Sem-G-RAG: Combining graph-based semantic symbolism and LLM for RAG

Anonymous ACL submission

Abstract

We propose a new methodology to add an explicit semantic representation (based on Framenet and the Penman graph format) to the instruction given to an LLM during answer generation. We have analysed its benefits on the SQuad dataset. If adding these information did not significantly improve the performance in our experiments, our qualitative analysis shows that in some cases the factual precision of the answer can be better. We also observed that having only the semantics graph instead of the textual context induces a clear degradation of performance. On the other hand, we did not find a significative difference between having a complete Framenet graph and a filtered version according to the entities present in the associated text query, opening the way for optimizating our methodology by semantic filtering of the graph.

1 Introduction

To broaden the capabilities of Retrieval-Augmented Generation (RAG) systems, knowledge graphs and semantic representations have been extensively exploited in recent work. The approaches proposed in (Han et al., 2025) and (Guo et al., 2025b) demonstrate the effectiveness of knowledge graphs for structuring information and improving retrieval relevance. With the rapid advances in Information Retrieval (IR) and Natural Language Processing (NLP), several techniques make possible a significant acceleration and improvement of the search process. For example, encoders based on the Transformer architecture improve result relevance by capturing the semantic and contextual dimension of queries and documents (Vaswani et al., 2023), while vector indexing methods such as FAISS enable large-scale similarity search in an efficient manner. However, the majority of work focuses primarily on improving the retrieval module, while the generation component remains comparatively less explored.

In particular, few studies explicitly address the integration of the semantic dimension within the generation process itself. Consequently, this work aims to explore the explicit integration of the semantic dimension into the generation module of RAG architectures, in order to assess its impact on the coherence and relevance of the produced responses. Concretely, we propose to augment the textual content of the context retrieved during the information retrieval phase with its corresponding semantic graph before proceeding to the generation step. We will successively present the state of the art, the proposed methodology, the experiments carried out, and the analysis of results.

2 Related Work

Contextual semantics has demonstrated its effectiveness in numerous works on RAG. In Sem-RAG, the authors propose to segment texts according to the cosine similarity between sentence vectors, which allows more relevant information to be grouped within the same segment (Zhong et al., 2025). On the other hand, Multi-Semantic RAG (MS-RAG) (You et al., 2025) and SPR-RAG (Wang et al., 2025) exploit cosine similarity by comparing entities and relations from the question with those extracted from knowledge graphs. This approach contributes to accelerating search and improves the relevance of retrieved content.

Despite these advances, these approaches predominantly rely on vector similarity, which mainly captures semantic proximity without explicitly modelling the conceptual relationships between pieces of information. This limitation also affects the performance of LLMs. Indeed, although these large models are capable of effectively understanding implicit semantics through their pre-trained parameters, they do not have access to the meaning of the input text in terms of explicit semantic representation (Cuskley et al., 2024). Furthermore,

the IR step succeeds in providing relevant context, but the possibility of the LLM misinterpreting that context remains.

Consequently, we wish to integrate the explicit semantic dimension into RAG, specifically at the generation step. To this end, we rely on the theory of Frame Semantics. This theory suggests that the best way to understand the meaning of words is to refer to the semantic and conceptual frames or schemas in which they are anchored (Fillmore et al., 2006). Building on this theory, FrameNet[1] is a semantic representation model that describes the meaning of words through structures called frames. A frame corresponds to a type of situation, such as an event, a relation, or a state, involving several participants playing different roles. These participants are called Frame Elements (FE). The role of FrameNet is to identify and define these frames and their elements, and then to describe how these elements are organised syntactically in a sentence around the lexical unit that evokes the frame.

A recent work has also explored the effectiveness of FrameNet in RAG systems (Madabushi, 2024). This approach is motivated by the observation that contextual vector representations primarily capture semantic similarity between texts, without necessarily modelling all logical relationships between pieces of information. As a result, retrieved content may be semantically close to the query without being sufficient to support multi-fact reasoning. In contrast to our approach, this work focuses on the information retrieval component. The authors propose annotating atomic facts with the main semantic frames (Frames) they express. During the retrieval phase, the system first identifies the main frame of the question as well as the frames associated with it. This step makes it possible to determine the frames corresponding to the facts likely to answer the question. Information retrieval is then guided by these frames, with the aim of improving the relevance and efficiency of information retrieval. In that article, the experimental results show that this approach outperforms keyword-based retrieval methods, while remaining significantly less effective than approaches relying solely on lexical embeddings. Nevertheless, this work highlights the value of the explicit semantics provided by FrameNet. Given that LLMs are trained on natural language text, it may be more relevant to integrate this explicit semantic representation into the generation phase.

Moreover, another study focused on the generation phase (Jin et al., 2024). Rather than relying on FrameNet, this approach exploits another form of semantics, AMR (Abstract Meaning Representation). In order to assess the impact of this representation on LLM performance, the authors simultaneously provide the model with the input text and its associated AMR representation. However, the experiments did not yield promising results. The integration of AMR graphs does not globally improve LLM performance and even causes a slight degradation on several tasks, including paraphrase detection, machine translation, event extraction, and logical fallacy detection. These results suggest that AMR does not offer a clear advantage to modern LLMs. Yet this limitation may be explained by certain characteristics of AMR. Indeed, this representation focuses primarily on the predicate and essentially models the relations between it and its arguments. Moreover, these relations are generally designated by abstract labels (ARG0, ARG1, ARG2, etc.), which indicate the existence of a structural link without making its precise semantic nature explicit (Banarescu et al., 2013).

These limitations could explain why AMR does not provide a significant benefit to LLMs. In contrast, FrameNet proposes semantic concepts covering not only the predicate, but also other textual units such as the subject, the object, or prepositions, thus capturing richer semantic information. Furthermore, the concepts and relations are expressed using a concrete vocabulary enriched with natural language examples, which facilitates their exploitation by LLMs. In this work, we adopt a similar approach consisting of combining the text with its corresponding semantic graph, substituting AMR with FrameNet.

In order to represent FrameNet information, this work will use the Penman format. This format represents graphs as rooted, directed, and labelled structures, and is designed to model complex semantic structures in a human-readable textual representation (Goodman, 2020). The Penman format encodes the hierarchy and tree structure of concepts using nested parentheses, thus making it possible to represent the dependency structure of a semantic graph. Admittedly, this format is primarily known for encoding AMR graphs or graphs for other NLU tasks, as in the works of (Kang et al., 2025; Bevilac-

[1] https://framenet.icsi.berkeley.edu/WhatIsFrameNet

qua et al., 2021; Bonial et al., 2020; Abrougui et al., 2023). Nevertheless, the penman library (a Python tool)[2] is not strictly limited to AMR, and is also compatible with FrameNet annotations, provided that these respect the structure and grammar of the graph. This appears relevant for FrameNet, whose organisation into frames presents structural similarities with AMR. In the context of FrameNet annotations expressed in Penman, nodes correspond to concepts or entities, while edges represent the semantic relations between these elements.

The Penman format constitutes an intermediate representation between a tree structure and a full semantic graph. It combines the readability of a linear textual notation with the properties of a directed graph, making it well suited to visualisation and automatic NLP processing.

3 Methodology

To study the contributions of FrameNet graphs in the RAG generation phase, we propose concatenating the question, the context, and its corresponding FrameNet graph into the instruction sent to the LLM. In what follows, we explain how the FrameNet semantic information is obtained for the retrieved context and how it is integrated into our RAG architecture.

Once the information retrieval system obtains the relevant textual context for the received question, an automatic parser will extract the semantic frames. This annotation is performed by the Frame Semantic Transformer (FST), which is based on a fine-tuned version of the T5 (Text-to-Text Transfer Transformer) model. Unlike approaches based on recurrent neural networks (RNN) such as OpenSESAME (Swayamdipta et al., 2017), FST reformulates the entire FrameNet semantic annotation process as a sequence-to-sequence task (Chanin, 2023). Concretely, FrameNet semantic parsing is decomposed into three sub-tasks performed in sequence:

1. Trigger identification: locating the words or expressions that evoke a semantic frame.

2. Frame classification: assigning a frame element (FE) to the identified trigger.

3. Argument extraction: detecting the frame elements and their values in the sentence.

[2] https://penman.readthedocs.io/en/latest/

In general, FrameSemanticTransformer-base (the basic pre-trained version of FST) is an automatic parser capable of annotating natural language text without external dependencies. We will focus on this model in the context of this work to perform FrameNet annotation on the retrieved RAG context. The generated annotation will subsequently be converted into a list of RDF triples, and then into a Penman-format graph. Below is an example of the complete transformation of our example sentence in English: "Nanjing is an important railway hub in eastern China."

1. FrameNet annotation by FST: Nanjing is an $_{\textbf{Importance}}$ IMPORTANT $_{\textbf{Roadways}}$ RAILWAY $_{\textbf{Political_locales}}$ HUB in $_{\textbf{Part_orientational}}$ EASTERN China.
2. RDF triple list (detail in Appendix A):

```
[[("important", "evokes",
↪  "Frame:Importance"),
↪  ("Frame:Importance", "hasRole:Factor",
↪  "railway")], ...]]
```

3. Final Penman graph:

```
(Sent1 / Sentence
    :frame (Im1 / Importance
        :text "important"
        :Factor "railway hub")
    :frame (Ro1 / Roadways
        :text "railway"
        :Roadway "railway")
    :frame (Po1 / Political_locales
        :text "hub"
        :Locale "Nanjing"
        :Descriptor "railway"
        :Relative_location "in eastern China")
    :frame (Pa1 / Part_orientational
        :text "eastern"
        :Whole "China))
```

Unlike the approach proposed by (Jin et al., 2024), which consists of inserting the full semantic graph of the text after it, we juxtapose the graph associated with each sentence immediately after that sentence. We call this representation "HybridGraph" (see Figure 1).

The Penman graph could be too voluminous to be incorporated as-is into the context if the corresponding sentence is too long. We observe that the Penman graph generally contains about twice as many tokens as the original text. In order to allow the inclusion of a larger volume of context — and therefore more retrieved information — without exceeding the LLM's token limit (e.g. 32,768 tokens for the quantised Mistral 8×7B), we apply

```
Nanjing is an important railway hub in eastern
↪  China.
(Sent1 / Sentence
    :frame (Im1 / Importance
        :text "important"
        :Factor "railway hub")
    ...)
It serves as rail junction for the
↪  Beijing-Shanghai (Jinghu).
(Sent2 / Sentence
    :frame (As1 / Assistance
        :text "serves"
        :Helper "It"
        :Goal "as rail junction for the
        ↪  Beijing-Shanghai (Jinghu)")
    :frame (Ro2 / Roadways
        :text "rail"
        :Roadway "rail"))
```

Figure 1: Illustration of the HybridGraph representation for the example sentences *"Nanjing is an important railway hub in eastern China. It serves as rail junction for the Beijing-Shanghai (Jinghu)."*

filtering of FrameNet annotations in the contextual representations used for generation, as described in the next section 3.1.

3.1 Semantic Annotation Filtering (NE Filtering)

This filtering depends on the question asked. Named entity recognition is first performed on the question using the SpaCy **en_core_web_trf**[3] model, based on **roberta-base**, in order to obtain a robust analysis. Then, among the FrameNet annotations of the retrieved context, only frames containing at least one of the detected named entities are retained. The resulting graph is much more concise. Frames containing the named entities found in the questions generally encapsulate a high density of relevant semantic information, which makes it possible to retain the most informative elements while reducing contextual noise. In general, when the information retrieval system identifies and retrieves the relevant context for the query, this context is then enriched with FrameNet graphs. The question is then concatenated with this augmented context, and the whole is transmitted to the LLM to generate the response. A detailed example of the filtering is provided in the appendix (see B.1)).

3.2 Evaluation Method

To evaluate the generation quality of LLMs, we used the following measures: ROUGE-1, MNLI, BertScore, Exact Inclusion, and LLM Score. ROUGE-1 computes the overlap of individual lexical units (unigrams) between the generated text and the reference text, without taking word order or syntax into account (Lin, 2004). The second measure used is MNLI (Multi-Genre Natural Language Inference) (Williams et al., 2018), specifically the **facebook/bart-large-mnli** model (Lewis et al., 2019). Based on the task of inferring logical relations between two texts (Natural Language Inference), it is used to quantify model performance on the understanding of complex semantic relations between two sentences. The use of **facebook/bart-large-mnli** allows the metric to be applied not only to isolated sentences but also to slightly longer text segments, unlike the first MNLI models based on RNN architectures. This metric is particularly relevant here because it evaluates the semantic coherence between the generated response and the reference, and makes it possible to test the model's capacity to reason, paraphrase, and infer. The third metric used is BertScore (Zhang et al., 2020), which evaluates semantic similarity using BERT-type embeddings between the generated text and the reference. The Exact Inclusion metric is an adaptation of the Exact Match metric proposed in SQuAD (Rajpurkar et al., 2016), which checks whether the generated text exactly matches the reference. In this work, we adopt a more lenient variant consisting of checking whether the reference is entirely contained in the generated response, rather than requiring an exact match. Finally, we use an LLM-as-a-judge metric, which we call LLM Score, inspired by (Wang et al., 2023). We instruct an LLM (here Mistral 24B[4]) to evaluate the factual quality of the text generated with respect to the reference. The instruction used is provided in the appendix (see C.1).

4 Experiment

4.1 Corpus

The corpus used in our experiments is SQuAD 1.1 (Rajpurkar et al., 2016), a standard reference corpus commonly used for evaluating RAG-type approaches on the question-answering task (Sawarkar et al., 2025; Veturi et al., 2024; Zhu, 2025; Zhang et al., 2025; Finardi et al., 2024; Guo et al., 2025a). It is an English reading comprehension corpus designed to evaluate a system's ability to read a text and answer associated questions. In version 1.1,

[3] https://spacy.io/models/en#en_core_web_trf

[4] https://huggingface.co/dphn/Dolphin-Mistral-24B-Venice-Edition

the corpus contains more than 100,000 question-answer pairs. In the context of this study, we randomly select 600 entries from SQuAD 1.1, each containing fields such as **question**, **context**, and **answers**. The answer always corresponds to a text segment extracted directly from the associated paragraph. Since SQuAD 1.1 covers a wide thematic diversity such as science, culture, history, entertainment, etc., a test corpus thus formed makes it possible to test RAG systems across various domains.

4.2 Experimental Protocol

The textual RAG implemented in our experiments is the classical version, composed of a lexical embedding database, an information retrieval system, and an LLM. The information retrieval system is based on vector search applied to the lexical embedding database. Each received question is transformed into a vector by a single encoder common to all embeddings in the database, in order to retrieve the most similar relevant texts for the question.

To build this lexical embedding database, the context provided in the SQuAD 1.1 dataset is extracted and vectorised by the "E5-large-v2" encoder, a bidirectional encoder producing high-quality lexical embeddings for a wide variety of tasks (Wang et al., 2024). No text chunking was performed during database creation, as each context already presents a reasonable size (on average 138 tokens per text), sufficient to be representative while remaining compatible with the encoder's capacity. Moreover, this simplifies processing and avoids the complexity related to text segmentation. After each retrieval, the three most relevant texts are retained as context.

In order to determine the different impacts of the components of our HybridGraph architecture, we integrated different context resources in our experiments. First, we completely remove the textual context and keep only the Penman graph. The second resource retains only a version of the graph filtered according to the named entities present in the question. The third is HybridGraph Penman, associating a textual context with the graph representing its semantics. The fourth resource is identical to the previous one except that the graph has been filtered according to entities. Finally, a purely textual context is used as a reference for comparison with these four types of contextual resources.

We thus obtain five different prompt types for generation:

1. Penman (see Appendix 3)
2. Penman NE filtered (see 3.1)
3. HybridGraph Penman (see Figure 1)
4. HybridGraph Penman NE filtered (see 3.1)
5. Text only (baseline)

The Penman format makes it possible to assess the impacts of replacing textual context with its FrameNet graph. The HybridGraph Penman format, in which the sentence is concatenated with its graph, will determine whether adding a FrameNet graph actually enriches the textual context, and whether replacing the text entirely with a graph leads to a loss of information not represented in the FrameNet graphs. Finally, the filtered variants consist of studying whether a focus on the named entities in the question could yield equivalent or even better performance than the unfiltered version.

The different context resources are concatenated with the question before being transmitted to the LLM to generate the response. In the experiments, the LLM used is always the quantised Mistral 8×7B[5]. We calculated statistical significance (P-value < 0.05) of the results with the Wilcoxon signed-rank test (Hae-Young, 2014). Non-significant results are displayed with a grey background in Table 1.

5 Results Analysis

We conducted quantitative and qualitative analyses on the entire test corpus (600 entries) to rigorously study the generation results. According to Table 1, the evaluation results suggest that in our complete RAG system, the LLM with a context composed solely of FrameNet graphs is not capable of generating a high-quality response. Moreover, filtering such a graph to retain only the sub-graphs containing the named entities present in the query significantly degrades LLM performance.

However, the context we call HybridGraph, associating the textual context from the information retrieval phase with complete or filtered FrameNet graphs, allows the LLM to achieve performances globally close to those obtained with a purely textual context, and even superior according to certain measures. Unfortunately, this slight performance improvement observed in our experiments

[5] https://huggingface.co/casperhansen/mixtral-instruct-awq

Metric %	Texte	Penman	Penman (NE filtered)	HybridGraph Penman	HybridGraph Penman (NE filtered)
ROUGE-1 Precision	17.3	14.7	9.8	**18.6**	16.9
ROUGE-1 Recall	**86.0**	65.6	42.5	83.3	84.9
ROUGE-1 F1	26.3	21.8	14.5	**27.4**	25.7
MNLI Entailment	43.9	38.5	29.0	**47.3**	43.2
MNLI Contradiction	**6.8**	17.8	34.6	7.5	7.9
MNLI Neutral	49.3	43.8	36.5	45.3	49.0
BertScore Precision	82.9	82.8	82.4	**83.2**	82.8
BertScore Recall	**87.6**	86.2	84.5	**87.6**	87.5
BertScore F1	85.2	84.4	83.4	**85.3**	85.1
Exact Inclusion	76.2	55.7	32.2	75.0	**76.7**
LLM Score	**84.9**	69.0	47.9	82.5	83.7

Table 1: With 3 retrieved documents: comparison of **Text** vs. **Graph** and **HybridGraph**; statistically significant results are shown with a light background and best performance in bold.

is not statistically significant. Nevertheless, based on our qualitative analysis, we observe that HybridGraph contributes to improving the factual accuracy of the responses generated by the LLM. Among the twenty-two questions for which the LLM responded less well with the textual context than with HybridGraph, the types of inaccuracy are:

1. Omission: the model omits an important element from the expected answer.
   - for the reference "the old Jinghu and Huning railway lines", the adjective "old" is absent in the LLM's response with the textual context.
2. Semantic confusion: the model replaces a precise semantic relation with a close but incorrect one.
   - for the question "At what period was Antarctica still connected to Australia?", the LLM confuses the end boundary with the time interval by answering "during the Cretaceous" instead of "at the end of the Cretaceous".
   - the LLM confuses the author of "The BFG" (Roald Dahl) with the screenwriter of its film adaptation (Melissa Mathison).
3. Over-generalisation: the model proposes several possible answers.
   - the LLM answers "late 18th century or early 19th century" whereas the exact answer is "early 19th century".
4. Contextual confusion: the LLM answers with a nearby incorrect element in the context.
   - in a sentence mentioning "Basel II recommendations", the LLM chooses "European legislators", which appears nearby but is not the correct answer.

In contrast, inserting FrameNet graphs between sentences seems to help highlight essential information, such as the roles of people (e.g. the inventor). We also identified a case in which the LLM was unable to distinguish the screenwriter from the author of the original work. Furthermore, with a purely textual context, the LLM tends to rely on its own knowledge when the context does not provide relevant information, despite an explicit instruction asking it not to do so. Moreover, with HybridGraphs, the performance difference between the filtered and unfiltered versions is not statistically significant according to three of the five measures. This thus demonstrates the contribution of entity filtering in terms of reducing the context size required for generation when combining a textual context with an explicit semantic representation. Our qualitative analysis also shows that the LLM makes errors on largely similar questions with or without entity filtering.

Our human evaluation also reveals several observations that could explain why HybridGraph is not clearly superior to using the textual context alone. Although the semantic graphs derived from FrameNet encode much information through Frames and Frame Elements, they omit certain information present in the original sentence. For example, the concession "But" and the negation "didn't" appear neither explicitly nor implicitly in the graphs. This absence results from the limitations of the FrameSemanticTransformer-base model; the training corpus contains little data with negation. More generally, certain types of linguis-

tic information are only partially represented in FrameNet. Modality (probability, obligation, possibility, etc.) is only annotated when it is lexicalised by a word triggering a modal Frame. The automatic annotation model (here based on T5) is therefore limited by its training corpus. Furthermore, FrameNet does not encode grammatical auxiliaries such as "will" (future) or "have" (perfect aspect) when they serve to mark tense, aspect, or other syntactic functions that are independent of autonomous lexical meaning. In our test corpus, 239 contexts out of 600 present some form of modality. Of these, only 58 instances are effectively annotated by the automatic model. Analysis of surface forms shows that modal auxiliaries are the most frequent category, representing 33.44% of the modality occurrences detected via a regular expression.

Thus, representing a text solely by its FrameNet semantic graph likely leads to ignoring information essential for a complete interpretation and degrades linear comprehension of the original text. Adding named entity-based filtering further accentuates this information loss. For these reasons, it appears relevant to explore the combination of text and its FrameNet graph within the RAG, which corresponds precisely to HybridGraph.

Focusing on the questions for which the LLM performed less well with a HybridGraph context, we also observe several typical errors:

1. Over-reliance on the graph: the LLM tends to privilege information from the graph, even when the answer is explicitly present in the context sentences.
    - for a question about the year of Tito's involvement in Israeli-Arab peace processes, the model concludes that information is absent from the graph, whereas the answer (1967) is present in the text.
    - similarly, for the question "How many male collegiate sports teams does Northwestern University field?", the model fails to exploit the context sentences and limits itself to the graph.
2. Continuity break in the context: the insertion of graphs within the text can disrupt sequential reading and hinder the model's ability to connect complementary information.
    - in a context excerpt about the RIBA library, two key pieces of information appear in separate sentences separated by the inserted graph, preventing the LLM from combining them to answer correctly.

Finally, we carried out complementary experiments to ensure the robustness of our experimental results. It is known that LLMs are sometimes sensitive to small variations in the form of the queries they receive (Pecher et al., 2026). Given a possible dependence on the format of semantic representation of the context information used for generation, we first conducted the same experiments, but systematically representing the Penman information as RDF triple lists rather than in Penman format. The results obtained show no significant difference compared to those obtained with Penman. Then, in a second experiment, a variant of HybridGraph was evaluated by concatenating each sentence with its associated graph represented as a JSON key-value structure (the sentence serving as the key and the corresponding list of triples as the value). Here too, LLM performance remains virtually equivalent to that observed with our HybridGraph Penman format.

6 Evaluation on Complex QA Benchmark

Since SQuAD 1.1 questions are exclusively extractive, requiring little reasoning, and the dataset predates our LLM (contamination risk of our LLM), we evaluate HybridGraph on a more challenging benchmark to assess its broader applicability.

GraphRAG Benchmark[6] (Xiang et al., 2025) is designed for knowledge-graph-based RAG evaluation. It covers four task types across varying reasoning levels. We use its literary subset (1,943 questions: 971 Fact Retrieval, 610 Complex Reasoning, 362 Contextual Summarize) and exclude Creative Generation (3.3% of the corpus), whose boundary with Contextual Summarize is unclear. Since SQuAD experiments showed no significant difference between filtered and unfiltered variants, we evaluate only the NE filtered HybridGraph Penman configuration against a text only baseline, using the chunking setup of the best GraphRAG system reported in the original paper.

We adapt the evaluation protocol: Exact Inclusion is dropped (rare in open-ended generation), and LLM Score is replaced by Answer Accuracy and Evidence Coverage, two LLM-as-a-judge metrics from the benchmark paper, in order to al-

[6] https://github.com/GraphRAG-Bench/GraphRAG-Benchmark

low direct score comparison. We also applied the Wilcoxon signed-rank test in the tables 2, 3 and 4.

Fact Retrieval (%)	Text	HybridGraph
ROUGE-L	24.4	21.8
ROUGE-1 Precision	22.0	19.6
ROUGE-1 F1	25.8	22.9
MNLI Entailment	12.5	**13.7**
MNLI Contradiction	48.7	**42.2**
MNLI Neutral	38.8	44.2
BertScore F1	87.1	86.7

Table 2: Significant results for Fact Retrieval

Complex Reasoning (%)	Text	HybridGraph
ROUGE-L	21.0	19.2
ROUGE-1 F1	23.9	22.0
MNLI Entailment	15.9	**21.5**
BertScore F1	87.0	86.4

Table 3: Significant results for Complex Reasoning

Contextual Summarize (%)	Text	HybridGraph
BertScore F1	86.3	85.8

Table 4: Significant results for Contextual Summarize

The Wilcoxon test confirms that HybridGraph significantly improves MNLI scores: entailment increases by 6% on Complex Reasoning (table 3) and contradictions decrease by 6% on Fact Retrieval (table 2). This corresponds with FrameNet's ability to encode conceptual situations independently of surface wording. Frame Elements make semantic roles and logical relations explicit, supporting inference beyond lexical overlap. Inversely, ROUGE scores always slightly decrease (>1%), as FrameNet leads the LLM toward semantic concepts rather than textual phrasing. BertScore is less affected (0.5% reduction) owing to its higher tolerance for paraphrase and synonymy.

According to the paper, traditional KG-based GraphRAG achieves better generation performance (ROUGE-L, Answer Accuracy, Coverage Score) on Complex Reasoning and Contextual Summarize tasks. This conclusion partially aligns with ours, although the KG-based GraphRAG differs from our HybridGraph approach, where FrameNet graphs intervene at generation time rather than retrieval time. Notably, the authors also observe that KG-based retrieval produces substantially longer contexts than standard RAG, due to its additional knowledge aggregation steps, which can introduce information overload and hurt context relevance. This mirrors a limitation shared by our HybridGraph approach: both methods face the challenge of context length growth, albeit through different mechanisms.

7 Limitations

Our study has two main limitations. First, all experiments were conducted with a single LLM (quantised Mistral 8×7B), and it remains to be seen whether the observed results generalise to other models. Second, we used training data rather than test data for our measures, which may introduce a bias.

8 Conclusion

This work proposes an approach called HybridGraph, which consists of enriching the context information from the information retrieval phase before its integration into the generation phase of the RAG-LLM system. In this approach, we associate textual content with its corresponding semantic graph based on FrameNet frames. This format allows us to appreciate the contribution of making the semantics of the textual context explicit when it is provided to the LLM for response generation. Although LLM performance with such a context is not statistically different in our experiments from that obtained with a purely textual context, our qualitative analysis shows that this explicit semantics can, in certain cases, improve the accuracy of the generated responses.

In contrast, the complete replacement of the textual context by the FrameNet graph leads to a significant degradation of the quality of the responses produced by LLMs. On the other hand, in the case of HybridGraph experiments associating a textual context and a FrameNet graph filtered according to the entities present in the question, the performance of the filtered version shows no statistically significant difference compared to an unfiltered approach, thus proving the value of entity filtering in the HybridGraph approach with regard to reducing the size of the explicit semantic representation.

In order to deepen the study of the impacts of HybridGraph in the RAG framework, it would be relevant to fine-tune the FrameNet annotator, FST, to obtain more accurate and consistent annotations. Finally, it would also be conceivable to introduce entity chaining and coreference resolution in the graphs, with the objective of producing graphs that are more compact, coherent, and less redundant.

References

Rim Abrougui, Géraldine Damnati, Johannes Heinecke, and Frédéric Béchet. 2023. Abstract representation for multi-intent spoken language understanding. In *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5.

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013. Abstract meaning representation for sembanking. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pages 178–186. Association for Computational Linguistics.

Michele Bevilacqua, Rexhina Blloshmi, and Roberto Navigli. 2021. One spring to rule them both: Symmetric amr semantic parsing and generation without a complex pipeline. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(14):12564–12573.

Claire Bonial, Lucia Donatelli, Mitchell Abrams, Stephanie M. Lukin, Stephen Tratz, Matthew Marge, Ron Artstein, David Traum, and Clare Voss. 2020. Dialogue-AMR: Abstract Meaning Representation for dialogue. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 684–695, Marseille, France. European Language Resources Association.

David Chanin. 2023. Open-source frame semantic parsing. *arXiv preprint arXiv:2303.12788*.

Christine Cuskley, Rebecca Woods, and Molly Flaherty. 2024. The limitations of large language models for understanding human language and cognition. *Open Mind*, 8:1058–1083.

Charles J Fillmore and 1 others. 2006. Frame semantics. *Cognitive linguistics: Basic readings*, 34:373–400.

Paulo Finardi, Leonardo Avila, Rodrigo Castaldoni, Pedro Gengo, Celio Larcher, Marcos Piau, Pablo Costa, and Vinicius Caridá. 2024. The chronicles of rag: The retriever, the chunk and the generator. *Preprint*, arXiv:2401.07883.

Michael Wayne Goodman. 2020. Penman: An open-source library and tool for AMR graphs. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 312–319, Online. Association for Computational Linguistics.

Shuyu Guo, Shuo Zhang, and Zhaochun Ren. 2025a. Enhancing rag efficiency with adaptive context compression. *Preprint*, arXiv:2507.22931.

Zirui Guo, Lianghao Xia, Yanhua Yu, Tu Ao, and Chao Huang. 2025b. Lightrag: Simple and fast retrieval-augmented generation. *Preprint*, arXiv:2410.05779.

Kim Hae-Young. 2014. Statistical notes for clinical researchers: Nonparametric statistical methods: 1. nonparametric methods for comparing two groups. *Restor Dent Endod*, 39(3):235–239.

Haoyu Han, Yu Wang, Harry Shomer, Kai Guo, Jiayuan Ding, Yongjia Lei, Mahantesh Halappanavar, Ryan A. Rossi, Subhabrata Mukherjee, Xianfeng Tang, Qi He, Zhigang Hua, Bo Long, Tong Zhao, Neil Shah, Amin Javari, Yinglong Xia, and Jiliang Tang. 2025. Retrieval-augmented generation with graphs (graphrag). *Preprint*, arXiv:2501.00309.

Zhijing Jin, Yuen Chen, Fernando Gonzalez, Jiarui Liu, Jiayi Zhang, Julian Michael, Bernhard Schölkopf, and Mona Diab. 2024. Analyzing the role of semantic representations in the era of large language models. *Preprint*, arXiv:2405.01502.

Jeongwoo Kang, Maximin Coavoux, Cédric Lopez, and Didier Schwab. 2025. Reassessing graph linearization for sequence-to-sequence amr parsing: On the advantages and limitations of triple-based encoding. *Preprint*, arXiv:2505.08504.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *Preprint*, arXiv:1910.13461.

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.

Harish Tayyar Madabushi. 2024. Fs-rag: A frame semantics based approach for improved factual accuracy in large language models. *Preprint*, arXiv:2406.16167.

Branislav Pecher, Michal Spiegel, Robert Belanec, and Jan Cegin. 2026. Revisiting prompt sensitivity in large language models for text classification: The role of prompt underspecification. *Preprint*, arXiv:2602.04297.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Kunal Sawarkar, Shivam R. Solanki, and Abhilasha Mangal. 2025. Metagen blended rag: Unlocking zero-shot precision for specialized domain question-answering. *Preprint*, arXiv:2505.18247.

Swabha Swayamdipta, Sam Thomson, Chris Dyer, and Noah A. Smith. 2017. Frame-Semantic Parsing with Softmax-Margin Segmental RNNs and a Syntactic Scaffold. *arXiv preprint arXiv:1706.09528*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2023. Attention is all you need. *Preprint*, arXiv:1706.03762.

Sriram Veturi, Saurabh Vaichal, Reshma Lal Jagadheesh, Nafis Irtiza Tripto, and Nian Yan. 2024. Rag based question-answering for contextual response prediction system. *Preprint*, arXiv:2409.03708.

Jiaan Wang, Yunlong Liang, Fandong Meng, Zengkui Sun, Haoxiang Shi, Zhixu Li, Jinan Xu, Jianfeng Qu, and Jie Zhou. 2023. Is chatgpt a good nlg evaluator? a preliminary study. *Preprint*, arXiv:2303.04048.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2024. Text embeddings by weakly-supervised contrastive pre-training. *Preprint*, arXiv:2212.03533.

Yufang Wang, Tongtong Xu, and Yihui Zhu. 2025. Sprrag: Semantic parsing retriever-enhanced question answering for power policy. *Algorithms*, 18(12).

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

Zhishang Xiang, Chuanjie Wu, Qinggang Zhang, Shengyuan Chen, Zijin Hong, Xiao Huang, and Jinsong Su. 2025. When to use graphs in rag: A comprehensive analysis for graph retrieval-augmented generation. *Preprint*, arXiv:2506.05690v1.

Xiaozhou You, Yahui Luo, and Lihong Gu. 2025. MS-RAG: Simple and effective multi-semantic retrieval-augmented generation. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 22609–22625, Suzhou, China. Association for Computational Linguistics.

Ningning Zhang, Chi Zhang, Zhizhong Tan, Xingxing Yang, Weiping Deng, and Wenyong Wang. 2025. Credible plan-driven rag method for multi-hop question answering. *Preprint*, arXiv:2504.16787.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. *Preprint*, arXiv:1904.09675.

Kezhen Zhong, Basem Suleiman, Abdelkarim Erradi, and Shijing Chen. 2025. Semrag: Semantic knowledge-augmented rag for improved question-answering. *Preprint*, arXiv:2507.21110.

Yuzhou Zhu. 2025. From static to dynamic: A streaming rag approach to real-time knowledge base. *Preprint*, arXiv:2508.05662.

A RDF triples list

```
[[("important", "evokes", "Frame:Importance"),
  ("Frame:Importance", "hasRole:Factor",
  ↪  "railway")],
  [("railway", "evokes", "Frame:Roadways"),
  ("Frame:Roadways", "hasRole:Roadway",
  ↪  "railway")],
  [("hub", "evokes", "Frame:Political_locales"),
  ("Frame:Political_locales", "hasRole:Locale",
  ↪  "Nanjing"),
  ("Frame:Political_locales",
  ↪  "hasRole:Descriptor", "railway"),
  ("Frame:Political_locales",
  ↪  "hasRole:Relative_location",
  "in eastern China")],
  [("eastern", "evokes",
  ↪  "Frame:Part_orientational"),
  ("Frame:Part_orientational", "hasRole:Whole",
  ↪  "China")]]
```

B Semantic annotation filtering (NE filtering)

B.1 Exemple of HybridGraph filtering

B.1.1 Context

Nanjing is an important railway hub in eastern China. It serves as rail junction for the Beijing-Shanghai (Jinghu) (which is itself composed of the old Jinpu and Huning Railways), Nanjing-Tongling Railway (Ningtong), Nanjing-Qidong (Ningqi), and the Nanjing-Xian (Ningxi) which encompasses the Hefei-Nanjing Railway. Nanjing is connected to the national high-speed railway network by Beijing-Shanghai High-Speed Railway and Shanghai-Wuhan-Chengdu Passenger Dedicated Line, with several more high-speed rail lines under construction.

B.1.2 Before the filtering

```
Nanjing is an important railway hub in
↪  eastern China.
(Sent1 / Sentence
    :frame (Im1 / Importance
        :text "important"
        :Factor "railway hub")
    :frame (Ro1 / Roadways
        :text "railway"
        :Roadway "railway")
    :frame (Po1 / Political_locales
        :text "hub"
        :Locale "Nanjing"
        :Descriptor "railway"
       :Relative_location "in eastern China")
    :frame (Pa1 / Part_orientational
        :text "eastern"
        :Whole "China"))
It serves as rail junction for the
↪  Beijing-Shanghai (Jinghu) (which is
↪  itself composed of the old Jinpu and
↪  Huning Railways), Nanjing-Tongling
↪  Railway (Ningtong), Nanjing-Qidong
↪  (Ningqi), and the Nanjing-Xian (Ningxi)
↪  which encompasses the Hefei-Nanjing
↪  Railway.
```

```
(Sent2 / Sentence
    :frame (As1 / Assistance
        :text "serves"
        :Helper "It"
        :Goal "as rail junction for the
        ↪  Beijing-Shanghai (Jinghu) (which
        ↪  is itself composed of the old
        ↪  Jinpu and Huning Railways ),
        ↪  Nanjing-Tongling Railway
        ↪  (Ningtong ), Nanjing-Qidong
        ↪  (Ningqi ), and the Nanjing-Xian
        ↪  (Ningxi ) which encompasses the
        ↪  Hefei-Nanjing Railway")
    :frame (Ro1 / Roadways
        :text "rail"
        :Roadway "rail")
    :frame (Ag1 / Age
        :text "old"
        :Entity "Jinpu and Huning Railways")
    :frame (Ro2 / Roadways
        :text "Railways"
        :Name "Jinpu and Huning"
        :Roadway "Railways")
    :frame (Ro3 / Roadways
        :text "Railway"
        :Name "Nanjing-Tongling"
        :Roadway "Railway")
    :frame (Ro4 / Roadways
        :text "Railway"
        :Name "Hefei-Nanjing"
        :Roadway "Railway"))
Nanjing is connected to the national
↪  high-speed railway network by
↪  Beijing-Shanghai High-Speed Railway and
↪  Shanghai-Wuhan-Chengdu Passenger
↪  Dedicated Line, with several more
↪  high-speed rail lines under construction.
(Sent3 / Sentence
    :frame (At1 / Attaching
        :text "connected"
        :Item "Nanjing"
        :Goal "to the national high-speed
        ↪  railway network"
        :Connector "by Beijing-Shanghai
        ↪  High-Speed Railway and
        ↪  Shanghai-Wuhan-Chengdu Passenger
        ↪  Dedicated Line")
    :frame (Po1 / Political_locales
        :text "national"
        :Locale "national")
    :frame (Ro1 / Roadways
        :text "railway"
        :Roadway "railway")
    :frame (Ne1 / Network
        :text "network"
        :Possessor "national"
        :Nodes "high-speed railway"
        :Network "network")
    :frame (Ro2 / Roadways
        :text "Railway"
        :Name "Beijing-Shanghai High-Speed"
        :Roadway "Railway")
    :frame (Ro3 / Roadways
        :text "Line"
        :Characteristic "Dedicated"
        :Roadway "Line")
    :frame (Ro4 / Roadways
        :text "rail"
        :Roadway "rail")
    :frame (Bu1 / Building
        :text "construction"
        :Created_entity "several more
        ↪  high-speed rail lines"))
```

B.1.3 After the filtering

```
Nanjing is an important railway hub in
↪  eastern China.
It serves as rail junction for the
↪  Beijing-Shanghai (Jinghu) (which is
↪  itself composed of the old Jinpu and
↪  Huning Railways), Nanjing-Tongling
↪  Railway (Ningtong), Nanjing-Qidong
↪  (Ningqi), and the Nanjing-Xian (Ningxi)
↪  which encompasses the Hefei-Nanjing
↪  Railway.
(S2 / Sent
    :frame (As1 / "Assistance"
        :text "serves"
        :Helper "It"
        :Goal "as rail junction for the
        ↪  Beijing-Shanghai (Jinghu) (which
        ↪  is itself composed of the old
        ↪  Jinpu and Huning Railways ),
        ↪  Nanjing-Tongling Railway
        ↪  (Ningtong ), Nanjing-Qidong
        ↪  (Ningqi ), and the Nanjing-Xian
        ↪  (Ningxi ) which encompasses the
        ↪  Hefei-Nanjing Railway"))
Nanjing is connected to the national
↪  high-speed railway network by
↪  Beijing-Shanghai High-Speed Railway and
↪  Shanghai-Wuhan-Chengdu Passenger
↪  Dedicated Line, with several more
↪  high-speed rail lines under construction.
(S3 / Sent
    :frame (At1 / "Attaching"
        :text "connected"
        :Item "Nanjing"
        :Goal "to the national high-speed
        ↪  railway network"
        :Connector "by Beijing-Shanghai
        ↪  High-Speed Railway and
        ↪  Shanghai-Wuhan-Chengdu Passenger
        ↪  Dedicated Line")
    :frame (Ro1 / "Roadways"
        :text "Railway"
        :Name "Beijing-Shanghai High-Speed"
        :Roadway "Railway"))
```

C Prompts

C.1 LLM Score metric

```
[INST]You are an expert evaluator. Your task is
↪  to judge how well the Actual Answer responds
↪  to the Question, using the Reference Answer
↪  as the ground truth.

Guidelines:
- Return only a single integer score from 0 to
↪  10.
- 0 = completely incorrect (does not answer the
↪  question or contradicts the reference).
- 10 = fully correct, answers the question, and
↪  matches the reference meaning.
- Intermediate scores = partially correct,
↪  incomplete, or somewhat aligned with the
↪  reference.
- The Actual Answer may be a full sentence, while
↪  the Reference Answer may be a phrase or
↪  segment of the correct information.
- Penalize if the Actual Answer introduces
↪  irrelevant or incorrect information.

Question: {entry['question']}

Actual Answer: {entry['generated_answer']}

Reference Answer: {entry['answer']}

Score (0-10):[/INST]
```