Integrating Large Language Models with Graph-based Reasoning for Conversational Question Answering

Parag Jain **Mirella Lapata**
Institute for Language, Cognition and Computation
School of Informatics, University of Edinburgh
parag.jain@ed.ac.uk mlap@inf.ed.ac.uk

Abstract

We focus on a conversational question answering task which combines the challenges of understanding questions in context and reasoning over evidence gathered from heterogeneous sources like text, knowledge graphs, tables, and infoboxes. Our method utilizes a graph structured representation to aggregate information about a question and its context (i.e., the conversation so far and evidence retrieved to find an answer), while also harnessing the reasoning and text generation capabilities of large language models (LLMs). Graph embeddings are directly injected into the LLM, bypassing the token embedding layers, and learned end-to-end by minimizing cross-entropy. Our model maintains a memory module to track and update past evidence, thus influencing the graph's structure, as the conversation evolves. Experimental results on the ConvMix benchmark (Christmann et al., 2022a) show that graph embeddings enhance the LLM's ability to reason, while the memory module provides robustness against noise and retrieval errors.

1 Introduction

Conversational question answering is an information seeking task where users engage in interactive conversations with AI systems (Choi et al., 2018; Reddy et al., 2019; Dalton et al., 2022). Unlike traditional question answering applications (Rajpurkar et al., 2016), conversational systems are expected to track the context of a conversation, i.e., remember previous questions and answers to provide relevant responses in an ongoing dialogue. The majority of prior work has studied different instantiations of conversational question answering, based on the simplifying assumption that answers can be found in a *single* information source. Examples include querying knowledge graphs such as Wikidata (Perez-Beltrachini et al., 2023; Christmann et al., 2022a; Saha et al., 2018), identifying answer spans in Wikipedia articles (Reddy et al., 2019; Choi et al., 2018), and searching for answers in table cells (Iyyer et al., 2017).

In this paper we focus on conversational question answering over *multiple* and *heterogeneous* information sources. Figure 1 shows an example interaction from ConvMix (Christmann et al., 2022b), a recently curated dataset, which combines the challenges of understanding questions in context, and retrieving their answers from multiple sources. As can be seen, answers are located in knowledge base triples (response to Q1), infoboxes (responses to Q4 and Q5), and tables (responses to Q2 and Q3). It is also possible for an answer to be found in different sources which may in turn disagree. Moreover, the interaction in Figure 1 displays the hallmarks of naturalistic dialogue. The second question (*Fact Rank?*) can only be interpreted by taking into account the topic of the conversation (i.e., the album *Kid A*) mentioned in the previous utterance. Follow-on questions are short and may seem ungrammatical taken out of context. As the conversation unfolds, the topic shifts from the album *Kid A* to the *Rolling Stone* magazine; Q4 in Figure 1 has no dependencies on previous utterances and a hypothetical system would have to recognize that a new topic is being introduced.

We propose a modeling approach to conversational question answering which integrates large language models (LLMs) with graph-based reasoning. The core idea is to represent information about a question and its context — such as the conversation so far and sources retrieved to find an answer — through a dynamically generated graph and size varies with each utterance. Our method utilizes a graph structured representation (Gori et al., 2005; Scarselli et al., 2009) to aggregate information (and resolve conflicts) from multiple sources, while also harnessing the reasoning and text generation capabilties of LLMs. Our graph network is efficiently trained using gradients from the LLM. Graph embeddings are directly injected into the

Q1: What is the release date of album Kid A?
A1: 2 October 2000

Q2: Fact Rank?
A2: 7

Q3: Ranking on Rolling Stone in 2009?
A3: 1

Q4: Editor?
A4: Noah Shachtman

Q5: Category?
A5: Popular culture

Query at Q3: What is the release date of album Kid A? 2 October 2000 Fact Rank? 7 Ranking on Rolling Stone in 2009?

Example retrieved evidence:

- Rolling Stone, Editor, Noah Shachtman (Infobox)
- Rolling Stone, inception, 1967 (Triple)
- Kid A, publication, 2 October 2000 (Triple)
- Kid A, Publication Rolling Stone, Country US, Accolade The 100 Best Albums of Decade, Year 2009, Rank 1 (Table)
- Kid A, Kid A is the fourth studio album by the English rock band Radio head, released on 2 October 2000 by Parlophone (Text)
- Kid A, Rolling Stone described the Kid A tour as "a revelation, exposing rock and roll humanity" in the songs. (Text)

Figure 1: Example interaction (left) from the ConvMix development set (Christmann et al., 2022b) and relevant evidence at query Q3 (right). Utterances Q1–Q3 explore the topic of album *Kid A*. Q4 transitions to the topic of *Rolling Stone* magazine. The evidence is retrieved from diverse sources highlighted in red. Wikipedia text and tables are prepended with their respective article title. Known entities are shown in blue. Underlined entities are identified through string matching.

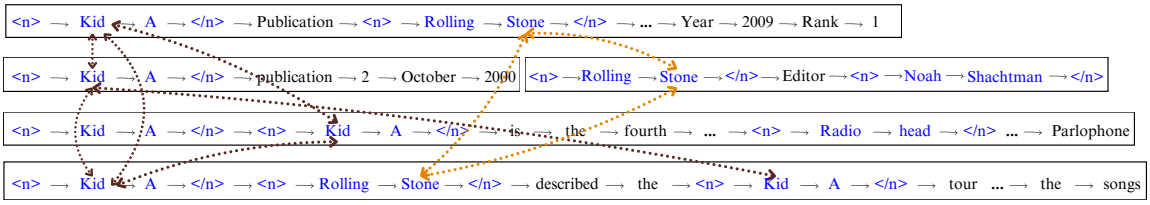


Figure 2: Graph for retrieved evidence (subset) from Figure 1. Tokens within each instance create local subgraphs in the form of a linear chain. Local subgraphs are connected through common entities (within <n> – </n>) to build a global graph. Same color highlights connections between similar entities (some edges are omitted for clarity).

LLM, bypassing the token embedding layers, and learned end-to-end by minimizing cross-entropy loss. To manage topic shifts and keep track of the conversation flow, we introduce a *memory* module that stores evidence used to answer previous questions, thus allowing to re-use past information for answering future questions. Our contributions are:

- A method to aggregate evidence from multiple sources into a dynamic graph representation for conversational question answering.
- We efficiently integrate the evidence-based graph with LLMs for end-to-end training.
- We keep track of past evidence in a memory module which is updated as the conversation evolves and influences the graph structure and its representation.
- Extensive experiments on the ConvMix dataset (Christmann et al., 2022b), demonstrate that graph structure enhances the LLM's ability to reason over multiple sources, while the memory module affords robustness to noise and retrieval errors.

2 Related Work

Conversational Question Answering Most previous work on conversational question answering operates over a *single* infromation source such as a knowledge graph, text passage, or table (Choi et al., 2018; Reddy et al., 2019; Perez-Beltrachini et al., 2023; Iyyer et al., 2017). Existing models tend to be specialized, catering to isolated modalities (e.g., text or tables), while a few approaches adopt graph-based representations to organize the conversation and available information (Shen et al., 2019; Jain and Lapata, 2023; Kacupaj et al., 2021; Mueller et al., 2019). A notable exception are Christmann et al. (2023) who propose an end-to-end model for *multiple* information sources. Specifically, their method constructs a heterogeneous graph based on evidence retrieved from tables, infoboxes, text snippets, and Wikidata triples. This graph is iteratively pruned at inference time to a smaller subgraph containing the answer (i.e., an entity node) to the question.

Our work also integrates information from multiple sources into a graph. However, we do not model question answering as a classification task,

but instead propose a generative model. We leverage graph representations and the reasoning capabilities of language models, without relying on specialized inference procedures.

LLMs with Graphs A common approach to encoding graph structure for LLMs involves describing the graph in natural language so that it resembles text (Ye et al., 2023; Wang et al., 2024). There is no agreed consensus on how to convert graphs to text, and most methods rely on hand-crafted rules. Previous efforts have shown it is challenging for LLMs to reason over graph representations (Fatemi et al., 2024; Huang et al., 2024), even when explicit prompts are given that describe the structure of the graph in natural language (Huang et al., 2024). Performance tends to be brittle and task dependent (Wang et al., 2024; Fatemi et al., 2024).

Our work proposes a parameter-efficient method for learning *task-specific* graph representations. It is closest to Perozzi et al. (2024), who use graph embeddings as soft-prompts to represent structured data for LLMs. In a similar vein, Chai et al. (2023) use prefix-tuning to integrate graph embeddings with LLM attention layers. Their approach shows promising results on small graphs with a few nodes ($\sim$20) and limited variability. It also relies on the architecture of the LLM and may not seamlessly integrate with other models, e.g., Mixture-of-Experts (MoE; Shazeer et al. 2017; Jacobs et al. 1991).

Retrieval-augmented Generation Our work integrates LLMs with graph structural information based on evidence retrieved from the Wikidata knowledge graph (Vrandečić and Krötzsch, 2014), Wikipedia text, tables, and infoboxes. Although we do not focus on retrieval as such, it plays a key role in identifying information for building the graph. Our approach can thus be viewed as a variant of retrieval augmented generation (RAG), since it conditions generation on freshly retrieved evidence based on user queries (Izacard et al., 2024; Khandelwal et al., 2020; Guu et al., 2020).

3 Overview

We assume a conversational question answering setting (Christmann et al., 2022b) that requires resoning over Wikipedia facts attested in multiple sources such as text, tables, infoboxes, and the Wikidata knowledge graph (KG). Given interaction I, our task is to answer question q_t at turn t, taking into account retrieved evidence r_t and previous turns $\mathrm{I}[:t-1]$ which consist of questions and their answers $\langle q_t, a_t \rangle$ (see Figure 1). To accomodate information from the conversation so far, we concatenate question q_t at turn t with previous question-answer pairs, i.e., $Q_t = [q_1, a_1 \ldots q_{t-1}, a_{t-1}, q_t]$, and use this to retrieve evidence.

As depicted in Figure 3, we adopt a modular approach. Given query Q_t, we retrieve and rank relevant evidence (Section 4.1). We next organize retrieved information into a graph (Section 4.3) and learn graph embeddings using Graph Attention Networks (GAT; Velickovic et al. 2018; Brody et al. 2022). Finally, graph embeddings are injected in a LLM by skipping the token embeddings layer (Section 4.5). Unlike Christmann et al. (2023) who *extract* answers from retrieved evidence, we *generate* them. Our model $\mathcal{M}$ is thus formulated as:

$$a_t = \mathcal{M}\left(\mathrm{I}[:t-1], q_t, r_t; \Theta\right) \quad (1)$$

where q_t is the current question, r_t is the graph representing retrieved evidence, $\mathrm{I}[:t-1]$ are previous turns, and Θ the parameters of our model which are fine-tuned on task-specific data (Section 4.6).

4 Model

4.1 Evidence Retrieval

We adopt the retrieval pipeline outlined in Christmann et al. (2022b). As mentioned earlier, information is obtained from Wikipedia pages and the Wikidata KG using a query based on the current question concatenated with previous question-answer pairs. Retrieval takes place in two stages. Initially, evidence is retrieved from the Wikidata KG, and then followed by retrieval from Wikipedia.

We extract Wikidata triples (see ② in Figure 3) using CLOCQ (Christmann et al., 2022a), a retrieval engine specifically tailored to question answering over knowledge bases. It preprocesses the knowledge graph in a memory efficient manner and returns the top-k triples based on query terms. Figure 3, shows a subset of relevant triples retrieved for Q3 along with the KG entities $E_{\mathcal{E}}$.

We next obtain evidence pertaining to additional Wikipedia sources by retrieving articles corresponding to the entities in $E_{\mathcal{E}}$. These pages are subsequently processed to extract text, tables, and infoboxes (see ③ in Figure 3). Tables are linearized by individually transforming each row into text and concatenating it with corresponding column headers. Infoboxes are linearized in a similar fashion

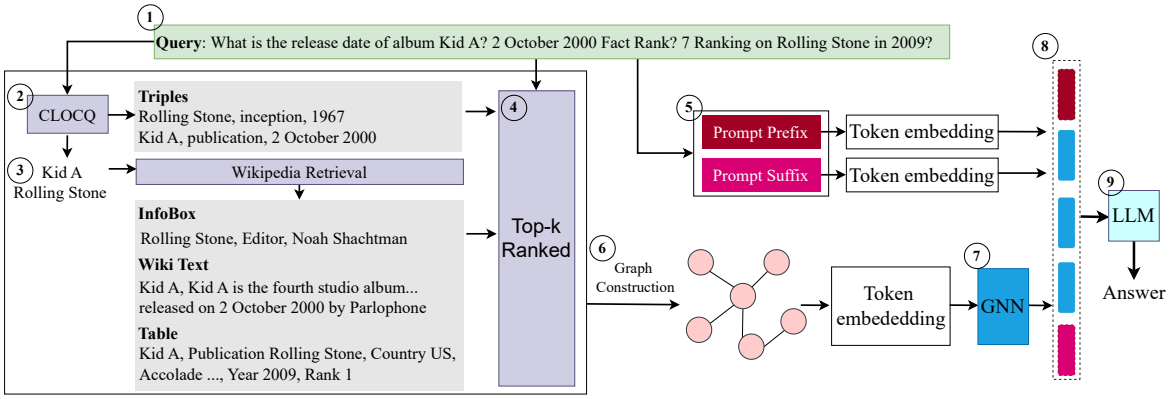


Figure 3: Sketch of proposed architecture. ① shows query Q3 from the interaction in Figure 1. ② shows KG triples retrieved with CLOCQ and their entities (③). Wikipedia articles for ③ are parsed to extract sentences, infoboxes and tables. In ④, retrieved evidence is ranked based on the current query using BM25. ⑤ creates an instruction prompt based on the input query (see Appendix A for the prompt template). In ⑥, a graph is constructed based on top ranked instances. ⑦ depicts the learned graph neural network. Graph node embeddings are initialized using LLM token embeddings that are separate from the base model. ⑧ shows the final embeddings which are passed to the LLM and are obtained by concatenating prompt (prefix, suffix) and graph embeddings (shown in different colors). ⑨ is the LLM without the token embedding layer.

by concatenating key-value pairs with header information (if available). KB triples are linearized by a simple concatenation of individual elements. Wikipedia text is split into sentences, each of which serves as a separate piece of evidence.

The evidence collected at this stage can be extensive, potentially comprising of several thousand instances, which would in turn lead to a very large graph (see Section 4.3). To manage this, we employ BM25 (Robertson and Zaragoza, 2009) to rank the evidence against the query and retain only the best scoring instances (see ④ in Figure 3). Let E_t denote the set of top-k retrieved instances at turn t.

4.2 Evidence Memory

By design, we retrieve new evidence at every turn t, which may suggest that every question introduces a new topic. However, a well-known property of conversational dialogue is *topic inertia* (Chai and Jin, 2004), i.e., users tend to explore the same topic for a while before switching to a new topic (see the interaction in Figure 1). We propose to keep track of past topics through a memory module which stores previously retrieved pieces of evidence to be re-utilized and re-ranked against Q_t. Specifically, at each turn t we define evidence memory M_t as,

$$M_t = \oplus \{E_j \mid j \in [1 \dots t-1]\} \qquad (2)$$

where $\oplus$ denotes concatenation. We replace a proportion (e.g., one third) of low-ranked instances from E_t with the top-ranking ones from M_t. We employ the Sentence-BERT model (Reimers and Gurevych, 2019) to re-rank the evidence stored in M_t, using Q_t as a query.

4.3 Graph Construction

Retrieved information is organized into a graph (see ⑥, Figure 3) by first converting individual pieces of evidence into a linear chain. *Local* subgraphs are then merged into a *global* graph by linking common entities between them. Figure 2 shows example graphs with *local* and *global* connections.

To construct a *local* graph, evidence from different sources is linearized (as discussed in Section 4.1) and tokenized using a base LLM tokenizer. Tokens within each instance are treated as graph nodes connected in a linear chain. In other words, evidence w with tokens $w_1 \dots w_{|w|}$ is represented by local sub-graph $w_1 \rightarrow w_2 \rightarrow \dots \rightarrow w_{|w|}$.

Connecting different pieces of evidence together is critical for enabling more global reasoning. We create a *global* graph by linking similar entities across *local* subgraphs. In this context, entities are KG items but also text spans in Wikipedia text, infoboxes, and tables gathered during retrieval. We identify entity spans by performing string matching against KG entities. In Figure 2, such entities are encircled by <n> *node* </n> tags. Finally, entity spans referring to same entity are linked, thus creating a more globally connected graph.

4.4 Graph Encoder

Our model generates an answer at each turn t given query Q_t and graph $\mathcal{G}_t$ representing relevant evidence (see Figure 3). More formally, $\mathcal{G}_t = (\mathcal{V}, \mathcal{E})$ is a directed graph with nodes $\mathcal{V} = \{v_1, v_2, \ldots, v_n\}$ and edges $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$.

We do not learn graph node embeddings from scratch. Instead, we initialize them using token embeddings from a large language model (see ⑦, Figure 3). This step is crucial for achieving feature alignment between the evidence graph and the downstream LLM. Generally, integrating LLMs with information from a different modality necessitates aligning features between them. For example, vision-language models like BLIP-2 (Li et al., 2023) and LLaVA (Liu et al., 2023) perform feature alignment by heavily pretraining a network whose goal is to act as a bridge between a frozen image encoder and a frozen LLM. This approach requires large amounts of pretraining data (as well as computational resources) which are not readily available for our task. We found that simply initializing graph node embeddings with token embeddings from a base LLM is effective and crucial for achieving good performance.

Let $\{x_i \mid i \in [1, n]\}$ denote the set of initial node embeddings. We learn graph structure representations with the Graph Attention Network (GAT; Velickovic et al. 2018; Brody et al. 2022), a neural network architecture designed for handling graph-structured data. It is computationally efficient, it requires less memory and storage compared to other deep learning models, and is applicable to inductive problems. GAT uses the attention mechanism to weigh the importance of neighboring nodes when aggregating information in a graph. Attention between two nodes is calculated as:

$$\alpha_{ij} = \frac{\exp\big(\psi\big(x_i, x_j\big)\big)}{\sum_{k \in \mathcal{N}_i} \exp\big(\psi\big(x_i, x_k\big)\big)} \tag{3}$$

where $\mathcal{N}_i = \{v_j \in \mathcal{V} \mid \big(j, i\big) \in \mathcal{E}\}$ are the neighbors of node v_i, and α_{ij} is the attention score between node embeddings x_i and x_j. Following Brody et al. (2022), we compute the scoring function ψ as:

$$\psi\big(x_i, x_j\big) = a^T \,\mathrm{LeakyReLU}\big(W \cdot [x_i \oplus x_j]\big) \tag{4}$$

where $\cdot^T$ represents transposition and $\oplus$ is the concatenation operation. Attention coefficients corresponding to each node i are then used to compute a linear combination of the features corresponding to neighboring nodes as:

$$x_i = \sigma\left(\sum_{j \in \mathcal{N}_i} \alpha_{ij} W x_j\right) \tag{5}$$

4.5 Integration with LLMs

The LLM takes as input a composite embdedding consisting of the graph embeddings discussed above, and embeddings corresponing to a prompt prefix $\mathsf{P}_{\text{prefix}}$, and a prompt suffix $\mathsf{P}_{\text{suffix}}$ (see ⑤ in Figure 3). $\mathsf{P}_{\text{prefix}}$ is an initial instruction prompt and $\mathsf{P}_{\text{suffix}}$ represents the conversational query at turn t to be answered. See Appendix A(Figure 6) for an example prompt. More formally, LLM input embeddings are obtained as:

$$\mathsf{H} = \mathsf{H}_{\text{prefix}} \oplus \mathsf{H}_g \oplus \mathsf{H}_{\text{suffix}} \tag{6}$$

where H_g is the list of embeddings of all graph nodes and $\mathsf{H}_{\text{prefix}}$ is the text embedding of $\mathsf{P}_{\text{prefix}}$:

$$\mathsf{H}_{\text{prefix}} = \mathrm{Embed}(\mathrm{Tok}(\mathsf{P}_{\text{prefix}})) \tag{7}$$

where Tok and Embed are the base LLM tokenizer and embedding layer, respectively. $\mathsf{P}_{\text{suffix}}$ is encoded in a similar manner using Equation (7) to obtain $\mathsf{H}_{\text{suffix}}$. We use the embeddings obtained with Equation (6) as the initial token embeddings for the pretrained LLM.

4.6 Training

Our model is trained end-to-end by optimizing cross-entropy loss. For all variants (with and without graph structure), the loss is calculated on completion tokens only, i.e., prompt tokens do not observe any loss. This is similar to setting the prompt loss weight to 0 (Wang et al., 2023).

Given training instance $\langle \mathrm{I}[:t-1], q_t, r_t; \Theta \rangle$, and sequence of gold output tokens $\langle a_t^1, a_t^2, \ldots, a_t^{|a_t|} \rangle$, we minimize token-level cross-entropy as:

$$\mathcal{L}\left(\hat{a}_t^i\right) = -\log p\left(a_t^i \mid \mathrm{I}[:t-1], q_t, r_t; \Theta\right) \tag{8}$$

where $\hat{a}_t^i$ denotes the predicted output token at decoder step i. We use a mixed approach for training the whole network. Our graph network is trained from scratch, however, the base LLM is updated using LoRA (Hu et al., 2022) in a parameter efficient manner. We perform inference based on the conversation context (i.e., $\mathrm{I}[:t-1]$) and current query q_t.

ConvMix-5T	
Entities covered	5,418
Long-tail entities	2,511
conversations	2,800
Number of turns	5
Split ratio	60:20:20
ConvMix-10T test set	
Conversations	200
Number of turns	10
Domains: Books, Movies, Music, TV series, Soccer	
Answer Source: Text, Tables, Infobox Wikidata	

Table 1: ConvMix dataset statistics. Long tail entities are those attested in less than 50 KG facts.

5 Experimental Setup

We use Mistral-7B-Instruct-v0.2 (Jiang et al., 2023) as our base model, given its good performance across complex reasoning tasks, and wider context window of 32K tokens. Recall that we retrieve and encode a large number of instances as evidence for a question. Our implementation predominantly relies on PyTorch (Paszke et al., 2019). We adapt the Mistral implementation available at the HuggingFace Transformers library (Wolf et al., 2020). For developing the graph neural network, we utilize PyTorch Geometric (PyG; Fey and Lenssen 2019). We use Hugging Face's TRL (Transformer Reinforcement Learning) library (von Werra et al., 2020) for fine-tuning model without graph. Additional training parameters and prompts can be found in Appendices B and A, respectively.

5.1 Dataset

We evaluate our work on ConvMix (Christmann et al., 2022b), a conversational question answering dataset that requires reasoning over heterogeneous sources, specifically Wikipedia text, infoboxes, tables, and the Wikidata KG. Aside from reasoning, the conversational nature of ConvMix requires handling discourse phenomena, such as coreference, ellipsis, and topic-shift (Sun and Chai, 2007; Jain and Lapata, 2021). Table 1 summarizes various dataset statistics. As can be seen (first block), the main dataset (CovMix-5T) contains 2,800 conversations, each with five turns (i.e., question-answer pairs), split into training, development, and test set. In addition, ConvMix-10T is a *separate* test set used to measure generalization on longer interactions. It contains 200 conversations, each 10 turns long (see last block in Table 1). We follow the splits provided in Christmann et al. (2022b) and report results on both test sets combined.

5.2 Evaluation Metrics

Our model generates answers which may be valid but not identical to the gold standard (e.g., *United States*, *United States of America*, and *USA* are all paraphrases of the same concept). When there is no exact match, we follow previous work (Christmann et al., 2022b) and try to normalize the answer to its canonical form. We use the Levenshtein distance (Levenshtein, 1965) to measure the similarity of the generated answer with entities in our retrieved evidence set. The entity with the smallest distance is used as the answer in such cases. We report H@1 (i.e., precision at 1) and H@5 (i.e., whether an answer match is found within the top 5 matching entities).

6 Results

Our experiments were designed to assess whether graph structure enhances LLM performance for our conversational question-answering task. Our results are summarized in Table 2.

We evaluate our approach against Mistral-7B variants without graph structure. Specifically, we compare against (a) Mistral-7B zero-shot prompted with top-k retrieved instances and the conversational history, i.e., the current query concatenated with previous QA pairs (see Appendix A for the prompt); and (b) Mistral-7B fine-tuned on the ConvMix training set using LoRA (Mistral-7B + FT) and top-k retrieved instances. We present three variants of our model, fine-tined with graph embeddings (Mistral-7B + Graph) and additionally with a memory management component (+Memory, +Rand Memory).

We also compare with several state-of-the-art systems built on top of T5 (Raffel et al., 2020). T5-FiD (Christmann et al., 2022b) is a fusion-in-decoder model which acts as a "generative reader" and is trained on (top-k) retrieved instances and gold answers. Specifically, query-evidence pairs are encoded independently, and passed on to the decoder to generate an answer. We also report results with a T5-based model (T5-FiD + Question rewriting) which rewrites the question based on the conversational history context (Raposo et al., 2022; Elgohary et al., 2019) and a related approach (T5-FiD + Question resolution) which performs query resolution, i.e., by appending relevant terms from previous question-answer pairs to the current

Models	H@1	H@5
Mistral-7B zero-shot	0.292	0.346
Mistral-7B + FT	0.350	0.400
Mistral-7B + Graph	0.425	0.459
Mistral-7B + Graph + Memory	**0.445**	**0.512**
Mistral-7B + Graph + Rand Memory	0.425	0.461
T5-FiD	0.300	0.350
T5-FiD + Question resolution	0.282	0.297
T5-FiD + Question rewriting	0.271	0.285
Convinse T5-FiD	0.342	0.386
EXPLAIGNN	0.406	0.561

Table 2: Model performance on the ConvMix dataset (results are averaged for ConvMix-5T and convMix-10T test sets). H@1 represents precision at 1 and H@5 represents a match at 5. A fine-tuned Mistral-7B with graph embeddings and a memory module performs best.

question (Voskarides et al., 2020).[1]

Finally, although not directly comparable, we report the performance of EXPLAIGNN (Christmann et al., 2023) and Convinse T5-FiD (Christmann et al., 2022b). EXPLAIGNN is a classification model that identifies entity nodes in a graph as answer predictions. It learns a task specific structured representation optimized for better retrieval and query understanding. The learned representation is used to train a classification model based on graph neural networks tying both of them together. Convinse T5-FiD is similar in that it also learns a task-specific structured representation for retrieval and query understanding, without, however, creating a graph.

All models in Table 2 use the same retrieval engine (i.e., CLOCQ; Christmann et al. 2022a) which allows us to focus on architectural differences and compare models on equal footing.

Integrating LLMs with graph-based reasoning boosts conversational QA performance. As shown in Table 2, Mistral-7B + Graph is superior to a plain fine-tuned version of Mistral-7B (+ FT) by a large margin. This suggests that organizing and representing retrieved evidence as a graph improves reasoning compared to processing pieces of evidence independently. Perhaps unsurprisingly, fine-tuning generally improves Mistral's performance on the conversational QA task over a zero-shot model. This is due to an improved understanding of task requirements, like regular shift in focus and answer format. For example, the model learns to avoid verbosity in answers and respond using dataset-specific conventions such as spelling out the month in dates (e.g., *2 October 2002* instead of *2/10/2002*). The performance of the T5-FiD systems is comparable to zero-shot Mistral-7B. In general, we observe that performance improvements are not simply due to increased model size. Rather, it is important to model the conversational nature of the task and interpret the retrieved information more globally.

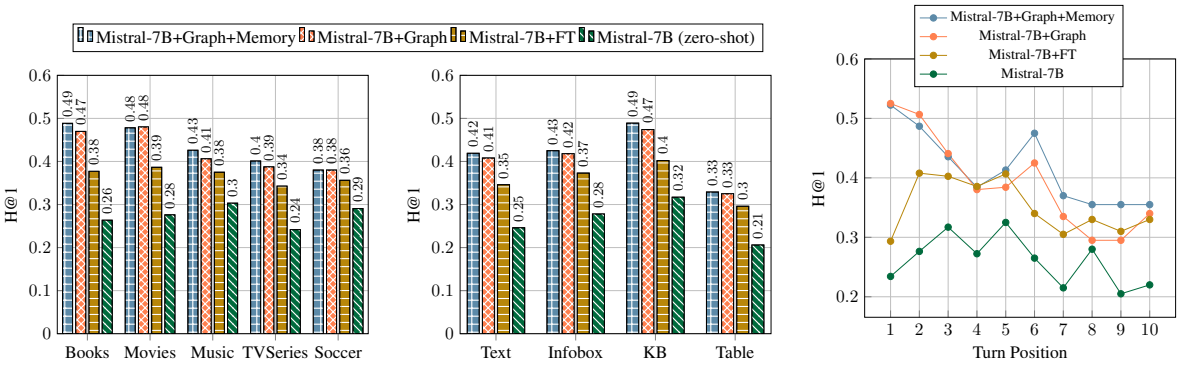
Adding a memory module improves QA precision. Table 2 shows that results further improve when a memory module is added to our model (+Graph +Memory). Recall that previously retrieved instances are kept in memory and re-reranked against the current query. To further assess the usefulness of re-ranking, we conducted a controlled experiment where evidence was selected randomly from the memory. We observe that random selection (+Rand Memory) amounts to not having a memory component at all.

It is challenging to provide accurate answers to questions that require numerical responses. Figure 4a shows model performance broken down by question domain. Overall, we observe similar trends across domains, with `TV Series` and `Soccer` being most challenging. Performance for these domains decreases by ∼10 percentage points, e.g., in comparison to `Books`. To uncover the reason for this gap, we further investigate whether there is an effect of answer type. We automatically annotate[2] the ConvMix development set with the following answer categories: strings, dates, and numbers. The results in Table 3 (top) show average H@1 stratified by different answer types.

We observe that questions with numeric answers are harder compared to other categories. There are several reasons for this, including variability in numerical reasoning performance due to the choice of numeric data tokenization by the base model (Singh and Strouse, 2024; Sun et al., 2023). As well as the effect of pre-training data on the output predictions and their probability (McCoy et al., 2023). Table 3b (bottom) reveals that the proportion of instances with numeric answers is highest for the `TV Series` and `Soccer` domains, thus explaining why performance drops for these domains.

[1]All FiD models are based on T5-base (Christmann et al., 2022b).

[2]We use `regex` and `python-dateutil` to automatically categorize the answers.



(a) Model performance across domains (ConvMix test set).

(b) Model performance across modalities (ConvMix test set).

(c) Model performance at different turn positions (ConvMix test set).

Figure 4: Analysis experiments for different model variants based on Mistral-7B prompted in a zero-shot setting, fine-tuned on ConMix without graph embeddings (+FT), with graph embeddings (+Graph), and with a memory module (+Graph +Memory). Performance degrades with numbers, tables, and later conversation turns.

(a)

Answer Type	Date	String	Number
H@1	0.50	0.45	0.14

(b)

Domain	Books	Movies	Music	TV series	Soccer
% Number	3.9	2.1	5.0	10.0	7.9

Table 3: Model performance (Mistral-7B + Graph + Memory) across answer types (top) and proportion of numeric answers per domain (ConvMix dev set).

It is challenging to extract accurate information from tables. Figure 4b, shows how performance varies depending on the source of the answer. Across models, we observe that performance deteriorates when the answers are located in tables. On the contrary, performance is generally better when answers are found in the knowledge graph. We believe this performance gap is due to how tabular information is linearized. In contrast to the knowledge graph from which facts can be easily extracted, Wikipedia tables often have complex hierarchical structure (Parikh et al., 2020) making it challenging to achieve clean and robust linearization (Alonso et al., 2023).

It is more difficult to answer questions occurring later in the conversation. In Figure 4c we examine how performance varies with conversation length. Ideally, a model should be able to answer questions irrespective of where these occur (e.g., beginning or end). As mentioned in Section 5.1, ConvMix contains conversations with a maximum length of 10 turns. The results in Figure 4c show a general decrease in performance as the dialogue progresses. Initial questions tend to be more complex while follow-on questions often extend or elaborate upon the initial topic (Chai and Jin, 2004; Jain and Lapata, 2021). Our results show that graph enhanced models generally outperform LLM variants which do not organize the retrieved information in any way. Furthermore, we observe that having a memory (of previously retrieved instances) is particularly helpful in longer interactions. Keeping track of past evidence helps ameliorate retrieval errors which might erroneously steer the model towards new topics. Aside from contextual factors, the quality of retrieval largely influences model precision, as approximately half of the answers cannot be found even at the beginning of the dialogue (see turn 1 in Figure 4c).

7 Conclusion

In this paper we propose a method to aggregate evidence from multiple sources into a dynamic graph representation for conversational question answering. We demonstrate how this graph can be efficiently integrated with large language models (LLMs) for end-to-end training, enhancing the model's ability to handle evolving conversational contexts. Our approach maintains a memory module to track and update past evidence, thus influencing the graph's structure and representation, as the conversation evolves. Experiments on the ConvMix dataset show that the graph enhances the LLM's ability to reason over multiple modalities, while the memory module provides robustness against noise and retrieval errors. In the future, we would like to improve information retrieval for our task, through using pretrained embeddings for better entity linking. We could also adopt a structured memory module for more complex reasoning.

8 Limitations

Our experiments are limited to one dataset (i.e., ConvMix) and one language, namely English. It would be interesting to see if our findings generalize to other datasets which are conversational in nature but do not target our specific question answering task. For example, SPICE (Perez-Beltrachini et al., 2023) is a recently released conversational semantic parsing dataset where utterances are translated into executable semantic parses (in this case SPARQL queries). It would also be interesting to examine how our model handles languages other than English, however, we are not aware of any multilingual or cross-lingual datasets for conversational question answering.

In this work, we do not study the effect of various prompting techniques on our task. In experiments, we found Mistral-7B's performance superior to Llama2-7B (Touvron et al., 2023), however, we did not perform an in-depth study on prompts and models. Measuring the effect of these factors on our task and model performance is non-trivial and a topic for future work.

References

Iñigo Alonso, Eneko Agirre, and Mirella Lapata. 2023. Pixt3: Pixel-based table to text generation. *arXiv preprint arXiv:2311.09808*.

Shaked Brody, Uri Alon, and Eran Yahav. 2022. How attentive are graph attention networks? In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Joyce Y. Chai and Rong Jin. 2004. Discourse structure for context question answering. In *Proceedings of the Workshop on Pragmatics of Question Answering at HLT-NAACL 2004*, pages 23–30, Boston, Massachusetts, USA. Association for Computational Linguistics.

Ziwei Chai, Tianjie Zhang, Liang Wu, Kaiqiao Han, Xiaohai Hu, Xuanwen Huang, and Yang Yang. 2023. Graphllm: Boosting graph reasoning ability of large language model. *arXiv preprint arXiv:2310.05845*.

Eunsol Choi, He He, Mohit Iyyer, Mark Yatskar, Wentau Yih, Yejin Choi, Percy Liang, and Luke Zettlemoyer. 2018. QuAC: Question answering in context. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2174–2184, Brussels, Belgium. Association for Computational Linguistics.

Philipp Christmann, Rishiraj Saha Roy, and Gerhard Weikum. 2022a. Beyond ned: Fast and effective search space reduction for complex question answering over knowledge bases. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*, WSDM '22. ACM.

Philipp Christmann, Rishiraj Saha Roy, and Gerhard Weikum. 2022b. Conversational question answering on heterogeneous sources. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '22, page 144–154, New York, NY, USA. Association for Computing Machinery.

Philipp Christmann, Rishiraj Saha Roy, and Gerhard Weikum. 2023. Explainable conversational question answering over heterogeneous sources via iterative graph neural networks. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '23, page 643–653, New York, NY, USA. Association for Computing Machinery.

Jeffrey Dalton, Sophie Fischer, Paul Owoicho, Filip Radlinski, Federico Rossetto, Johanne R. Trippas, and Hamed Zamani. 2022. Conversational information seeking: Theory and application. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '22, page 3455–3458, New York, NY, USA. Association for Computing Machinery.

Ahmed Elgohary, Denis Peskov, and Jordan Boyd-Graber. 2019. Can you unpack that? learning to rewrite questions-in-context. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5918–5924, Hong Kong, China. Association for Computational Linguistics.

Bahare Fatemi, Jonathan Halcrow, and Bryan Perozzi. 2024. Talk like a graph: Encoding graphs for large language models. In *The Twelfth International Conference on Learning Representations*.

Matthias Fey and Jan E. Lenssen. 2019. Fast graph representation learning with PyTorch Geometric. In *ICLR Workshop on Representation Learning on Graphs and Manifolds*.

M. Gori, G. Monfardini, and F. Scarselli. 2005. A new model for learning in graph domains. In *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.*, volume 2, pages 729–734 vol. 2.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. 2020. Retrieval augmented language model pre-training. In *International conference on machine learning*, pages 3929–3938. PMLR.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Jin Huang, Xingjian Zhang, Qiaozhu Mei, and Jiaqi Ma. 2024. Can llms effectively leverage graph structural information through prompts, and why? *Preprint*, arXiv:2309.16595.

Mohit Iyyer, Wen-tau Yih, and Ming-Wei Chang. 2017. Search-based neural structured learning for sequential question answering. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1821–1831, Vancouver, Canada. Association for Computational Linguistics.

Gautier Izacard, Patrick Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. 2024. Atlas: few-shot learning with retrieval augmented language models. *J. Mach. Learn. Res.*, 24(1).

Robert A Jacobs, Michael I Jordan, Steven J Nowlan, and Geoffrey E Hinton. 1991. Adaptive mixtures of local experts. *Neural computation*, 3(1):79–87.

Parag Jain and Mirella Lapata. 2021. Memory-Based Semantic Parsing. *Transactions of the Association for Computational Linguistics*, 9:1197–1212.

Parag Jain and Mirella Lapata. 2023. Conversational semantic parsing using dynamic context graphs. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 8667–8679, Singapore. Association for Computational Linguistics.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Endri Kacupaj, Joan Plepi, Kuldeep Singh, Harsh Thakkar, Jens Lehmann, and Maria Maleshkova. 2021. Conversational question answering over knowledge graphs with transformer and graph attention networks. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 850–862, Online. Association for Computational Linguistics.

Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. 2020. Generalization through memorization: Nearest neighbor language models. In *International Conference on Learning Representations*.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Vladimir I. Levenshtein. 1965. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet physics. Doklady*, 10:707–710.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pages 19730–19742. PMLR.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023. Visual instruction tuning. In *Advances in Neural Information Processing Systems*, volume 36, pages 34892–34916. Curran Associates, Inc.

R. Thomas McCoy, Shunyu Yao, Dan Friedman, Matthew Hardy, and Thomas L. Griffiths. 2023. Embers of autoregression: Understanding large language models through the problem they are trained to solve. *Preprint*, arXiv:2309.13638.

Thomas Mueller, Francesco Piccinno, Peter Shaw, Massimo Nicosia, and Yasemin Altun. 2019. Answering conversational questions on structured data without logical forms. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5902–5910, Hong Kong, China. Association for Computational Linguistics.

Ankur Parikh, Xuezhi Wang, Sebastian Gehrmann, Manaal Faruqui, Bhuwan Dhingra, Diyi Yang, and Dipanjan Das. 2020. ToTTo: A controlled table-to-text generation dataset. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1173–1186, Online. Association for Computational Linguistics.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32.

Laura Perez-Beltrachini, Parag Jain, Emilio Monti, and Mirella Lapata. 2023. Semantic parsing for conversational question answering over knowledge graphs. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2507–2522, Dubrovnik, Croatia. Association for Computational Linguistics.

Bryan Perozzi, Bahare Fatemi, Dustin Zelle, Anton Tsitsulin, Mehran Kazemi, Rami Al-Rfou, and Jonathan Halcrow. 2024. Let your graph do the talking: Encoding structured data for llms. *arXiv preprint arXiv:2402.05862*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

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.

Gonçalo Raposo, Rui Ribeiro, Bruno Martins, and Luísa Coheur. 2022. Question rewriting? assessing its importance for conversational question answering. In *European Conference on Information Retrieval*, pages 199–206. Springer.

Siva Reddy, Danqi Chen, and Christopher D. Manning. 2019. CoQA: A conversational question answering challenge. *Transactions of the Association for Computational Linguistics*, 7:249–266.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Stephen Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: Bm25 and beyond. *Found. Trends Inf. Retr.*, 3(4):333–389.

Amrita Saha, Vardaan Pahuja, Mitesh M. Khapra, Karthik Sankaranarayanan, and Sarath Chandar. 2018. Complex sequential question answering: towards learning to converse over linked question answer pairs with a knowledge graph. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence*, AAAI'18/IAAI'18/EAAI'18. AAAI Press.

Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. 2009. The graph neural network model. *IEEE Transactions on Neural Networks*, 20(1):61–80.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. 2017. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538*.

Tao Shen, Xiubo Geng, Tao Qin, Daya Guo, Duyu Tang, Nan Duan, Guodong Long, and Daxin Jiang. 2019. Multi-task learning for conversational question answering over a large-scale knowledge base. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2442–2451, Hong Kong, China. Association for Computational Linguistics.

Aaditya K. Singh and DJ Strouse. 2024. Tokenization counts: the impact of tokenization on arithmetic in frontier llms. *Preprint*, arXiv:2402.14903.

Kaiser Sun, Peng Qi, Yuhao Zhang, Lan Liu, William Wang, and Zhiheng Huang. 2023. Tokenization consistency matters for generative models on extractive NLP tasks. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 13300–13310, Singapore. Association for Computational Linguistics.

Mingyu Sun and Joyce Y Chai. 2007. Discourse processing for context question answering based on linguistic knowledge. *Knowledge-Based Systems*, 20(6):511–526.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph attention networks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Leandro von Werra, Younes Belkada, Lewis Tunstall, Edward Beeching, Tristan Thrush, Nathan Lambert, and Shengyi Huang. 2020. Trl: Transformer reinforcement learning. `https://github.com/huggingface/trl`.

Nikos Voskarides, Dan Li, Pengjie Ren, Evangelos Kanoulas, and Maarten de Rijke. 2020. Query resolution for conversational search with limited supervision. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, pages 921–930.

Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: a free collaborative knowledgebase. *Commun. ACM*, 57(10):78–85.

Heng Wang, Shangbin Feng, Tianxing He, Zhaoxuan Tan, Xiaochuang Han, and Yulia Tsvetkov. 2024. Can language models solve graph problems in natural language? *Advances in Neural Information Processing Systems*, 36.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. Self-instruct: Aligning language models with self-generated instructions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13484–13508, Toronto, Canada. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame,

Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Ruosong Ye, Caiqi Zhang, Runhui Wang, Shuyuan Xu, and Yongfeng Zhang. 2023. Natural language is all a graph needs. *arXiv preprint arXiv:2308.07134*.

A Prompt Description

Figure 5 shows an example prompt for the Mistral-7B model without graph embeddings (see Mistral-7B zero-shot in Table 2). The prompt includes a sequence of retrieved and ranked pieces of evidence, each encapsulated within `<evidence>-</evidence>` tags. We represent the past interaction $\mathrm{I}[:t-1]$ as a series of question and answer pairs. The same prompt is used for fine-tuning (see Mistral-7B + FT in Table 2) with the subsequent response as the gold output tokens (see Section 4.6 for details).

Figure 6 shows an example prompt for the graph-based model (all model variants with +Graph in Table 2). The prompt consists of three parts, the initial instructions which we refer to as $\mathsf{P}_{\text{prefix}}$, a sequence of graph node embeddings represented as `graph_node_embedding`, and the conversational query which we denote as $\mathsf{P}_{\text{suffix}}$.

B Training Details

Table 4 list the hyper-parameters employed to train our model. Implementation details are discussed in Section 5. During the fine-tuning of the base language model, only the query, key, and value projection parameters are updated.

Parameter	**Value**
Graph layers	2
Graph heads	2
Lora rank	128
Lora α	32
Lora dropout	0.05
GAT Dropout	0.5
Optimizer	Adam (Kingma and Ba, 2015)
Learning rate	5e-5
Batch size	1
Gradient accumulation	4

Table 4: Hyperparameter values used for our model.

Prompt: Mistral-7B zero shot and fine-tuned without graph embeddings

[INST]

You are a helpful assistant. Using the following facts:

<evidence>Kid A, publication, 2 October 2000</evidence>

<evidence>Rolling Stone, Editor, Noah Shachtman</evidence>

<evidence>Rolling Stone, Catgories, Popular culture</evidence>

<evidence>Publication Fact, Country UK, Accolade The 100 Best Albums of the 2000s, Year 2010, Rank 7</evidence>

<evidence>Publication Rolling Stone, Country US, Accolade The 100 Best Albums of the decade, Year 2009, Rank 1</evidence>

<evidence>Rolling Stone was founded in San Francisco in 1967 by Jann Wenner and Ralph J. Gleason.</evidence>

Answer the following conversational query as a simple key fact without description:

[/INST]

Question: What is the release date of album Kid A?
Answer: 2 October 2000
Question: Fact Rank?
Answer: 7
Question: Ranking on Rolling Stone in 2009?
Answer:

Figure 5: Example prompt for models which do not employ graph embeddings. Only a few relevant pieces of evidence are shown, for the sake of brevity.

Prompt: Mistral-7B fine-tuned with graph embeddings

[INST]

You are a helpful assistant. Using the following facts:

`[graph_node_embedding_1, graph_node_embedding_2, ... , graph_node_embedding_n]`

Answer the following conversational query as a simple key fact without description:

[/INST]

Question: What is the release date of album Kid A?
Answer: 2 October 2000
Question: Fact Rank?
Answer: 7
Question: Ranking on Rolling Stone in 2009?
Answer:

Figure 6: Example prompt for graph-based models. We use $\mathsf{P}_{\text{prefix}}$ and $\mathsf{P}_{\text{suffix}}$ to denote the instruction before and after the `graph_node_embeddings` respectively. The umber of graph node embeddings is dynamic and varies based on evidence that has been retrieved.