system prompt regarding the possible entities.

Model	RAG	LoRA+	Reorder	Low t	Entities	P	R	F_1	P_{micro}	R_{micro}	$F_{1,micro}$
Hermes 3B	×	×	×	×	✓	0.00	0.00	0.00	0.00	0.00	0.00
				✓	×	0.00	0.00	0.00	0.00	0.00	0.00
					✓	0.00	0.00	0.00	0.00	0.00	0.00
			✓	×	✓	0.00	0.00	0.00	0.00	0.00	0.00
				✓	✓	0.00	0.00	0.00	0.00	0.00	0.00
		✓	×	×	×	0.06	0.03	0.04	0.17	**0.08**	0.11
				✓	×	0.08	0.03	0.04	0.19	0.07	**0.11**
					✓	0.06	0.03	0.04	0.16	0.05	0.08
			✓	✓	✓	0.06	0.03	0.04	0.16	0.05	0.08
	✓	×	×	✓	×	0.03	0.01	0.01	0.06	0.02	0.03
					✓	0.03	0.01	0.02	0.08	0.03	0.05
			✓	✓	✓	0.03	0.01	0.02	0.08	0.03	0.05
		✓	×	×	×	0.08	**0.03**	**0.04**	0.17	0.08	0.11
				✓	×	0.06	0.03	0.03	0.17	0.06	0.09
					✓	**0.08**	0.03	0.03	**0.19**	0.07	0.10
			✓	✓	✓	**0.08**	0.03	0.03	**0.19**	0.07	0.10

Table 17
Dev Set Result for Task 6.1.1 (NER) for various models and approaches.

Model	RAG	LoRA+	Reorder	Low t	Entities	P	R	F_1	P_{micro}	R_{micro}	$F_{1,micro}$
Hermes 3B	×	×	×	×	✓	0.05	0.02	0.03	0.06	0.02	0.03
				✓	×	0.02	0.02	0.01	0.04	0.01	0.02
					✓	0.05	0.02	0.02	0.06	0.02	0.03
			✓	×	✓	0.05	0.02	0.03	0.06	0.02	0.03
				✓	✓	0.05	0.02	0.02	0.06	0.02	0.03
	✓	×	×	×	✓	0.21	**0.08**	**0.11**	0.25	**0.12**	**0.16**
				✓	×	**0.32**	0.07	0.10	0.29	0.10	0.15
					✓	0.25	0.05	0.08	**0.31**	0.08	0.13
			✓	×	✓	0.21	**0.08**	**0.11**	0.25	**0.12**	**0.16**
				✓	✓	0.25	0.05	0.08	**0.31**	0.08	0.13

in this combined setting. The Union performs significantly better, suggesting that our model indeed produces a few very good results. This is even more evident when the precision for the intersection is investigated, reaching a score of 0.96 for Subtasks 6.2.1 and 6.2.2, which is significantly higher than any other model on the leaderboards.

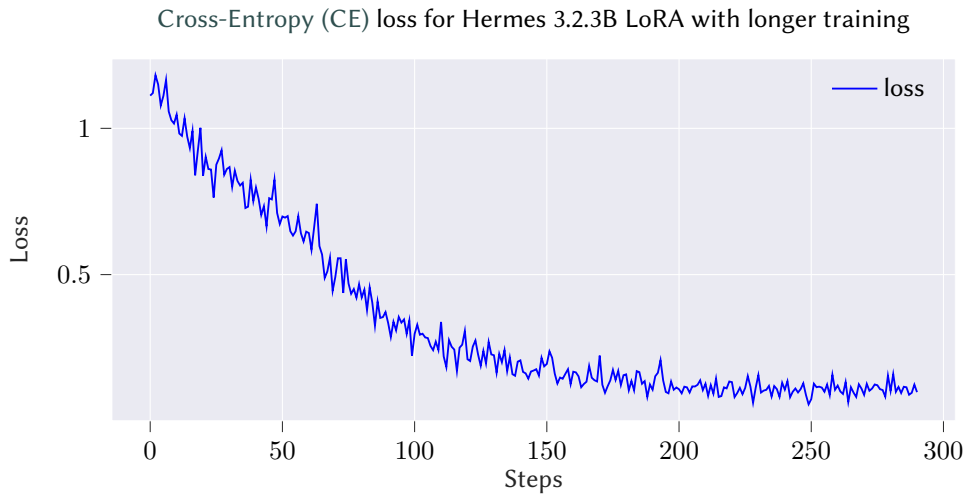

Figure 3: Loss over a longer learning period (3 epochs).

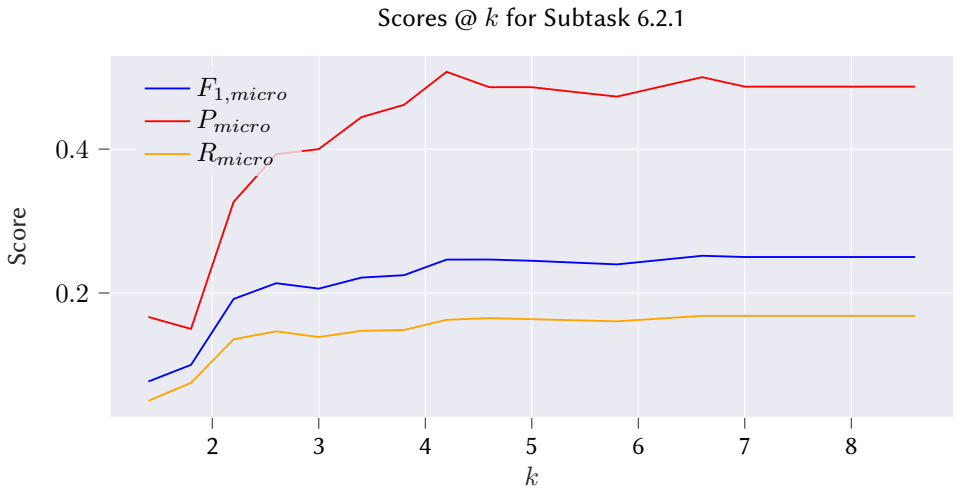

Figure 4: Scores over different capped outputs and ground truth lengths k of retrieved relations.

Table 18
Dev Set Result for Subtask 6.2.1 for various models and approaches from the intersection (∩) and union (∪) between the Graphwise submission.

Set	Model	RAG	LoRA	Reorder	P	R	F_1	P_{micro}	R_{micro}	$F_{1,micro}$
∩	Hermes 3B	×	×	×	0.03	0.01	0.02	0.67	0.01	0.02
				✓	0.01	0.00	0.01	0.50	0.00	0.01
			✓	×	0.18	0.04	0.06	**0.96**	0.11	0.20
				✓	0.18	0.04	0.06	**0.96**	0.11	0.20
		✓	×	×	0.25	0.07	0.10	0.85	0.15	0.25
				✓	0.13	0.05	0.07	0.69	0.12	0.21
			✓	×	0.17	0.04	0.06	0.89	0.11	0.19
				✓	0.15	0.04	0.06	0.86	0.11	0.19
∪	Hermes 3B	×	×	×	0.55	0.53	0.51	0.65	0.62	0.64
				✓	0.54	0.53	0.51	0.65	0.62	0.64
			✓	×	**0.59**	**0.54**	**0.54**	0.71	**0.65**	**0.67**
				✓	**0.59**	**0.54**	**0.54**	0.71	**0.65**	**0.67**
		✓	×	×	0.53	0.53	0.51	0.63	0.64	0.63
				✓	0.56	0.53	0.52	0.62	0.63	0.63
			✓	×	0.59	0.53	0.53	0.68	0.63	0.65
				✓	0.59	0.54	0.54	0.69	**0.65**	0.67

Table 19
Dev Set Result for Subtask 6.2.2 from the intersection (∩) and union (∪) between the Graphwise submission.

Set	Model	RAG	LoRA	Reorder	P	R	F_1	P_{micro}	R_{micro}	$F_{1,micro}$
∩	Hermes 3B	×	×	×	0.03	0.01	0.02	0.67	0.01	0.02
				✓	0.01	0.00	0.01	0.50	0.00	0.01
			✓	×	0.17	0.04	0.06	**0.96**	0.11	0.20
				✓	0.17	0.04	0.06	**0.96**	0.11	0.20
		✓	×	×	0.24	0.07	0.10	0.85	0.14	0.25
				✓	0.13	0.05	0.06	0.69	0.12	0.20
			✓	×	0.16	0.04	0.06	0.89	0.10	0.19
				✓	0.14	0.04	0.06	0.86	0.10	0.19
∪	Hermes 3B	×	×	×	0.55	0.53	0.51	0.66	0.60	0.63
				✓	0.54	0.53	0.51	0.65	0.60	0.62
			✓	×	**0.59**	**0.53**	**0.54**	0.70	**0.62**	**0.66**
				✓	**0.59**	**0.53**	**0.54**	0.70	**0.62**	**0.66**
		✓	×	×	0.52	0.53	0.50	0.62	0.61	0.62
				✓	0.55	0.53	0.51	0.62	0.61	0.61
			✓	×	0.58	0.52	0.53	0.68	0.60	0.64
				✓	0.58	0.53	0.54	0.69	**0.62**	0.65

Table 20
Dev Set Result for Subtask 6.2.3 from the intersection (∩) and union (∪) between the Graphwise submission.

Set	Model	RAG	LoRA	Reorder	P	R	F_1	P_{micro}	R_{micro}	$F_{1,micro}$
∩	Hermes 3B	×	×	×	0.00	0.00	0.00	0.02	0.00	0.00
				✓	0.00	0.00	0.00	0.00	0.00	0.00
			✓	×	0.06	0.04	0.04	0.34	0.13	0.18
				✓	0.06	0.04	0.04	0.34	0.13	0.18
		✓	×	×	0.11	0.05	0.06	0.27	0.13	0.18
				✓	0.02	0.03	0.02	0.18	0.10	0.13
			✓	×	0.05	0.03	0.03	0.31	0.10	0.15
				✓	0.06	0.03	0.03	0.32	0.11	0.16
∪	Hermes 3B	×	×	×	0.28	0.29	0.25	0.31	0.36	0.34
				✓	0.27	0.29	0.25	0.28	0.36	0.31
			✓	×	0.33	**0.30**	0.29	0.36	**0.39**	0.38
				✓	0.33	**0.30**	0.29	0.36	**0.39**	0.38
		✓	×	×	0.28	0.29	0.27	0.28	0.37	0.32
				✓	0.30	0.29	0.27	0.27	0.37	0.31
			✓	×	0.33	0.29	0.29	0.36	0.37	0.36
				✓	**0.34**	0.30	**0.30**	**0.37**	0.38	**0.38**