

GETALP@AutoMin 2025: Leveraging RAG to Answer Questions based on Meeting Transcripts

Jeongwoo Kang, Markarit Vartampetian, Felix Herron, Yongxin Zhou,
Diandra Fabre, Gabriela Gonzalez-Saez

Univ. Grenoble Alpes, CNRS, Grenoble INP*, LIG, 38000 Grenoble, France

Correspondence: firstname.lastname@univ-grenoble-alpes.fr

Abstract

This paper documents GETALP’s submission to the Third Run of the Automatic Minuting Shared Task at SIGDial 2025. We participated in Task B: question-answering based on meeting transcripts. Our method is based on a retrieval augmented generation (RAG) system and Abstract Meaning representations (AMR). We propose three systems combining these two approaches. Our results show that incorporating AMR leads to high-quality responses for approximately 35% of the questions and provides notable improvements in answering questions that involve distinguishing between different participants (e.g., who questions).

1 Introduction

The 2025 edition of the Automatic Minuting (AutoMin) Shared Task introduces, for the first time, a question-answering challenge based on extensive meeting transcripts. This task (task B) involves generating accurate answers grounded in long conversational data.

To address this challenge, we propose a retrieval-augmented generation (RAG) approach enriched with Abstract Meaning Representation (AMR). Specifically, we leverage Information Retrieval (IR) techniques to identify and extract relevant passages from large transcripts based on a given question. Relevant passages are identified using both dense sentence embeddings and synthetic queries generated via the Doc2Query model (Nogueira et al., 2019). To represent the relationships described in the meeting, we include a Knowledge graph from an AMR of the retrieved sentences. These graphs are then translated into natural language descriptions. Finally, we utilize the capabilities of large language models (LLMs) to generate accurate responses using both the user question and the retrieved context. Our approach thus consists of

two main stages: (1) context construction, and (2) answer generation.

To analyze the impact of AMR-based context, we develop and evaluate three system variants:

1. **IR-only:** Using only the retrieved sentences (from sentence and Doc2Query representations).
2. **IR+AMR:** Using both the retrieved sentences and their AMR natural language descriptions.
3. **AMR-only:** Using only the AMR natural language descriptions of the retrieved sentences.

Finally, we evaluate each variant using the LLM-as-Judge metric (Kim et al., 2023), and we further conduct a manual evaluation to qualitatively assess the performance of the systems using the same scale.

2 Related Work

2.1 QA based on Meeting Transcripts

Previous work on question answering from meeting transcripts has explored both extractive and generative approaches. Apel et al. (2023) address real questions in meeting dialogues using an extractive model that jointly predicts answers and detects when no answer is present; the authors report moderate performance and note the difficulty of handling ambiguous or unanswered questions. Prasad et al. (2023) use models like Longformer and RoBERTa to extract multi-span answers from full or partial transcripts, but highlight that performance remains well below human level due to the complexity of long, dispersed dialogues. Pan et al. (2024) propose a two-step approach that first compresses transcripts using summarization, then applies QA models to the shortened text; results improve with compression, though performance depends heavily on the quality of the summaries. Golany et al. (2024) introduce a RAG pipeline

*Institute of Engineering Univ. Grenoble Alpes

where relevant segments are retrieved and used to generate answers; this approach improves handling of dispersed information but can be sensitive to retrieval errors.

2.2 RAG

Retrieval-Augmented Generation (RAG) (Lewis et al., 2020) is a framework designed to enhance the performance of LLMs by incorporating external knowledge through information retrieval. Instead of relying solely on the model’s parametric memory, RAG systems retrieve relevant documents from a collection —such as a database or the Internet— and use these documents as additional context to ground the model’s generation. This paradigm has proven effective for injecting up-to-date or domain-specific knowledge into LLMs and improving factual consistency in their outputs. In typical RAG pipelines, user queries are first augmented with retrieved passages, which are then fed into the LLM to generate responses that are both informative and grounded in external sources. A key advantage of RAG is its ability to mitigate the "lost in the middle" phenomenon (Liu et al., 2024) —where LLMs overlook relevant content located in the middle of long contexts — by ensuring that only the most relevant content is presented to the model. However, RAG systems also face notable challenges, notably in effectively managing long contexts and multi-document question answering.

2.3 Meaning representation for question answering

Previous work leveraged meaning representations for question answering tasks (Kapanipathi et al., 2021; Wang et al., 2023). Meaning representation represents meanings of a text in a structured form such as a graph, tree, or formal logic expressions. Incorporating structured information into QA systems provides a few advantages. First, meaning representation reduces ambiguity by explicitly encoding one plausible interpretation among many others. For example, in the following sentence “Kevin told Tom that he broke the glass,” it is unclear whether ‘he’ refers to Kevin or Tom. This ambiguity can be resolved by explicitly representing its meaning. Second, meaning representation provides information in canonical form regardless of the surface-level variations-especially syntactic ones. For example, “Mary bought the flower.” and “The flower was bought by Mary” are expressed identically in a meaning representation, thereby

reducing the search space in information retrieval systems. Because of these advantages, meaning representation is widely adopted in traditional QA systems.

Among many meaning representation frameworks, Abstract Meaning Representation (Banasescu et al., 2013, AMR) has gained popularity due to its broad semantic coverage and availability of annotated data. AMR encodes meaning of texts as a rooted, directed and acyclic graph (see Figure 1). In AMR graph, the graph nodes are either: Propbank predicate (*e.g.*, sell-01 in Figure 1) or English words (*e.g.*, man and flower in Figure 1) or AMR-specific entities (*e.g.*, date-entity and ordinal-entity). Edges between nodes are labeled to indicate semantic relations between the connected nodes. For example, in Figure 1, :ARG0 and :ARG1 respectively indicates that man is the agent of sell-01 and flower is the object of the same predicate. AMR graph can also be serialized in a textual format (see Figure 2), which is both human and machine-readable. AMR also uses variables to identify each node, *e.g.*, s, m and f in Figure 2. It can also be decomposed into a set of triples that represent the underlying graph structure.

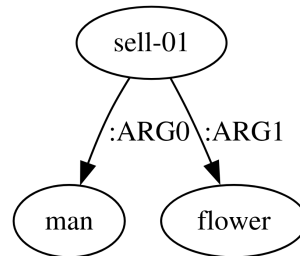


Figure 1: AMR graph for “A man breaks a window.”

```
(s / sell-01
  :ARG0 (m / man)
  :ARG1 (f / flower))
```

Figure 2: AMR graph linearized in text format.

With ongoing paradigm shift with large language model, however, the advantage of using AMR as an input for downstream tasks has been questioned. For example, Jin et al. (2024) argues that AMR, in its traditional graph form, is not optimal for LLMs, showing that incorporating it offers no improvement across five different NLP subtasks. On the contrary, Zhang et al. (2025) presents evidence supporting the usefulness of AMR when its format is adapted for LLMs. They argue that since LLM is

heavily trained with human languages, the structured format of AMR may not align well with their training. To address it, they propose *translating* the graph into a set of textual descriptions by converting each triple of an AMR graph into a natural language sentence. They show that these natural language descriptions of an AMR graph improve the performance of various downstream tasks, both in zero-shot and fine-tuning scenarios. Following their work, we incorporate AMR into QA systems while converting its structured form into natural language descriptions.

3 Methodology

3.1 Dataset Description

We use the two datasets provided for AutoMin TaskB: the ELITR Minuting Corpus and the ELITR-Bench Dataset.

ELITR Minuting Corpus (Nedoluzhko et al., 2022) consists of transcripts of meetings in Czech and English. On average, each transcript contains 7,000 words, involves 5.9 speakers, and includes 727 speaker turns. ELITR-Bench Dataset (Thonet et al., 2024)¹ contains questions to be answered using the English transcripts from the ELITR Minuting Corpus, splitted in two corpus: Dev and Test. In total, the *Dev* split comprises 10 meetings with 141 questions and is used for model validation prior to submission. The *Test* split, used for the final evaluation in the shared task, includes 8 meetings with 130 questions. While only the English transcripts are used for the task, the questions are in English (monolingual setting) or in Czech (cross-lingual setting).

3.2 RAG System Overview

Our system follows a two-stage RAG architecture (1) context construction, and (2) Answer generation. Given an input Question, denoted as Q , and a meeting transcript, denoted as DOC , the system produces an answer A that is based on the content of the transcript.

In the context construction stage, we apply information retrieval (IR) techniques to identify and extract relevant passages from the transcript DOC , based on the input question Q . We denote this context as the relevant context C_r . Using C_r we construct a second context using AMR which is

finally translated in natural language, we denote this context as C_{amr} .

In the Answer generation stage, an LLM reads the context C and the query Q to generate the final answer A following a specific prompting strategy.

Context C_r construction: IR To construct the relevant context C_r , we implement an information retrieval setup that combines two complementary strategies: dense sentence embeddings and Doc2Query-based document expansion. Each sentence in the transcript is not only encoded as a dense vector but also represented by a set of synthetic queries generated using the Doc2Query model (Nogueira et al., 2019). We index both the sentence embeddings and the synthetic queries using FAISS (Douze et al., 2025) for efficient similarity search.

At retrieval time, given a question Q , we retrieve (1) the most similar sentences based on the dense embedding similarity to Q , and (2) the sentences whose generated queries are most similar to Q in the Doc2Query index. The union of these results forms the initial set of relevant sentences. To improve coherence, we expand each selected sentence with its immediate context: one preceding and one following sentence from the transcript. We observed that, in some cases, the answer to the question was actually contained in the sentence closest to the most similar one. The final IR-based context C_r consists of this expanded set of relevant passages, ordered by their original position in the transcript to preserve the sequential structure of the transcript.

Context C_{amr} construction: AMR for QA To enrich the retrieved context and improve answer generation, we incorporate AMRs, derived from the selected sentences in C_r . Following the work of Zhang et al. (2025) as described in 2.3, we convert an AMR graph into its natural language descriptions. Specifically, we apply this conversion to the context retrieved in the information retrieval step (C_r). Since the code has yet to be provided by Zhang et al. (2025),² we use our own implementation for this process. We refer the readers to the original article for detailed description and examples.

Converting AMR into its natural language descriptions consists of 3 steps: 1) Extracting a set of triples from a given AMR graph 2) Translate each

¹<https://github.com/utter-project/ELITR-Bench/tree/main>

²At the moment of writing, June 2025.

triple into a sentence 3) Polish each sentence using LLM. For the first step, we used library PENMAN (Goodman, 2020). The second step requires pre-defined rules to translate each semantic role into a sentence, *e.g.*, (John, :ARG0, rob-01) → ‘John is the doer of rob-01 (to engage in or commit robbery)’. This may produce an unnatural text that needs to be polished for natural effect. This step is done in the third step using LLM. Following the original work, we provide some examples for the prompt to polish the text. As a result, for example, ‘John is the doer of rob-01 (to engage in or commit robbery)’ is polished as ‘John robs something.’

The natural descriptions of AMR graphs form the C_{amr} context, which can be provided either alone or alongside with the original sentences depending on our system variant. This is further detailed in the next section.

Answer A generation: Prompting LLM We use a large language model (LLM) as the backbone of the answer generation component. Given the constructed context C , which may include the IR-based context C_r , the AMR-derived context C_{amr} , or both, and the question Q , the LLM generates the final answer A .

We experiment with three variants of the input context provided to the LLM:

1. **IR-only:** Using only the retrieved sentences based on sentence and Doc2Query representations (C_r), along with the question Q .
2. **IR+AMR:** Using both the retrieved sentences (C_r) and their AMR-based natural language descriptions (C_{amr}), along with the question Q .
3. **AMR-only:** Using only the AMR-based natural language descriptions of the retrieved sentences (C_{amr}), along with the question Q .

4 Experiments

4.1 Models

We implement our RAG pipeline using the following components :

Context Construction: IR For the sentence-level representation in the IR module, we use the all-MiniLM-L6-v2 sentence embedding model³.

³<https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2>

For Doc2Query, we use the DocTTTTQuery model trained on the MS-MARCO dataset⁴.

Context Construction: AMR For AMR-to-text conversion, we use the meta-llama/Llama-3.1-8B-Instruct model⁵, prompted with the following instruction:

You are an AI language assistant. Your job is to improve and rewrite a list of sub-sentences (input_sub_sentences) so they flow naturally and resemble fluent, natural language. Use the input_original_sentence as context to guide your rewrites. Follow the format and style shown in the examples. Only output the final polished sentences. Do not include any explanations.

Answer Generation We use the same meta-llama/Llama-3.1-8B-Instruct model to generate answers, with the following instruction:

You are an AI assistant that answers questions using retrieved meeting information. Provide only the most relevant 1-2 sentence answer extracted directly from the content. Follow these rules: Be extremely concise - just the core fact, Use exact terms/phrases from the retrieved content, and never add analysis, disclaimers or "Based on...".

From English to Czech We control the output language by adding the instruction “Answer in Czech.” to the prompt when needed.

4.2 Evaluation

For the evaluation of Task B, predictions are evaluated by the LLM-as-a-judge metric, which uses large language models as automated judges to assess the quality of responses. As used in (Thonet et al., 2025; Zheng et al., 2023; Chiang and Lee, 2023), these models will compare the system-generated answers with the human-crafted gold reference answer for each given query. In this experiment, Prometheus model was used as implemented by the authors of (Kim et al., 2023). Prometheus is a 13B open-source language model fine-tuned to serve as an evaluator capable of assessing long-form responses based on user-provided rubrics and reference answers. We follow the Prometheus scale of 0 to 5, where 0 is when an answer is not generated in the intended language, and 5 is when the response to evaluate is essentially equivalent to the reference answer. As the final evaluation of

⁴<https://huggingface.co/castorini/doc2query-t5-base-msmarco>

⁵<https://huggingface.co/meta-llama/Llama-3.1-8B-Instruct>

Model	Mean \pm 1 std
GETALP@AutoMin	5.15 \pm 3.73
GETALP@AutoMin_amr	4.97 \pm 3.77
GETALP@AutoMin_amr_only	4.31 \pm 3.52

Table 1: Mean and standard deviation for LLM-as-judge evaluation on Czech Answers.

the task is in range 0 to 10, we rescale our scores accordingly.

5 Results

In this section, we propose an evaluation of the model based on LLM-as-judges for the Czech dataset only, and both LLM-as-judges and human evaluation for the English dataset. We evaluate the significant difference between the different experiments using a t-test.

Czech results Table 1 shows the mean and standard deviation from the scores as provided by LLM-as-judges. Figure 3 displays violin plots of the score distribution. We decided to remove the 0 score, as it was also triggered in cases where the reference answer was in English (e.g., [ORGANIZATION1]).

No significant difference was observed between the results obtained for each of our three proposed architectures.

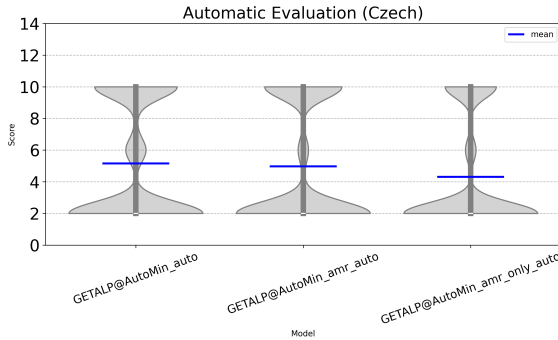


Figure 3: Automatic evaluation using LLM-as-judges. Scores are between 0 and 10. Violin plot with mean distribution as blue line. *** for $p \leq 0.005$

English results We used different human annotators to manually evaluate the performance of each of our three configurations. Each annotator evaluated only a part of the dataset, and there was no cross-over between annotators. Table 2 shows the mean and standard deviation of the scores provided by both LLM and Human evaluators. Human scores are higher than automatic LLM-as-judges

Model	Mean \pm 1 std
LLM-as-Judge	
GETALP@AutoMin	4.09 \pm 3.16
GETALP@AutoMin_amr	3.35 \pm 2.54
GETALP@AutoMin_amr_only	2.46 \pm 1.75

Human evaluation	
GETALP@AutoMin	5.65 \pm 3.06
GETALP@AutoMin_amr	5.55 \pm 2.95
GETALP@AutoMin_amr_only	3.94 \pm 2.69

Table 2: Mean and standard deviation for LLM-as-judge and Human evaluation on English Answers.

scores. We can observe from Figure 4 and Figure 5 that the evaluation is consistent between humans and LLMs. In both cases, GETALP@Automin and GETALP@AutoMin_amr obtain higher scores than GETALP@AutoMin_amr_only, and no significant difference is observed between GETALP@Automin and GETALP@AutoMin_amr. When comparing both automatic and manual scores for the two best configurations, as displayed in Figure 6, we could not identify a model that outperforms the other.

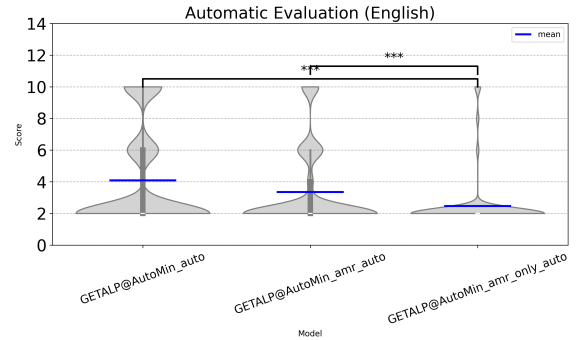


Figure 4: Automatic evaluation using LLM-as-judges. Scores are between 2 and 10. Violin plot with distribution mean as blue line. *** for $p \leq 0.005$

Ground-truth answers provided by the dataset are not always complete sentences, but are often sentence fragments or short pieces of information, as shown in Table 4 and Table 3. However, given the text generation capabilities of LLMs, we would expect a correct answer to be a full sentence conveying the correct information. Out of 130 questions, 46 received a human evaluation score of 8 or 10 for GETALP@AutoMin_amr, and 49 out of 130 for GETALP@AutoMin. The two systems obtained the same score for 91 of the 130 questions, while in 18 questions the AMR-based solution performed better than IR-only. Interestingly, half of these 18

Ground Truth	Ours
Who is leaving the project?	
[PERSON10]	[PERSON10] will be leaving.
What is the application deadline for the PhD program?	
End of the year	The PhD application deadline is the end of the year.
Who had to pause the meeting for a few minutes?	
[PERSON2]	PERSON2 had to pause the meeting for a few minutes.

Table 3: Examples where LLM-as-judge gave a score of 2 while evaluators gave a score of either 8 or 10, for both GETALP@Automin and GETALP@Automin_amr models. The answer displayed comes from one of the two models.

Ground Truth	Ours
What is [PERSON4] focusing on now?	
Multi-source machine translation	[PERSON4] is putting together and also working on the censorship component, which is one part of the pipeline after the [PROJECT4].
What is the current delay of the language id system?	
3 seconds	The current delay of the language ID system is not explicitly stated.
How did [PERSON7] qualify the experience of the latest recording session assessment?	
It was qualified as "disastrous".	Unfortunately, the provided information does not contain a direct quote from [PERSON7] regarding the experience of the latest recording session assessment.

Table 4: Examples where both LLM-as-judge and evaluators gave a score 2, for both GETALP@Automin and GETALP@Automin_amr models. The answer displayed comes from one of the two models.

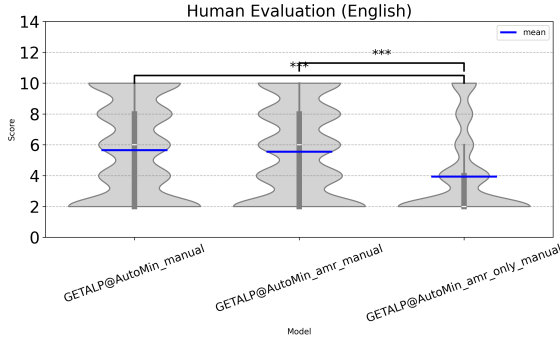


Figure 5: Human evaluation. Scores are between 2 and 10. Violin plot with distribution mean as blue line. *** for $p \leq 0.005$

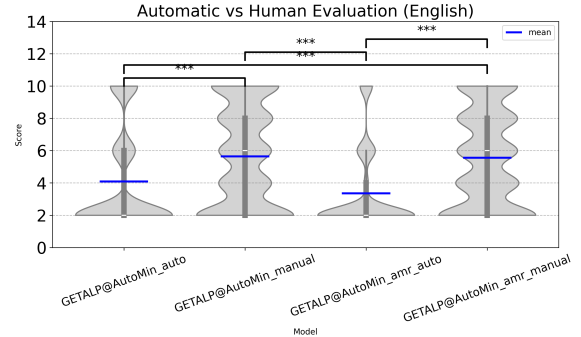


Figure 6: Human versus automatic evaluation. Scores are between 2 and 10. Violin plot with distribution mean as blue line. *** for $p \leq 0.005$

questions correspond to WHO questions. Among the 45 WHO questions in total, AMR achieved the same or a better score in 39 of them.

6 Conclusion

Our participation in the AutoMin 2025 Shared Task focused on developing RAG system for question answering over long meeting transcripts. To address the challenges of this task, we combined dense retrieval with Doc2Query-based document expansion and enriched the retrieved content using AMR. We

explored three variants of our system: using only the retrieved passages, combining them with their AMR-based natural language descriptions, and using only the AMR descriptions.

Our results suggest that AMR contexts can improve the quality of generated answers, particularly for questions involving entity resolution or semantic roles, such as identifying the responsible person for a task or determining who is experiencing an issue (e.g., "Who is experiencing disk space issues?"). Future work includes refining the AMR-

to-text generation process, better integrating AMR into context construction, and selectively applying AMR in question types where structured semantic information offers the most benefit.

Limitations

Our approach relies on a large language model (Llama 3.1 8B) for both AMR-to-text generation and final answer generation. This significantly increases computational demands and limits the feasibility of our system in resource-constrained environments. To carry out our experiments, we required high-performance GPUs, including an NVIDIA RTX A6000 and NVIDIA H100. Furthermore, although we prompt the model to produce answers in Czech, many of the underlying components, such as sentence embeddings and the Doc2Query model, are primarily trained on English data. This can result in reduced answer quality in non-English outputs and potential inconsistencies in multilingual behavior.

Acknowledgments

This work is being carried out as part of the AugmentIA Chair and supported by the Grenoble INP Foundation, thanks to sponsorship from the Artelia Group. This chair also receives state funding managed by the French National Research Agency under France 2030, reference ANR-23-IACL-0006 (MIAI Cluster). In addition, this work is supported by the CREMA project (Coreference REsolution into MACHine translation) funded by the French National Research Agency (ANR), contract number ANR-21-CE23-0021-01.

References

- Reut Apel, Tom Braude, Amir Kantor, and Eyal Kolman. 2023. [Meeqa: Natural questions in meeting transcripts](#). *Preprint*, arXiv:2305.08502.
- 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, Sofia, Bulgaria. Association for Computational Linguistics.
- Cheng-Han Chiang and Hung-yi Lee. 2023. [Can large language models be an alternative to human evaluations?](#) In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15607–15631, Toronto, Canada. Association for Computational Linguistics.
- Matthijs Douze, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvasy, Pierre-Emmanuel Mazaré, Maria Lomeli, Lucas Hosseini, and Hervé Jégou. 2025. [The faiss library](#). *Preprint*, arXiv:2401.08281.
- Lotem Golany, Filippo Galgani, Maya Mamo, Nimrod Parasol, Omer Vandsburger, Nadav Bar, and Ido Dagan. 2024. [Efficient data generation for source-grounded information-seeking dialogs: A use case for meeting transcripts](#). In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 1908–1925, Miami, Florida, USA. Association for Computational Linguistics.
- 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.
- Zhijing Jin, Yuen Chen, Fernando Gonzalez Adauto, 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](#). In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 3781–3798, Mexico City, Mexico. Association for Computational Linguistics.
- Pavan Kapanipathi, Ibrahim Abdelaziz, Srinivas Ravishankar, Salim Roukos, Alexander Gray, Ramón Fernandez Astudillo, Maria Chang, Cristina Cornelio, Saswati Dana, Achille Fokoue, Dinesh Garg, Alfio Gliozzo, Sairam Gurajada, Hima Karanam, Naweel Khan, Dinesh Khandelwal, Young-Suk Lee, Yunyao Li, Francois Luus, and 11 others. 2021. [Leveraging Abstract Meaning Representation for knowledge base question answering](#). In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3884–3894, Online. Association for Computational Linguistics.
- Seungone Kim, Jamin Shin, Yejin Cho, Joel Jang, Shayne Longpre, Hwaran Lee, Sangdoo Yun, Seongjin Shin, Sungdong Kim, James Thorne, and 1 others. 2023. Prometheus: Inducing fine-grained evaluation capability in language models. In *The Twelfth International Conference on Learning Representations*.
- Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, and 1 others. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459–9474.

- Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024. [Lost in the middle: How language models use long contexts](#). *Transactions of the Association for Computational Linguistics*, 12:157–173.
- Anna Nedoluzhko, Muskaan Singh, Marie Hledřková, Tirthankar Ghosal, and Ondřej Bojar. 2022. [ELITR minuting corpus: A novel dataset for automatic minuting from multi-party meetings in English and Czech](#). In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 3174–3182, Marseille, France. European Language Resources Association.
- Rodrigo Nogueira, Wei Yang, Jimmy Lin, and Kyunghyun Cho. 2019. Document expansion by query prediction. *arXiv preprint arXiv:1904.08375*.
- Zhuoshi Pan, Qianhui Wu, Huiqiang Jiang, Menglin Xia, Xufang Luo, Jue Zhang, Qingwei Lin, Victor Rühle, Yuqing Yang, Chin-Yew Lin, H. Vicky Zhao, Lili Qiu, and Dongmei Zhang. 2024. [LLMLingua-2: Data distillation for efficient and faithful task-agnostic prompt compression](#). In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 963–981, Bangkok, Thailand. Association for Computational Linguistics.
- Archiki Prasad, Trung Bui, Seunghyun Yoon, Hanieh Deilamsalehy, Franck Dernoncourt, and Mohit Bansal. 2023. [MeetingQA: Extractive question-answering on meeting transcripts](#). In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15000–15025, Toronto, Canada. Association for Computational Linguistics.
- Thibaut Thonet, Laurent Besacier, and Jos Rozen. 2025. [ELITR-bench: A meeting assistant benchmark for long-context language models](#). In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 407–428, Abu Dhabi, UAE. Association for Computational Linguistics.
- Thibaut Thonet, Jos Rozen, and Laurent Besacier. 2024. Elitr-bench: A meeting assistant benchmark for long-context language models. *arXiv preprint arXiv:2403.20262*.
- Cunxiang Wang, Zhikun Xu, Qipeng Guo, Xiangkun Hu, Xuefeng Bai, Zheng Zhang, and Yue Zhang. 2023. [Exploiting Abstract Meaning Representation for open-domain question answering](#). In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 2083–2096, Toronto, Canada. Association for Computational Linguistics.
- Jiahuan Zhang, Tianheng Wang, Hanqing Wu, Ziyi Huang, Yulong Wu, Dongbai Chen, Linfeng Song, Yue Zhang, Guozheng Rao, and Kaicheng Yu. 2025. [Sr-llm: Rethinking the structured representation in large language model](#). *Preprint*, arXiv:2502.14352.
- Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhaghao Wu, Yonghao Zhuang, Zi Lin,
- Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. In *Proceedings of the 37th International Conference on Neural Information Processing Systems, NIPS ’23*, Red Hook, NY, USA. Curran Associates Inc.