Computational Linguistics and Intellectual Technologies:
Proceedings of the International Conference "Dialogue 2025"

April 23–25, 2025

Enhancing RAG and Knowledge Graphs with Discourse

Boris Galitsky
Knowledge Trail, San Jose, CA, USA
bgalitsky@hotmail.com

Dmitry Ilvovsky
HSE University, Moscow, Russia
dilvovsky@hse.ru

Anton Morkovkin
HSE University, Moscow, Russia
ag.morkovkin@gmail.com

Abstract

We consider a number of Retrieval Augmented Generation (RAG) architectures to address a lack of specific information and hallucination issues of Large Language Models (LLM)—based question answering. We start with conformal prediction which acts on top of LLM and maintains a set of generations instead of a single one and attempts to find the best element of this set, which is assumed to be the "most average one". We then proceed to LLM self-reflection series of RAG architectures predicting the multi-hop question answering session before actual search for an answer. After that, we propose a mechanism for LLM to filter out answers inappropriate with respect to style. All these components need discourse-level analysis for more robust functioning. Knowledge graph (KG) and Abstract Meaning Representation (AMR)-based knowledge graph construction follow. We evaluate the contribution of all of these components to overall answer relevance and also zoom in on the role of discourse-based subsystem in each of these components. There is a substantial improvement of performance due to the four-component architecture introduced in this paper; the contribution of discourse-based subsystems is fairly modest.

DOI: 10.28995/2075-7182-2025-23-103-116

1 Introduction

Large Language Models (LLMs), such as GPT-4 and LLaMA, demonstrate strong performance in natural language understanding and generation, including in conversational search and question answering (QA). However, they often suffer from hallucinations and reduced accuracy when addressing queries involving rare, domain-specific, or long-tail entities. To address these limitations, Retrieval-Augmented Generation (RAG) approaches (Lewis et al., 2021) enhance LLMs by injecting retrieved textual information into the generation process. This combination of retrieval-based precision and generative fluency significantly improves the robustness of QA systems.

Recent work has extended RAG methods to broader domains and architectures. For instance, Feng et al. (2023) explore tighter integration between retrieval and generation components, improving LLM responsiveness to complex queries. Guo et al. (2023) demonstrate RAG's utility even in non-knowledge-intensive scenarios. More dynamic methods like active RAG [Jiang et al., 2023] adapt retrieval to evolving query intent in real time. Despite these advances, general-purpose RAG systems often fail in specialized domains. For example, RAG with GPT-4-Turbo struggles to answer financial questions derived from SEC filings, correctly handling only one out of five prompts. This indicates the necessity for more domain-aware methods, including fine-tuned LLMs and improved retrieval components.

Moreover, current RAG pipelines have difficulty performing complex reasoning, particularly in multi-hop QA tasks (Jeong et al., 2024; Zhang et al., 2024). Retrieval mechanisms often return noisy or irrelevant passages (Shi et al., 2023), and generated answers may contradict the retrieved evidence (Gao et al., 2023) or override LLMs' parametric knowledge (Parvez, 2024). While strategies like re-ranking (Nogueira and Cho, 2020) and conditional or active retrieval (Mallen et al., 2023) help alleviate some of these issues, they remain sensitive to annotation quality, may exclude useful information, and are computationally expensive. These limitations underscore the need for more robust integration of external knowledge.

To enhance the reasoning capabilities of RAG models, incorporating non-parametric knowledge is essential. Two primary sources are unstructured textual data (Izacard et al., 2022) and structured knowledge graphs (KGs). KGs provide compact and less noisy representations compared to full text and are more suitable for long-tail QA (Huang et al., 2024).

In this work, we also argue for the inclusion of discourse-level information. Discourse analysis examines how sentences relate to one another within a dialogue or text. It helps systems understand not only topical relevance, but also coherence, rhetorical structure, and intent. For example, rhetorical relations (e.g., Explanation, Cause, Elaboration) enable better matching between questions and answers. Discourse-level features are particularly beneficial for dialogue systems, where context extends beyond single utterances and includes stylistic and pragmatic dimensions.

We show that discourse analysis can:

- Improve the contextual understanding of dialogue history;
- Aid in structuring retrieved and generated content;
- Enhance answer filtering by identifying intent, tone, and rhetorical fit.

Our integrated system combines conformal prediction, self-reflective retrieval, discourse-aware QA coordination, knowledge graph integration, and AMR-based graph reasoning. Discourse elements enhance answer coherence and contextual appropriateness throughout the pipeline, contributing to the robustness of the overall architecture.

2 Conformal predictions for answer relevance improvement

Conformal prediction is a statistical technique for constructing prediction sets that are guaranteed to contain the semantically correct response with high probability. We apply it to improve the reliability of answers generated by RAG-based question answering systems. Instead of returning a single prediction, this approach constructs a set of candidates, such that the true answer is included in this set with a user-specified confidence level (e.g., 95%). This is especially important in RAG pipelines, where either the retrieved passage may be irrelevant or the LLM may produce an incorrect response even given a relevant passage (Fig. 1).

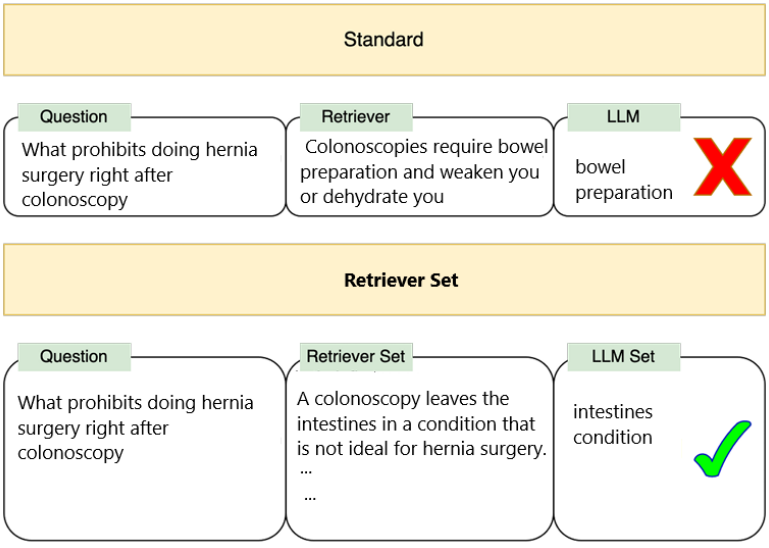

Figure 1. Proceeding from a single candidate to the candidate set for conformal prediction

The method builds two types of conformal prediction sets:

- Retriever set (CRet): ensures the relevant passage is included among those retrieved.
- LLM set (CLLM): ensures the generated answer is semantically correct with respect to the question and context.

These sets are then aggregated into an overall prediction set (C_{Agg}) that guarantees the presence of a correct answer with high probability. To construct the retriever set, we follow (Li et al., 2024) and use the negative inner product between the question and the annotated most relevant passage as the nonconformity measure. Given a calibration set and error tolerance α, a threshold τ_{Ret} is computed as the corresponding quantile over calibration scores, and all passages below this threshold are included in $C_{Ret}(q)$.

On the LLM side, prediction sets are constructed using a generation-and-clustering approach. For each <question, passage> pair, the LLM generates multiple responses (e.g., 30). These responses are clustered based on semantic similarity or entailment using an NLI model; responses are merged if they are semantically close or logically entailed. The confidence of each response is approximated as the ratio of its cluster size to the total number of generated responses. A threshold is applied to construct the prediction set $C_{LLM}(q, p)$from responses with sufficiently high confidence.

This methodology is extended to discourse-level representations. For each question, we construct prediction sets not only over generated answers but also over their discourse structures. The coordination between the question and the discourse structure of the answer is treated analogously to semantic coordination: a separate nonconformity score is defined for this dimension, and a threshold is applied to form a discourse-level prediction set. This allows the system to align both content and rhetorical structure with high confidence (Fig. 2).

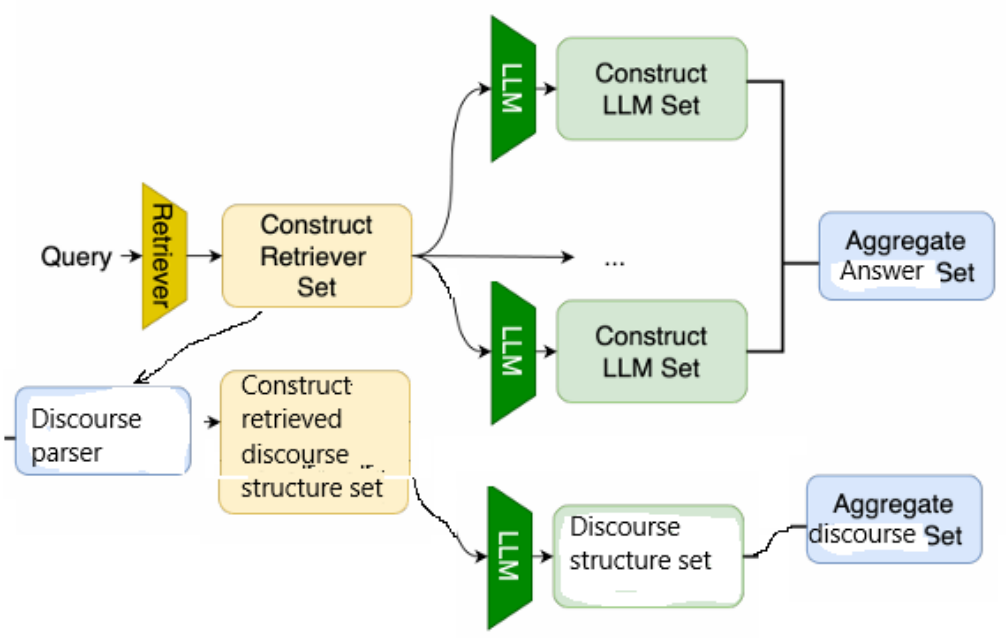

Figure 2. Conformal prediction for answers and discourse coordination of answers

To improve semantic coverage when constructing prediction sets, we apply Latin Hypercube Sampling, which ensures more uniform exploration of the confidence space than basic Monte Carlo sampling. Under mild assumptions such as data exchangeability, this conformal framework provides statistical guarantees for both semantic and discourse-level correctness of the predicted answers.

3 Self-reflection for RAGs

To address and control these behaviors such as retrieval frequency of the RAG model and guide the generation to be contextually consistent, Self RAG and its variants (Asai et al., 2024; Jeong et al., 2024) adopt a self-reflection based method. During training, these models learn to generate both task output and intermittent special reflection/critique tokens (e.g., *is_supported, is_relevant*, etc.), leveraging knowledge distillation from proprietary models like GPT-4. At inference, these generated tokens determine the usability of each candidate output. While these methods enable the model to effectively

rank candidate outputs from different retrievals and partially improve grounded/contextual generation, they struggle with navigating irrelevant or misleading information, especially when dealing with complex queries such as multi-hop retrieval tasks. This limitation arises since the models are not explicitly trained to contrast harder distractor passages and adhere to the facts from the retrievals.

To address this issue, Islam et al. (2024) reformulate the problem as a sparse mixture of experts (MoE) setup, where the model can selectively activate reasoning pathways that better differentiate subtle semantic contrasts in retrieved content. In our system, we extend this idea by incorporating discourse-aware self-reflection, enabling the model to not only evaluate semantic grounding but also analyze the rhetorical relation between the question and candidate answers.

In this paper, we enable self-reflection with discourse analysis of an occurrence of a potential answer in a passage obtained by retrieval. For different types of questions, the discourse structure of paragraphs containing answers significantly varies. So it is important not only to predict if a given question needs a retrieval session beyond what is encoded in LLM, but also to predict a discourse-level occurrence of a candidate answer in a text. A precise answer A occurs under a specific rhetorical relation to the phrase best matching the question (PBMQ).

During inference (Fig. 3), the model starts in a *no_retrieval* mode. It generates a preliminary answer and estimates its confidence. If no retrieval is needed, the model returns the answer based solely on parametric knowledge. Otherwise, for both single- or multi-hop queries from an external knowledge source $K = \{d_i\}_{i=1}^{N_k}$, we use a retriever R to obtain the top-m documents $S = \{s_t\}_{t=1}^{k}$, where each passage contains $\{r_j\}_{j=1}^{N_H}$ with $r_j \in K$ and N_H is the hop size. For each retrieved passage, the model M produces the output response y_t, along with the following tokens:

(1) The *relevance* tokens ([*relevant/irrelevant*]) indicate if s_t is relevant to q,
(2) the *grounding* tokens ([*fully supported/ps$_t$artially supported/no support*]) indicate if y_t is supported by s_t,
(3) the discourse relation specified how is the retrieved context unit is s_t related to the output response y_t rhetorically, and
(4) the *utility* tokens ([U:1]-[U:5]) define *how* useful y_t is to q.

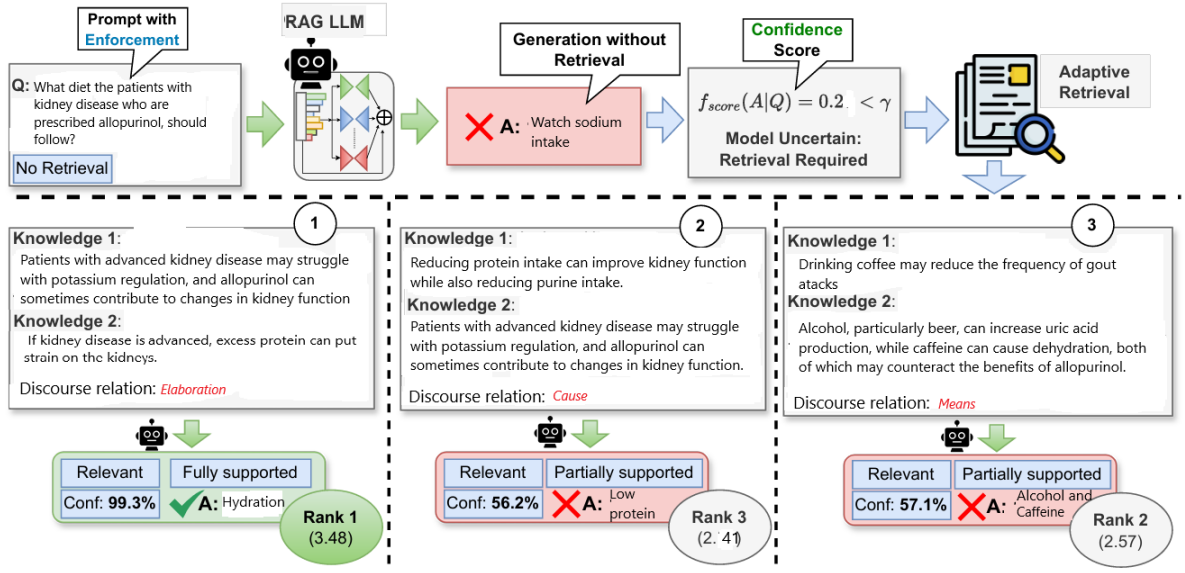

Figure 3. Inference pipeline in RAG self-reflection framework. For each knowledge set, discourse relation between the elements (PBMQ) and the candidate answer is specified.

We also evaluate the rhetorical relation between the Phrase Best Matching the Question (PBMQ) and each candidate answer. This relation helps determine whether the discourse structure of the answer matches the expected structure for the given question type. Specifically:

- Elaboration is expected for attribute-value questions;
- Explanation or Cause for *Why* questions;
- Enablement for procedural or *How* to achieve questions;
- Means for tool or method selection queries.

These discourse patterns guide filtering and selection of grounded answers that are not only semantically relevant but also contextually appropriate. As illustrated in Fig. 4, this approach enables the LLM to contrast distractors and reflect on answer quality during ranking and generation.

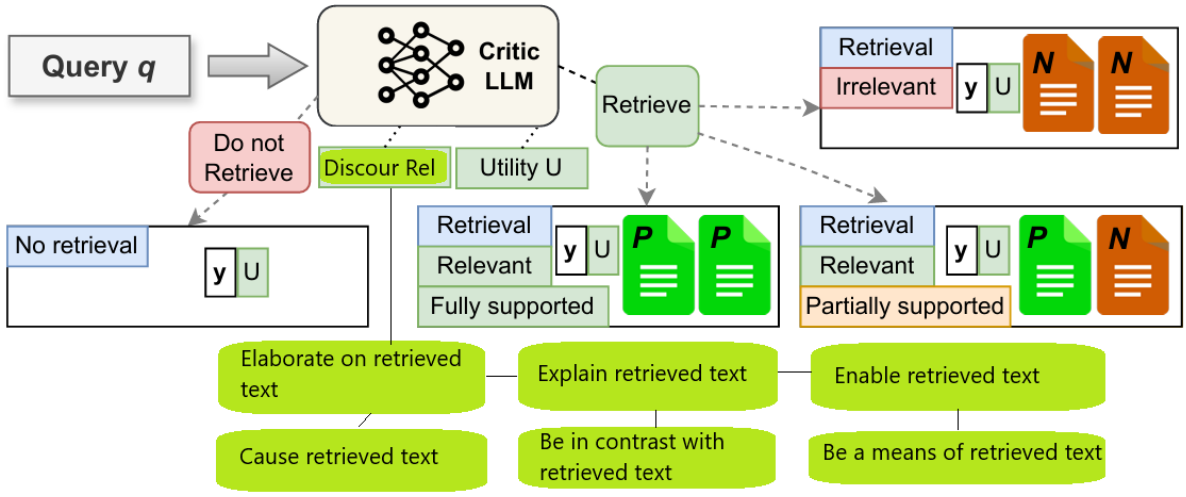

Figure 4. Enabling an LLM to reflect on an answer quality and to build contrast distractors

4 Enabling RAG with Knowledge Graph

While LLMs achieve strong performance across NLP tasks, they struggle with long-tail questions requiring domain-specific knowledge. To mitigate this, RAG architectures are often enhanced with non-parametric sources such as textual passages and knowledge graphs (KGs). Recent findings (Huang et al., 2024) demonstrate that prompting LLMs with KG triples can outperform passage-based retrieval, particularly in reducing hallucinations. Combining both KG and passage-based inputs may not consistently improve recall, but often enhances factual accuracy and answer grounding.

In our system, we apply this idea to the medical domain, using a two-phase architecture for question answering and treatment plan recommendation. During the KG construction phase, historical issue records (e.g., EHR-like customer support logs) are parsed into hierarchical tree structures, where each tree T_i represents an individual health issue. These trees are then interconnected to form a KG, combining both:

- Explicit links E_{exp}, specified in the documentation;
- and Implicit links E_{imp}, derived via semantic similarity between node embeddings (Xu et al., 2024).

An overview of RAG+KG framework is shown in Fig. 5. KG construction is shown on the left and Q/A and recommendation – on the right.

Each issue tree includes structured sections like “summary”, “description”, and “priority”. Fields suitable for rule-based parsing are extracted directly, while others are handled by an LLM guided by a template $T_{template}$. Embeddings are generated for semantically rich nodes and stored in a vector database. This dual-level KG structure allows us to preserve both intra-issue hierarchy and inter-issue relationships.

During the question-answering phase, user queries are parsed into entity-intent maps P of type Map($N \rightarrow V$), where N corresponds to a KG section (e.g., “issue summary”) and V is the extracted value. This parsing is performed by an LLM using prompts aligned with the template $T_{template}$.

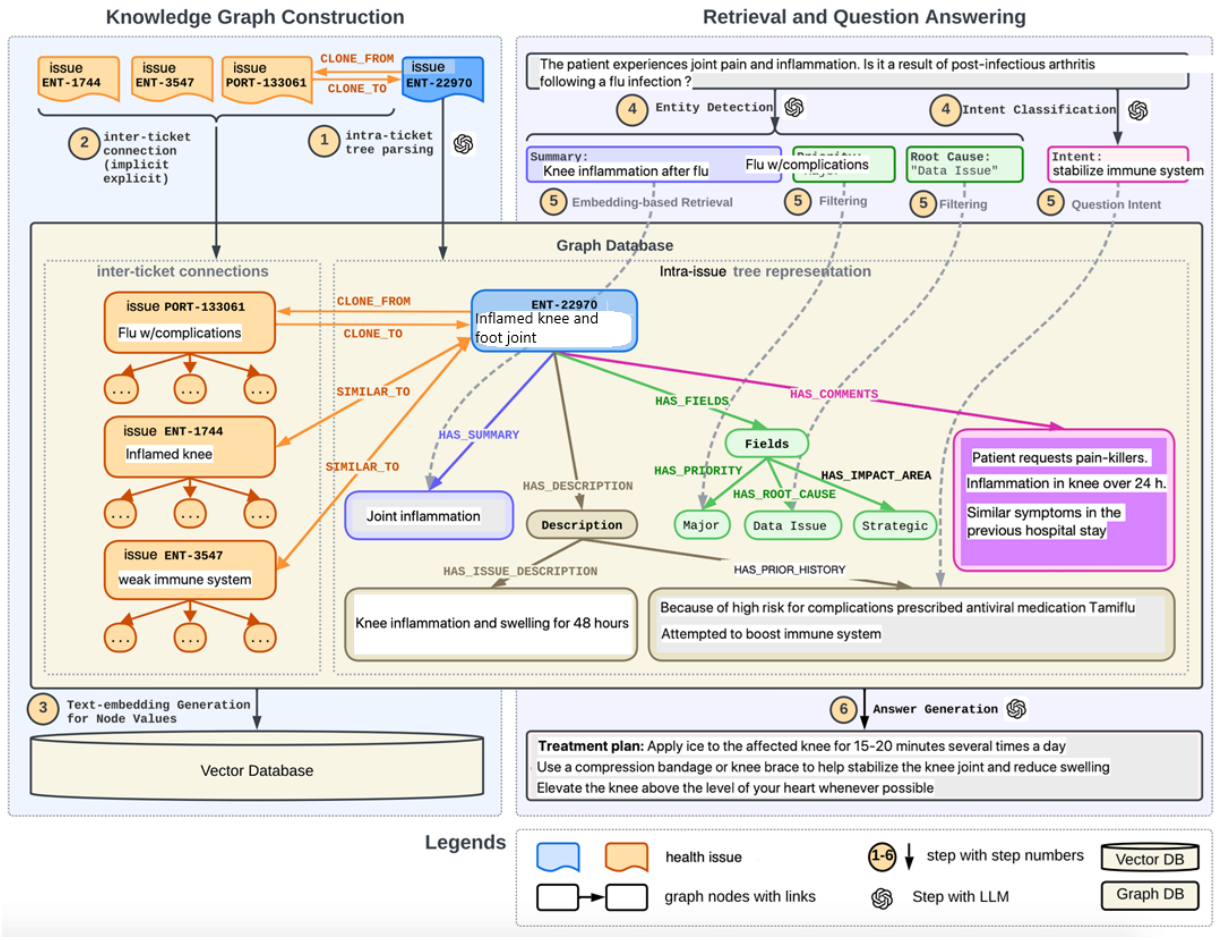

Figure 5. RAG+KG framework

For example, given the query “How to cure an inflamed knee which occurred as a complication of a flu?”, we extract:

- *Pe* = Map("*issue summary*" → "*inflamed knee*", "*issue description*" → "*knee inflammation after the flu with high temperature and dizziness*")
- and the intents *P=Set*(*elevation*, *ice*, *compress*, *rest*)

LLMs are generally robust in this task, and hallucinations are minimized through alignment with KG structure. Using these entity sets, the system retrieves top K most relevant historical cases based on cosine similarity of embeddings. A final score is aggregated per health issue using a sum over relevant section-node matches:

$$Score_{T_i} = \sum_{(k,v)\in P} \sum_{n\in T_i} TP\{section(n) = k\} * \cos\left(emdeb(v), embed\ (text(n)\right)$$

This score reflects both semantic proximity and entity occurrence frequency. Based on it, the most relevant subgraphs are selected for generation.

4.1 Leveraging LLMs to improve Graph Neural Network

Graph Neural Networks (GNNs) are widely used in graph machine learning. However, they often rely on shallow embeddings and struggle to generalize across diverse graph structures. LLMs can help by improving node representations, generating augmented features, and aligning feature spaces (Fan et al., 2024). Fig. 6 illustrates this idea.

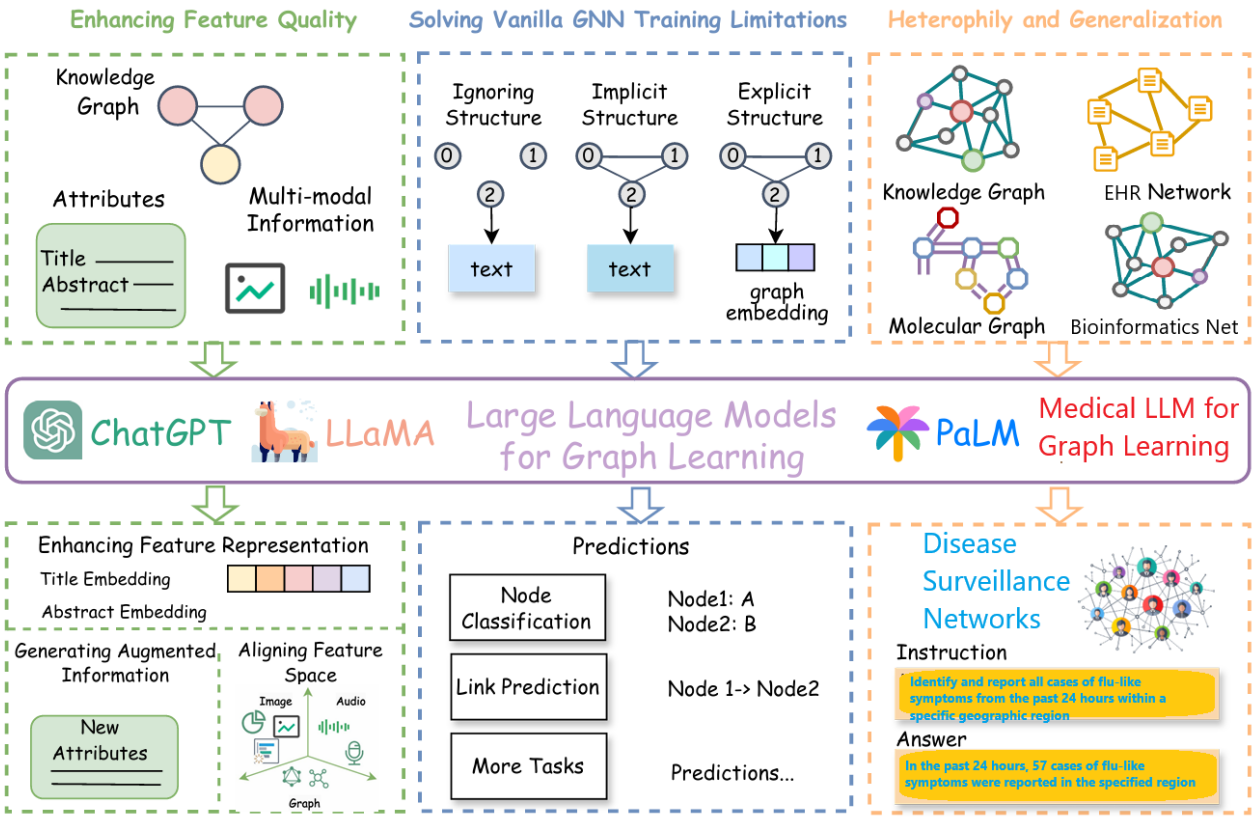

Figure 6. LLM assisting construction of KG

Challenges of vanilla GNNs include:

(1) Over-smoothing: mitigated with skip connections and batch normalization;
(2) Overfitting: addressed through node feature masking, dropout, and subgraph sampling;
(3) Heterophily: handled via adaptive aggregation (e.g., MixHop) or hybrid models (Abu-El-Haija et al., 2019);
(4) Low expressiveness: improved with k-GNNs, attention mechanisms, or higher-order message passing.

Recent works (He et al., 2023; Luo et al., 2023) also use LLMs to generate node labels or explanations from textual attributes (titles, abstracts), which are then encoded and combined with structural data in GNNs for downstream predictions.

4.2 Graph databases

The choice of graph vs. vector database plays a key role in RAG systems. Vector databases store embedded chunks and rely on semantic similarity for retrieval. This fuzzy matching improves flexibility but often introduces irrelevant or noisy passages.

In contrast, graph databases store structured entity-relation triples:

$$[ENTITY\ A] \rightarrow [RELATIONSHIP] \rightarrow [ENTITY\ B]$$

For example:

- *A* [*rust*] *EATS* [*metal*],
- *A* [*wolf*] *IS a* [*living thing*],

Benefits of graph databases for RAG include:

- Relational depth: better modeling of multi-hop reasoning;
- Reduced noise: fewer irrelevant results due to explicit structure;
- Explainability: reasoning paths are traceable and interpretable.

A limitation is rigid matching: if query entities don't exactly match stored nodes, relevant data may be missed. To mitigate this, hybrid systems combine graph structure with vector-based indexing. Entities and relations are encoded as embeddings, enabling approximate nearest-neighbor search over graph-like data.

This hybrid approach retains semantic flexibility while preserving explicit structure, improving retrieval quality and final LLM output.

5 AMR – based Knowledge Graph Construction

Retrieving relevant knowledge graph triples typically involves three steps: (1) tagging entities in the input query and aligning them with the KG; (2) retrieving candidate triples, often using SPARQL queries over RDF-based resources (Wang et al., 2021); and (3) ranking triples based on their relevance. To improve this ranking step, we incorporate Abstract Meaning Representation (AMR) to better capture the underlying relationships between entities.

AMR is a semantic representation framework that abstracts away from surface syntax and captures the meaning of sentences in a language-neutral graph format (Banarescu et al., 2013). AMR graphs are labeled, directed, rooted, and acyclic, where nodes represent events, entities, or properties, and edges denote semantic roles or relationships (e.g., ARG0, ARG1, temporal or modal relations). Compared to traditional syntactic parsing, AMR provides a unified view of meaning, supporting downstream tasks such as question answering, summarization, and information extraction (Naseem et al., 2021).

Fig. 7 illustrates an AMR expression in health domain *"We will send him to attend Dr. John Smith for a treatment review later this week following his testing"*.

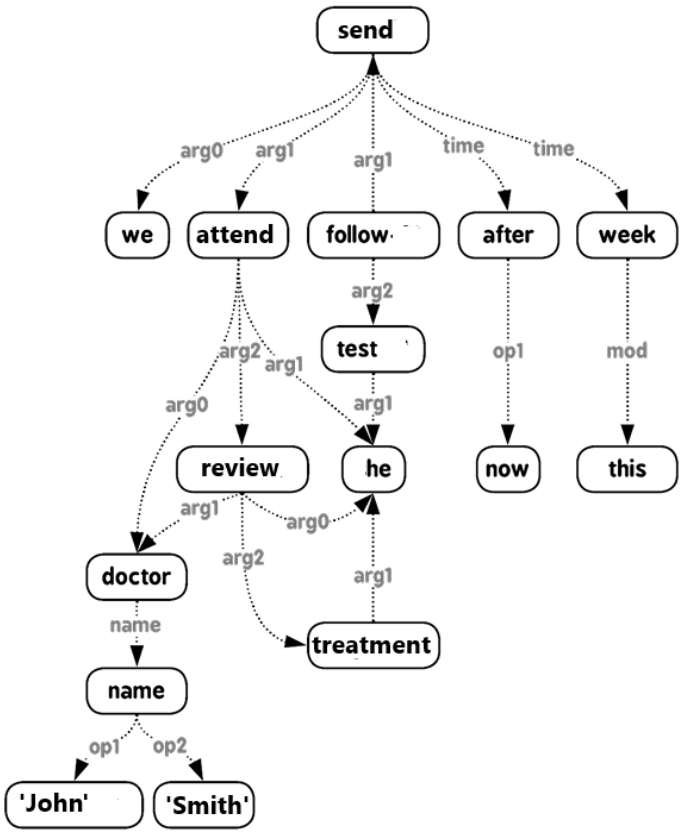

Figure 7. AMR representation for a sentence

In this AMR, the predicate attend represents the consultation event. Its ARG0 is “Dr. John Smith” (agent), ARG1 is “he” (patient), and the temporal modifiers express “later this week” via subgraphs like after now and within this week.

To apply AMR in triple ranking, we follow Huang et al. (2024) and use a dual-encoder architecture:

Entity-Relation Pairs Encoder. Given a sentence q, tagged entity e, and candidate relation r, the Entity-Relation Pairs as is defined as [TEXT]q[ENT]e[REL]r (Naseem et al.2021). [TEXT], [ENT], and [REL] are special tags inserted to mark the beginning of a sentence (q), tagged entity e, and candidate relations r respectively. This is passed through a BERT-based encoder to obtain representations $Q_{ERP} = [q_1,q_2,...,q_m]$, where each q_j corresponds to an entity-relation pair.

AMR-Enabled Multihead-Attention. The question is parsed into an AMR graph using parsers such as AMRLib (Jascob, 2024). Named entities are linked via BLINK (Wu et al., 2020), and the graph is encoded using a GNN. To avoid loss of edge semantics in message passing, all labeled edges are reified as nodes, preserving rich role information. Node embeddings $E_{AMR} = [e_1,e_2,...,e_n]$ are then computed via a Graph Attention Network

The two representations are fused using AMR-Multihead-Attention (AMA), which enables flexible alignment between semantic structures and candidate facts.

Let $Q = Q_{ERP}$, $K = V = E_{AMR}$. The AMR-Multihead-Attention (AMA) mechanism is defined as:

$$AMA(Q, K,V) = Concat(head_1,head_2,...,head_h)W^O$$

$$head_i = Attention(QW_i^Q, KW_i^K, VW_i^V) \text{ with } Attention(Q, K,V) = softmax(\frac{QK^T}{\sqrt{d_k}})V$$

where W_i^Q, W_i^K, and W_i^V are the parameter matrices for the i-th attention head, and W^O is a parameter matrix for linearly transforming the concatenated outputs of all heads. This attention formalization shows the connection between the AMR and each entity-relation pair. Using AMR attention mechanism supports a flexibility in interpretation of entity relations, expressing the underlying semantic structures within the sentences. AMR is great for normalizing a semantic representation, converting into a canonical form (Fig. 8).

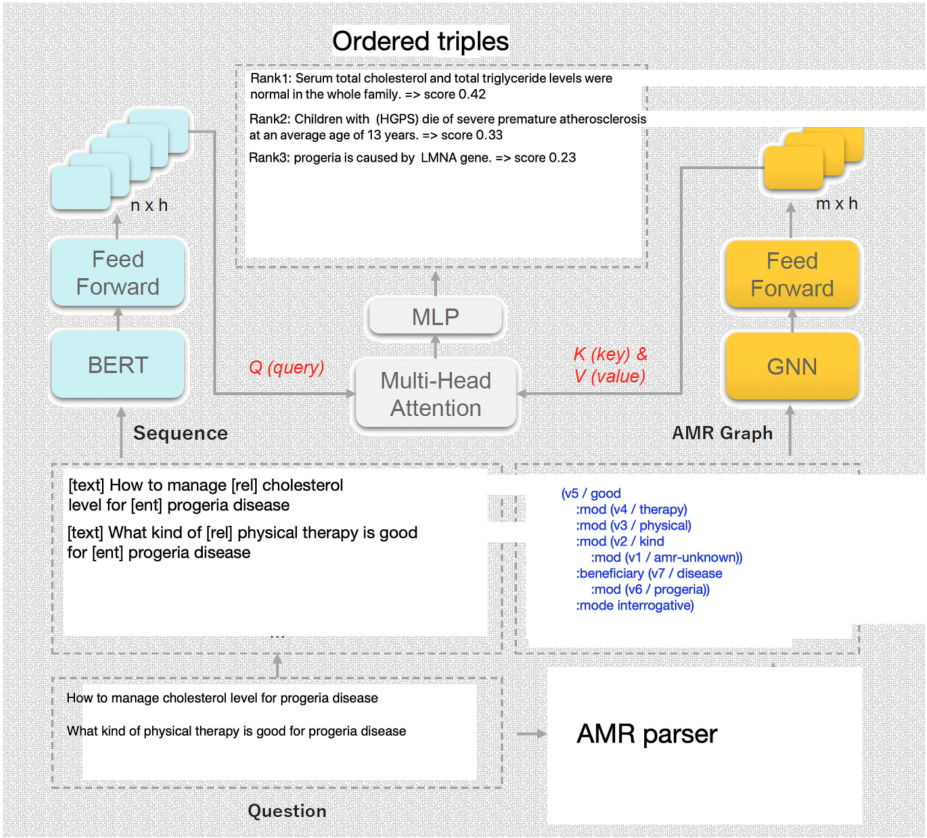

Figure 8. Obtaining knowledge from an AMR representation

6 Integrated system architecture

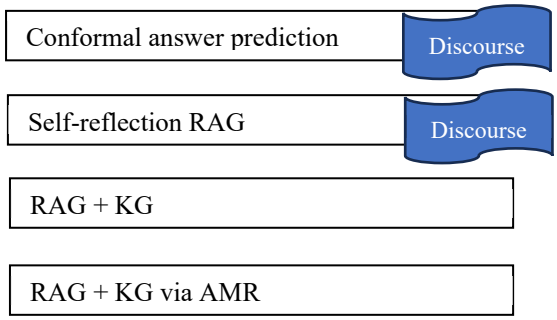

Figure 9. Four main components of the integrated RAG system

The integrated system architecture is shown in Fig. 9. Firstly, *Conformal prediction* is used to tackle hallucination and lower relevance answers. The *Self-reflection* RAG builds a retrieval scenario for each candidate in the answer set from the previous component. After that, for tail answers, KG is employed, if necessary, as determined by *Self-reflection.* Finally, long-tail answers, if determined by Self-reflection, are handled by RAG + KG via AMR.

7 Evaluation

We conducted a comprehensive evaluation of the contribution of each of the four components of our integrated system. For the first and second components, we also assessed the performance of their discourse analysis subsystems. To perform these assessments, we utilized GPT-4 alongside the embedding model E5 (Wang et al., 2022).

To evaluate retrieval efficacy, we employed the following key metrics:

(1) Mean Reciprocal Rank (MRR): This measures the average of the reciprocal ranks of the first correct answer in the result set, providing insight into the system's ability to prioritize relevant results.
(2) Recall@K: This metric assesses the probability that a relevant item appears within the top K retrieved results, indicating the system's ability to surface useful information.
(3) Normalized Discounted Cumulative Gain (NDCG@K): This metric evaluates the quality of the ranking by taking into account both the position of relevant items and their importance, offering a nuanced measure of retrieval precision.

To ensure the robustness of our findings, we compared our four-component system against three baselines:

(1) Stand-alone Large Language Model (GPT-4): This measures the performance of GPT-4 in isolation.
(2) Simple Retrieval-Augmented Generation (RAG) architecture: This serves as a baseline for comparison with more advanced architectures.
(3) Baseline from (Galitsky 2025): This provides a benchmark using the system optimized for medical discourse.

The evaluation was conducted on several key datasets:

(1) HotpotQA-Doc (Yang et al., 2018): A challenging dataset designed for multi-hop question answering based on two long documents. It tests the system's ability to handle complex reasoning tasks.
(2) ConditionalQA (Sun et al., 2022): This dataset serves as a benchmark for conditional question answering over long documents, pushing the system’s capacity for understanding dependencies across large amounts of text.
(3) Qasper (Dasigi et al., 2021): A dataset focusing on question answering from scientific papers, which evaluates the system's performance in extracting precise and contextually accurate information from dense, technical text.

The results of our evaluations are summarized in Table 1 (retrieval performance) and Table 2 (question-answering performance). Four runs were conducted for each evaluation setting. Across all metrics, our integrated system demonstrated consistent improvements over the baselines:

- MRR: Our system outperformed the averaged baseline by 12%, highlighting its enhanced ability to prioritize correct responses.
- Recall@K: Modest improvement in some cases, and loss for K = 1.
- NDCG@K: modest 8% improvement in for K = 2.
- BLEU Score: The system achieved an increase of 7% in BLEU score, further confirming its superior accuracy in question-answering tasks.
- ROUGE score: significant 24%.
- METEOR: significant 20%.

These results indicate that the four-component system, particularly when augmented with discourse-level analysis and advanced retrieval mechanisms, significantly improves both the retrieval and question-answering performance compared to simpler architectures. The integration of GPT-4 and the E5 embedding model ensures that the system delivers more relevant, accurate, and contextually appropriate answers, outperforming traditional baselines in complex, multi-hop reasoning tasks.

	MRR	**Recall@K**		**NDCG@K**	
		K = 1	**K = 2**	**K = 1**	**K = 2**
Stand-alone LLM	0.65	0.350	0.432	0.350	0.475
Simple RAG architecture	0.70	**0.450**	0.512	0.450	0.526
Baseline from (Galitsky 2025)	0.65	0.400	0.487	0.400	0.463
Four-component integrated system	**0.75**	0.400	**0.523**	**0.500**	**0.530**

Table 1. Accuracy of retrieval

	BLEU	**ROUGE**	**METEOR**
Stand-alone LLM	0.048	0.134	0.215
Simple RAG architecture	0.061	0.183	0.233
Baseline from (Galitsky 2025)	0.045	0.212	0.289
Four-component integrated system	**0.055**	**0.218**	**0.295**

Table 2. Accuracy of question answering

We now proceed to the assessment of each component out of four (Table 3), measuring an improvement compared to the averaged RAG baseline, the same as in Tables 1 and 2. We measure the boost of performance of the stand-alone first component, then the first and the second, then the first, the second and the third, etc. We use MRR for this ablation study.

	Averaged baseline per Table 1	**Conformal prediction**		**Self-reflection RAG**		**RAG + KG**	**RAG + KG via AMR**	**Total boost due to discourse**
			+ discourse		**+ discourse**			
HotpotQA-Doc	1	1.082	1.098	1.112	1.110	1.147	1.156	0.16
ConditionalQA	1	1.064	1.072	1.084	1.088	1.120	1.134	0.10
Qasper	1	1.051	1.050	1.059	1.064	1.093	1.121	0.03
PatientInfo (2024)	1	1.063	1.069	1.082	1.080	1.106	1.118	0.04

Table 3. Results of ablation / component contribution study

The impact of the discourse subsystems exhibits significant variability across different QA datasets, showing stronger variation compared to other accuracy metrics (Table 3). However, the overall contribution of these subsystems remains modest. Among the four components, the Self-reflection subsystem provides the least benefit to the model's performance, while the KG integration yields the most substantial improvement, enhancing both retrieval and response quality.

Considering the diverse metrics applied to evaluate our four-component architecture, we conclude that each component contributes a noticeable performance boost when compared to baseline RAG architectures. Despite varying levels of impact from each subsystem, the overall effectiveness of this multi-component architecture proves valuable in advancing the accuracy and robustness of the system.

8 Conclusions

The introduction of the four-component architecture presented in this paper has resulted in a significant performance boost, with improvements reaching up to 20%. This gain is attributed to the comprehensive integration of multiple system elements, which work together to enhance both retrieval and question-answering accuracy. Notably, the four-component system focuses on refining various stages of information processing, such as retrieval, discourse analysis, and the filtering of answers, which collectively contribute to the overall efficiency.

However, the contribution of the discourse-based subsystems is more modest in comparison, yielding improvements of up to 4%. While discourse analysis plays a crucial role in maintaining coherence and relevance within multi-hop question-answering tasks, its direct impact on retrieval performance remains relatively minor. This suggests that while discourse analysis enhances the system's overall flow and relevance, other components, such as retrieval and self-reflection mechanisms, play a more substantial role in driving the observed performance gains.

This assessment underscores the necessity of balancing multiple subsystems to achieve optimal performance, with discourse playing a supporting but essential role in ensuring the integrity of the system's responses.

In this paper, we explored multiple RAG architectures aimed at addressing key challenges in LLM-based question answering, specifically the issues of hallucination and lack of specific information. We began with conformal prediction, which enhances answer accuracy by maintaining a set of possible generations and selecting the most "average" one. We then introduced LLM self-reflection architectures that predict multi-hop question-answering sessions prior to engaging the retrieval system, improving the quality and relevance of generated answers. Furthermore, we proposed a mechanism for filtering answers based on stylistic appropriateness, ensuring that the generated responses align with the desired tone and format.

A central component in each of these architectures is discourse-level analysis, which provides a more robust framework for managing answer generation. Additionally, the integration of knowledge graphs (KG) and AMR-based knowledge graph construction strengthens the retrieval process by improving the relevance and contextual accuracy of the information retrieved. Our evaluations demonstrate the significant contributions of these mechanisms, particularly the discourse-based subsystems, to enhancing overall answer relevance and mitigating hallucination in LLM-based systems. This layered approach to RAG, incorporating self-reflection, discourse, and knowledge graph elements, offers a scalable solution to improving the quality of question-answering systems.

RAG combined with a knowledge graph is a practical implementation of the neuro-symbolic paradigm, blending neural networks (data-driven, statistical AI) and symbolic AI (logic-based, structured reasoning):

(1) Neural Component: The RAG model incorporates LLMs which use neural networks to generate text based on patterns in data. LLMs handle unstructured data (like natural language) and excel at understanding broad, context-driven information. When a question is posed, the RAG system uses vector-based retrieval techniques to find semantically similar chunks of text from a large corpus. This part represents the neural approach—flexible, probabilistic, and able to handle noisy or incomplete data.
(2) Symbolic Component: The knowledge graph represents the symbolic side. It structures data as nodes (entities) and edges (relationships), providing a clear, formal representation of knowledge. This allows for logical reasoning over the data, making it possible to retrieve precise and contextually related information based on the relationships between entities. For example, if a user asks a question about a specific historical event, the knowledge graph can retrieve related facts (entities) and their connections, ensuring accuracy and structured reasoning.

Knowledge graphs support multi-hop reasoning, meaning that the system can answer complex queries that require chaining together several facts or relationships. Neural models often struggle with multi-step logic, but the symbolic structure of the graph allows for deductive reasoning across entities, making it easier for the LLM to generate precise, well-founded responses.

References

[1] Abu-El-Haija S, Bryan Perozzi, Amol Kapoor, Nazanin Alipourfard, Kristina Lerman, Hrayr Harutyunyan, Greg Ver Steeg, Aram Galstyan (2019) MixHop: Higher-Order Graph Convolutional Architectures via Sparsified Neighborhood Mixing. arXiv:1905.00067v3

[2] Asai A, ZeqiuWu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. (2024) Self-RAG:Learning to retrieve, generate, and critique through self-reflection. In The Twelfth International Conference on Learning Representations.

[3] Asher N, Lascarides A (2003) Logics of conversation. Cambridge University Press, Cambridge UK

[4] Barzilay R and Mirella Lapata. 2008. Modeling Local Coherence: An Entity-Based Approach. Computational Linguistics, 34(1):1–34.

[5] Banarescu L, C. Bonial, S. Cai, M. Georgescu, K. Griffitt, U. Hermjakob, K. Knight, P. Koehn, M. Palmer, N. Schneider, Abstract Meaning Representation for sembanking, in: A. Pareja-Lora, M. Liakata, S. Dipper (Eds.), Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse, Association for Computational Linguistics, Sofia, Bulgaria, 2013, pp. 178–186.

[6] Benveniste D (2024) Augmenting LLMs with a Graph Database. https://newsletter.theaiedge.io/p/augmenting-llms-with-a-graph-database

[7] Dasigi P, Kyle Lo, Iz Beltagy, Arman Cohan, Noah A. Smith, and Matt Gardner. 2021. A dataset of information-seeking questions and answers anchored in research papers. In Proceedings of the 2021 Conference of the North American Paper of the Association for Computational Linguistics: Human Language Technologies, pages 4599–4610

[8] Fan, W., Wang, S., Huang, J., Chen, Z., Song, Y., Tang, W., Mao, H., Liu, H., Liu, X., Yin, D., & Li, Q. (2024). Graph Machine Learning in the Era of Large Language Models (LLMs). ArXiv, abs/2404.14928

[9] Feng Z, Xiaocheng Feng, Dezhi Zhao, Maojin Yang, and Bing Qin. 2023.Retrieval-generation synergy augmented large language models.arXiv, abs/2310.05149.

[10] Galitsky B, Dobrocsi G, de la Rosa JL, Kuznetsov SO (2011) Using generalization of syntactic parse trees for taxonomy capture on the web. ICCS:104–117

[11] Galitsky B, Ilvovsky D, Kuznetsov SO, Strok F (2013) Finding maximal common sub-parse thickets for multi-sentence search. IJCAI Workshop on Graphs and Knowledge Representation. LNAI,volume 8323, 39-57

[12] Galitsky B (2017) Discovering rhetoric agreement between a request and response. Dial Disc 8(2)

[13] Galitsky B (2019) Rhetorical agreement: maintaining cohesive conversations. In developing enterprise chatbots. Springer, Cham Switzerland, pp 327–363

[14] Galitsky B and Ilvovsky D (2019) Two Discourse Tree - Based Approaches to Indexing Answers. In Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019), pages 367–372, Varna, Bulgaria

[15] Galitsky B (2020) A Virtual Social Promotion Chatbot with Persuasion and Rhetorical Coordination

[16] Artificial Intelligence for Customer Relationship Management. Springer, Cham Switzerland. 129 -180

[17] Galitsky B and [Ilvovsku D (2022b) Building ontologies relying on communicative discourse trees. Artificial Intelligence for Healthcare Applications and Management. Elsevier.

[18] Galitsky B (2022) Relying on discourse analysis to answer complex questions by neural machine reading comprehension. US Patent App. 17/505,462

[19] Galitsky B (2023) Converting a document into a chatbot-accessible form via the use of communicative discourse trees. US Patent 11,615,145

[20] Galitsky B (2024) Relying on discourse trees to build ontologies. US Patent 11,914,961

[21] Gao T, Howard Yen, Jiatong Yu, and Danqi Chen. 2023a. Enabling large language models to generate text with citations. In Proceedings of the 2023 EMNLP, pages 6465–6488, Singapore. ACL

[22] He X, X. Bresson, T. Laurent, A. Perold, Y. LeCun, and B. Hooi (2023) Harnessing explanations: LLM-to-LM interpreter for enhanced text attributed graph representation learning," Oct. 2023.

[23] Huang W, Guancheng Zhou, Mirella Lapata, Pavlos Vougiouklis, Sebastien Montella, Jeff Z. Pan. Prompting Large Language Models with Knowledge Graphs for Question Answering Involving Long-tail Facts.

[24] Islam SB, Md Asib Rahman, K S M Tozammel Hossain, Enamul Hoque, Shafiq Joty, and Md Rizwan Parvez (2024) Open-RAG: Enhanced Retrieval Augmented Reasoning with Open-Source Large Language Models. EMNLP Findings.

[25] Izacard G, M. Caron, L. Hosseini, S. Riedel, P. Bojanowski, A. Joulin, E. Grave, Unsupervised dense information retrieval with contrastive learning, Trans. Mach. Learn. Res. 2022 (2022)

[26] Jascob B. AMRLib - A python library that makes AMR parsing, generation and visualization simple. https://github.com/bjascob/amrlib (2024)

[27] Jeong S, Jinheon Baek, Sukmin Cho, Sung Ju Hwang, and Jong C Park. 2024. Adaptive-rag: Learning to adapt retrieval-augmented large language models through question complexity. arXiv preprint arXiv:2403.14403.

[28] Jiang JJ, Conrath DW (1997) Semantic similarity based on corpus statistics and lexical taxonomy. In: Proceedings of the International Conference on Research in Computational Linguistics
[29] Joshi M, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. 2020. SpanBERT: Improving pre-training by representing and predicting spans. Transactions of the Association for Computational Linguistics, 8:64–77.
[30] Karpukhin V, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense Passage Retrieval for Open-Domain Question Answering. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 6769–6781, Online. Association for Computational Linguistics.
[31] Khashabi D, Sewon Min, Tushar Khot, Ashish Sabharwal, Oyvind Tafjord, Peter Clark, Hannaneh Hajishirzi (2020) UnifiedQA: Crossing Format Boundaries with a Single QA System. arXiv:2005.00700
[32] Komatsuzaki A, Joan Puigcerver, James Lee-Thorp, Carlos Riquelme Ruiz, Basil Mustafa, Joshua Ainslie, Yi Tay, Mostafa Dehghani, and Neil Houlsby. 2022. Sparse upcycling: Training mixture-of experts from dense checkpoints. arXiv preprint 2212.05055.
[33] Luo Y, J. Zhang, S. Fan, K. Yang, Y. Wu, M. Qiao, and Z. Nie (2023) BiomedGPT: Open multimodal generative pre-trained transformer for biomedicine," arXiv preprint arXiv:2308.09442
[34] Mallen A, A. Asai, V. Zhong, R. Das, D. Khashabi, H. Hajishirzi. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories, in: A. Rogers, J. Boyd-Graber, N. Okazaki (Eds.), Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Toronto, Canada, 2023, pp. 9802–9822. URL: https://aclanthology.org/2023.acl-long.546. doi:10.18653/v1/2023.acl-long.546.
[35] Mallen A, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. arXiv preprint arXiv:2212.10511.
[36] Naseem T, S. Ravishankar, N. Mihindukulasooriya, I. Abdelaziz, Y.-S. Lee, P. Kapanipathi, S. Roukos, A. Gliozzo, A. Gray (2021) A semantics-aware transformer model of relation linking for knowledge base question answering, in: C. Zong, F. Xia, W. Li, R. Navigli (Eds.), Proceeding ACL, pp. 256–262.
[37] Nogueira R and Kyunghyun Cho. 2020. Pas sage re-ranking with BERT. arXiv preprint arXiv:1901.04085.
[38] Palmer M, Daniel Gildea, and Paul Kingsbury. 2005. The Proposition Bank: An annotated corpus of semantic roles. Computational Linguistics, 31(1):71 p106.
[39] Parvez MR 2024. Evidence to generate (e2g): A single-agent two-step prompting for context grounded and retrieval augmented reasoning. arXiv preprint arXiv:2401.05787
[40] PatientInfo (2024) https://patient.info/forums/discuss
[41] Shi F, Xinyun Chen, Kanishka Misra, Nathan Scales, David Dohan, Ed H Chi, Nathanael Schärli, and Denny Zhou. 2023. Large language models canbe easily distracted by irrelevant context. In International Conference on Machine Learning, pages 31210–31227. PMLR.
[42] Sun H, William Cohen, and Ruslan Salakhutdinov. 2022. ConditionalQA: A complex reading comprehension dataset with conditional answers. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 3627–3637.
[43] Wang L, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2022. Text embeddings by weakly-supervised contrastive pre-training. arXiv preprint arXiv:2212.03533.
[44] Wang Y, Hanghang Tong, Ziye Zhu, and Yun Li. 2022. Nested named entity recognition: A survey. ACM Trans. Knowl. Discov. Data, 16(6).
[45] Wang Z, Jun Araki, Zhengbao Jiang, Md Rizwan Parvez, and Graham Neubig. 2023. Learning to filter context for retrieval-augmented generation. arXiv preprint arXiv:2311.08377.
[46] Wang Z, P. Ng, R. Nallapati, B. Xiang, Retrieval, re-ranking and multi-task learning for knowledge-base question answering, in: P. Merlo, J. Tiedemann, R. Tsarfaty (Eds.), Proceedings of the 16th Conference of the European Paper of the Association for Computational Linguistics: Main Volume, Association for Computational Linguistics, Online, 2021, pp. 347–357.
[47] Wu L, F. Petroni, M. Josifoski, S. Riedel, L. Zettlemoyer, Scalable zero-shot entity linking with dense entity retrieval, in: B. Webber, T. Cohn, Y. He, Y. Liu (Eds.), Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Association for Computational Linguistics, Online, 2020, pp. 6397–6407.
[48] Xu Z, Mark Jerome Cruz, Matthew Guevara, Tie Wang, Manasi Deshpande, Xiaofeng Wang, Zheng Li (2024) Retrieval-Augmented Generation with Knowledge Graphs for Customer Service Question Answering arxiv.org 2404.17723
[49] Yang Z, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A Dataset for Diverse, Explainable Multi-hop Question Answering. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 2369–2380, Brussels, Belgium. Association for Computational Linguistics.