박사학위논문
Ph.D. Dissertation

보편적 검색과 검증을 통한 파운데이션 모델의 맥락화

Beyond Parameters: Contextualizing Foundation Models with Universal Retrieval and Verification

2026

백진헌 (白 珍 憲 Baek, Jinheon)

한국과학기술원

Korea Advanced Institute of Science and Technology

박사학위논문

보편적 검색과 검증을 통한 파운데이션 모델의 맥락화

2026

백진헌

한국과학기술원

김재철AI대학원

보편적 검색과 검증을 통한
파운데이션 모델의 맥락화

백 진 헌

위 논문은 한국과학기술원 박사학위논문으로
학위논문 심사위원회의 심사를 통과하였음

2025년 11월 19일

심사위원장	Sung Ju Hwang	(인)
심 사 위 원	Jinwoo Shin	(인)
심 사 위 원	Minjoon Seo	(인)
심 사 위 원	Yejin Choi	(인)
심 사 위 원	Sujay Kumar Jauhar	(인)

Beyond Parameters: Contextualizing Foundation Models with Universal Retrieval and Verification

Jinheon Baek

Advisor: Sung Ju Hwang

A dissertation submitted to the faculty of
Korea Advanced Institute of Science and Technology in
partial fulfillment of the requirements for the degree of
Doctor of Philosophy in AI

Daejeon, Korea
November 19, 2025

Approved by

Sung Ju Hwang
Professor in Kim Jaechul Graduate School of AI

The study was conducted in accordance with Code of Research Ethics[1].

[1] Declaration of Ethical Conduct in Research: I, as a graduate student of Korea Advanced Institute of Science and Technology, hereby declare that I have not committed any act that may damage the credibility of my research. This includes, but is not limited to, falsification, thesis written by someone else, distortion of research findings, and plagiarism. I confirm that my thesis contains honest conclusions based on my own careful research under the guidance of my advisor.

DAI

백진헌. 보편적 검색과 검증을 통한 파운데이션 모델의 맥락화. 김재철AI 대학원 . 2026년. 156+ix 쪽. 지도교수: 황성주. (영문 논문)
Jinheon Baek. Beyond Parameters: Contextualizing Foundation Models with Universal Retrieval and Verification. Kim Jaechul Graduate School of AI . 2026. 156+ix pages. Advisor: Sung Ju Hwang. (Text in English)

초록

파운데이션 모델은 비약적인 발전을 이루어 왔지만, 매개변수 확장만으로는 내부에 내재된 지식의 부정확성·불완전성·시의성 문제를 해결하지 못하며, 모델의 본질적 특성인 상태 비저장성 또한 바꾸지 못한다. 이러한 모델의 행동은 궁극적으로 주어진 컨텍스트에 의해 결정되므로, 컨텍스트를 얼마나 잘 구성해 제공하느냐가 중요하다. 본 논문은 지식의 검색·구조화·검증·통합을 포함한 컨텍스트화가 파운데이션 모델의 능력을 강화하는 핵심임을 주장하며, 세 가지 연구 방향을 제시한다. 첫째, 구조적·비구조적·멀티모달 지식을 모델에 주입하고 검색 및 근거 오류를 탐지·교정하는 검증 메커니즘을 제안한다. 둘째, 컨텍스트화의 성능은 결국 검색되는 지식의 품질에 좌우되므로, 멀티모달 문서·비디오·지식 그래프·데이터베이스 등 이질적인 지식원을 아우르는 표현 및 검색 기법을 개발하고, 소스별 질의를 자동 생성하는 범용 검색 프레임워크를 제안한다. 셋째, 개인화, 과학적 발견, 구조적 엔터프라이즈 데이터 접근 등 실제 응용에서 컨텍스트화가 파운데이션 모델의 정확성과 도메인 적합성을 개선함을 보인다. 이를 통해, 더 큰 모델을 만드는 것보다 적절한 지식을 효과적으로 검색·검증·활용하는 능력이 차세대 인공지능의 핵심임을 강조한다.

핵심낱말 머신러닝, 파운데이션 모델, 컨텍스트화, 검색, 검증

Abstract

Foundation models have become remarkably powerful, and much of their recent progress has been driven by parameter scaling. However, scaling cannot address the limitations of the knowledge encoded in their parameters, which is often inaccurate, incomplete, and outdated, nor can it alter the intrinsic statelessness of these models. The behavior of foundation models is fundamentally governed by the context they receive: as stateless functions, they transform inputs into outputs, and therefore the effectiveness, reliability, and applicability of these models hinge on how well we construct and supply their context. In light of this, the dissertation argues that contextualization, which involves retrieving, structuring, verifying, and integrating knowledge, is the key to advancing foundation model capabilities beyond parameter size, and in pursuit of this goal, I advance three interconnected research directions. First, I introduce methods that augment models with structured, unstructured, and multimodal knowledge, and propose verification mechanisms that detect and correct retrieval and grounding errors. Second, recognizing that contextualization is, to a large extent, as strong as the knowledge it retrieves, I develop retrieval techniques capable of representing and retrieving diverse knowledge sources, from unstructured multimodal documents and videos to structured knowledge graphs, and broaden this to include databases through a universal retrieval framework that generates source-specific queries. Finally, I demonstrate that contextualization can drive real-world impact by enabling foundation models to support downstream applications, such as personalization, scientific discovery, and natural language access to structured enterprise data, showing that appropriately formulated context allows models to produce accurate and targeted outputs across domains. Taken together, this dissertation positions contextualization as a foundational principle for the next generation of AI agents and systems, where the challenge is no longer how large a model can be, but how effectively it can retrieve, verify, and utilize the knowledge required for reliable decision-making.

Keywords Machine Learning, Foundation Models, Contextualization, Retrieval, Verification

Contents

List of Tables

List of Figures

Chapter 1. Introduction

1.1 Background and Motivation

Over the past several years, foundation models [7, 8, 9, 10, 11, 12, 13, 14] have rapidly reshaped the landscape of artificial intelligence. In particular, powered by large-scale pre-training over diverse corpora and refined through alignment techniques, these models demonstrate strong generalization abilities across various tasks ranging from open-ended dialogue and code generation to complex reasoning. It is worth noting that this progress has been driven largely by scaling (e.g., expanding model parameters, training data, and compute budgets), which has produced models of unprecedented capability.

Yet, scaling alone does not overcome two fundamental limitations. First, the knowledge internalized in the model parameters is inevitably inaccurate, incomplete, and outdated: no matter how large a model is, its internalized knowledge reflects only the distribution on which it was trained. Second, foundation models are inherently stateless: they process inputs purely as functions of their current context, without persistence or access to (evolving) external information. For instance, in practical deployment scenarios of foundation models (such as ChatGPT or Gemini), their outputs are heavily influenced by the context they receive at inference time, not merely by the number of parameters behind them.

This observation motivates a crucial shift in perspective: rather than asking how large a model must be, we should instead ask and answer how effectively a model can be contextualized. Contextualization refers to the process of retrieving, structuring, verifying, and integrating knowledge into a model context so that the model can process and reason over diverse information that may be only partially encoded in its parameters or not encoded at all. As a result, in modern AI systems built on top of foundation models, the quality, accuracy, and relevance of this contextual information could directly shape the downstream predictions, often far more than the sheer number of parameters alone.

However, contextualization itself introduces multiple substantial challenges: it requires identifying and retrieving the right information from vast and heterogeneous knowledge sources; structuring and representing this information in formats suitable for model consumption; verifying retrieved facts and generated outputs to mitigate hallucinations and grounding failures holistically; and integrating knowledge from diverse modalities, schemas, and domains into the unified context. These challenges highlight that the frontier of model capability hinges not only on architectural improvements with increased sizes but also on the mechanisms through which models interface with, and reason over, various knowledge.

1.2 Thesis Scope and Research Directions

This dissertation investigates contextualization as a fundamental principle for advancing foundation model capabilities beyond parameter scaling. It argues that the future of foundation models (and potentially the agentic systems built upon them) will be determined not by how much information they can encode internally, but by how effectively they can retrieve, verify, and utilize the knowledge required for reliable output generation and decision-making. To support this argument, the dissertation develops a coherent line of works organized into three interconnected directions:

Chapter 2: Foundation Model Contextualization with Verification. The first part of the dissertation focuses on augmenting foundation models with external knowledge without modifying their parameters. This includes methods that incorporate structured, unstructured, and multimodal information into model contexts [15, 16, 17, 18], allowing models to leverage knowledge that would otherwise remain inaccessible. However, an additional challenge here is that contextualization alone does not guarantee correctness: the evidence retrieved may be incomplete or irrelevant, and models may fail to ground their answers in the retrieved sources (instead relying on unsupported or spurious internal associations). To address this, the dissertation further introduces verification mechanisms that detect and correct two critical error types, namely retrieval errors and grounding errors [19, 20]. In particular, by iteratively verifying retrieved facts and generated outputs and further correcting them whenever errors are detected, the methods significantly improve the accuracy and robustness of contextualized predictions.

Chapter 3: Universal Knowledge Retrieval for Contextualization. The next part of the dissertation aims to enhance contextualization by improving the quality and coverage of the knowledge being retrieved. In other words, since the effectiveness of contextualization is largely bounded by the relevance and fidelity of the retrieved information, this part develops retrieval techniques capable of handling a diverse set of knowledge sources, including unstructured text corpora, multimodal documents that interleave images and tables, videos with temporal structure, and structured resources such as knowledge graphs and relational databases. Specifically, since each of these sources requires distinct representational assumptions and retrieval interfaces, the dissertation introduces strategies that transform heterogeneous sources into formats compatible with conventional retrieval pipelines; for example, linearizing relational facts from knowledge graphs to make them accessible through conventional text retrieval methods [21], and encoding multimodal and temporal signals from documents and videos into embedding spaces [22, 17] so that retrieval behaves consistently across different modalities. Extending this direction further, the dissertation proposes a unified data access framework that generates source-specific queries while maintaining a common interface across knowledge types, enabling foundation models to retrieve information from diverse knowledge sources in a coherent and modality-agnostic manner.

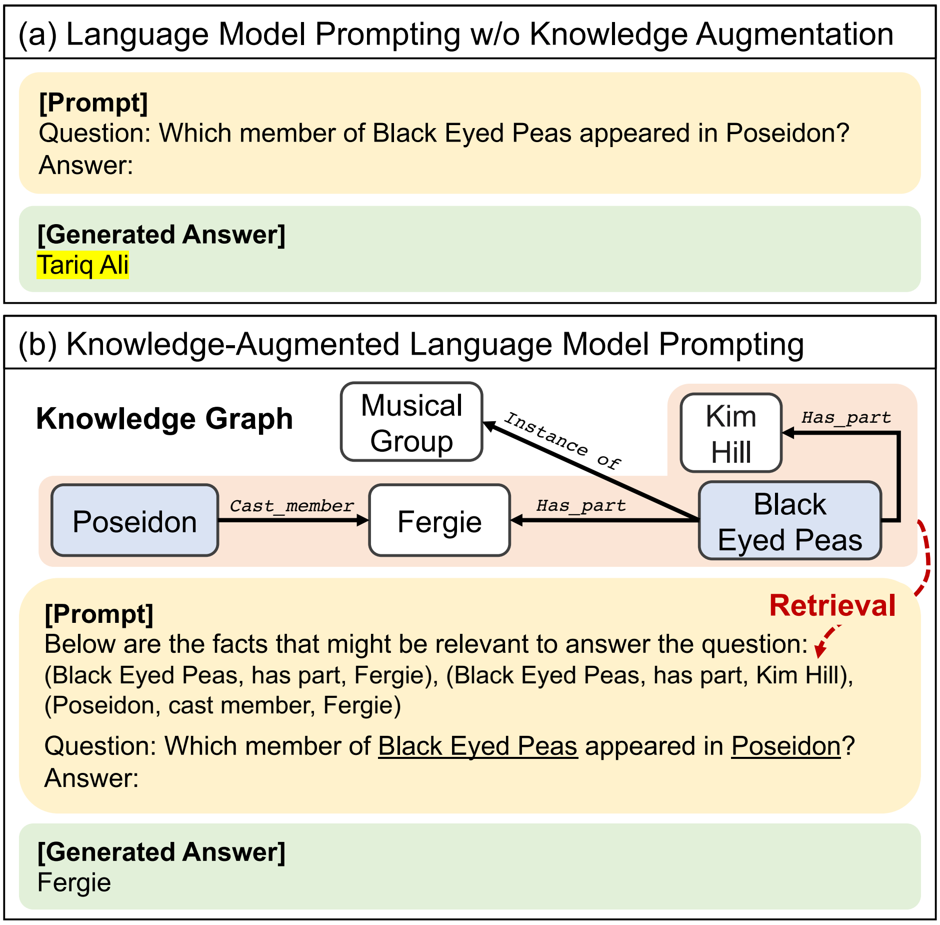
Chapter 4: Contextualization in Real-World Applications. The third part of the dissertation shows how contextualization enables foundation models to support real-world applications that require domain-specific information and grounding. In this part, the dissertation presents frameworks that apply contextualization to settings such as personalization, scientific discovery, and natural-language access to structured enterprise data (i.e., domains in which models must operate over specialized information that may not be in their parameters). Specifically, for personalization, the dissertation introduces an entity-centric knowledge augmentation approach that constructs lightweight user-specific knowledge stores from search and browsing histories to generate contextually relevant and tailored query suggestions (and, in general, model outputs) [23]. For scientific discovery, the dissertation develops a research idea generation and refinement framework that contextualizes models with papers, citations, and concept-level knowledge to produce more creative, valid, and specific research hypotheses [24]. Lastly, in natural-language access to structured databases, the dissertation proposes a knowledge-base construction method for text-to-SQL, which retrieves schema-aware and domain-relevant knowledge to guide models to generate more accurate SQL queries [25]. Taken together, these applications show that contextualization is not merely a technical mechanism for augmenting model inputs, but a practical pathway for enabling foundation models to operate reliably in specialized domains, grounding their outputs in task-relevant knowledge.

This dissertation takes the position that advancing foundation models requires shifting focus from parameter scaling to strengthening their capability to retrieve, verify, and utilize knowledge. The three research directions (such as contextualizing models with diverse knowledge, developing universal retrieval techniques, and demonstrating real-world impact), together establish contextualization as a principle for next-generation AI agents and systems. In Chapter 5, I revisit these contributions and discuss broader implications. I also outline promising avenues for future work, including conflict-aware contextualization, memory-enhanced agentic systems, and more robust verification methods for reasoning-intensive domains (AI for Science), which may further shape the role of contextualization in building more capable AI.

Chapter 2. Foundation Model Contextualization with Verification

2.1 Knowledge-Augmented Language Model Contextualization

2.1.1 Motivation



[Prompt]
Question: Which member of Black Eyed Peas appeared in Poseidon?
Answer:

[Generated Answer]
Tariq Ali

[Prompt]
Below are the facts that might be relevant to answer the question:
(Black Eyed Peas, has part, Fergie), (Black Eyed Peas, has part, Kim Hill),
(Poseidon, cast member, Fergie)

Question: Which member of Black Eyed Peas appeared in Poseidon?
Answer:

[Generated Answer]
Fergie

Figure 2.1: (a) For the input question in the prompt, the large language model (such as GPT-3 [5]) can generate the answer based on its internal knowledge in parameters, but hallucinates it (which is highlighted in yellow). (b) Our Knowledge-Augmented language model PrompTING (KAPING) framework first retrieves the relevant facts in the knowledge graph from the entities in the question, and then augments them to the prompt, to generate the factually correct answer.

Pre-trained Language Models (LMs) [26, 27], which are trained on a large amount of text corpora with self-supervised learning, can perform closed-book Question Answering (QA) tasks that aim to answer the user's question based only on their internal knowledge in parameters, without using any external knowledge [28, 29]. Also, when we increase the LM sizes, Large Language Models (LLMs) can generate the answer for the question without any additional fine-tuning steps, called *LM prompting* [5, 30]. However, since the knowledge in LLMs might be incomplete, incorrect, and outdated, they often generate factually wrong answers, known as *hallucination* [31] (See Figure 2.1a). Also, refining the knowledge in

LLMs with parameter updates is costly, especially when knowledge is constantly changing (e.g., exchange rates of money). Lastly, whether LLMs are fetching the correct knowledge for QA is unclear.

To overcome those limitations, we propose to retrieve and inject the relevant knowledge directly as an input, called a *prompt*, to LLMs (Figure 2.1b). As a knowledge source, we use a Knowledge Graph (KG) consisting of symbolic knowledge in the form of a triple: (head entity, relation, tail entity). Therefore, to extract the relevant facts to the input question, we first match entities in the question with entities in the KG. After that, triples associated to entities in the KG are verbalized (i.e., transforming the symbolic relational knowledge to the textual string) and prepended to the input question, which are then forwarded to LLMs to generate the answer. Consequently, LLMs conditioned on the factual knowledge are able to generate the factual answers, alleviating the hallucination issue, while keeping LLMs' parameters unchanged: fine-tuning is not required for knowledge updates. We refer to our overall framework as **K**nowledge-**A**ugmented language model **P**rompt**ING** (**KAPING**), which is completely *zero-shot* and can be done with any off-the-shelf LLMs, without additional training.

While the above scheme looks simple yet effective, there is a couple of challenges. First, most retrieved triples associated with the question entities are unrelated to answer the given question. For example, when we retrieve the associated triples for the question entity (e.g., Poseidon) in Figure 2.1 in the Wikidata KG [32], there exist 60 triples, and most of them (e.g., genre, publication date, to name a few) are irrelevant to answer the question. Therefore, they might mislead the model into generating incorrect answers. On the other hand, the number of triples for the question entities is occasionally large (e.g., 27% samples for the WebQSP dataset [33] have more than 1,000 triples), thereby encoding all triples including unnecessary ones yields high computational costs, especially on LLMs.

To overcome such challenges, we further propose to filter out unnecessary triples based on their semantic similarities to the input question, inspired by the information retrieval [34]. To be specific, we first represent the question and its associated verbalized triples in the embedding space. Then, we retrieve the small number of triples whose embeddings are more close to the input question's embedding than others. By doing so, we can prepend only the more relevant triples to the given question, which can effectively prevent LLMs from generating irrelevant answers with high computational efficiencies, unlike the one that augments all triples. Note that, our filtering approach uses off-the-shelf sentence embedding models [1, 35]; thus no additional training is required in every part of our pipeline.

We then validate our KAPING framework on Knowledge Graph Question Answering (KGQA) tasks. The results show that our KAPING significantly outperforms relevant zero-shot baselines. Also, the detailed analyses support the importance of knowledge retrieval and augmentation schemes.

2.1.2 Related Work

Language Model Prompting Language model pre-training, which trains Transformers [36] on unannotated text corpora with auto-encoding [26, 37] or auto-regressive [38, 39] objectives, becomes an essential approach for natural language tasks. Also, Large Language Models (LLMs) [5, 27, 40, 41] are able to perform zero-shot learning, for example, generating the answer for the input textual prompt, based on the knowledge stored in pre-trained parameters [28, 29, 42], without additional parameter updates as well as labeled datasets. To further improve their performances, some work [43, 44] proposes retrieving relevant samples to the input question from the training dataset and prepending them in the prompt under few-show learning. Recent few work [45, 46] further shows that, when LLMs are fine-tuned on a collection of instructions phrased from natural language tasks, they can have strong generalization performance on unseen zero-shot tasks. However, the knowledge inside LMs might be insufficient to tackle factual questions, which gives rise to knowledge-augmented LMs. Notably, our LM prompting is different from prompt-tuning literature [47, 48] that additionally tunes LMs with model training.

Knowledge-Augmented LMs Recent work proposes to integrate the knowledge, such as documents from unstructured corpora (e.g., Wikipedia) and facts from Knowledge Graphs (KGs), into LMs. To mention a few, REALM [49] and RAG [50] learn to retrieve documents and augment LMs with them. In addition, KGs could be another knowledge source, where the knowledge is succinctly encoded in the most compact form, and some methods augment such facts in KGs into LMs [51, 52, 53]. However, all aforementioned approaches require massive amount of training data and model updates for downstream tasks. While more recent work [54] shows retrieval-augmented LM can have strong performance with few-shot learning, it still requires extra training steps, which is different from ours focusing on *LM prompting* for entirely zero-shot. Recently, there are few studies augmenting the knowledge in the LM prompting scheme. Specifically, some work proposes to extract the knowledge in the parameters of LLMs themselves via prompting, and then use the extracted knowledge to answer the question [55, 56, 57, 58]. However, since LLMs' parameters might be insufficient to store all the world knowledge, the extracted knowledge and generated answers might be inaccurate. On the other hand, Lazaridou et al. [59] propose to use the Google Search to retrieve documents on the Web, and then prepend the retrieved documents to the input question along with few-shot demonstrations, to answer the question under few-shot LLM prompting schemes. Yet, our focus on *zero-shot prompting with KGs* is orthogonal to the previous study working on documents with few-shot prompting, and leveraging KGs can bring additional advantages. Specifically, since KGs can succinctly encode the knowledge in the compact triple form, for QA tasks, ours makes LLM prompting more efficient (i.e., reducing the input sequence compared to the document case), as well as more effective on the zero-shot QA scheme: LLMs need to select one triple containing the answer entity in the prompt, instead of looking through lengthy documents having various entities.

Knowledge Graph Question Answering The goal of our target Knowledge Graph Question Answering (KGQA) tasks is to answer the input question based on a set of facts over KGs [60, 61]. Previous approaches are broadly classified into neural semantic parsing-based methods [62, 63, 64], information retrieval-based methods [65, 66, 67], and differentiable KG-based methods [68, 69, 70], which, however, require annotated data with additional model training. While Zhou et al. [71] aim to transfer the KGQA model to the target language domains without any training data on them, this work indeed needs the labeled data to train the model on data-rich source domains first before transferring the model to the target domains. In contrast to all the aforementioned methods, we explore the novel zero-shot KGQA mechanism, which does not require any annotated QA pairs and training, leveraging LM prompting.

2.1.3 Approach

We now describe our Knowledge-Augmented language model PromptING (KAPING) framework.

2.1.3.1 LM Prompting for Zero-Shot QA

We begin with the zero-shot question answering, and then explain the language model prompting.

Zero-Shot Question Answering Given an input question $\boldsymbol{x}$, the Question Answering (QA) system returns an answer $\boldsymbol{y}$, where $\boldsymbol{x}$ and $\boldsymbol{y}$ consist of sequences of tokens: $\boldsymbol{x} = [w_1, w_2, \ldots, w_{|\boldsymbol{x}|}]$. Let P be a QA model based on the Language Model (LM) [27, 5], which generates the conditional probability of answer

$\boldsymbol{y}$ for question $\boldsymbol{x}$ as follows: $P(\boldsymbol{y}|\boldsymbol{x})$. Then, in contrast to supervised learning that trains model P with a set of annotated ($\boldsymbol{x}$, $\boldsymbol{y}$) pairs, zero-shot learning does not use any labeled samples and model training. Notably, we are interested in this zero-shot QA, since collecting the dataset and then fine-tuning the existing LMs for every new domain are known to be expensive and sometimes infeasible [72, 73].

LM Prompting LMs are often pre-trained by predicting the next token based on previous tokens, which is known as auto-regressive language modeling [39, 27]. Then, thanks to this pre-training objective, LLMs can perform zero-shot instruction learning. Specifically, when we provide a question as well as an instruction (e.g., "Please answer the question: Who is the author of Lady Susan?") to the LLM (i.e., P), such the LLM, conditioned by the input text, can sequentially generate the probability of output tokens, which might be an answer, "Jane Austen".

To be more formal, for every input question $\boldsymbol{x}$, we first modify it with a particular instruction template T into a textual string $\boldsymbol{x}'$ called a *prompt*, as follows: $T : \boldsymbol{x} \mapsto \boldsymbol{x}'$. For example, if we have the previous question $\boldsymbol{x}$ = "Who is the author of Lady Susan?" along with the previous instruction template "Please answer the question:", the resulting prompt $\boldsymbol{x}'$ would be $T(\boldsymbol{x})$ = "Please answer the question: Who is the author of Lady Susan?". Then, we forward the prompt $\boldsymbol{x}'$ to the LLM (i.e., P), which then generates the answer (i.e., $\boldsymbol{y}$) through $P(\boldsymbol{y}|\boldsymbol{x}')$. Note that this LM prompting scheme does not require any additional model parameter updates (i.e., fine-tuning) on the labeled data, thus appropriate for the target zero-shot QA task.

However, there are multiple challenges in this naive zero-shot prompting for QA. First, LLMs, which rely on the knowledge in parameters, are vulnerable from generating the factually incorrect answer, since the knowledge in LLMs might be inaccurate, and outdated: knowledge can be emerged and changed over time. Also, refining the internalized knowledge with additional parameter updates is expensive, while it is necessary to reflect the wrong and ever growing knowledge. Lastly, which knowledge LLMs memorize and utilize when generating the answer to the question prompt is unclear, which limits their explainability on the outputs.

2.1.3.2 Knowledge-Augmented LM Prompting

In order to tackle the aforementioned limitations of the existing LM prompting scheme, we propose to inject the relevant knowledge to the input question from the Knowledge Graph (KG), which we refer to as Knowledge-Augmented language model PromptING (KAPING). In this subsection, we first define the main objective of our KAPING framework, and then introduce the ingredients for augmenting the knowledge over KGs to LM prompts.

LM Prompting with Knowledge Graphs Instead of relying on the knowledge internalized in parameters, we propose to additionally access and inject the knowledge from the external KG, which contains accurate and up-to-date facts helpful to answer the question. Formally, a knowledge graph $\mathcal{G}$ consists of a set of factual triples $\{(s, r, o)\}$, where s and o denote subject and object entities, and r is a specific type of a relation between them. For example, one relational knowledge "Lady Susan was written by Jane Austen" can be represented as a triple consisting of two entities s = "Lady Susan" and o = "Jane Austen" along with a relation r = "written by". Then, for the question prompt $\boldsymbol{x}'$ transformed from the example question $\boldsymbol{x}$ = "Who is the author of Lady Susan?" via the template T, we additionally augment its relevant triple: (Lady Susan, written by, Jane Austen), to the LM prompting scheme. By doing so, LLMs can generate the answer with regard to the augmented knowledge from KGs, formalized as follows:

$P(\boldsymbol{y}|\boldsymbol{x}', \mathcal{G})$. Note that, since we can provide specific and valid facts in KGs to LLMs whenever they exist, our framework can alleviate hallucination issue, originated from inaccurate and outdated knowledge in LLMs, without costly updating their model parameters. Furthermore, we can confirm whether LLMs generate answers based on augmented facts, thus improving the explainability of LM prompting.

The remaining questions are then how to *access* the relational symbolic facts over the KG from the input question, *verbalize* the symbolic knowledge to the textual string, and *inject* the verbalized knowledge into the LM prompting scheme. We explain them one by one in the following paragraphs.

Knowledge Access In order to utilize the related facts to the input question, we first extract the entities in the question. For example, for the question "Who is the author of *Lady Susan*?", we extract the entity "Lady Susan". Then, based on the extracted entity, we find its corresponding entity over the KG, whose incident triples then become associated facts to the input question. Note that entity matching can be done by existing entity linking techniques [74, 75, 2].

Knowledge Verbalization LLMs are working on textual inputs, whereas factual triples are represented over the symbolic graph. Therefore, before injecting the symbolic fact from KGs to LLMs, we first transform the triple consisting of (s, r, o) into its textual string, called verbalization. While there exists recent methods [76, 77] that particularly design or even learn the graph-to-text transformation, in this work, we use the linear verbalization: concatenating the subject, relation, and object texts in the triple, which we observe works well in LM prompting. For instance, one triple (Lady Susan, written by, Jane Austen) is used as is: "(Lady Susan, written by, Jane Austen)", for an LLM's input.

Knowledge Injection Based on verbalized facts associated with the input question, the remaining step is to realize the knowledge injection mechanism, which allows LLMs to be grounded on the external knowledge, useful to generate the answer. Let assume we have a set of N associated triples $\boldsymbol{k} = \{(s_i, r_i, o_i)\}_{i=1}^{N}$ for question $\boldsymbol{x}$. Then, similar to instruction template $T : \boldsymbol{x} \mapsto \boldsymbol{x}'$ described in Section 2.1.3.1, we modify N verbalized triples $\boldsymbol{k}$ along with the instruction for the knowledge injection into the knowledge prompt $\boldsymbol{k}'$, as follows: $T : \boldsymbol{k} \mapsto \boldsymbol{k}'$. One particular template we use for constructing the prompt is that, we first enumerate N verbalized triples line-by-line and then add the specific instruction: "Below are facts in the form of the triple meaningful to answer the question.", at the top of the prompt. After that, such the knowledge prompt string, $\boldsymbol{k}'$, is prepended to the question prompt $\boldsymbol{x}'$, and LLMs conditioned by knowledge and question prompts then sequentially generate the answer tokens, formalized as follows: $P(\boldsymbol{y}|[\boldsymbol{k}', \boldsymbol{x}'])$, where $[\cdot]$ denotes concatenation.

2.1.3.3 Question-Relevant Knowledge Retrieval

The proposed KAPING framework in Section 2.1.3.2, allows LLMs to leverage the knowledge from KGs for zero-shot QA. However, there are critical challenges that the number of triples associated to questions is often too large to forward in LLMs. Also, most of them are unrelated to the question, misleading LLMs into generating the irrelevant answer.

Knowledge Retriever To overcome those limitations, we further propose to retrieve and augment only the relevant triples to the question. Note that there exists a document-retrieval scheme [78], whose goal is to retrieve relevant documents for the given query based on their embedding similarities, which motivates us to retrieve, in our case, the triples for the user's question. In particular, thanks to the verbalizer defined in Section 2.1.3.2, we can play with triples, obtained from symbolic KGs, over the text space. Therefore, for the verbalized triple and the question, we first embed them onto the representation space with off-the-shelf sentence embedding models for text retrieval [1, 79, 80], and then calculate their similarities. After that, we use only the top-K similar triples, instead of using all N triples, associated to the given question. Note that, unlike few recent studies [76, 77, 53] that aim at improving KG retrievers themselves under supervised training, we focus on zero-shot LM prompting with KGs, thus we use any off-the-shelf retrievers as a tool to filter out unnecessary triples for questions.

2.1.4 Experiments

We first explain datasets, models, metrics, and implementations.

2.1.4.1 Datasets

We evaluate our Knowledge-Augmented language model PromptING (KAPING) framework on two Knowledge Graph Question Answering (KGQA) datasets, namely WebQuestionsSP and Mintaka.

WebQuestionsSP (WebQSP) This dataset [81, 33] is designed with a Freebase KG [82]. It consists of 1,639 test samples, which we use for zero-shot evaluation. Additionally, since Freebase is outdated, we further use the Wikidata KG [32] by using available mappings from Freebase ids to Wikidata [83]. This additional dataset consists of 1,466 samples.

Mintaka This dataset [84] is recently designed with the Wikidata KG for complex KGQA tasks. Among 8 different languages, we use English test sets consisting of 4,000 samples.

2.1.4.2 Large Language Models

To verify the performance of our KAPING framework on Large Language Models (LLMs), as well as benchmarking them on zero-shot KGQA, we use various LLMs with different sizes. Specifically, we use T5 [27] (0.8B, 3B, 11B), T0 [45] (3B, 11B), OPT [85] (2.7B, 6.7B) and GPT-3 [5] (6.7B, 175B).

2.1.4.3 Baselines and Our Model

In this subsection, we explain four zero-shot LM prompting baselines and our KAPING framework.

No Knowledge This is a naive LM prompting baseline, which generates answers from input questions without knowledge augmentation from KGs.

Random Knowledge This is an LM prompting baseline, which additionally augments the randomly sampled K triples, associated to the entities appeared in the question, to the prompt.

Popular Knowledge This is an LM prompting baseline, which augments K popular triples among all triples from the question entities, based on relations that appear the most frequently in the KG.

Generated Knowledge This is an LM prompting baseline, which first extracts the knowledge from LLMs themselves based on prompting, and then augments them as the form of the prompt [56], which is similar to Kojima et al. [55].

Table 2.1: Main results of language model prompting, where we report the generation accuracy. The number inside the parentheses in the first row denotes the parameter size of language models, and the best scores are emphasized in bold.

Datasets	Methods	T5 (0.8B)	T5 (3B)	T5 (11B)	OPT (2.7B)	OPT (6.7B)	OPT (13B)	T0 (3B)	T0 (11B)	GPT-3 (6.7B)	GPT-3 (175B)	AlexaTM (20B)	Average
WebQSP w/ Freebase	No Knowledge	6.95	13.40	9.48	19.85	29.77	28.38	21.43	40.77	44.63	63.59	46.79	29.55
	Random Knowledge	21.55	19.15	17.57	28.07	31.73	33.31	32.62	51.20	51.01	65.87	57.37	37.22
	Popular Knowledge	15.30	16.88	18.39	28.32	28.13	24.21	27.05	47.22	45.58	62.26	54.91	33.48
	Generated Knowledge	6.19	7.84	6.76	7.46	11.50	8.22	19.41	38.81	45.89	62.14	35.13	22.67
	KAPING (Ours)	**34.70**	**25.41**	**24.91**	**41.09**	**43.93**	**40.20**	**52.28**	**62.85**	**60.37**	**73.89**	**67.67**	**47.94**
WebQSP w/ Wikidata	No Knowledge	10.30	18.42	15.21	23.94	33.77	32.40	24.56	44.20	48.50	67.60	42.41	32.85
	Random Knowledge	17.94	22.78	24.28	37.24	35.61	38.27	28.85	47.68	52.05	60.64	55.63	38.27
	Popular Knowledge	15.35	20.80	20.74	30.83	30.01	27.83	24.83	48.02	47.41	63.37	53.92	34.83
	Generated Knowledge	11.94	13.30	12.28	11.26	17.53	14.19	22.92	41.34	48.77	65.89	31.16	26.42
	KAPING (Ours)	**23.67**	**40.38**	**35.47**	**49.52**	**53.34**	**51.57**	**49.86**	**58.73**	**60.44**	**69.58**	**65.04**	**50.69**
Mintaka w/ Wikidata	No Knowledge	11.23	14.25	17.06	19.76	27.19	26.83	14.75	23.74	34.65	56.33	41.97	26.16
	Random Knowledge	17.59	18.19	18.83	28.11	26.58	28.36	16.10	26.15	32.98	51.56	46.02	28.22
	Popular Knowledge	17.56	18.09	18.73	26.97	27.08	23.10	16.74	27.15	32.48	53.16	46.41	27.95
	Generated Knowledge	13.61	14.61	14.29	11.87	14.96	16.24	14.46	23.13	33.12	55.65	34.58	22.41
	KAPING (Ours)	**19.72**	**22.00**	**22.85**	**32.94**	**32.37**	**33.37**	**20.68**	**29.50**	**35.61**	**56.86**	**49.08**	**32.27**

KAPING (Ours) This is our Knowledge Augmented language model PromptING (KAPING) framework, which first retrieves the top-K similar triples to the question with the knowledge retriever, and then augments them as the form of the prompt.

2.1.4.4 Evaluation Metrics

Generation Following the evaluation protocol of generative KGQA [86, 84, 87], we use accuracy, which measures whether the generated tokens from the given prompt include one of the answer entities. Note that we further consider *aliases* – a set of alternative names – of answer entities available in Freebase and Wikidata KGs, for evaluation.

Retrieval We also measure the retriever performance, to see how much the retrieved triples are helpful for answer generation. As metrics, we use Mean Reciprocal Rank (MRR) and Top-K accuracy (Top-K), which are calculated by ranks of correctly retrieved triples containing answer entities among all triples associated to question entities.

2.1.4.5 Implementation Details

For the knowledge injection, we set the number of retrieved facts as 10 ($K = 10$), and the hop for triple retrieval as one. For the text-based retriever, we experiment with MPNet [1] that uses the same encoder for embedding question and triples.

2.1.4.6 Experimental Results and Analyses

We now provide the overall results of our KAPING framework along with its comprehensive analyses.

Main Results As shown in Table 2.1, our KAPING framework significantly outperforms all LM prompting baselines, on zero-shot KGQA tasks. In particular, the generated knowledge model mostly degenerates the performance compared to the no knowledge model, since the extracted knowledge from LLMs themselves might be inaccurate. On the other hand, the random and popular knowledge baselines bring performance improvements, since the augmented knowledge from KGs are sometimes useful to answer the question. However, ours outperforms them, which suggests that, for zero-shot LM prompting

Table 2.2: Retriever results, where we compare random model, popular model, and MPNet [1].

		1-Hop Retrieval				2-Hop Retrieval			
Datasets	**Retrievers**	**MRR**	**Top-1**	**Top-10**	**Top-30**	**MRR**	**Top-1**	**Top-10**	**Top-30**
WebQSP w/ Freebase	Random	12.50	7.21	25.09	34.64	1.50	0.70	2.65	5.37
	Popular	8.58	5.31	15.93	24.53	1.59	0.95	2.72	4.68
	MPNet	**47.27**	**40.27**	**60.56**	**64.48**	**41.64**	**33.12**	**58.47**	**65.23**
WebQSP w/ Wikidata	Random	9.50	3.62	22.58	40.72	1.31	0.00	2.80	8.59
	Popular	8.52	4.57	15.89	35.47	4.63	4.02	5.53	6.62
	MPNet	**43.46**	**33.36**	**64.39**	**70.67**	**40.42**	**30.56**	**62.62**	**71.56**
Mintaka w/ Wikidata	Random	4.80	1.85	11.48	22.03	0.91	0.14	1.78	5.15
	Popular	6.09	3.09	12.51	20.47	0.24	0.04	0.28	1.24
	MPNet	**13.01**	**7.50**	**25.44**	**35.43**	**13.00**	**6.82**	**26.65**	**40.01**

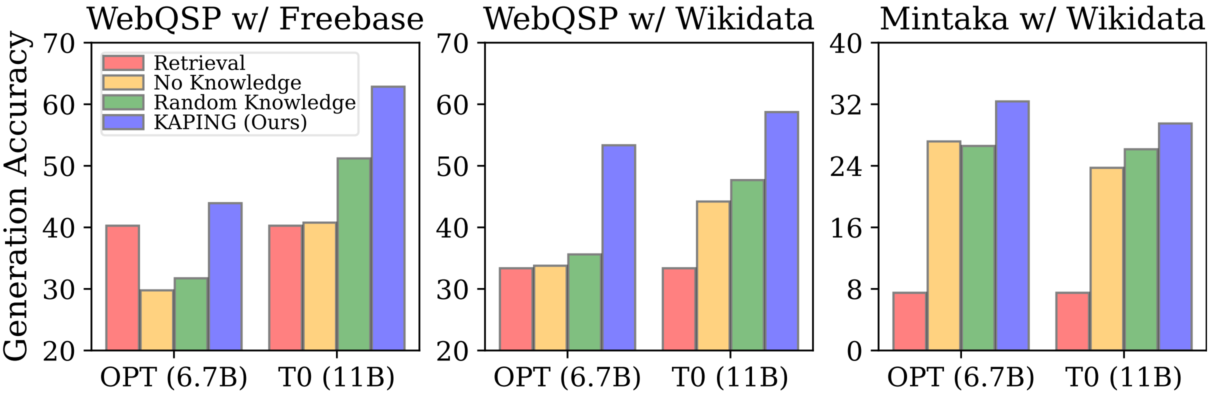


Figure 2.2: Comparisons of retrieval and LM prompting, where retrieval is the Top-1 result of MPNet [1].

for QA, the knowledge internalized in LLMs is insufficient to generate factual answers, and it is important to use only the relevant facts.

In addition, we also observe larger performance improvements when LMs are relatively small. In other words, since smaller models have insufficient parameter spaces to memorize the knowledge during pre-training, they are more likely to generate factually incorrect answers. However, when the appropriate knowledge is given to them, their performances sometimes become similar to larger models (e.g., different sizes of OPT have similar performances by our KAPING). Therefore, for tasks that require factual knowledge under low-resource setups (e.g., production), augmenting the knowledge would be beneficial, instead of increasing model sizes to handle the huge volume of knowledge.

Retriever Results To see how relevant the augmented knowledge is, we further measure the retrieval performances. As shown in Table 2.2, the existing retrieval model (i.e., MPNet) shows superior performances against naive models: random and popular retrievers. This result suggests that our simple graph-to-text verbalization works well with the existing retriever, which further confirms that our KAPING augments useful facts in the LM prompt. Regarding the number of hops for the candidate triples to retrieve, we observe that, when we increase the hop-size from one to two, the retriever is more likely to retrieve irrelevant triples that does not include answer entities, as shown in Table 2.2. Therefore, in our experiments, we retrieve knowledge among 1-hop triples of question entities.

Additionally, since we can alternatively answer the input question based on entities in the Top-1 triple from the retriever, we compare the generation performance of LLMs to the retrieval performance. As

Figure 2.3: Comparisons of correct and incorrect retrieval for the generation performance on the GPT-3 (6.7B) model.

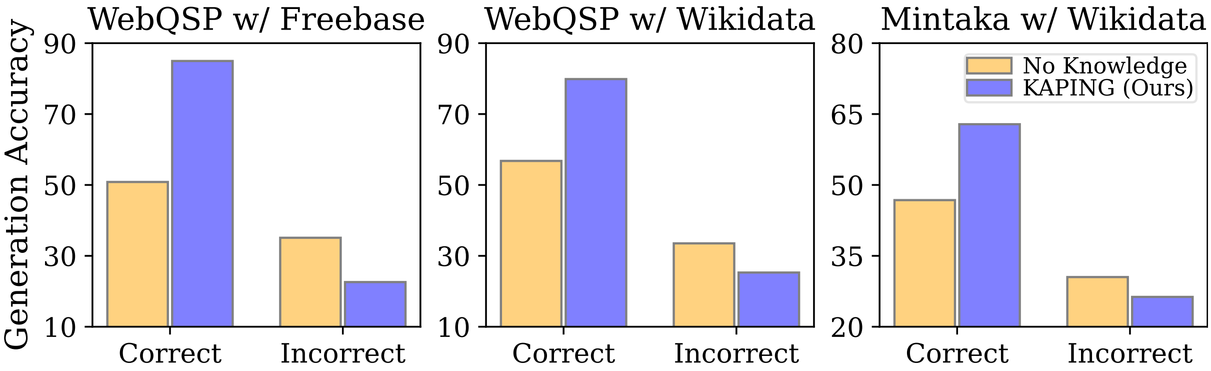
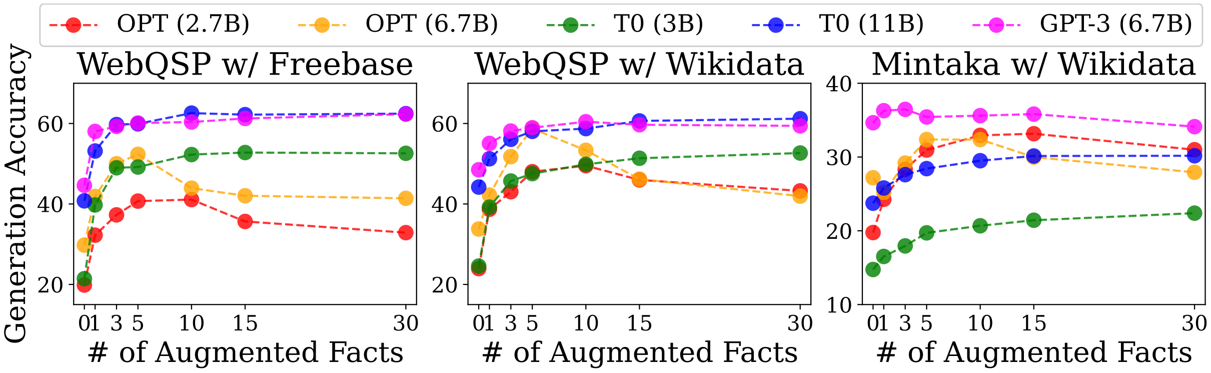


Figure 2.4: Performances with varying the knowledge amount, where we change the number of retrieved triples to augment.



shown in Figure 2.2, LM prompting schemes even without knowledge augmentation (i.e., no knowledge) are superior than simply answering with the entity in the retrieved triple, except for the WebQSP w/ Freebase dataset. Also, we observe huge gaps between our KAPING framework and the simple retrieval scheme on all datasets. These results suggest that, for zero-shot KGQA, it would be helpful to leverage LLMs to generate answers based on their internalized and external facts, instead of directly searching answer entities over KGs.

Impact of Correct & Incorrect Retrievals We conduct analyses on how much the correctly retrieved triples, having answer entities, bring performance improvements, and how performances are affected by the incorrectly retrieved triples, which do not include answer entities. As shown in Figure 2.3, when retrieved triples contain answer entities, performances of LLMs are significantly improved, compared to models without knowledge augmentation. However, when retrievers fail, performances are lower than models of no knowledge augmentation. These results suggest, when relevant knowledge is augmented, LLMs can contextualize and generate answers accurately. Meanwhile, incorrectly retrieved knowledge makes LLMs condition on irrelevant facts, and generate wrong answers.

Varying the Amount of Knowledge We change the number of facts, to see which triple amounts are optimal to augment in the prompt, by comparing trade-off between the generation performance and the wall-clock time. First of all, as shown in Figure 2.4, most LLMs reach the somewhat highest performance, when the number of triples is 5 or 10. Also, when we further increase the augmented triple

Table 2.3: Efficiencies with varying the knowledge amount, where we measure the wall-clock time of every model for generating the answer on the WebQSP w/ Wikidata dataset.

		Relative Time	
Models	**# of Retrieved Facts**	**T0 (3B)**	**OPT (2.7B)**
No Knowledge	0	1.00	1.00
KAPING (Ours)	1	0.49	1.12
	5	0.73	1.48
	10	1.07	1.89
	15	1.54	2.36
	30	2.49	3.77

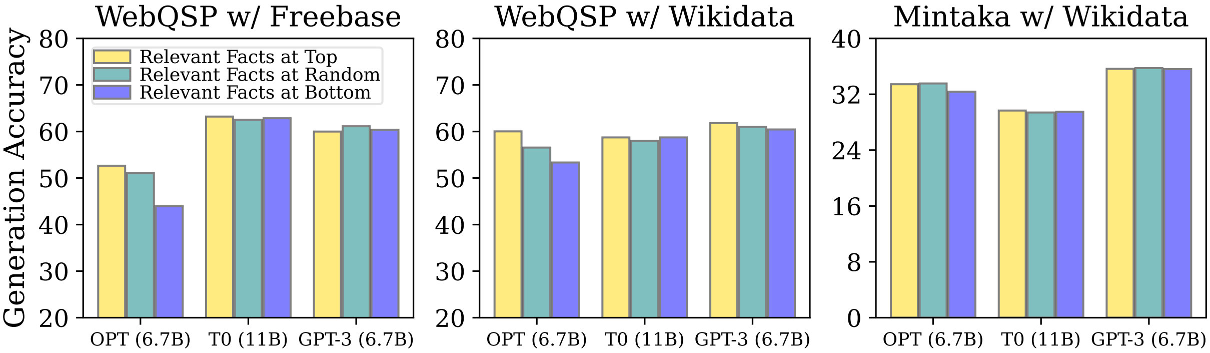


Figure 2.5: Performances with varying the knowledge order, where we change the location (namely, top, bottom, or random) of more relevant triples for the question in the prompt of LLMs.

size to 15 and 30, performances of OPT models are largely decreasing. This result suggests that some LMs might be distracted by irrelevant triples when their volumes are high, therefore, failing to select and generate the answer entity.

We then measure the wall-clock time of the answer generation, for the encoder-decoder (T0) and decoder-only (OPT) models with varying the number of augmented triples in the prompt. As shown in Table 2.3, regarding the encoder-decoder model, our KAPING framework with less than 10 triples is faster than the model without knowledge augmentation. We observe this is because, when the knowledge is augmented to the model, the model tends to generate shorter answers, which can reduce the decoding time. More specifically, the length of generated tokens for the T0 model with 10 triples is 15, whereas, the no knowledge model generates 32 tokens on average. Yet, for OPT, the more knowledge we augment, the slower the model becomes, because of its auto-regressive characteristic for digesting the input.

Impact of Orders of Retrieved Triples In few-shot LM prompting where LLMs additionally observe few examples in the prompt, they are known to be sensitive to the order of examples [88], and they tend to follow the answer in the last example [89]. Based on those observations, we also conduct an analysis on whether the order of retrieved triples affects the performance. In particular, we vary the location of more similar triples for the question, by locating them at the Top, Bottom, or Random position of the prompt. As shown in Figure 2.5, our KAPING is not sensitive to the location of retrieved triples, except for the OPT model on the WebQSP dataset. In other words, the OPT model tends to generate the entity located at the first part of the prompt input. Meanwhile, other LLMs can contextualize the entire prompt input, and generate the entity regardless of its position.

Table 2.4: Results with entity linking, where the model w/ EL uses entities extracted from the entity linking technique [2], instead of using labeled ones, on Mintaka.

Models	**T5** (3B)	**T5** (11B)	**OPT** (2.7B)	**OPT** (13B)	**T0** (3B)	**T0** (11B)
No Knowledge	14.25	17.06	19.76	26.83	14.75	23.74
Random Knowledge	18.19	18.83	28.11	28.36	16.10	26.15
Random Knowledge w/ EL	15.99	17.98	23.10	26.47	15.60	24.66
KAPING	22.00	22.85	32.94	33.37	20.68	29.50
KAPING w/ EL	18.94	20.58	26.87	28.39	18.51	27.11

Table 2.5: Generation examples of the prompted GPT-3 for the input question with augmented triples from the retriever, where, in the last row, we change the knowledge of augmented facts to see whether the model is able to adapt to the changed knowledge.

Question: Where did Alex Chilton die?
Retrieved triples: (Alex Chilton, place of death, New Orleans), (Alex Chilton, manner of death, natural causes), (Alex Chilton, cause of death, myocardial infarction), (Alex Chilton, date of death, time: +2010-03-17), ...
Answer: New Orleans

	Prompts	**Outputs**
No Knowledge	Question: Where did Alex Chilton die? Answer:	Alex Chilton died in a ~~car accident~~ on ~~February 10, 1979~~.
KAPING (Ours)	Below are facts in the form of the triple meaningful to answer the question. ... (Alex Chilton, date of death, time: +2010-03-17) (Alex Chilton, cause of death, myocardial infarction) (Alex Chilton, manner of death, natural causes) (Alex Chilton, place of death, New Orleans) Question: Where did Alex Chilton die? Answer:	Alex Chilton died on March 17, 2010 in New Orleans, Louisiana due to a myocardial infarction.
KAPING (Ours) **w/ Knowledge Change**	Below are facts in the form of the triple meaningful to answer the question. ... (Alex Chilton, date of death, time: +2000-09-01) (Alex Chilton, cause of death, pancreatic cancer) (Alex Chilton, manner of death, natural causes) (Alex Chilton, place of death, Los Angeles) Question: Where did Alex Chilton die? Answer:	Alex Chilton died in Los Angeles, California on September 1, 2000 from pancreatic cancer.

Effectiveness with Entity Linking Following the conventional KGQA evaluation [68], we use question entities labeled in datasets, to retrieve facts in KGs. However, to see the performance with entities identified by Entity Linking (EL) technique, we further conduct experiments with the EL model, namely ReFinED [2]. As shown in Table 2.4, while the performance of KAPING w/ EL is slightly decreasing from the model with labeled entities due to the performance of EL, we consistently observe meaningful performance improvements from a No Knowledge model.

Case Study We conduct a case study in Table 2.5. In particular, when the knowledge is not given to the LM, it hallucinates the factually incorrect answer. However, when related facts are retrieved and augmented in the prompt, it can generate the correct answer. In addition, we analyze whether our KAPING can adapt to the updated knowledge, motivated by that some knowledge can be changed over time, while the knowledge in LMs remains static. To do so, as shown in the last row of Table 2.5, we replace object entities of triples, and then forward the prompt with the modified facts to the LM. Then, the result shows that the LM can generate the output based on the updated facts, which suggests the potential of adapting LMs without costly updating their parameters.

2.1.5 Summary

In this work, we focused on the limitation of existing LM prompting schemes, which rely on the static knowledge internalized in parameters; therefore, when such knowledge are incomplete, inaccurate, and outdated, LLMs may generate factually incorrect answers. To tackle this challenge, we introduced a novel Knowledge-Augmented language model PrompTING (KAPING) framework, which augments the knowledge for the input question from KGs directly in the input prompt of LLMs, with the fact retriever to inject only the relevant knowledge. The proposed framework is completely zero-shot, and versatile with any LMs, without additional parameter updates and training datasets. We validated that KAPING yields huge performance gaps from the LM prompting model relying on its internal knowledge, especially with smaller LMs, on the KGQA tasks. We believe our new mechanism for augmenting facts from KGs to the LM prompt will bring substantial practical impacts in generating knowledge-grounded answers.

2.1.6 Extension: Adaptive Model Contextualization

While KAPING demonstrates that large language models can effectively leverage externally retrieved knowledge without parameter updates, it still treats all queries uniformly, applying the same retrieval and augmentation strategy regardless of their complexity. Yet, real-world queries vary widely, from straightforward factual lookups that LLMs can answer without retrieval, to multi-step, compositional questions that require iterative reasoning and multiple rounds of retrieval. To address this more realistic spectrum of query types, we propose the adaptive model contextualization framework (Adaptive-RAG) [16], which extends the initial prompting-based framework by introducing an adaptive mechanism that dynamically selects the most suitable contextualization strategy based on the predicted query complexity. Additionally, this is operationalized through a lightweight classifier, trained automatically using model predictions and dataset-induced inductive biases, which categorizes each incoming query into a complexity level and routes it accordingly: no-retrieval prompting for simple queries, single-step retrieval-augmented prompting for moderate ones, and multi-step iterative retrieval for challenging questions. We then demonstrate that this adaptive formulation preserves the (zero-shot) training-free spirit of the model contextualization while substantially improving efficiency: avoiding unnecessary retrieval or multi-hop reasoning for easy queries, while maintaining strong accuracy on complex queries. Ultimately, the proposed adaptive model contextualization framework advances the core principle that foundation models should be augmented not only with the right knowledge but also with the right contextualization strategy tailored to each query, moving toward more flexible and efficient retrieval-augmented systems.

2.2 Knowledge-Augmented Language Model Verification

2.2.1 Motivation

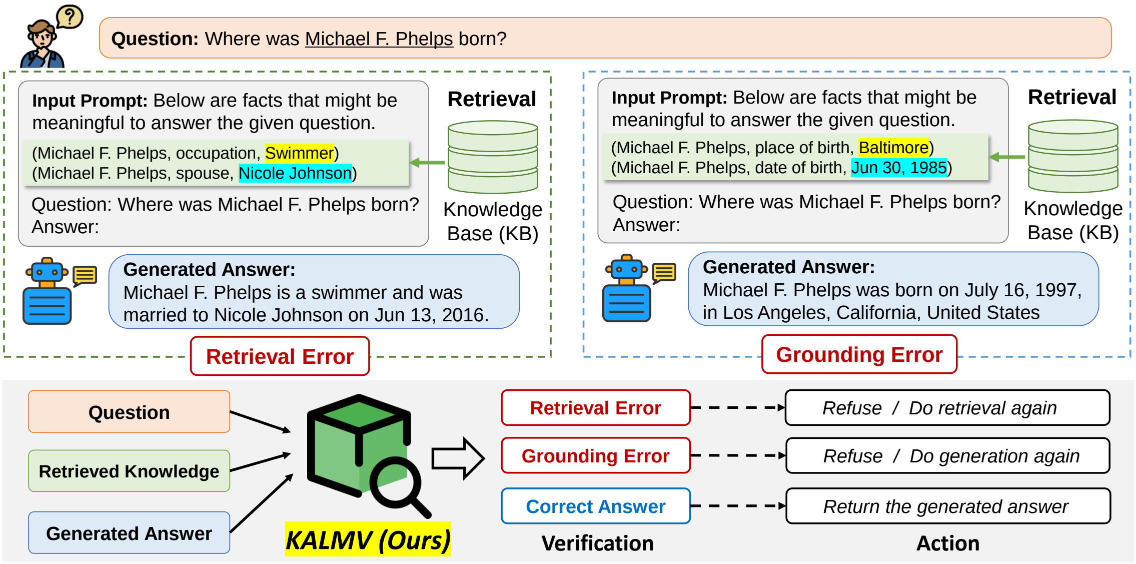

Question: Where was Michael F. Phelps born?
Input Prompt: Below are facts that might be meaningful to answer the given question.
(Michael F. Phelps, occupation, Swimmer)
(Michael F. Phelps, spouse, Nicole Johnson)
Question: Where was Michael F. Phelps born?
Answer:
Retrieval
Knowledge Base (KB)
Generated Answer:
Michael F. Phelps is a swimmer and was married to Nicole Johnson on Jun 13, 2016.
Retrieval Error
Input Prompt: Below are facts that might be meaningful to answer the given question.
(Michael F. Phelps, place of birth, Baltimore)
(Michael F. Phelps, date of birth, Jun 30, 1985)
Question: Where was Michael F. Phelps born?
Answer:
Retrieval
Knowledge Base (KB)
Generated Answer:
Michael F. Phelps was born on July 16, 1997, in Los Angeles, California, United States
Grounding Error
Question
Retrieved Knowledge
Generated Answer
KALMV (Ours)
Retrieval Error
Grounding Error
Correct Answer
Verification
Refuse / Do retrieval again
Refuse / Do generation again
Return the generated answer
Action

Figure 2.6: Existing knowledge-augmented language models first retrieve the relevant knowledge to the given query from the external knowledge base and augment LMs with the retrieved knowledge to generate the factually correct responses. However, there are two types of errors: 1) the retrieved knowledge might be irrelevant to the given query (retrieval error); 2) the generated answer might not be grounded in the retrieved knowledge (grounding error). Our proposed KALMV can detect those two types of errors in knowledge retrieval and grounding, and also iteratively rectify them, reducing hallucinations.

Recent Language Models (LMs) [5, 40, 90], which have a large number of parameters and are further instruction-finetuned on massive datasets, have achieved remarkable successes on various language tasks. For example, they are able to perform closed-book zero-shot question answering, which aims to provide an answer to a user's query without updating the LM parameters while using only the knowledge internalized in their parameters. However, while the generated answers from LMs look plausible and sound, they are often factually incorrect, which is a problem widely known as *hallucination* [31, 91, 92]. Hallucination is a critical problem when deploying LMs, since it poses a risk of spreading misinformation, potentially misleading users who rely on the information.

To mitigate hallucination of LMs, recent works have proposed to augment LMs with the knowledge retrieved from external knowledge sources (e.g., Wikipedia and Wikidata) [93, 94, 15]. Moreover, some other works have proposed to check the factuality of generated texts and refine them by using the knowledge in LMs themselves or from the external knowledge sources [95, 96, 97, 98, 99, 100]. However, while the aforementioned knowledge-augmentation strategies are effective in reducing hallucinations, we find that there still exists a couple of challenges: 1) the retrieved knowledge may not be relevant to the given question from the user, and 2) the generated answer may not be grounded in the retrieved knowledge, as illustrated in Figure 2.6 (and shown in Figure 2.7).

In this work, we aim to overcome these suboptimalities of knowledge-augmented LMs. In other words, our goal is to verify whether the retrieved knowledge used for augmenting LMs is related to generating the answers for the given questions and whether the generated answers include the relevant parts of the retrieved knowledge. To this end, we propose to train a small, tailorable LM that is able to verify the aforementioned two failure cases of knowledge-augmented LMs in retrieval and generation steps. More specifically, we first automatically construct the training labels by categorizing the failure of knowledge-augmented LMs into two cases: retrieval error and generation error, based on the triplet of the input question, retrieved knowledge, and generated answer. Then, we instruction-finetune the LM with pairs of a certain verification instruction and its associated label, during verifier training. At the inference step, we validate the generated texts through our verifier, to filter out potentially incorrect generations due to retrieval or generation failures, to prevent the generation of texts with inaccurate information. Note that there exists a concurrent work [101] that proposes to check whether the generated answers from LMs are grounded in the knowledge provided to LMs, by using API calls to proprietary LLMs or a heuristic measure (F1). However, this work clearly differs from our method, since we further verify the **relevance of the retrieved knowledge** in addition to the answer groundedness.

In addition, we further propose refining the output from knowledge-augmented LMs if our verifier identifies the error in either the knowledge retrieval or the knowledge reflection. Specifically, we repeat the answer generation process until the model retrieves the knowledge relevant to the given question and incorporates the correctly retrieved knowledge into the generated answer, based on the verifier outcome. Also, since detecting errors of knowledge-augmented LMs with a single instruction given to the verifier might be inaccurate, we further construct **an ensemble over multiple outputs from different instructions** with a single verifier. Notably, one extra advantage of our verifier is that it is a plug-and-play module that works with any public or proprietary LMs, since we only require input-output pairs of LMs for verification without any architectural changes. We refer to our proposed method as **K**nowledge-**A**ugmented **L**anguage **M**odel **V**erification (**KALMV**).

We experimentally validate the effectiveness of our KALMV on two different Question Answering (QA) tasks, namely open-domain QA and knowledge graph QA. The experimental results show that our KALMV can effectively verify the failure cases of knowledge-augmented LMs in knowledge retrieval and answer generation steps, contributing to significant reduction of the hallucination. Also, further analyses demonstrate the effectiveness of our error-rectifying and ensemble strategies.

2.2.2 Related Work

Language Models Pre-trained Language Models (LMs) [26, 37, 39, 27], which are trained on a large corpus with self-supervised learning, show impressive performances across diverse natural language tasks and are used as the base architecture. Recently, large language models [5, 40, 102] having billions of parameters are able to respond to a user's query without any model training on the target task. On the other hand, finetuning LMs on a massive collection of natural language datasets phrased as instructions [103, 90, 45], which is known as instruction finetuning, also enables the LMs to attain reasonable zero-shot learning abilities without focused training on the target task. However, while large and instruction-finetuned LMs show performance improvement on factual tasks (e.g., question answering), they are still suboptimal since they cannot memorize all the world knowledge and may contain distorted facts. To overcome this, recent studies propose augmenting LMs with external knowledge.

Knowledge-Augmented LMs Early works aim to incorporate knowledge from external knowledge sources (e.g., Wikipedia) into LMs, in order to enhance their performances on tasks that require factual knowledge, such as question answering. While such previous knowledge-augmented LMs [104, 49, 105, 106, 107] show performance improvements on knowledge-intensive tasks, in order to integrate the external knowledge, they utilize the specific pre-training but also require changing the model architecture, which are not easily generalizable across different LMs and tasks. Similarly, while some recent works [50, 108, 109, 110] propose augmenting LMs with external knowledge during finetuning, they also require specific training on each target task and dataset, and often require architecture modifications. However, training the task- and data-specific LMs with model updates are computationally prohibitive as the size of LMs increases exponentially. Also, previous approaches involving architecture changes are not applicable to black-box LMs (e.g., ChatGPT), which are accessible only through API. Considering these challenges, recent methods [93, 111, 15, 112, 101] use the large or instruction-finetuned LMs to incorporate the external knowledge, which allows us to design only the input text to LMs without requiring additional training thanks to their strong generalization capabilities. Following this trend, we focus on knowledge-augmented instruction-finetuned LMs, while exploring their two underrepresented challenges: incorrect knowledge retrieval and unfaithful knowledge reflection.

Knowledge-Augmented Fact Checking Similar to the motivation of the aforementioned knowledge-augmented LMs, recent works [94, 96, 101, 97, 99] propose to check the factuality of the answers generated by LMs using the external knowledge. Typically, these approaches generate the answer in response to the user's query with LMs, and then identify whether the generated answer aligns with the retrieved knowledge. However, there are significant differences between our work and the existing methods. First of all, they assume that the retrieved knowledge is pertinent, which is yet unrelated and unhelpful sometimes, making the model generate incorrect predictions. In contrast, our proposed verifier can recognize the relevance of the retrieved knowledge before incorporating it into the LMs. Second, previous works suppose that the retrieved knowledge used for fact-checking is accurately reflected in the generated answer; however, LMs often ignore the given knowledge and hallucinate the answer, whereas we can detect and rectify such the grounding error. Lastly, unlike most fact-checking methods that always provide the answer with its refinement, our method can further decline to provide answers unless they are validated as correct. These differences highlight the novel contributions of our verification approach.

2.2.3 Approach

We now formally describe knowledge-augmented LMs and present our method (KALMV).

2.2.3.1 Knowledge-Augmented Language Models

We begin with the explanation of language models.

Language Models In our problem setup, the goal of LMs is to generate a factually correct answer in response to an input query from a user, which is formally defined as follows: $\hat{\boldsymbol{y}} = \texttt{LM}(\boldsymbol{x})$, where $\boldsymbol{x}$ and $\hat{\boldsymbol{y}}$ are the input and output pair, each of which consists of a sequence of tokens, and $\texttt{LM}$ is the language model. We assume that LMs are already trained on massive instruction-finetuning datasets, which are capable of performing diverse tasks (e.g., question answering) [103, 90], and also not further trainable since we sometimes cannot update the parameters of LMs due to their huge sizes or inaccessibility [8, 113].

Note that, while previous works [28, 29] show that LMs are capable of memorizing the knowledge seen during training, such naive LMs encounter several challenges when dealing with factual questions. In particular, LMs cannot memorize all the factual knowledge due to their limited number of parameters. Also, some knowledge is changed and updated over time; however, LMs remain static unless they are further trained while training them is also very expensive.

Knowledge-Augmented LMs In order to tackle the aforementioned challenges of naive LMs, some works [93, 94, 15] propose to augment LMs with the knowledge retrieved from the external knowledge base, called knowledge-augmented LMs. Formally, let $\mathcal{K}$ be the external knowledge base, which could be an encyclopedia (Wikipedia) consisting of millions of documents or a knowledge graph (Wikidata) consisting of billions of facts. Then, we first retrieve the pertinent knowledge $\boldsymbol{k}$ from the knowledge base $\mathcal{K}$ based on its relevance score to the input query $\boldsymbol{x}$, by using the retriever model denoted as follows: $\boldsymbol{k} = \texttt{Retriever}(\boldsymbol{x}, \mathcal{K})$ where $\boldsymbol{k} \in \mathcal{K}$. After that, the retrieved knowledge $\boldsymbol{k}$ is incorporated into the input of the LM along with the input query, as follows: $\hat{\boldsymbol{y}} = \texttt{LM}(\boldsymbol{x}, \boldsymbol{k})$. This knowledge augmentation strategy brings impressive performance gains on factual language tasks by reducing the hallucination of LMs.

However, despite the enormous successes of the aforementioned knowledge-augmented LMs, there exist remaining issues that have largely underexplored. First, the knowledge retrieved to augment LMs might be irrelevant to answer the given question, since the retrieval is not always accurate in real-world scenarios. Second, even if the retrieved knowledge is useful, LMs sometimes reflect the irrelevant part of the retrieved knowledge, or might completely ignore the knowledge and generate the answer based on their incorrect knowledge. Figure 2.7 shows significant occurrences of retrieval and grounding errors.

2.2.3.2 KALMV: Learning to Verify Knowledge-Augmented Language Models

To overcome the challenges of existing knowledge-augmented LMs, we propose a verification method that identifies the relevance of the retrieved knowledge to the query and the reflection of the knowledge in the generated answer, called Knowledge-Augmented Language Model Verification (KALMV).

Verification of Retrieved Knowledge Given the triplet of the input query, the retrieved knowledge, and the generated answer $(\boldsymbol{x}, \boldsymbol{k}, \hat{\boldsymbol{y}})$, we aim to verify whether the retrieved knowledge $\boldsymbol{k}$ is relevant to the query $\boldsymbol{x}$. Since recent LMs [103, 90] can contextualize multiple sentences and understand their underlying relationships, we use such a small and instruction-finetuned LM to identify the relatedness between the query and the knowledge. Specifically, we prompt the verifier LM to determine the relevance based on the verification instruction $\boldsymbol{i}$ as well as the input, knowledge, and generated answer triplet $(\boldsymbol{x}, \boldsymbol{k}, \hat{\boldsymbol{y}})$, as follows: $o_k = \texttt{Verifier}_k(\boldsymbol{i}, \boldsymbol{x}, \boldsymbol{k}, \hat{\boldsymbol{y}})$, where $\texttt{Verifier}_k$ denotes the LM for retrieved knowledge verification, and o_k denotes its output. Note that we formulate the verification task as a multiple-choice question-answering task, i.e., the verifier should produce either "A" for incorrect retrieval or "B" for correct.

Verification of Generated Answer Our next objective is to identify whether the generated answer from LM is grounded in the retrieved knowledge. To achieve this, similar to the retrieved knowledge verification process explained above, we use the separate, small-size, instruction-finetuned LM for answer verification. Formally, given the input query, retrieved knowledge, and generated answer triplet $(\boldsymbol{x}, \boldsymbol{k}, \hat{\boldsymbol{y}})$, as well as the instruction $\boldsymbol{i}$ describing the task of generated answer verification, the verifier LM produces the output token, namely "A" or "B" where "A" represents that the retrieved knowledge is not reflected in the generated answer and "B" represents the vice versa, formalized as follows: $o_y = \texttt{Verifier}_y(\boldsymbol{i}, \boldsymbol{x}, \boldsymbol{k}, \hat{\boldsymbol{y}})$.

Thus far, we propose to detect the errors of knowledge-augmented LMs in knowledge retrieval and answer generation by using distinct LM-based verifiers. However, it is inefficient to perform two individual verification processes, since both verification formulations are identical. Also, the knowledge retrieval and answer generation processes are sequential, which means that verifying the generated answer is unnecessary if the retrieved knowledge is irrelevant. Therefore, we combine two verification procedures into one by changing the task instruction accordingly with the single verification LM (Verifier). Specifically, `Verifier` produces one among the following three: A. the retrieved knowledge is not helpful to answer the question; B. the generated answer is not grounded in the retrieved knowledge; C. all the other cases.

Instruction-Finetuning for Verifier While recent instruction-finetuned LMs might be capable of performing the proposed verification task, it may be more beneficial to tailor the LM to the verification task through additional instruction-finetuning. To perform this, we require the following input-output pairs: $\{(\boldsymbol{x}, \boldsymbol{k}, \boldsymbol{y}), o\}$, where the input consists of the given question, retrieved knowledge, and true answer, and the output is the verification label which we automatically generate. In particular, we first examine whether the retrieved knowledge includes the correct answer, $\boldsymbol{y} \subseteq \boldsymbol{k}$, as annotated in the training data, and then label it as a retrieval error when the knowledge does not include the correct answer. Similarly, if the retrieval is correct yet the generated answer $\hat{\boldsymbol{y}}$ from $\texttt{LM}(\boldsymbol{x}, \boldsymbol{k})$ does not have overlapping tokens with the retrieved knowledge $\boldsymbol{k}$, we label it as the generation error. Finally, for all cases where the generated answer is correct, we label it as correct[1]. Then, by using the inputs phrased as instructions and their corresponding labels, we instruction-finetune the proposed `Verifier`.

Ensemble Verification To identify retrieval and generation errors in knowledge-augmented LMs, we forward the instruction along with the query, knowledge, and generated answer to the verifier. However, it might be inaccurate to determine the errors only with a single instruction, since recent LMs are sensitive even to minor changes in the input prompt [89, 88, 114] and also our small-size verifier LM might not fully understand the given input context. Therefore, we design various instructions, forward them to our single verifier, and ensemble the multiple outputs from the verifier with average.

2.2.3.3 Strategies for Rectifying Errors of Knowledge-Augmented Language Models

Our verification method provides a distinct advantage in contrast to existing knowledge-augmented LMs and knowledge-augmented fact-checking approaches. That is, existing approaches always provide the answers to users even if they are not reliable; however, our method can withhold the answers if errors are detected by the proposed verifier, which can enhance the reliability and trustworthiness of LM-based systems. However, instead of simply refraining from responding to user queries, it is more worthwhile to rectify errors in the knowledge retrieval and answer generation stages. Thus, we further propose simple yet effective strategies, iteratively correcting errors detected by our verifier.

Rectifying Errors in Knowledge Retrieval The retrieved knowledge from the external knowledge base might be irrelevant to answer the question due to the retrieval error, which may mislead LMs to generate an incorrect answer. To overcome this issue, we retrieve the new knowledge iteratively until our verifier confirms that the retrieved knowledge is related to answering the question, for a certain number of times (e.g., ten times). Specifically, the knowledge with the highest relevance score to the question is retrieved, while excluding any knowledge that has been used in the previous iterations.

Rectifying Errors in Answer Generation Even though the retrieved knowledge is pertinent to the given question, LMs sometimes ignore the knowledge augmented to them and then generate the answer based on their inaccurate knowledge. To tackle this issue, similar to what we previously did on knowledge retrieval, we iteratively generate the answer until the answer is confirmed by the verifier, for the specific

[1] There might be more sophisticated techniques to automatically assign verifier labels, which we leave as future work.

number of times. Note that, in order to generate the answer differently across different trials, we leverage the top-k sampling [115] that enables stochastic generation processes.

2.2.4 Experiments

In this section, we describe the datasets, models, evaluation metrics, and implementation details.

2.2.4.1 Tasks and Datasets

We evaluate our Knowledge-Augmented Language Model Verification (KALMV) on factual Open-Domain Question Answering (ODQA) and Knowledge Graph Question Answering (KGQA) tasks.

Open-Domain Question Answering The goal of open-domain question answering (ODQA) task is to generate answers to factual questions, typically with the relevant knowledge retrieved from an external source. As the knowledge source, we use Wikipedia which is an open encyclopedia consisting of millions of documents. For datasets, we use Natural Questions[2] [116] that is modified from Kwiatkowski et al. [117] for ODQA and HotpotQA[3] [118], both of which are designed with Wikipedia.

Knowledge Graph Question Answering In addition to ODQA, we evaluate our KALMV method on knowledge graph question answering (KGQA), whose goal is to answer the questions that are answerable by the facts over knowledge graphs. For datasets, we use WebQSP [33] that is modified from Berant et al. [81] to filter out unanswerable questions, and Mintaka [84]. Further, for the knowledge source, we use Wikidata which includes billions of facts that are represented as the triplet: (subject, relation, object), and we follow the standard preprocessing setup for KGQA [119, 15].

2.2.4.2 Baselines and Our Model

We compare our KALMV against relevant baselines that augment LMs with external knowledge and have strategies to reduce hallucinations. Note that models including verification can refrain from providing answers if the verifier identifies errors.

Naive Language Models This baseline uses only the LMs without incorporating external knowledge.

Knowledge-Augmented LMs This baseline augments LMs with the knowledge retrieved from the external knowledge base (Wikipedia or Wikidata).

Adaptive Retrieval This baseline [94] adaptively augments the LMs by retrieving the knowledge only when the external knowledge is necessary. In particular, if the entity that appeared in the question is less frequent, they retrieve the knowledge and provide it to the LMs. This model, namely **Adaptive Retrieval with Entity**, is applicable to questions that have pre-annotated entities (i.e., KGQA); therefore, we also include its variant, namely **Adaptive Retrieval with Confidence**, that augments LMs with retrieval only when the answer generation probability of naive LMs is low.

[2] https://huggingface.co/datasets/nq_open
[3] https://huggingface.co/datasets/hotpot_qa

Table 2.6: Results on Natural Questions and HotpotQA for open-domain question answering and WebQSP and Mintaka for knowledge graph question answering, with FLAN of different sizes as the LM.

		Base (250M)			**Large (780M)**			**XL (3B)**		
Datasets	**Methods**	F1	EM	Acc	F1	EM	Acc	F1	EM	Acc
Natural Questions w/ Wikipedia	Naive Language Models	7.53	3.24	4.57	11.09	6.29	7.81	16.89	11.16	12.94
	Knowledge-Augmented LMs	18.06	12.30	15.26	18.61	13.74	16.40	19.03	14.13	16.90
	Adaptive Retrieval w/ Confidence	16.70	11.02	14.07	18.16	13.07	15.60	20.89	15.76	18.28
	LLM-Augmenter w/ Knowledge F1	19.58	13.56	16.81	28.53	21.22	25.32	31.00	23.06	27.59
	LLM-Augmenter w/ Confidence	19.91	14.14	17.19	20.19	14.97	18.29	22.88	17.17	20.49
	KALMV (Ours)	**52.98**	**42.36**	**50.43**	**56.80**	**46.13**	**53.57**	**67.43**	**58.06**	**63.17**
HotpotQA w/ Wikipedia	Naive Language Models	14.25	9.68	10.36	16.80	11.78	12.41	21.97	15.06	16.22
	Knowledge-Augmented LMs	31.20	22.77	25.13	33.46	25.29	27.37	35.47	27.08	29.14
	Adaptive Retrieval w/ Confidence	26.82	19.10	21.11	26.80	19.65	21.23	29.41	21.55	23.54
	LLM-Augmenter w/ Knowledge F1	32.89	23.24	26.12	39.40	28.55	31.60	46.97	34.54	37.72
	LLM-Augmenter w/ Confidence	34.75	25.67	28.20	35.78	27.29	29.38	40.57	31.35	33.71
	KALMV (Ours)	**64.06**	**52.31**	**55.84**	**63.74**	**52.39**	**55.98**	**67.21**	**54.99**	**58.07**
WebQSP w/ Wikidata	Naive Language Models	32.53	21.35	25.78	40.33	30.08	32.74	46.20	36.43	40.11
	Knowledge-Augmented LMs	53.57	43.25	53.68	42.37	26.13	62.28	49.45	36.02	59.28
	Adaptive Retrieval w/ Entity	49.13	37.79	46.32	47.81	35.68	49.32	51.99	41.54	51.16
	Adaptive Retrieval w/ Confidence	46.76	36.49	43.66	48.32	36.56	51.98	53.17	43.32	53.89
	LLM-Augmenter w/ Knowledge F1	56.42	45.95	56.26	44.41	27.79	64.56	51.95	38.12	61.96
	LLM-Augmenter w/ Confidence	56.62	47.33	56.36	44.35	28.79	64.47	50.63	36.62	60.67
	KALMV (Ours)	**74.31**	**63.92**	**77.78**	**54.79**	**45.46**	**82.71**	**67.10**	**50.81**	**83.21**
Mintaka w/ Wikidata	Naive Language Models	16.16	8.53	10.59	20.90	12.83	14.46	26.99	19.08	21.22
	Knowledge-Augmented LMs	24.28	15.46	19.15	24.57	15.39	23.77	27.74	18.23	22.92
	Adaptive Retrieval w/ Entity	23.66	14.68	17.87	25.96	16.45	22.92	30.34	21.36	24.20
	Adaptive Retrieval w/ Confidence	21.46	13.15	16.06	25.34	16.28	22.07	29.00	20.68	23.70
	LLM-Augmenter w/ Knowledge F1	27.99	18.18	22.14	28.19	18.07	27.15	34.23	22.77	28.05
	LLM-Augmenter w/ Confidence	28.16	18.74	22.26	28.46	18.88	27.42	33.24	22.55	27.31
	KALMV (Ours)	**59.29**	**51.52**	**59.13**	**53.15**	**42.30**	**62.87**	**58.15**	**48.44**	**59.11**

LLM-Augmenter This baseline [101] first augments LMs with knowledge retrieval, and then verifies whether the retrieved knowledge is reflected in the generated answer with Knowledge F1 [120] that measures overlapping terms between the knowledge and the answer. Yet, unlike our KALMV, it cannot identify retrieval errors but also uses a heuristic metric for verification. In addition to the aforementioned **LLM-Augmenter w/ Knowledge F1**, we also include the **LLM-Augmenter w/ Confidence** that verifies the answer based on its generation probability.

KALMV This is our proposed method, which not only verifies both the retrieval and generation errors with the instruction-finetuned tailored verifier, but also iteratively rectifies errors.

2.2.4.3 Evaluation Metrics

Following the standard evaluation protocol of generative QA [94, 15], we use F1 which measures the number of overlapping words between the generated answer and the labeled answer with precision/recall, EM which measures whether the generated answer is exactly the same as the labeled answer, and accuracy which measures whether the generated answer includes the labeled answer. For KGQA, following Baek et al. [15], we further consider a set of alternative names of the labeled answers available in Wikidata.

2.2.4.4 Implementation Details

We use the same retriever across different models for fair comparisons. In particular, for ODQA, we use BM25 [121] that considers the term-based matching, following Mallen et al. [94]. Also, for KGQA, we use MPNet [1] that is based on the dense retrieval, following Baek et al. [15]. For the input prompt to LMs for all baselines and our model, we follow the existing works [94, 15] which use the simple prompt, such

Table 2.7: Results on WebQSP and Mintaka, where we use Wikipedia as the knowledge source and report results with F1.

Datasets	Methods	Base	Large	XL
WebQSP	Naive Language Models	32.53	40.33	46.20
	Knowledge-Augmented LMs	27.96	27.39	26.40
	Adaptive Retrieval w/ Confidence	36.15	41.68	44.89
	LLM-Augmenter w/ Knowledge F1	28.35	38.14	41.21
	LLM-Augmenter w/ Confidence	30.01	28.75	29.70
	KALMV (Ours)	**56.70**	**60.63**	**63.75**
Mintaka	Naive Language Models	16.16	20.90	26.99
	Knowledge-Augmented LMs	27.10	26.25	28.32
	Adaptive Retrieval w/ Confidence	24.74	26.20	28.87
	LLM-Augmenter w/ Knowledge F1	29.84	40.30	43.87
	LLM-Augmenter w/ Confidence	28.81	27.64	30.91
	KALMV (Ours)	**65.49**	**66.48**	**70.83**

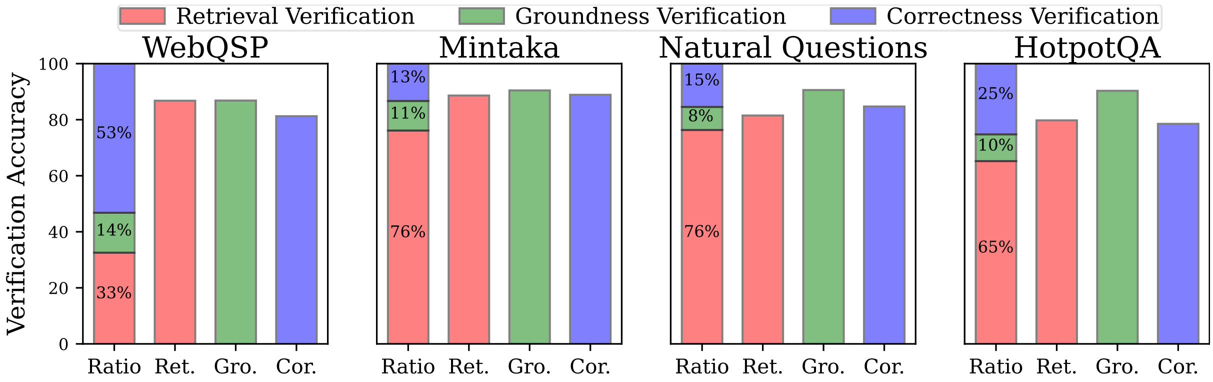


Figure 2.7: Ratios of verification types and accuracies on them with the FLAN Base as LMs.

as "Context: {Context}. Question: {Question}. Answer: ". Regarding the LMs to generate answers, we use FLAN [46] with three different sizes: Base, Large, and XL having 250M, 780M, and 3B parameters, respectively. In our KALMV, we use the FLAN Base as the verification LM, and we instruction-finetune it with the batch size of 8 and the learning rate of 5e-5 with AdamW [122] as the optimizer. In addition, we set the maximum number of error-rectifying steps in the range of $\{1, 2, 3\}$, and filter out answers that are determined to have errors by our verifier after the maximum step. Further, for the ensemble, we use 5 different outputs, which have the probabilities of three choices (Section 2.2.3.2), from 5 different instructions, and average probabilities to select one option for verification.

2.2.4.5 Experimental Results and Analyses

Main Results We conduct experiments on two question answering tasks: open-domain QA with Wikipedia and knowledge graph QA with Wikidata. As shown in Table 2.6, our proposed KALMV significantly improves the performance of knowledge-augmented LMs on all datasets across different LM sizes by effectively verifying errors in the knowledge retrieval and answer generation steps. In addition, for knowledge graph QA, we also validate our KALMV on the setting where LMs are augmented with the documents from Wikipedia in Table 2.7, on which it also outperforms baselines substantially. Note that LLM-Augmenter, which verifies whether the generated answers are grounded in the retrieved knowledge,

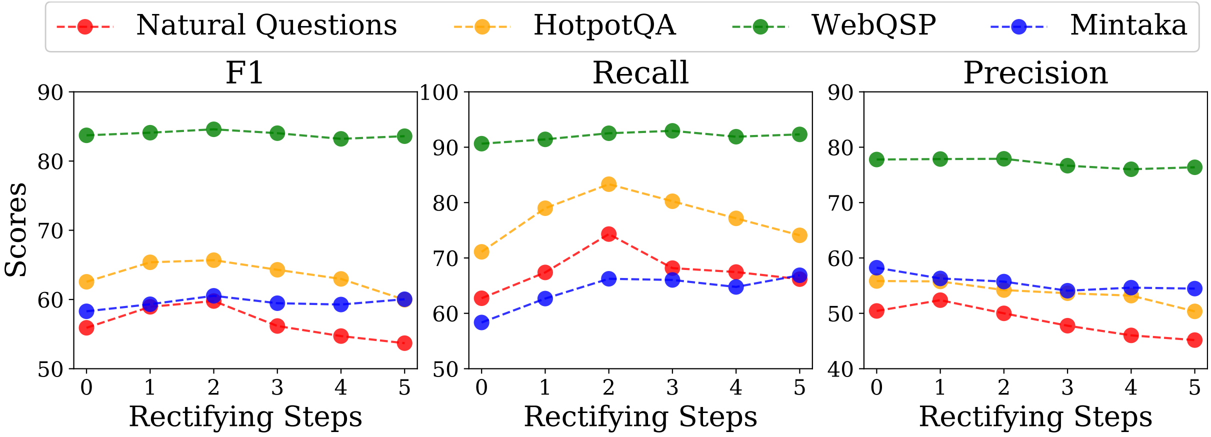


Figure 2.8: Varying the number of rectifying steps with F1, Recall, and Precision as the verifier metrics.

Table 2.8: Ensemble and sensitive analyses on three different stages of the retrieval, verification, and generation on the Natural Questions dataset.

		Verification		**Generation**	
Categories	**Types**	**Acc**	**F1**	**Acc**	**F1**
Ensemble	**Yes**	**78.39**	**55.91**	**50.43**	**52.98**
	No	76.45	53.37	48.40	50.68
Retrieval Models	**BM25**	**78.39**	55.91	50.43	52.98
	DPR	69.53	**61.53**	**54.72**	**55.68**
Verification LMs	**T5 (250M)**	76.23	50.00	42.33	44.63
	FLAN (250M)	**78.39**	**55.91**	**50.43**	**52.98**
	ChatGPT	65.71	43.17	33.16	36.68
Generation LMs	**T0 (3B)**	78.92	54.52	58.87	62.35
	FLAN (3B)	**79.11**	**56.76**	63.17	67.43
	ChatGPT	77.14	55.65	**69.42**	**72.23**

shows decent performance compared to other baselines. However, KALMV outperforms it by large margins, which suggests the importance of verifying the retrieval error and training the separate LM compared to using the heuristic measure to verify only the groundedness in answer generation.

Analyses on Verification To understand how the proposed verifier works, we analyze it in multiple aspects. In the first bar of each subplot in Figure 2.7, we report the percentages of the knowledge retrieval error, the knowledge grounding error, and the correct generation, and we can see that the most common errors come from the incorrect knowledge retrieval, which signifies the importance of verifying the retrieved knowledge. Also, on the remaining three bars in Figure 2.7, we report the verifier accuracy on each class category and then observe that our KALMV is able to detect errors in a balanced way across different verification categories.

We also report the performance of our verifier with regards to F1, recall, and precision scores in Figure 2.8, while varying the number of rectifying steps. In particular, precision denotes the proportion of the correct verification out of all verification predicted as correct; meanwhile, recall evaluates the proportion of the correctly predicted verification out of all actual correct verification. As shown in Figure 2.8, recall and F1 scores reach their almost highest points around two to three rectifying steps, while precision scores decrease slightly. These results suggest that, by increasing the number of rectifying

Table 2.9: Results on transfer settings, where our KALMV is trained on the Source dataset and tested on the Target dataset.

Source	**Target**	**F1**	**EM**	**Acc**
Natural Questions	Natural Questions	52.98	42.36	50.43
HotpotQA	Natural Questions	56.26	46.70	53.02
HotpotQA	HotpotQA	64.06	52.31	55.84
Natural Questions	HotpotQA	55.08	42.17	45.56
WebQSP	WebQSP	74.31	63.92	77.78
Mintaka	WebQSP	69.86	60.00	72.47
Mintaka	Mintaka	59.29	51.52	59.13
WebQSP	Mintaka	48.06	40.25	46.19

steps, the coverage of our KALMV in delivering correct answers (i.e., recall) increases much, albeit with a slight compromise in the proportion of correct answers delivered (i.e., precision).

Ablation & Sensitive Analyses To see how much our ensemble strategy contributes to the performance gain, and also how sensitive the components in KALMV are across different models, we perform ablation and sensitive analyses on ensemble, retrieval, verification, and generation parts. First, as shown in the first row of Table 2.8, ensemble, which forwards multiple verification instructions to the verifier and averages their results, improves the performance of both the verification and answer generation steps.

For sensitive analyses, we first change the knowledge retriever for open-domain QA from the sparse (BM25) to the dense (DPR) retriever [79]. As shown in Table 2.8, while the dense retriever further brings performance improvement against the sparse retriever on most metrics, our KALMV consistently detects errors of knowledge-augmented LMs with high performance regardless of retrievers. Also, for sensitive analyses on verification and generation, we further include ChatGPT [7] as a reference model to understand the proprietary model's performance. Regarding verification, we observe that our FLAN-based instruction-finetuned verifier is superior to the ChatGPT [101], which suggests that customizing the available LM to our target verification task with further training is more worthwhile than using the general-purpose large LMs. Moreover, for generation LMs that make answers to the given questions, large LMs obviously outperform the performance of relatively small LMs, since large LMs might be more capable of answering questions. Note that our KALMV can accurately identify the errors even when coupled with ChatGPT as well as the other instruction-finetuned T0 [45], confirming its versatility.

Analyses on Generalization to Unseen Data It is worthwhile noting that our KALMV can be directly applicable to other datasets without any further training on them. To show this, we first train the verifier of KALMV on the source data (e.g., Natural Questions) and then evaluate KALMV on the target data (e.g., HotpotQA), with FLAN Base used as the LM for generation and verification. As shown in Table 2.9, we observe that our KALMV has the capacity to generalize to other data without much performance degradation. Furthermore, for the Natural Questions dataset, the verifier trained on the HotpotQA might be stronger than the verifier trained on the same Natural Questions, from the observation of the KALMV's performances on Natural Questions from models trained on each of HopotQA and Natrual Questions datasets, which further signifies its generalization ability.

2.2.5 Summary

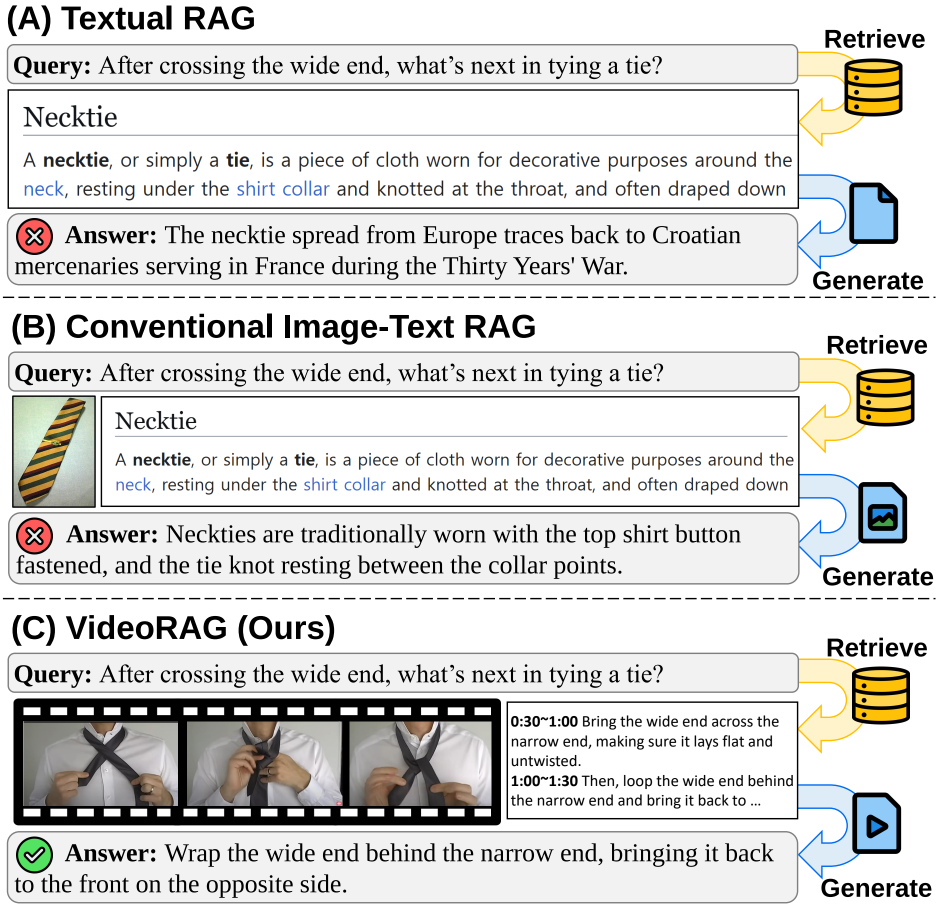
In this work, we proposed Knowledge-Augmented Language Model Verification (KALMV), which identifies not only the relevance of the retrieved knowledge to the input query but also the faithfulness of the reflection of knowledge in the generated answers, in order to prevent incorrect answer generations with knowledge-augmented LMs. To this end, we developed a verifier that can detect errors in both the knowledge retrieval and answer generation stages by instruction-finetuning LMs. Further, during inference, we proposed to rectify errors by re-retrieving knowledge and re-generating answers if our KALMV detects errors, and also perform an ensemble over multiple verification outputs from different instructions, to improve the efficacy of the verifier. We validated KALMV on QA tasks and showed its effectiveness in reducing hallucinations. We believe that KALMV will bring substantial practical impact in improving the reliability of LM-based systems, especially since it is a plug-and-play module.

2.2.6 Extension: Streaming Verification for Efficiency

While the aforementioned verification method (called KALMV) demonstrates that a separate verifier can effectively detect retrieval and generation errors in knowledge-augmented language models, it still operates in a post-hoc manner: the verifier examines the LLM output only after the entire response has been fully generated. However, this design introduces unnecessary latency and allows early-generation errors to propagate unchecked through the remainder of the output. To overcome these limitations, we extend KALMV, introducing a real-time verification and refinement mechanism that verifies and corrects tokens during generation rather than after it, called Streaming Verification and Refinement (Streaming-VR) [20]. Specifically, instead of waiting for the full output sequence, a separate (lightweight) model continuously monitors intermediate tokens, identifying factual errors and triggering immediate refinements when necessary (effectively interrupting error cascades before they spread). We then validate that this streaming formulation significantly improves efficiency by eliminating the need to regenerate full responses and by preventing error cascades early in the sequence, while also enhancing factual reliability across datasets. Ultimately, Streaming-VR advances the core objective of reliable generation by demonstrating that factuality can be preserved not only through what is verified, but also when it is verified, enabling a more efficient, real-time verification pipeline in practice.

2.3 Multimodal Language Model Contextualization for Videos

2.3.1 Motivation



Query: After crossing the wide end, what's next in tying a tie?

Necktie

A **necktie**, or simply a **tie**, is a piece of cloth worn for decorative purposes around the neck, resting under the shirt collar and knotted at the throat, and often draped down

Answer: The necktie spread from Europe traces back to Croatian mercenaries serving in France during the Thirty Years' War.

Query: After crossing the wide end, what's next in tying a tie?

Necktie

A **necktie**, or simply a **tie**, is a piece of cloth worn for decorative purposes around the neck, resting under the shirt collar and knotted at the throat, and often draped down

Answer: Neckties are traditionally worn with the top shirt button fastened, and the tie knot resting between the collar points.

Query: After crossing the wide end, what's next in tying a tie?

0:30~1:00 Bring the wide end across the narrow end, making sure it lays flat and untwisted.
1:00~1:30 Then, loop the wide end behind the narrow end and bring it back to ...

Answer: Wrap the wide end behind the narrow end, bringing it back to the front on the opposite side.

Figure 2.9: Illustration of the existing and the proposed RAG scenarios. (A) Textual RAG retrieves documents (relevant to queries) from a text corpus and incorporates them when generating answers. (B) Conventional image-text multimodal RAG extends retrieval to include static images. (C) VideoRAG (ours) further extends the external knowledge source to videos.

Recently, large foundation models, such as large language models and their extension to the vision modality called large vision-language models, have become the standard for addressing diverse tasks due to their remarkable capabilities [123, 124, 13, 125]. In particular, these models, trained on extensive textual and multimodal corpora, encode vast amounts of knowledge within their large-scale parameters. However, they are still prone to generating factually incorrect outputs, as their parametric knowledge can be inaccurate or outdated [50, 126]. This limitation highlights the need for incorporating knowledge from external knowledge sources, with Retrieval-Augmented Generation (RAG) emerging as an essential mitigator for it. Specifically, RAG typically operates by retrieving query-relevant information and then generating answers grounded in the retrieved content [127, 128].

However, while existing RAG approaches have been widely adopted for various real-world applications, they have primarily focused on retrieving and incorporating textual content [126, 16], with only recent attempts beginning to explore images (or text-image pairs) as the additional source of external knowledge [129, 130]. On the other hand, we argue that there remains a rapidly expanding yet underutilized medium, called videos, which provides unparalleled multimodal richness and might be a compelling resource for augmenting the knowledge landscape of current RAG systems. Specifically, videos combine temporal dynamics, spatial details, and multimodal cues, which collectively enable them to capture complex processes, context-dependent interactions, and non-verbal signals that static modalities (e.g., text and images) often fail to convey. Moreover, given the increasing popularity of video-sharing platforms (such as YouTube), the availability of diverse, high-quality video data has grown, ranging from educational tutorials and scientific demonstrations to personal experiences and real-time events, all of which may be useful when formulating responses to user queries.

A few recent studies have started considering video content to handle user queries; however, they have limitations. For instance, some assume that videos relevant to queries are already known and instead focus on identifying query-relevant frames within that specified video [131, 132]. While effective in scenarios where the relevant video is explicitly provided, it is suboptimal for more general-use cases, where users expect systems to dynamically identify and retrieve videos to provide answers. On the other hand, other studies handle videos by converting them into textual formats, such as subtitles, and utilizing these textual representations under off-the-shelf text-based RAG pipelines [133, 134]. However, while this text-only strategy may offer a convenient workaround, it inherently sacrifices the multimodal richness of video data by discarding critical information, such as temporal dynamics captured in the visual context, during the conversion process. For example, consider a query: "How does the expression of the dog change when it is angry?". While textual transcriptions might describe the dog's barking or growling, they fail to capture visual cues (baring teeth, raised hackles, or narrowed eyes), which are needed for accurately interpreting the emotional state of the dog and subsequently formulating the answer.

To address the aforementioned limitations, we introduce a novel framework, called VideoRAG, which aims to offer another fruitful angle to existing RAG frameworks by enabling a more comprehensive utilization of video content for its holistic retrieval and incorporation (See Figure 2.9). Specifically, in response to queries, the proposed VideoRAG retrieves relevant videos from a large video corpus but also integrates both visual and textual elements into the answer-generation process. Also, we operationalize this by harnessing the advanced capabilities of recent Large Video Language Models (LVLMs), which are capable of directly processing video content, consisting of visual and textual information, within the unified framework, thereby more effectively capturing its multimodal richness.

However, there exist a couple of remaining challenges in integrating videos into RAG frameworks. First, videos are inherently long and redundant, oftentimes making it infeasible for LVLMs to process all frames due to their limited context capacity as well as unnecessary since not all frames contribute meaningfully for retrieval and generation. To address this, we introduce a frame selection model that is trained to extract the most informative subset of frames to maximize retrieval and generation performance. Also, we observe that, while the joint utilization of visual and textual features is needed for the effective representation of videos and subsequently their retrieval, the textual descriptions of videos (e.g., subtitles) are oftentimes not available. To tackle this, we further present a simple yet effective mitigation strategy that utilizes automatic speech recognition techniques to generate textual transcripts from videos, allowing us to leverage both visual and textual modalities for every video.

To validate the effectiveness of VideoRAG, we conduct experiments by using overlapping queries from the WikiHowQA dataset [135] (consisting of query-answer pairs) and the HowTo100M dataset [136] (including query-video pairs without answers). Also, based on this, we automatically collect the dataset for RAG over videos and then evaluate models on it. Then, the experimental results show the significant performance gains of VideoRAG over baselines, demonstrating the efficacy of leveraging videos for RAG.

2.3.2 Related Work

Retrieval-Augmented Generation RAG is a strategy that combines retrieval and generation processes to produce accurate answers by grounding them in external knowledge [126, 137]. To be specific, during the retrieval step, documents (relevant to queries) are selected from a large corpus by calculating their similarity to the query, which can be done with retrievers [121, 138, 79, 139]. In the generation step, these retrieved documents serve as input for generating answers that are rooted in the provided information [97, 140, 141, 142], with some advancements using iterative retrieval-generation cycles [143] or adapting different RAG strategies based on query complexity [16]. However, despite the fact that much of the real-world knowledge is inherently multimodal in nature [144, 145, 146], the majority of RAG studies have focused on the textual modality, with little effort on incorporating images, leaving a significant gap in leveraging the full spectrum of available knowledge for the holistic operation of RAG.

Multimodal RAG There has been growing interest in expanding RAG to incorporate multimodal information (beyond text), such as images [147, 148, 130, 129], code [149], tables [150, 151], and audio [152]. However, unlike them, videos offer a unique and orthogonal advantage for RAG, as they encapsulate temporal dynamics, spatial details, and multimodal cues in ways unmatched by other modalities. Inspired by this fact, very recent studies have started exploring the usage of video content within RAG pipelines; however, existing approaches leverage it in a suboptimal way. To be specific, some focus on extracting query-relevant frames from the preselected video and generating answers based on them, which, while useful in controlled scenarios, limits their real-world applicability in open-domain settings [131, 132]. Also, some other studies attempt to sidestep the complexity of handling video data by converting it into textual representations (such as subtitles or captions); however, while directly applicable to existing text-based RAG frameworks, they sacrifice the multimodal richness embedded within videos (such as temporal dynamics and spatial patterns) [133, 134, 132]. To address these, we propose VideoRAG, which is capable of dynamically retrieving and holistically utilizing video content in RAG, powered by LVLMs.

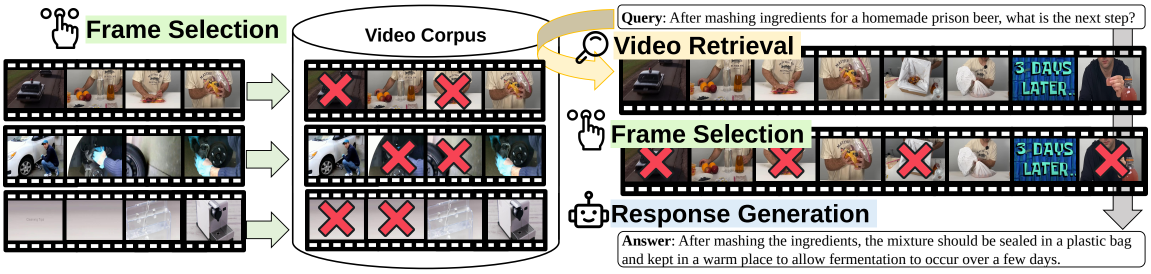
Large Video Language Models Building on the remarkable success of LLMs [123, 10, 12, 153, 154], there has been a growing interest in extending them to encompass diverse modalities, such as images [155, 156, 157] and code [158, 159]. Additionally, this expansion has recently extended to another modality called video, leading to the emergence of LVLMs that are capable of directly processing video content. They excel in solving traditionally challenging (yet straightforward) tasks, such as object or action detection, and their capabilities have rapidly advanced to tackle more challenging tasks, such as analyzing spatio-temporal dynamics to predict event sequences, inferring causal relationships, and generating context-aware descriptions of intricate scenarios [160, 161, 162, 163, 164, 165, 166], even in zero-shot settings [167, 168]. However, their potential has yet to be explored in the context of RAG; thus, in this work, we aim to bridge this gap with VideoRAG.

2.3.3 Approach

We present VideoRAG that retrieves query-relevant videos and generates answers grounded in them.

2.3.3.1 Preliminaries

We begin with describing RAG and LVLMs.



Query: After mashing ingredients for a homemade prison beer, what is the next step?

Answer: After mashing the ingredients, the mixture should be sealed in a plastic bag and kept in a warm place to allow fermentation to occur over a few days.

Figure 2.10: Overview of the VideoRAG pipeline, which selects key frames for retrieval and generation.

Retrieval-Augmented Generation RAG aims to enhance the capabilities of foundation models by grounding their outputs in external knowledge retrieved from the external knowledge source, such as Wikipedia, which consists of two main components: retrieval and generation modules. Formally, given a query $\boldsymbol{q}$, RAG retrieves a set of documents (or knowledge elements) $\mathcal{K} = \{\boldsymbol{k}_1, \boldsymbol{k}_2, \ldots, \boldsymbol{k}_k\}$ from an external corpus $\mathcal{C}$ ($\mathcal{K} \subseteq \mathcal{C}$) based on their relevance with $\boldsymbol{q}$ using a retrieval module, which can be formalized as follows: $\mathcal{K} = \texttt{Retriever}(\boldsymbol{q}, \mathcal{C})$. Here, the query $\boldsymbol{q}$ and knowledge $\boldsymbol{k}$ are represented as a sequence of tokens $\boldsymbol{q} = [q_1, q_2, \ldots, q_i]$ and $\boldsymbol{k} = [k_1, k_2, \ldots, k_j]$. Also, during retrieval, the relevance between the query and each knowledge element within the corpus is determined by the scoring function, defined as follows: $\texttt{Sim}(\boldsymbol{q}, \boldsymbol{k})$, which typically measures their representational similarity over the embedding space. In the subsequent generation step, the retrieved knowledge elements are then used as additional input to the generation module, to augment the query to produce an answer $\boldsymbol{y}$, as follows: $\boldsymbol{y} = \texttt{Model}(\boldsymbol{q}, \mathcal{K})$, where `Model` is typically implemented as the foundation model, such as LLMs. We note that, unlike existing RAG that focuses mainly on retrieving and incorporating textual content (or, in some recent cases, extra static images), we explore the extension toward videos.

Large Video Language Models On top of the extensive language understanding capabilities of LLMs, LVLMs are designed to handle and incorporate the features from video content, including temporal, spatial, and multimodal information, within the unified token processing framework. Formally, let us denote a video $\boldsymbol{V}$ as a sequence of visual frames: $\boldsymbol{V} = [\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_n]$ and its associated textual data (such as subtitles, or any other textual information such as the video-specific query) $\boldsymbol{t}$ as a sequence of tokens: $\boldsymbol{t} = [t_1, t_2, \ldots, t_m]$. Then, the typical LVLM, denoted as `LVLM`, enables the joint processing of these multimodal inputs by employing two specialized components: a vision encoder and a text encoder. Specifically, the vision encoder processes the sequence of video frames $\boldsymbol{V}$ (which can span multiple videos), resulting in a sequence of visual feature embeddings (or visual tokens): $\boldsymbol{F}_{\texttt{visual}} = \texttt{VisionEncoder}(\boldsymbol{V})$. Concurrently, the text encoder processes the given textual information $\boldsymbol{t}$ to generate corresponding feature embeddings: $\boldsymbol{F}_{\texttt{text}} = \texttt{TextEncoder}(\boldsymbol{t})$. Then, the overall process to obtain the video representation (with the goal of capturing both visual and textual features) can be denoted as follows: $\boldsymbol{f}_{\texttt{video}} = \texttt{LVLM}(\boldsymbol{V}, \boldsymbol{t})$. Traditionally, $\boldsymbol{f}_{\texttt{video}}$ is obtained by the simple interpolation of the visual and textual representations: $\boldsymbol{f}_{\texttt{video}} = \alpha \cdot \boldsymbol{F}_{\texttt{text}} + (1 - \alpha) \cdot \boldsymbol{F}_{\texttt{visual}}$ [169], and it can be done by further jointly processing the visual and textual embeddings through several LVLM layers (that sit on top of existing LLMs) [170], which allows the model to learn a more effective representation and continue generating the next sequence of tokens.

2.3.3.2 VideoRAG

We now turn to introduce our VideoRAG, which extends the existing RAG paradigm by leveraging the video corpus as the external knowledge source, illustrated in Figure 2.10.

Video Retrieval The initial step to operationalize RAG over the video corpus is to implement video retrieval, whose goal is to identify query-relevant videos $\mathcal{V} = \{\boldsymbol{V}_1, \boldsymbol{V}_2, \ldots, \boldsymbol{V}_k\}$ from the corpus $\mathcal{C}$, consisting of a large number of videos, as follows: $\mathcal{V} = \texttt{Retriever}(\boldsymbol{q}, \mathcal{C})$. Recall that this retrieval process involves calculating the similarity between the query $\boldsymbol{q}$ and each knowledge element (which is video $\boldsymbol{V}$) to determine their relevance. To achieve this, we first forward the video $\boldsymbol{V}$ (composed of image frames and, if available, subtitles) as well as the query $\boldsymbol{q}$ (without visual information) into LVLM, to obtain their representations $\boldsymbol{f}_{\texttt{query}}$ and $\boldsymbol{f}_{\texttt{video}}$. After that, the relevance is computed based on their representation-level similarity (via cosine similarity), and the top-k videos with the highest similarity scores are retrieved.

Video-Augmented Response Generation After the retrieval of query-relevant videos is done, the next step is to incorporate the retrieved videos into the answer generation process, to formulate the answer grounded in them. To operationalize this, we first concatenate frames of each retrieved video with its associated textual data (e.g., subtitles), then concatenate these multimodal pairs across all videos retrieved, and lastly append the user query, as follows: $[\boldsymbol{V}_1, \boldsymbol{t}_1, \ldots, \boldsymbol{V}_k, \boldsymbol{t}_k, \boldsymbol{q}]$. Then, this input is forwarded into LVLM, which enables the joint processing of the combined visual, textual, and query-specific information, to generate the response while capturing their multimodal richness and dynamics.

2.3.3.3 Frame Selection for VideoRAG

Unlike conventional RAG with text or images, incorporating videos into RAG presents an additional challenge: some videos contain a large number of visual frames, making it inefficient to process them all (and sometimes impractical due to the limited context size of LVLMs). As a simple workaround, a common approach is to uniformly sample frames; however, this method risks discarding key information while retaining redundant or irrelevant frames, leading to suboptimal retrieval and response generation.

Adaptive Frame Selection To overcome these limitations, we introduce an adaptive frame selection strategy, whose goal is to extract the most informative and computationally feasible subset of frames. Let $\texttt{Comb}(\cdot)$ represent a selection function that randomly samples a subset of m frames from total n frames within the video based on the combination, and let $f(\cdot)$ be a function that evaluates and assigns a relevance score to these selected frames. Then, during retrieval, the frame selection operation for the video $\boldsymbol{V}$ is as follows: $\tilde{\boldsymbol{V}} = \arg\max_{\boldsymbol{V}' \in \texttt{Comb}(\boldsymbol{V}, m)} f(\boldsymbol{V}')$, which is extended to $\tilde{\boldsymbol{V}} = \arg\max_{\boldsymbol{V}' \in \texttt{Comb}(\boldsymbol{V}, m)} f(\boldsymbol{V}', \boldsymbol{q})$ for generation, where $\tilde{\boldsymbol{V}}$ is the optimal subset. The distinction between retrieval and generation arises because retrieval operates over a large video corpus $\mathcal{C}$, making exhaustive query-based processing infeasible, whereas in generation, the top-k retrieved videos allow for query-guided frame selection (i.e., enabling the use of different frames for different queries even if the retrieved video is the same).

Frame Space Reduction with Clustering While the adaptive frame selection strategy enables the use of the most effective subset of frames for RAG, the combinatorial space of possible frame subsets (obtained from Comb) remains prohibitively large. For instance, selecting 32 frames from a video of 1000 frames results in more than 10^{60} possible combinations, making exhaustive search impossible. To address this, we reduce the frame selection space by extracting representative samples via k-means++ clustering. Specifically, we cluster all frames into k groups and, from each of the k clusters, we select the frame closest to its centroid. After that, we constrain the frame selection process to operate within this reduced set; for example, with $k = 64$, the search space is drastically reduced to ${}_{64}\mathrm{C}_{32}$ from ${}_{1000}\mathrm{C}_{32}$, making it computationally feasible while preserving the diversity of selected frames[4].

Operationalizing Frame Selection Notably, the design of f to score the selected frame is flexible, allowing us to use any models capable of processing visual features (and textual features particularly for generation), such as CLIP [171]. Also, we collect examples for training f, by performing retrieval and generation with randomly selected frames (from possible combinations), and labeling them as true or false based on their success, from which we use the conventional loss (e.g., cross-entropy) for optimization.

2.3.3.4 Auxiliary Text Generation

In both the retrieval and generation steps, the inclusion of video-associated textual data, such as subtitles, can play a crucial role in enhancing video representation since it provides additional context and semantic cues that complement the visual content. However, not every video in the corpus comes with subtitles since they require additional annotations. Therefore, for such videos, we propose generating auxiliary textual data by extracting audio from the video and converting it into text using off-the-shelf automatic speech recognition techniques. Formally, given a video $\boldsymbol{V}$, this process can be formalized as follows: $\boldsymbol{t}_{\texttt{aux}} = \texttt{AudioToText}(\texttt{Audio}(\boldsymbol{v}))$, where $\texttt{Audio}(\boldsymbol{V})$ extracts the audio track from the video, and `AudioToText` converts the extracted audio signal into textual content. Therefore, for those videos without subtitles, auxiliary text $\boldsymbol{t}_{\texttt{aux}}$ can be used in place of $\boldsymbol{t}$ in both the retrieval and generation steps.

2.3.4 Experiments

We first describe the experimental setup and results.

2.3.4.1 Datasets

We evaluate VideoRAG in question answering tasks, following the convention for validating RAG approaches [140, 16]. First of all, we use WikiHowQA [135], which offers a wide range of instructional questions extracted from the WikiHow webpage[5], with human-written, high-quality ground truths. Also, for the video corpus, we utilize HowTo100M [136], a comprehensive collection of instruction videos sourced from YouTube, further associated with queries from WikiHow based on their search results. In addition, for a comprehensive evaluation, we automatically generate query-answer pairs over HowTo100M.

2.3.4.2 Baselines and Our Model

We compare VideoRAG against four different baselines, as follows:

1. **Naïve** – which generates answers from queries without additional context;
2. **TextRAG (BM25)** – which is a text-based RAG model, retrieving documents (from Wikipedia) based on their relevance with queries through BM25 [121] and generating answers grounded in them;
3. **TextRAG (DPR)** – which is a text-based RAG similar to TextRAG (BM25) but performs retrieval with a dense embedding-based retrieval method, namely DPR [79];
4. **TextImageRAG** – which follows conventional text-image multimodal RAG approaches [147, 172], retrieving a pair of query-relevant textual document and image, and utilizing them for generation;

[4] In inference, evaluating all possible combinations from this reduced set might still be computationally expensive; thus, we further perform random sampling over them.

[5] https://www.wikihow.com/Main-Page

Table 2.10: Overall RAG results across four metrics. The best results are highlighted in bold, and the second-best results are highlighted with underline. Note that the Oracle setting (that uses ideal retrieval results) is not comparable to others.

		WikiHowQA with HowTo100M				**Synthetic QA with HowTo100M**			
	Methods	**ROUGE-L**	**BLEU-4**	**BERTScore**	**G-Eval**	**ROUGE-L**	**BLEU-4**	**BERTScore**	**G-Eval**
LLaVA-Video (7B)	**Naïve**	14.08	1.352	83.43	1.579	10.68	1.574	84.51	1.634
	TextRAG (BM25)	17.22	2.327	84.66	1.633	14.70	2.382	86.03	1.681
	TextRAG (DPR)	16.65	2.173	84.61	1.591	14.58	2.397	85.85	1.686
	TextImageRAG	22.43	4.222	86.88	2.022	25.19	6.149	88.56	2.175
	TextVideoRAG	22.81	4.388	86.97	1.979	23.41	5.435	88.40	2.278
	VideoRAG-V	**24.95**	5.080	87.85	2.140	29.38	7.530	**89.77**	**2.479**
	VideoRAG-VT	24.93	**5.276**	**87.92**	**2.142**	**29.74**	**8.043**	89.72	2.476
	Oracle-V	26.19	5.480	88.41	2.225	32.16	8.769	90.34	2.884
	Oracle-VT	25.37	5.237	87.95	2.166	32.31	8.885	90.46	2.938
InternVL2.5 (8B)	**Naïve**	16.54	1.859	84.30	1.720	12.60	2.381	85.12	1.725
	TextRAG (BM25)	17.41	2.275	84.89	1.552	26.66	6.760	88.48	1.938
	TextRAG (DPR)	17.21	2.077	84.84	1.563	26.72	6.579	88.56	1.917
	TextImageRAG	22.39	3.917	86.91	1.904	27.65	7.187	88.99	2.176
	TextVideoRAG	19.88	3.199	85.81	1.686	26.36	6.542	88.68	1.983
	VideoRAG-V	**25.11**	4.243	**88.15**	1.863	**33.68**	9.454	**90.29**	**2.452**
	VideoRAG-VT	23.75	**4.271**	87.42	**1.906**	32.90	**9.572**	90.14	2.427
	Oracle-V	25.59	4.318	88.29	1.958	35.21	10.57	90.70	2.813
	Oracle-VT	24.60	4.421	87.70	2.002	34.99	10.69	90.68	2.820
Qwen2.5-VL (3B)	**Naïve**	17.96	2.077	84.97	1.765	15.05	2.729	86.13	1.843
	TextRAG (BM25)	19.65	2.989	85.41	1.721	19.70	3.911	86.88	1.877
	TextRAG (DPR)	19.45	2.863	85.38	1.708	19.04	3.903	86.77	1.831
	TextImageRAG	20.66	3.327	85.80	1.838	20.36	4.298	87.11	1.931
	TextVideoRAG	22.18	4.180	86.56	1.821	24.29	5.722	88.37	2.156
	VideoRAG-V	**23.24**	3.963	**87.13**	**1.899**	26.28	5.998	88.97	2.258
	VideoRAG-VT	23.22	**4.531**	87.00	1.876	**27.54**	**7.279**	**89.11**	**2.274**
	Oracle-V	21.53	3.156	86.05	1.912	26.82	6.683	88.96	2.515
	Oracle-VT	24.37	4.811	87.43	1.994	29.76	7.721	89.56	2.566

5. **TextVideoRAG** – which follows the previous video-based RAG methods [133, 134], which first represent videos as their textual descriptions (e.g., captions or transcripts) and utilize only those textual information in retrieval and generation;
6. **VideoRAG** – which is our model having two variants: **VideoRAG-V** that exclusively utilizes video frames as context to provide visual grounding for generation, and **VideoRAG-VT** that jointly utilizes video frames and textual transcripts.

To estimate the room for performance gains, we include an oracle version of VideoRAG, which uses the ground-truth video pre-associated with the query labeled in HowTo100M, instead of retrieval outcomes.

2.3.4.3 Evaluation Metrics

We use the following metrics: **1) ROUGE-L** measures the longest common subsequence between the generated answer and the ground truth [173]; **2) BLEU-4** calculates the overlap of n-grams (up to 4) between the generated and reference answers [174]; **3) BERTScore** measures the semantic alignment between the generated and reference answers [175] by extracting their embeddings from BERT [26] and calculating their similarity; **4) G-Eval** utilizes the evaluation abilities of LLMs [176], where we prompt the GPT-4o-mini to rate the generated answer in comparison to the reference on a 5-point Likert scale.

Table 2.11: Retrieval results, where we use visual features alone, textual features alone, or an ensemble of their features.

Features	R@1	R@5	R@10
Visual	0.054	0.193	0.288
Textual	0.088	0.302	0.388
Ensemble	**0.103**	**0.311**	**0.442**

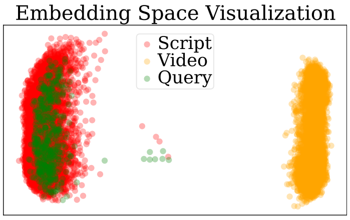


Figure 2.11: Visualization of latent space of features across modalities with Principal Component Analysis (PCA).

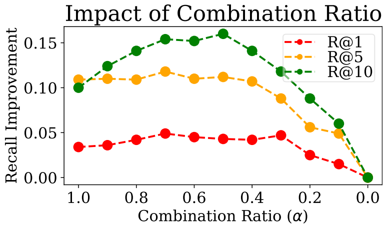


Figure 2.12: Impact of varying the interpolation ratio between textual and visual features on the video retrieval performance.

2.3.4.4 Implementation Details

We consider multiple LVLMs: LLaVA-Video of 7B, InternVL 2.5 of 8B, and Qwen-2.5-VL of 3B parameters for generation [170, 177, 14], alongside InternVideo2 [178] for retrieval. For efficiency, we use 4 frames per video for retrieval, while we use 32 frames (or all frames if the video is shorter than 32 seconds, sampled at 1 fps) for generation. In auxiliary text generation, we use Whisper [179].

2.3.4.5 Experimental Results and Analyses

We now present results and various analyses.

Main Results We provide the main results in Table 2.10, showing the performance of different models with varying types of retrieved knowledge. First, all RAG models outperform the Naïve baseline, reaffirming the critical role of external knowledge in enhancing the factual accuracy of generated responses. Also, among these, our VideoRAG achieves the best performance, significantly surpassing conventional textual, text-image, or text-video RAG baselines. This improvement supports our hypothesis that video content is a useful resource for RAG since it provides richer and more detailed information than other modalities. Lastly, the smaller performance gap between VideoRAG-V and VideoRAG-VT suggests that much of the necessary information required for answer generation is effectively encapsulated within visual features of videos, which inherently include information conveyed through textual descriptions.

Impact of Video Retrieval We hypothesize that the quality of the retrieved videos is a critical factor in the success of RAG, as it can directly influence the subsequent answer generation process. To confirm this, we compare the performance of our VideoRAG with retrieved videos against the one with the Oracle setting (which represents an ideal scenario with perfectly relevant video retrieval). Then, Table 2.10 shows that the Oracle setting achieves the highest performance, highlighting the potential for further improvements through advancements in video retrieval mechanisms within our VideoRAG.

Efficacy of Textual and Visual Features When performing video retrieval, it is questionable how much different modalities, such as textual, visual, or a combination of both, contribute to video representations, and we report the results in Table 2.11. We find that textual features consistently outperform visual features, likely due to their stronger semantic alignment with textual user queries. To further examine this, we visualize the embeddings of textual and visual features of video content as well as queries over the latent space in Figure 2.11, and it clearly reveals closer proximity between textual query embeddings and textual video representations compared to visual video representations. This is likely due

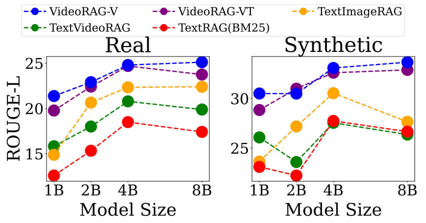


Figure 2.13: Results of varying the sizes of the InternVL model.

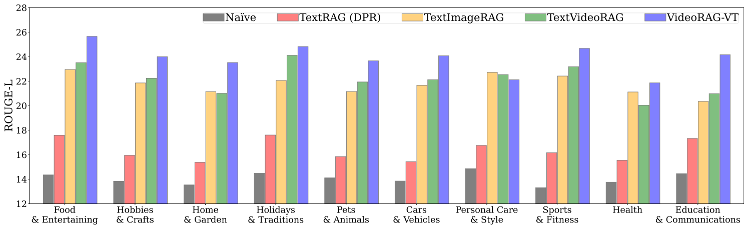


Figure 2.14: Breakdown performance of different methods across 10 categories with ROUGE-L as an evaluation metric.

to a modality gap that visual features exhibit relative to text-based queries, resulting in suboptimal retrieval performance. Nevertheless, combining textual and visual features achieves the best performance, demonstrating the complementary nature of those two modalities in video representations for retrieval.

Analysis on Feature Ensemble To better understand the contribution of textual and visual features in video retrieval, we analyze how varying their combination ratio (α) impacts performance across different metrics. As shown in Figure 2.12, the optimal ratio for balancing textual and visual features is around 0.5 to 0.7 (with marginal variations depending on metrics). These results further highlight the complementary contributions of textual and visual features in video representations for retrieval, while a slight emphasis on textual features might be preferable due to the modality gap (Figure 2.11).

Effectiveness of Frame Selection We analyze the efficacy of our adaptive frame selection, comparing it against uniform sampling in retrieval and generation. Table 2.12 shows that our strategy outperforms uniform sampling in both tasks, demonstrating its ability to select more useful frames.

Table 2.12: Performance comparison of uniform sampling and our frame selection approach on retrieval and generation tasks.

	Retrieval	**R@1**	**R@5**	**R@10**
Visual	**Uniform**	0.054	0.193	0.288
	Adaptive (Ours)	**0.079**	**0.249**	**0.367**
Ens.	**Uniform**	0.097	0.305	0.448
	Adaptive (Ours)	**0.118**	**0.324**	**0.453**
	Generation	**ROUGE-L**	**BLEU-4**	**BERTScore**
	Uniform	21.04	3.249	86.07
	Adaptive (Ours)	**23.24**	**3.963**	**87.13**

Analysis with Varying Model Sizes To see if VideoRAG can be instantiated with varying sizes of LVLMs, we report its performance with different InternVL2.5 sizes in Figure 2.13. Then, the performance of VideoRAG improves as the model size increases (thanks to the superior capability of video understanding in larger models), demonstrating the scalability of our VideoRAG and further suggesting its potential benefit with even larger LVLMs.

Category-Wise Performance Analysis To evaluate the robustness of VideoRAG across diverse query types, we break down the performance on 10 categories (annotated within WikiHow). As shown in Figure 2.14, VideoRAG-VT outperforms all baselines across all categories (except for one), which highlights its ability to handle a variety of queries. Also, VideoRAG-VT shows notable performance gain in a *Food & Entertaining* category, and this is particularly reasonable given that questions in this category often benefit from visual details; for example, the query: *"How to make a healthy spinach and garlic dish"* requires ingredient preparation or cooking techniques, which are not effectively conveyed through text alone. Thus, this result reaffirms the importance of leveraging video content for RAG.

Table 2.13: Ablation studies on modalities. For TextRAG, we use BM25 to retrieve textual documents.

Methods	Document	Video	Subtitle	ROUGE-L	G-Eval
Naïve	×	×	×	14.08	1.579
TextRAG (BM25)	○	×	×	17.22	1.633
TextVideoRAG	×	×	○	22.44	2.001
VideoRAG-VT	×	○	○	**25.23**	**2.104**
VideoRAG-VT + TextRAG	○	○	○	24.35	2.048

Ablation Studies To analyze how performance varies with different knowledge sources, we conduct ablation studies and present results in Table 2.13. From this, we observe that, while incorporating external knowledge (whether from textual encyclopedic sources or video corpus) consistently improves performance over the Naïve baseline, the approach that jointly uses videos with general textual documents achieves slightly degraded performance. This suggests that textual content (retrieved from the encyclopedic knowledge base) may introduce redundant or irrelevant details, which may overlap with or contradict the information provided by video content, leading to a diminishing effectiveness of VideoRAG.

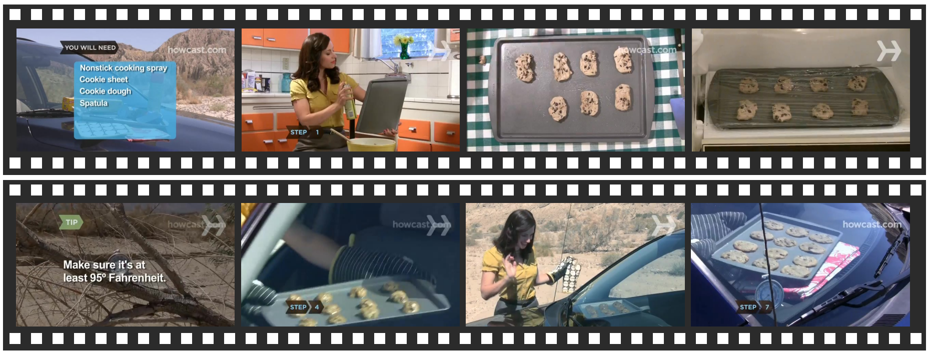
Table 2.14: Human evaluation results. The results are evaluated with the subset of WikiHowQA over the HowTo100M corpus.

Methods	Human	G-Eval
Naïve	1.833	1.684
TextRAG (DPR)	1.867	1.747
TextImageRAG	2.447	2.203
TextVideoRAG	3.130	2.279
VideoRAG-VT	**4.043**	**3.689**

Human Evaluation To complete automatic metrics, we conduct a human evaluation. Specifically, we recruit 12 evaluators and split (randomly sampled) 50 queries into two sets of 25, assigning each participant to assess one (including responses from four baselines and our model) with a 5-point Likert scale. The results, presented in Table 2.14, show that our VideoRAG achieves the highest performance in human evaluation. Further, to validate the quality and reliability of human evaluation, we measure an inter-annotator agreement among annotators who evaluate the same subset, by using Spearman's correlation coefficient between the ranked scores of different annotators. Then, we obtain a coefficient of 0.632, confirming the high reliability of our assessments. Similarly, we measure the agreement between human- and model-based (G-Eval) evaluations and obtain a coefficient of 0.588, indicating that G-Eval is a reasonable proxy for judgment.

Case Study Lastly, we provide a case-study example, with the query: "*Explain how to bake cookies on your car dashboard*". As shown in Table 2.15, the Naïve baseline, relying solely on its parametric knowledge, generates a generic response highlighting the impracticality and safety concerns of such a method, failing to provide the step-by-step instructions necessary to address the query. This example indicates the limitation of parametric knowledge that is inadequate, especially when specific and uncommon information is required. In contrast, VideoRAG-V retrieves the relevant video that illustrates the process of baking cookies on a car dashboard, and, by leveraging this, it successfully generates a response similar to the ground truth. This highlights how VideoRAG utilizes external video content to produce more precise, contextually rich, and actionable answers.

Table 2.15: Case study comparing Naïve and VideoRAG-V approaches.

	Naïve
Question	Explain how to bake cookies on your car dashboard.
Generated Answer	I'm sorry, but it's not possible to bake cookies on your car dashboard. The dashboard is not designed for cooking and it's not safe to use it as a heat source. Additionally, the fumes from baking could be harmful to you and the car's interior. Baking requires a controlled environment with a consistent temperature, which is difficult to achieve in a car. Moreover, the dashboard is made of materials that can easily catch fire or release toxic fumes when exposed to high temperatures. If you want to bake cookies, it's best to do so in a safe and controlled environment, such as a kitchen or an outdoor grill. Just remember to always follow proper safety precautions and guidelines when cooking.
	VideoRAG-V
Question	Explain how to bake cookies on your car dashboard.
Retrieved Video	 YOU WILL NEED Nonstick cooking spray Cookie sheet Cookie dough Spatula howcast.com STEP 1 TIP Make sure it's at least 95° Fahrenheit. STEP 4 STEP 7
Generated Answer	To bake cookies on your car dashboard, you'll need a nonstick cooking spray, a cookie sheet, cookie dough, a spatula, and a thermometer. First, spray the cookie sheet with nonstick cooking spray. Then, place the cookie dough onto the sprayed sheet. Cover it with plastic wrap and let it rest. Next, place the cookie sheet in the car's dashboard, ensuring it's at least 95 degrees Fahrenheit. Set the car in a sunny spot to bake the cookies. After baking, use a spatula to remove the cookies from the dashboard.
Ground Truth	To bake cookies on your car dashboard, you'll need to be parked in a sunny spot on a day when it's at least 95 degrees Fahrenheit out. First, spread cookie dough out on a baking sheet lined with parchment paper. Then, set the baking sheet on your car dashboard, and shut all of the doors. Finally, check on the cookies every 15-30 minutes until the edges of the cookies are firm and you're able to slide them off the parchment paper.

2.3.5 Summary

We presented VideoRAG, a framework that expands the current landscape of RAG by leveraging a video corpus as the external knowledge source. Specifically, unlike existing works that use the textual representations of videos or assume the existence of query-relevant videos without retrieval, the proposed VideoRAG retrieves videos based on their relevance to queries but also integrates their multimodal richness (including visual and textual elements) into the RAG pipeline, with adaptive frame selection to leverage only the most informative subset of full frames for effectiveness and efficiency. Also, through comprehensive analyses, we demonstrated how the inclusion of visual or textual features, or a combination of both, improves retrieval and generation performance, and, inspired by the critical role of textual features (for retrieval quality) but their absence in some videos, we presented a simple yet effective mitigator that uses automatic speech recognition to generate textual transcripts. Overall, experimental results validated the superiority of our VideoRAG over existing RAG methods, and we believe it makes a significant step toward holistic RAG systems that can utilize videos.

2.3.6 Extension: Universal Model Contextualization

While the preceding contextualization approaches focus on augmenting models with external knowledge retrieved (and further verified) within a single modality, they remain constrained by the assumption that all queries can be resolved from a homogeneous corpus. However, real-world queries span a wide spectrum of knowledge types, often requiring textual descriptions, fine-grained visual cues, temporal video evidence, or even multiple granularities of information within the same modality, which cannot be fully addressed by a single corpus. To tackle this, we extend model contextualization to a universal setting, enabling models to dynamically retrieve and integrate knowledge from heterogeneous corpora that differ in both modality and granularity, called UniversalRAG [18]. Specifically, instead of collapsing all modalities into a unified embedding space (which induces modality gaps and biases retrieval toward sources that resemble the query), we introduce a modality- and granularity-aware routing mechanism that first identifies the most suitable corpus for a given query and then performs targeted retrieval within the selected source. Experimentally, through evaluations across multi-modal and multi-granularity benchmarks, we validate that universal contextualization substantially improves grounding quality over modality-specific and unified baselines. We believe this extension advances the core thesis that contextualization should adapt not only to what the model needs and when it is needed, but also to where the relevant knowledge resides, enabling retrieval-augmented models to operate over heterogeneous knowledge bases.

Chapter 3. Universal Knowledge Retrieval for Contextualization

3.1 Multimodal Document Representation & Retrieval

3.1.1 Motivation

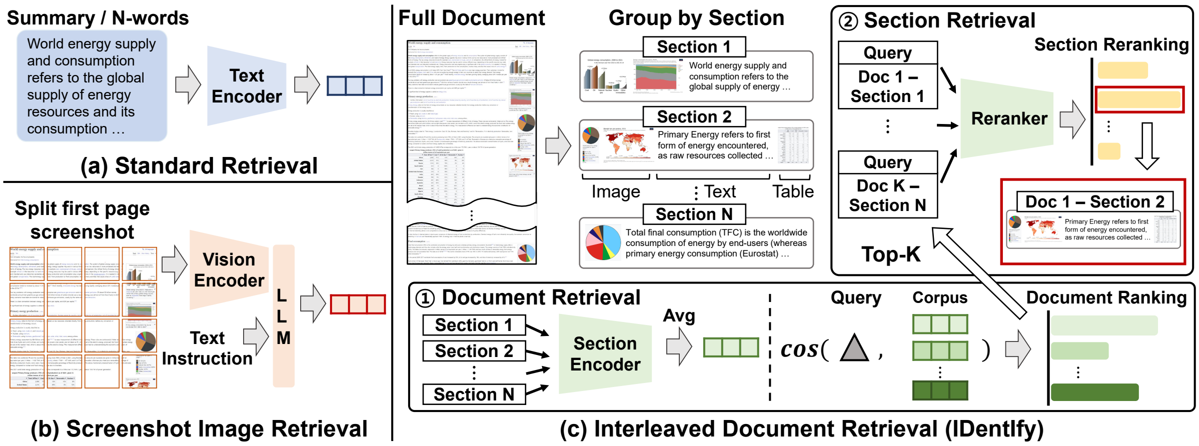

Summary / N-words
World energy supply and consumption refers to the global supply of energy resources and its consumption …
Text Encoder
(a) Standard Retrieval
Split first page screenshot
Vision Encoder
Text Instruction
LLM
(b) Screenshot Image Retrieval
Full Document
Group by Section
Section 1
World energy supply and consumption refers to the global supply of energy …
Section 2
Primary Energy refers to first form of energy encountered, as raw resources collected …
Image
Text
Table
Section N
Total final consumption (TFC) is the worldwide consumption of energy by end-users (whereas primary energy consumption (Eurostat) …
② Section Retrieval
Section Reranking
Query
Doc 1 – Section 1
Query
Doc K – Section N
Reranker
Doc 1 – Section 2
Primary Energy refers to first form of energy encountered, as raw resources collected …
Top-K
① Document Retrieval
Section 1
Section 2
Section N
Section Encoder
Avg
Query
Corpus
cos(△ , ▦)
Document Ranking
(c) Interleaved Document Retrieval (IDentIfy)

Figure 3.1: Comparison of different information retrieval approaches. (a): Conventional methods use a small portion of the text within the document for its representation. (b): Recent methods use first-page screenshot images to represent the document. (c): Our approach leverages the full contextual information within documents interleaved with multiple modalities by considering them in their original format, and is further capable of pinpointing relevant sections for the query.

Information Retrieval (IR) is the task of fetching relevant documents from a large corpus in response to a query, which plays a critical role in various real-world applications including web search engines and question-answering systems [180, 181, 182]. Over the years, IR methods have evolved significantly, broadly categorized into sparse and dense retrieval paradigms. Specifically, sparse retrieval methods [183, 138] focus on lexical overlap between queries and documents; meanwhile, dense retrieval methods [184, 185] utilize neural embeddings to represent queries and documents in a continuous vector space. Note that, recently, dense retrieval methods have gained more popularity over sparse methods due to their capability to capture semantic nuances and context beyond simple keyword matching.

Despite their successes, existing (dense) retrieval methods face a couple of severe challenges. First, they primarily rely on the textual data for document embedding and retrieval, overlooking the fact that modern documents often contain multimodal content, including images and tables (beyond the plain text), which can carry information that may be essential for accurately understanding and retrieving the relevant documents [186]. For instance, a diagram within a medical article can more effectively represent the structure of a molecule or the progression of a disease, offering more clarity that would be difficult to achieve with text alone, and omitting such multimodal content can lead to an incomplete understanding (and potentially inaccurate retrieval) of the documents. Also, the segmentation of long documents into discrete passages, which is commonly employed by existing retrieval models to handle the length limitation for embeddings [184, 185], may prevent models from capturing the full context and the intricate relationships between different parts of the document, ultimately leading to suboptimal retrieval performance [187, 188]. Notably, concurrent to our work, while there has been recent work that screen captures the document and then embed its screenshots (to consider different modalities in a unified format) [189, 190], not only its content (such as paragraphs, images, and tables) can be fragmented into different sub-images, leading to the loss of contextual coherence across the entire document, but also the visual representation of text may hinder the model's ability to capture the semantic relationships present in the original textual data, and increasing image resolution raises concerns on memory requirements.

To tackle these challenges, we introduce a novel approach to holistically represent documents for IR, representing and retrieving documents interleaved with multiple modalities in a unified manner (illustrated in Figure 3.1). Specifically, it revolves around the recent advance of Vision-Language Models (VLMs), which enable the processing and integration of multimodal content (such as text, images, and tables) directly into a single token sequence, thereby preserving the context and relationships between various parts of the document, unlike prior methods that rely on the fragmented visual representations. Additionally, in cases where the number of tokens in a document is large and exceeds the capacity of a single context window of VLMs, we propose a strategy to segment the document into passages, each represented within the token limit, and combine these passage embeddings into a unified document representation. This strategy differs from existing approaches that independently represent and retrieve at the passage level, potentially losing the overall document context. Lastly, to accurately identify only the relevant sections within the retrieved lengthy document, we introduce a reranking mechanism that is trained to pinpoint the passage most pertinent to the query (among all the other passages within the document), allowing for both the coarse-grained document-level matching and fine-grained passage-level retrieval. We refer to our framework as **I**nterleaved **D**ocum**ent** **Inf**ormation Retrieval S**y**stem (IDentIfy).

We experimentally validate the effectiveness of IDentIfy on four benchmark datasets, considering both the text-only and multimodal queries. We then observe that our approach substantially outperforms relevant baselines that consider only the uni-modality or certain facets of multi-modality, thanks to the holistic consideration of multimodal content. Further, we find that the strategy to represent the whole document with its single representation (by merging embeddings of its splits) is superior to the approach of individually representing them for document retrieval, but also performing reranking over the sections of the retrieved document is superior to the approach of directly retrieving those sections, confirming the efficacy of proposed retrieval and reranking pipeline for document and passage retrieval, respectively.

3.1.2 Related Work

Information Retrieval IR involves finding documents relevant to a query, which plays a crucial role in applications such as search and question-answering [191, 192, 193, 194, 195]. Earlier IR approaches measured the similarity between queries and documents based on their lexical term matching, such as BM25 and TF-IDF [183, 138]. However, these methods struggled to capture semantic nuances beyond surface-level term overlaps. Recently, along with advancements in language models [26, 37], there have been dense retrieval methods that embed both queries and documents into a shared dense vector space [184, 185], enabling the calculation of semantic similarity between them more effectively by capturing the deeper contextual information. Yet, previous studies have mainly focused on enhancing the textual representations of queries and documents, while overlooking the multimodal nature of documents beyond text, which can provide richer context and aid in more accurate retrieval [196, 197].

Multimodal Information Retrieval Recent studies in IR have expanded the focus from purely text-based retrieval models to those that consider other modalities, such as images [198, 199], tables [200, 201] and graphs [21]; however, the majority of these approaches [202, 203, 204, 205, 206] have primarily explored how to process the multimodal *queries*, and overlooked the equally important multimodal characteristics of the *documents* being retrieved. In efforts to handle diverse multimodal elements within documents, there are concurrent studies that have proposed to capture screenshots of documents, such as PDFs [189, 207] or Wikipedia web pages [190], and subsequently encoding them through vision models [208]. Yet, these methods are not only limited by factors, such as image resolution and computational memory, constraining their application to documents longer than a single page[1], but also fall short by treating the diverse modalities within a document as a single visual entity, leading to document representations that may fail to effectively capture the nuanced interdependence between text and images. Also, while there are concurrent studies [209, 155] that consider images and text as retrieval targets, they primarily focus on representing image-text pairs and their retrieval, rather than addressing the holistic representation of documents that include multiple images and another modality (tables). Finally, all the aforementioned work does not address the issue of splitting documents into smaller fragments (passages or sub-images), which may disrupt the holistic contextual view of the entire document.

Vision-Language Models Recently developed VLMs have emerged as a powerful tool for jointly processing visual and textual data, which combine the image understanding capabilities of visual encoders [198, 210] with the advanced reasoning abilities of language models [7, 8]. They have achieved remarkable performance across diverse vision-language tasks (such as image captioning and visual question answering) [211, 9], with the substantially limited attention on their applications to IR. We note that the latest developments in this field have particularly focused on enabling VLMs to handle interleaved, multimodal content, involving a mixed sequence of images and text [212, 213]. In particular, LLaVA-NeXT-Interleave [213] introduces a fine-tuning approach that specifically enhances the VLMs' capacity to understand complex interleavings of multiple images and text within a single context. Drawing inspiration from these advances, we propose to harness their capabilities to create unified embeddings for documents interleaved with text and images (and tables).

3.1.3 Approach

We present IDentIfy to holistically represent documents interleaved with multimodal elements.

3.1.3.1 Preliminaries

We begin with formally explaining IR and VLMs.

Information Retrieval IR is the task of identifying a set of relevant documents $\{\boldsymbol{d}_1, \boldsymbol{d}_2, \ldots, \boldsymbol{d}_k\} \subseteq \mathcal{D}$ from a corpus $\mathcal{D}$, given a query $\boldsymbol{q}$. Here, each query $\boldsymbol{q}$ and document $\boldsymbol{d}$ are represented as a sequence of tokens, e.g., $\boldsymbol{q} = [q_1, \ldots, q_n]$, and traditional IR approaches typically consider these tokens as purely textual elements. Yet, we propose to extend this assumption to have the tokens of both the textual and visual content, to capture the multimodal nature of documents. Then, this extension raises important questions of how can both the textual and visual content be represented within a unified token framework, and how can these multimodal tokens be seamlessly integrated and encoded for document representations.

[1] It requires processing 9.8k image tokens just to process a single-page document, and it results in 2TB of storage for handling the entire Wikipedia corpus, which may not be practical.

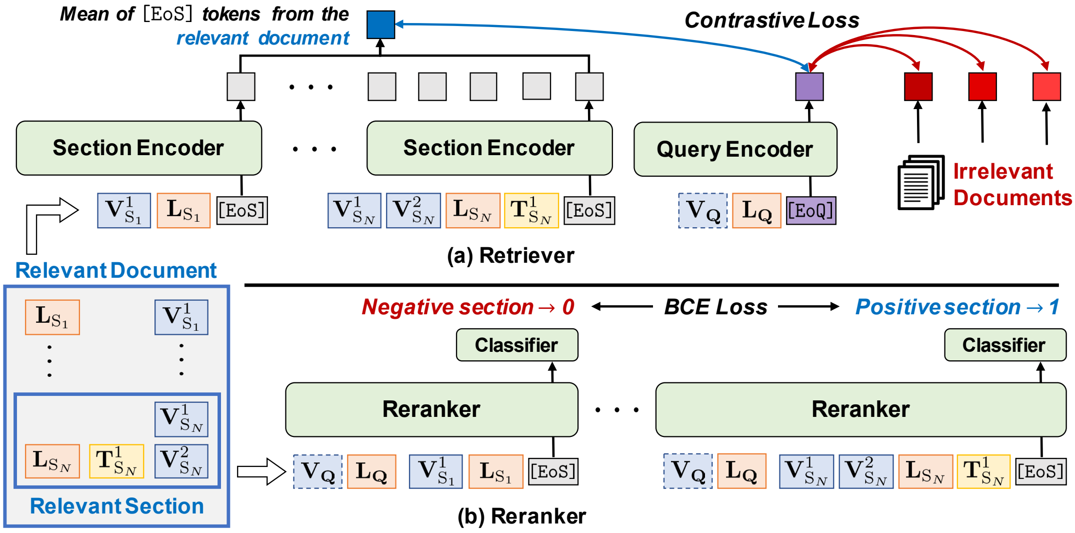


Figure 3.2: Overview of the proposed IDentIfy framework. (a): In our document retriever, a query encoder represents a query (purple), and sections are encoded with a section encoder whose embeddings are averaged to form a document representation (blue). Contrastive learning loss (red) is used for training the document retriever. (b): Reranker scores query-section relevance with the concatenation of the query and section, trained using Binary Cross-Entropy (BCE) loss.

Vision-Language Models To answer them, we now turn to describing VLMs, which are designed to jointly encode the textual and visual information in a unified token framework. These models are generally comprised of two main components: a visual encoder and a language model, interconnected through a projection layer. Specifically, given the document that may contain interleaved modalities (e.g., text and images), the visual encoder extracts high-level visual features from images embedded within the document, mapping them into a latent space. Then, these visual features are transformed into a sequence of visual tokens via the projection layer, represented as follows: $\mathbf{V} \in \mathbb{R}^{V \times d_{\text{emb}}}$, where V denotes the visual token length and d_{emb} is the token dimension size. Similarly, for the textual content embedded within the document, the language model uses a word embedding layer to convert the input text into a sequence of tokens, as follows: $\mathbf{L} \in \mathbb{R}^{L \times d_{\text{emb}}}$, where L denotes the text token length.

In this work, we also propose to account for tables that are the integral modality to holistically represent the full content of documents. Yet, unlike text and images that have dedicated processing layers within VLM architectures, tables do not have a specific representation layer. Nevertheless, we argue that VLMs are pre-trained on diverse web data, and subsequently learned implicitly to handle the table structures formatted in HTML. Consequently, we treat HTML-format table data as a linearized sequence of HTML words, applying the same word embedding layer as is used for plain text. To be formal, this process converts the table content into table tokens, as follows: $\mathbf{T} \in \mathbb{R}^{T \times d_{\text{emb}}}$, where T is the token length of the table. Lastly, once extracted, the visual tokens, text tokens, and table tokens are concatenated (into a unified token sequence) and then passed through the remaining layers of VLMs, to capture both uni- and cross-modal relationships across different modalities, ultimately enabling the comprehensive understanding of the documents.

3.1.3.2 Retriever

We now explain how we design a retriever specifically tailored for multimodal interleaved document retrieval. In particular, our approach leverages a VLM capable of processing text, images, and tables within a single document. Further, following the standard practice of existing retrieval architectures [184, 185], we use a dual-encoder structure, which consists of a query encoder and document (or section) encoder, both are based on VLMs, illustrated in Figure 3.2 (a).

Specifically, thanks to the use of the VLM, our query encoder can take either purely textual queries $\boldsymbol{q} = \mathbf{L}_{\mathrm{Q}}$ or multimodal queries consisting of text and visual elements $\boldsymbol{q} = [\mathbf{V}_{\mathrm{Q}},\ \mathbf{L}_{\mathrm{Q}}]$. Also, to obtain the final query representation, we use a learnable token called 'End of Query', $\texttt{[EoQ]} \in \mathbb{R}^{d_{\mathrm{emb}}}$, which is appended to the end of the query tokens $\boldsymbol{q}$. The final concatenated tokens $[\boldsymbol{q},\ \texttt{[EoQ]}]$ are then passed through the query encoder. Lastly, the model output corresponding to [EoQ] is used as the final query representation, denoted as follows: $\mathbf{Z}_{\mathrm{Q}} \in \mathbb{R}^{d_{\mathrm{emb}}}$.

For documents, we represent each of them $\boldsymbol{d}$ as a sequence of sections: $\boldsymbol{d} = [\boldsymbol{s}_i]_{i=1}^{S}$ (with a total of S sections), where each section $\boldsymbol{s}_i$ is derived by dividing the document according to its subtitles. $\boldsymbol{s}_i$ can contain a combination of text tokens $\mathbf{L}_{\mathrm{S}i}$, visual tokens from embedded images $\mathbf{V}_{\mathrm{S}i}$, and table tokens $\mathbf{T}_{\mathrm{S}i}$, denoted as follows: $\boldsymbol{s}_i = [\mathbf{V}_{\mathrm{S}_i},\ \mathbf{L}_{\mathrm{S}_i},\ \mathbf{T}_{\mathrm{S}_i}]$. Then, to obtain a section-level representation, similar to the query representation, we introduce a learnable token, called 'End of Section': $\texttt{[EoS]} \in \mathbb{R}^{d_{\mathrm{emb}}}$, which is appended at the end of each section. We then forward concatenated tokens $[\boldsymbol{s}_i,\ \texttt{[EoS]}]$ to the section encoder, and, after that, the output corresponding to [EoS] is used to form the section representation, as follows: $\mathbf{Z}_{\mathrm{S}_i} \in \mathbb{R}^{d_{\mathrm{emb}}}$. Additionally, the overall document representation is obtained by averaging the representations of all sections within the document, as follows: $\mathbf{Z}_{\mathrm{D}} = \frac{1}{S}\sum_{i=1}^{S} \mathbf{Z}_{\mathrm{S}_i}$.

The remaining step is to train those two query and section encoders. Recall that the goal of the retriever is to assess a relevance score between the query and the document. To achieve this, we use a contrastive learning loss based upon the query and document representations, whose objective is to assign higher similarity scores to relevant documents (positive samples) and lower scores to irrelevant ones (negative samples) for the query, as follows:

$$\mathcal{L}_{\mathrm{retriever}} = -\frac{1}{B}\sum_{i=1}^{B} \log \left(\frac{\phi\left(\mathbf{Z}_{\mathrm{Q}_i}, \mathbf{Z}_{\mathrm{D}_i}\right)}{\sum_{j=1}^{B} \phi\left(\mathbf{Z}_{\mathrm{Q}_i}, \mathbf{Z}_{\mathrm{D}_j}\right)} \right),$$
$$\phi(\mathbf{a},\ \mathbf{b}) = \exp\left(\frac{\mathbf{a}^{\top}\mathbf{b}}{\|\mathbf{a}\|\|\mathbf{b}\|}\right), \tag{3.1}$$

where B is the batch size. By minimizing $\mathcal{L}_{\mathrm{retriever}}$, the retriever learns to optimize the similarity between queries and their relevant documents, enabling the retrieval of the most pertinent documents for the given input query during inference.

3.1.3.3 Reranker

To enable fine-grained retrieval within documents beyond the retrieval of documents themselves, we introduce a section-level reranking mechanism that identifies the section most relevant to the query. In particular, once the document is retrieved, the objective of the reranker f_{R} is to pinpoint the specific sections within the document that best match the query. We also note that this reranker is similarly operationalized with the VLM along with a binary classifier on top of it, which directly measures the relevance of each query-section pair (Figure 3.2 (b)).

Formally, for a retrieved document, we take each of its sections $\boldsymbol{s}_i$ with a learnable token for section embedding [EoS] attached to the end and concatenate it with query $\boldsymbol{q}$, forming the input sequence of $[\boldsymbol{q},\ \boldsymbol{s}_i,\ \texttt{[EoS]}]$. The concatenated tokens are then processed through the reranker, and its output corresponding to [EoS] captures the relevance between the query and section, which is further subsequently passed to a binary classifier. Through this, the classifier outputs a probability score indicating the likelihood of the section being relevant to the query, *i.e.*, a score close to one denotes a high relevance.

To train this reranker, we use the binary cross-entropy loss, formulated as follow:

$$\begin{aligned} \mathcal{L}_{\text{reranker}} = \sum_{i=1}^{B}\sum_{j=1}^{S_i}\frac{1}{BS_i}\ell\left(\mathbf{y}_{\boldsymbol{s}_{i,j}},\ f_{\text{R}}\left([\boldsymbol{q},\ \hat{\boldsymbol{s}}_{i,j}]\right)\right), \\ \ell\left(y,\hat{y}\right) = -\left[y\log\hat{y} + (1-y)\log(1-\hat{y})\right], \end{aligned} \tag{3.2}$$

where S_i is the number of sections in the i-th document, $\mathbf{y}_{\boldsymbol{s}_{i,j}}$ is the label for the j-th section of the i-th document $\boldsymbol{s}_{i,j}$ (with its value of one if relevant to the query $\boldsymbol{q}$, otherwise zero), $\hat{\boldsymbol{s}}_{i,j} = [\boldsymbol{s}_{i,j},\ \texttt{[EoS]}]$, and B is the batch size during training. Also, during training, the sections not labeled as relevant to the query are considered negative samples. Then, by minimizing $\mathcal{L}_{\text{reranker}}$, the reranker learns to predict section relevance for any query, thereby refining our overall retrieval process by allowing the retrieval of not just whole documents but also their most relevant sections, for multiple use cases of IR.

3.1.4 Experiments

3.1.4.1 Datasets

We evaluate the proposed IDentIfy on four benchmark datasets designed for multimodal IR that require understanding of both textual and visual cues within queries and documents, as follows:

- **Encyclopedic-VQA** [214] is a benchmark for multimodal Visual Question Answering (VQA) with queries linked to specific Wikipedia sections and includes both textual and multimodal queries;
- **InfoSeek** [215] is a knowledge-intensive VQA dataset with multimodal questions generated from Wikidata triples that include diverse entities such as landmarks, animals, and food;
- **ViQuAE** [216] involves both text-based and multimodal queries about human entities, linked to annotated Wikipedia sections, making it ideal for evaluating section reranking;
- **Open-WikiTable** [217] extends WikiSQL [218] and WikiTableQuestions [219], targeting table QA (in the open-domain setup) by identifying documents or sections containing relevant tables.

3.1.4.2 Baselines

To comprehensively validate IDentIfy, we compare it against two categories of baselines:

- **Conventional VLM Baselines:** We consider earlier VLMs, which are not capable of jointly processing text and images, such as CLIP [198] and BLIP [220]. Also, we consider the approaches, such as UniIR [221], which is built on top of them and fine-tuned with a contrastive loss (Equation 3.1).
- **Baselines with Different Document Representations:** We further consider existing approaches, representing documents in various ways. **Entity** and **Abstract** baselines retrieve documents based on their titles and summaries, respectively, using high-level textual cues. **Text-only** baselines utilize the full textual content of documents for retrieval [206, 222]. **Text & Table** and **Text & Image** baselines leverage tables and first image of documents alongside the text, respectively [223, 155, 209]. These baselines, like our method, are built on the same recent VLMs for direct comparison. **IDentIfy** is our model that holistically represents multimodal content (text, images, and tables) in documents.

Table 3.1: Results with different document retrievers.

Method	**R@1**	**R@10**	**R@100**	**MRR@10**
		CLIP-VIT-L-14		
Zero-Shot	1.9	6.3	13.9	3.1
UniIR + Text-Only	3.8	20.6	50.3	7.7
UniIR + Text & Image	5.8	21.5	48.5	10.0
		BLIP-Large		
Zero-Shot	0.0	0.0	0.0	0.0
UniIR + Text-Only	9.8	36.9	71.4	16.3
UniIR + Text & Image	9.9	23.9	60.7	13.5
		LLaVA-NeXT-Interleave-0.5B		
Entity	3.1	15.5	39.7	6.1
Abstract	13.4	41.3	66.5	21.6
Text-Only	12.5	37.8	68.7	19.8
Text & Table	12.6	38.6	68.5	19.9
Text & Image	16.4	45.4	77.1	25.3
IDentIfy (Ours)	**20.5**	**50.0**	**78.0**	**29.4**

3.1.4.3 Evaluation Metrics

To evaluate our approach, we use standard metrics: Recall@K (R@K) measures whether the relevant document or section appears within the top-K results; MRR@K measures how early the first relevant item is ranked (within top-K) by averaging its inverse rank across queries.

3.1.4.4 Implementation Details

We use LLaVA-NeXT-Interleave [213] as the basis VLM for both the retriever and reranker, and also use LLaVA-OneVision [224] as an additional basis VLM to show the robustness of IDentIfy. Following the convention of using the basis of retrieval with less than 1B parameters to balance computational efficiency and retrieval performance [198, 202, 221], we choose 0.5B-parameter versions of the VLMs. During training, documents are represented using randomly selected four sections, while in inference, we consider all sections within each document. For section-level retrieval, all sections within the top 25 retrieved documents are reranked. Experiments are conducted on a single H100 GPU.

3.1.4.5 Experimental Results and Analyses

Main Results We report retrieval performance on the Encyclopedic-VQA dataset in Table 3.1, where queries include both text and images. IDentIfy significantly outperforms all baselines built on VLMs such as CLIP and BLIP, which are limited to handling a single image alongside text and encoding image-text representations independently, making them suboptimal for understanding multimodal interactions within documents. We also observe that IDentIfy achieves the best performance, improving R@1 scores by 53.0%, 64.0%, 62.7%, and 25.0% over Abstract, Text-Only, Text & Table and Text & Image retrieval baselines, respectively, with similar trends observed for other metrics. These results demonstrate the effectiveness of integrating multimodal content holistically into a unified representation.

We further examine the impact of our pipeline of document retrieval and section reranking. In

Table 3.2: Comparison of different IR strategies for section retrieval. In particular, Document (Ours) performs the document retrieval and section reranking, whereas Passage performs the passage retrieval and reranking. * denotes the model without reranking.

Granularity	**R@1**	**R@10**	**R@20**	**MRR@10**
Passage*	3.9	16.9	22.0	7.5
Passage	28.6	36.4	37.8	31.2
Document (Ours)	**35.1**	**50.8**	**53.6**	**40.3**

Table 3.3: Performance on document retrieval and section reranking for multimodal and textual queries on Encyclopedic-VQA (Enc-VQA), ViQuAE, and InfoSeek. We compare the approach that solely uses textual information from documents (Text-Only) and our approach of leveraging interleaved multimodal contents from documents (IDentIfy) over various scenarios.

			Document Retrieval				**Section Reranking**			
Dataset	**Query Type**	**Method**	**R@1**	**R@10**	**R@100**	**MRR@10**	**R@1**	**R@10**	**R@20**	**MRR@10**
Enc-VQA	Multimodal	Text-Only	12.5	37.8	68.7	19.8	40.7	52.8	55.5	44.8
		IDentIfy (Ours)	**20.5**	**50.0**	**78.0**	**29.4**	**42.4**	**53.6**	**55.7**	**46.3**
	Textual	Text-Only	62.7	76.3	87.4	67.0	68.1	79.4	80.2	72.3
		IDentIfy (Ours)	**65.4**	**76.8**	**87.8**	**69.0**	**69.7**	**80.1**	**80.6**	**73.6**
ViQuAE	Multimodal	Text-Only	13.5	40.4	67.4	20.9	**12.6**	31.7	37.7	**18.2**
		IDentIfy (Ours)	**17.5**	**46.0**	**69.4**	**26.3**	11.4	**32.1**	**39.2**	17.5
	Textual	Text-Only	55.8	71.5	83.0	60.9	27.8	50.2	57.7	35.0
		IDentIfy (Ours)	**56.5**	**72.2**	**83.0**	**61.6**	**29.9**	**50.9**	**59.8**	**36.7**
InfoSeek	Multimodal	Text-Only	6.8	23.6	52.5	11.2	N/A	N/A	N/A	N/A
		IDentIfy (Ours)	**10.2**	**30.4**	**57.3**	**15.7**	N/A	N/A	N/A	N/A

Table 3.2, the passage retriever represents individual sections as separate retrieval units, whereas the document retriever (ours) aggregates multiple section representations into a single representation. Then, we perform reranking over the retrieved sections or the sections from the retrieved documents, and then report the results in Table 3.2. From this, we observe that the passage retriever without reranking (Passage*) achieves suboptimal retrieval performance, highlighting the challenge in pinpointing the most relevant section within a document using traditional retrieval methods. In contrast, when the reranker is used alongside the document retriever, the performance significantly surpasses the passage retrieval, demonstrating the effectiveness of our coarse-to-fine document-to-section retrieval strategy.

Interleaved format enhances document retrieval across modalities. We further expand our experiments to two additional datasets, InfoSeek and ViQuAE, and report document retrieval results. As shown in Table 3.3 Left, our model consistently outperforms the Text-document model for both the multimodal and textual queries. We attribute these gains to the integration of multimodal content, allowing the VLM to capture richer alignments with pre-existing knowledge for document representation [225].

Interleaved format is also beneficial in section retrieval. Similarly, we evaluate section retrieval performance on Encyclopedic-VQA and ViQuAE datasets, for both multimodal and textual queries. As shown in Table 3.3 Right, our model outperforms the Text-document baseline in most cases. However, the performance gains over the baseline are smaller compared to the document retrieval setup. This is likely because section reranking focuses on evaluating the relationship between a single section and a

Table 3.4: Retrieval results for tables, where Zero-shot denotes a model trained on Encyclopedic-VQA but not on the target dataset. Finetuned refers to additional training of the model on the target dataset. (a): Results for tabular document retrieval on Open-WikiTable (OWT). (b): Textual and tabular section reranking results on ViQuAE and OWT datasets, respectively. (c): Reranker accuracy of a classification task that identifies the section containing the query-associated table given a gold document.

(a) Document Retrieval for Tables

Method	R@1	R@10	R@100	MRR@10
Zero-shot	29.4	58.0	86.0	38.1
Finetuned	**55.8**	**84.1**	**93.5**	**66.1**

(c) Tabular Classification

Method	Random	Zero-shot	Finetuned
Acc@1	11.9	9.3	**56.5**

(b) Section Reranking for Tables

Dataset	Target	Method	R@1	R@10	R@20	MRR@10
ViQuAE	Text	Zero-shot	20.3	49.0	57.7	28.9
		Finetuned	**29.9**	**50.9**	**59.8**	**36.7**
OWT	Table	Zero-shot	5.9	20.5	29.4	9.1
		Finetuned	**8.4**	**36.7**	**52.8**	**15.2**

Figure 3.3: Trade-off between the performance (MRR@10) and the training cost (GPU Memory) for retrieval with respect to the number of sections considered.

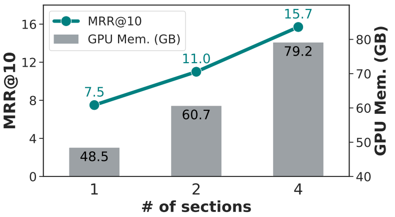


Table 3.5: Comparison of training objectives for the reranker: Contrastive uses contrastive loss similar to the document retriever training; Doc + BCE concatenates the query with multiple sections from the same document and uses the BCE loss; Sec + BCE trains the reranker by concatenating the query with each section individually.

Query Type	Train Loss	R@1	R@10	R@20	MRR@10
Multimodal	Contrastive	3.6	15.0	21.3	6.5
	Doc + BCE	13.6	29.6	32.9	24.1
	Sec + BCE (Ours)	**42.4**	**53.6**	**55.7**	**46.3**
Textual	Contrastive	13.6	37.7	45.1	20.6
	Doc + BCE	23.8	43.4	47.2	39.1
	Sec + BCE (Ours)	**69.7**	**80.1**	**80.6**	**73.6**

query (rather than leveraging the holistic context of the entire document), and individual sections may lack the diverse multimodal information necessary for fully capturing the intent of queries.

Retrieving tables interleaved within documents is challenging. We explore the retrieval task for tabular data, aiming to identify documents or sections containing query-relevant tables, and compare models trained on Encyclopedic-VQA (Zero-shot) with those additionally trained on Open-WikiTable (Finetuned). As shown in Table 3.4 (a), the Finetuned retriever outperforms the Zero-shot retriever on retrieving documents containing query-relevant tables. Yet, more fine-grained section reranking results (identifying sections containing query-relevant tables) in Table 3.4 (b) reveal a notable modality-specific challenge: the performance of Zero-shot and Finetuned rerankers is considerably lower on table retrieval compared to their performance on text retrieval, despite both the text and tables being represented with word tokens. To better understand this, we design a classification task, where rerankers are tasked with identifying the correct section containing the target table within the golden document. Then, as shown in Table 3.4 (c), the Zero-shot reranker performs comparably to random selection, while the Finetuned reranker shows modest improvements. These findings highlight the challenge of tabular retrieval, suggesting the need for table-specific modules to more holistically represent multimodal interleaved documents.

More sections enhance document retrieval performance but raise computational costs. To see how the number of sections used for representing each document impacts performance, we evaluate document retrieval on the InfoSeek dataset by varying the sections per document during training. As shown in Figure 3.3, incorporating more sections improves MRR@10 from 7.5 to 15.7 due to leveraging richer multimodal and contextual information. However, this comes at the cost of increased computational requirements, as processing more sections raises GPU memory consumption.

Table 3.6: Comparison of the negative sample selection strategies for reranker training: Top-K (top-k retrieved sections), In-batch (sections from other samples in the batch), and In-document (sections in the same document).

Negative	R@1	R@20	MRR@10
Top-K	38.1	55.3	44.4
In-batch	39.5	55.4	45.0
In-document (Ours)	**42.4**	**55.7**	**46.3**

Table 3.7: Results with another base model (namely, LLaVA-OneVision-0.5B) for document retrieval (with different document formats).

Format	R@1	R@10	R@100	MRR@10
Entity	2.3	10.3	29.7	4.3
Abstract	7.6	24.7	55.7	12.0
Text-Only	7.0	24.1	50.4	11.7
Text & Table	6.9	26.3	54.9	12.1
Text & Image	9.3	31.4	61.9	15.4
IDentIfy (Ours)	**12.1**	**36.1**	**62.5**	**18.2**

BCE loss is the most effective to train the section reranker. In our reranker design, we use a binary cross-entropy (BCE) loss by concatenating the query with each document section individually (Section + BCE), allowing the model to directly assess query-section relevance. As an alternative, we also explore a contrastive loss (Contrastive), which models section reranking similarly to document retrieval but uses sections as the retrieval units, and a variant of BCE loss (Document + BCE), where the query is concatenated with multiple sections (both positive and negative) from the same document. As shown in Table 3.5, the Section + BCE reranker outperforms both alternatives. Specifically, contrastive loss performs the worst, suggesting that direct concatenation of query and section provides clearer relevance signals, consistent with conventional reranking approaches. Moreover, while Document + BCE leverages inter-section context, its performance might be hindered by training constraints as the model processes fewer sections during training [226, 227], and addressing it would be interesting future work.

Sections from the same document act as effective negatives for reranker training. In training the reranker, we investigate whether considering sections from the same document as negative examples (called In-document) is effective than other strategies, such as Top-K negatives (top-K retrieved sections based on their similarity with the query) and In-batch negatives (positive sections from other samples in the same batch). As shown in Table 3.6, we observe that the In-document approach achieves superior performance especially on R@1, demonstrating its ability to effectively identify the most pertinent section among highly similar sections within the same document, i.e., its training objective can encourage the reranker to focus on fine-grained distinctions between closely related sections (within the same document).

IDentIfy is Versatile with Different VLMs. To ensure the robustness of IDentIfy across VLMs, we evaluate its performance with another VLM, LLaVA-OneVision [224], with 0.5 billion parameters, in addition to LLaVA-NeXT-Interleave [213] used in our main experiments. Results in Table 3.7 show that ours continues to outperform baselines, achieving a notable 30.1% gain in R@1 over the best baseline.

3.1.5 Summary

In this paper, we introduced IDentIfy, a novel IR framework designed to address the limitations of conventional methods that rely on textual content of documents and their segmented passages. Specifically, our approach sits on top of recent VLMs, which enables integration and representation of diverse multimodal content (including text, images, and tables) into a unified document representation. Also, unlike previous strategies that segment documents at the passage level, our method merges these segments to maintain the document's structural coherence, while further introducing a reranking strategy for precise identification of relevant sections. Extensive experiments across various IR datasets demonstrated that IDentIfy consistently outperforms existing baselines, confirming that the interleaved multimodal representation significantly enhances the quality of the document and section retrieval. We believe IDentIfy represents a crucial step toward more comprehensive and contextually aware IR systems, capable of handling the increasing multimodality of modern information sources.

3.1.6 Extension: Video Representation & Retrieval

While our unified multimodal document representation framework (namely, IDentIfy) provides a way to holistically embed and retrieve documents interleaved with textual, visual, and tabular information, it remains limited to static documents and cannot capture the temporal dynamics inherent to (for example) videos. However, real-world information needs often require understanding motion, temporal ordering, and dynamic visual cues that only the video data can provide. To bridge this gap, we extend the underlying principles of IDentIfy to the video domain, introducing a video-specific retrieval (and further contextualization) framework, VideoRAG, discussed in Section 2.3. Specifically, inspired by the interleaved-representation philosophy of IDentIfy, VideoRAG jointly embeds both visual and textual signals from videos, while handling video-specific challenges through a frame selection mechanism designed to extract the most informative frames and an auxiliary strategy for generating textual signals when subtitles are unavailable. This extension opens the path for retrieving temporally grounded, high-capacity video evidence as part of the contextualization process, enabling retrieval-augmented models to access and utilize dynamic, time-dependent knowledge that static multimodal documents cannot provide.

3.2 Fact Representation & Retrieval from Knowledge Graphs

3.2.1 Motivation

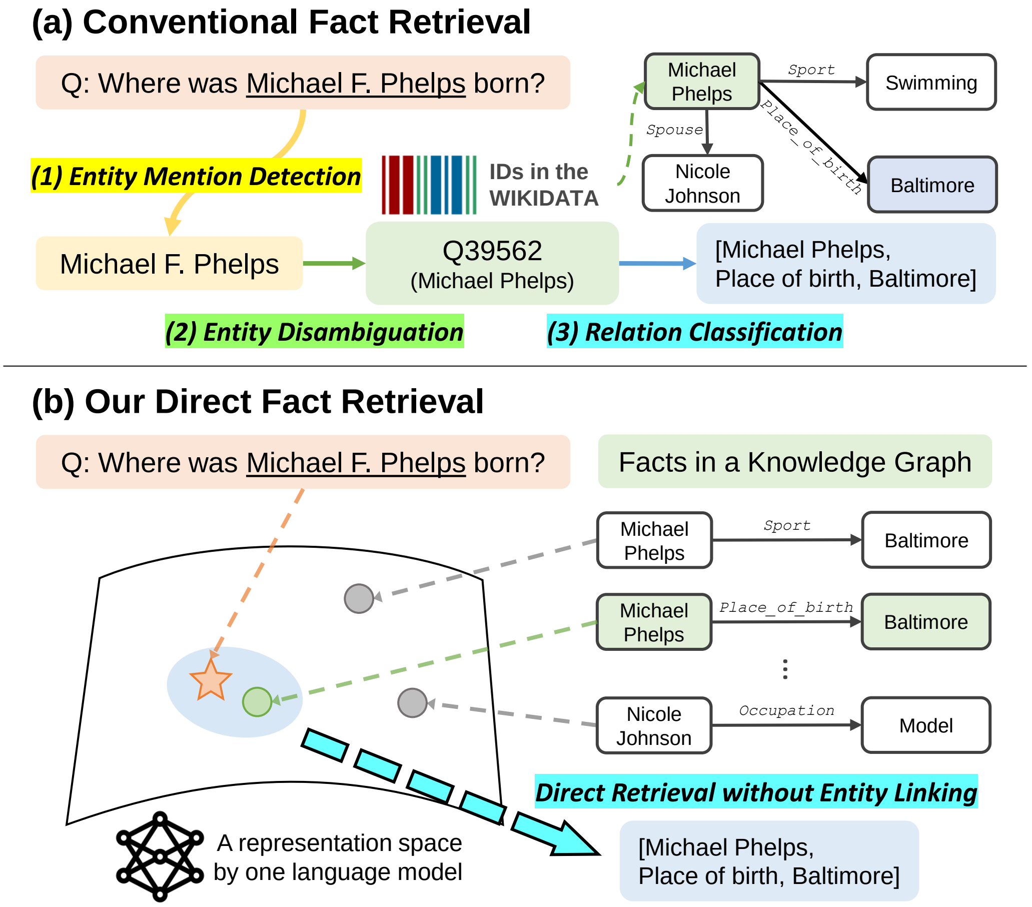

(a) Conventional Fact Retrieval
Q: Where was Michael F. Phelps born?
(1) Entity Mention Detection
IDs in the WIKIDATA
Michael Phelps
Sport
Swimming
Spouse
Place_of_birth
Nicole Johnson
Baltimore
Michael F. Phelps
Q39562 (Michael Phelps)
[Michael Phelps, Place of birth, Baltimore]
(2) Entity Disambiguation
(3) Relation Classification
(b) Our Direct Fact Retrieval
Q: Where was Michael F. Phelps born?
Facts in a Knowledge Graph
Michael Phelps
Sport
Baltimore
Michael Phelps
Place_of_birth
Baltimore
Nicole Johnson
Occupation
Model
Direct Retrieval without Entity Linking
A representation space by one language model
[Michael Phelps, Place of birth, Baltimore]

Figure 3.4: (a) A conventional fact retrieval from KGs involves three sequential steps: 1) entity mention detection to identify entities in queries; 2) entity disambiguation to match entities in input texts to KGs; 3) relation classification to select relevant relations. (b) Our fact retrieval directly retrieves relevant facts with their representational similarities to input queries.

Knowledge graphs (KGs) [82, 32, 228], which consist of a set of facts represented in the form of a (head entity, relation, tail entity) triplet, can store a large amount of knowledge. In many applications, language models (LMs) [26, 5] are commonly used; however, their knowledge internalized in parameters is often incomplete, inaccurate, and outdated. Therefore, several recent works suggest augmenting LMs with facts from KGs, for example, in question answering [229, 230] and dialogue generation [231, 53].

However, despite the broad applications of the KGs, the existing mechanism for retrieving facts from them are, in many cases, unnecessarily complex. In particular, to retrieve facts from KGs, existing work [232, 233, 234] relies on three sequential steps, consisting of span detection, entity disambiguation, and relation classification, as illustrated in Figure 3.4 (a). For example, given an input text: "Where was Michael Phelps born?", they first detect a span of an entity within the input, which corresponds to "Michael Phelps". Then, they match the entity mention in the input to an entity id in the KG. Those two steps are often called entity linking. Finally, among 91 relations associated with the entity of Michael Phelps, they select one relation relevant to the input, namely "place of birth".

The aforementioned approach has a couple of drawbacks. First, all three sub-modules in the pipeline require module-specific labels in addition to query-triplet pairs for training. However, in real-world, high-quality training data is limited, and annotating them requires significant costs. Second, such a pipeline approach is prone to error propagation across steps [235, 236]. For example, if the span detection fails, the subsequent steps, such as relation classification, are likely to make incorrect predictions as well. Third, certain modules, that match entities in queries to KGs or predict relations over KGs, are usually not generalizable to emerging entities and relations and cannot be applied to different KGs. It would be preferable to have a method that does not require KG-specific training and inference.

To tackle these limitations, we propose to directly retrieve the relevant triplets related to a natural language query by computing their similarities over a shared representation space (See Figure 3.4 (b)). The design of our direct retrieval framework is motivated by a pioneering work of open-domain question answering with documents [79], which showed the possibility of dense retrieval with simple vector similarities between the question and document embeddings. However, in contrast to the document retrieval scenario where documents have sufficient contexts to embed, it is unclear whether the LM can still effectively embed facts represented in the short triplet form for retrieval. Also, compared to the document retrieval which additionally requires a reader to extract only the relevant piece of knowledge, our fact retriever itself can directly provide the relevant knowledge.

To realize our fact retriever, we train it by maximizing similarities between representations of relevant pairs of input texts and triplets while minimizing irrelevant pairs, where we use LMs for encoding them. We note that this process requires only text-triplet pairs without using extra labels, unlike the conventional pipeline approach for fact retrieval. After training, we index all triplets in the KG with the trained encoder in an offline manner, and, given the input query, we return the nearest triplets over the embedding space. This procedure simplifies the conventional three steps for retrieving facts from KGs into one. To further efficiently search the relevant triplets, we approximate the similarity calculation with vector quantization and hierarchical search based on clustering [237]. We further note that, since we embed triplets using the LM, our retriever can generalize to different KGs without any modification, unlike some conventional retrieval systems that require additional training to learn new KG schema about distinct entities and relations types. We refer to our framework as **Di**rect **Fa**ct **R**etrieval (**DiFaR**).

We experimentally demonstrate that our direct retrieval on KGs works well; however, the fact represented in the triplet form has a limited context, since it consists of only two entities and one relation. Also, similarity calculation with the independently represented input text and triplets is arguably simple, and might be less effective. Therefore, to further improve the retriever performance, we additionally use a reranker, whose goal is to calibrate the ranks of retrieved triplets for the input text. In particular, we first retrieve k nearest facts with the direct retriever, and then use another LM which directly measures the similarity by encoding the input text and the triplet simultaneously. Moreover, another objective of the reranker is to filter out irrlevant triplets, which are the most confusing ones in the embedding space of the direct retriever. Therefore, to effectively filter them, we train the reranker to minimize similarities between the input text and the most nearest yet irrelevant triplets.

We evaluate our DiFaR framework on fact retrieval tasks across two different domains of question answering and dialogue, whose goals are to retrieve relevant triplets in response to the given query. The experimental results then show that our proposed DiFaR framework outperforms relevant baselines that use conventional pipeline approaches to retrieve facts on KGs, and also show that our reranking strategy significantly improves retrieval performances. The detailed analyses further support the efficacy of our DiFaR framework, with its great simplicity.

3.2.2 Related Work

Knowledge Graphs Knowledge Graphs (KGs) are factual knowledge sources [82, 32], containing a large number of facts, represented in a symbolic form: (head entity, relation, tail entity). Since some natural language applications require factual knowledge [238], existing literature proposes to use knowledge in KGs, and sometimes along with language models (LMs) [26]. To mention a few, in question answering domains, facts in KGs can directly be answers for knowledge graph question answering tasks [239, 240], but also they are often augmented to LMs to generate knowledge-grounded answers [104, 108]. Similarly, in dialogue generation, some existing work augments LMs with facts from KGs [231, 53]. However, prior to utilizing facts in KGs, fact retrieval – selection of facts relevant to the input context – should be done in advance, whose results substantially affect downstream performances. In this work, we propose a conceptually simple yet effective framework for fact retrieval, motivated by information retrieval.

Information Retrieval The goal of most information retrieval work is to retrieve relevant documents in response to a query (e.g., question). Early work relies on term-based matching algorithms, which count lexical overlaps between the query and documents, such as TF-IDF and BM25 [121, 241]. However, they are vulnerable to a vocabulary mismatch problem, where semantically relevant documents are lexically different from queries [242, 243]. Due to such the issue, recently proposed work instead uses LMs [26, 37] to encode queries and documents, and uses their representational similarities over a latent space [79, 80, 244]. They suggest their huge successes are due to the effectiveness of LMs in embedding documents. However, they focus on lengthy documents having extensive context, and it is unclear whether LMs can still effectively represent each fact, succinctly represented with two entities and one relation in the triplet form, for its retrieval. In this work, we explore this new direction by formulating the fact retrieval problem as the information retrieval problem done for documents.

Knowledge Retrieval from KGs Since KGs have a large number of facts, it is important to bring only the relevant piece of knowledge given an input query. To do so, one traditional approach uses neural semantic parsing-based methods [62, 245, 63, 64] aiming to translate natural language inputs into logical query languages, such as SPARQL[2] and λ-DCS [246], executable over KGs. However, they have limitations in requiring additional labels and an understanding of logical forms of queries. Another approach is to use a pipeline [247, 248, 249, 250, 234] consisting of three subtasks: entity span detection, entity disambiguation, and relation classification. However, they similarly require additional labels on training each subcomponent, and this pipeline approach suffers from errors that are propagated from previous steps [235, 236]. While recent work [229] proposes to retrieve textual triplets from KGs based on their representational similarities to the input text with the information retrieval mechanism, they still rely on entity linking (e.g., span detection and entity disambiguation) first, thus identically having limitations of the pipeline approach. Another recent work [230] merges a set of facts associated with each entity into a document and performs document-level retrieval. However, the document retrieval itself can be regarded as entity linking, and also the overall pipeline requires an additional reader to extract only the relevant entity in retrieved documents. In contrast to them, we directly retrieve facts from the input query based on their representational similarities, which simplifies the conventional three-step approach including entity linking into one single retrieval step.

[2] https://www.w3.org/TR/rdf-sparql-query/

3.2.3 Approach

3.2.3.1 Preliminaries

We formally define a KG and introduce a conventional mechanism for retrieving facts from the KG.

Knowledge Graphs Let $\mathcal{E}$ be a set of entities and $\mathcal{R}$ be a set of relations. Then, one particular fact is defined as a triplet: $t = (\texttt{e}_\texttt{h}, \texttt{r}, \texttt{e}_\texttt{t}) \in \mathcal{E} \times \mathcal{R} \times \mathcal{E}$, where $\texttt{e}_\texttt{h}$ and $\texttt{e}_\texttt{t}$ are head and tail entities, respectively, and $\texttt{r}$ is a relation between them. Also, a knowledge graph (KG) $\mathcal{G}$ is defined as a set of factual triplets: $\mathcal{G} = \{(\texttt{e}_\texttt{h}, \texttt{r}, \texttt{e}_\texttt{t})\} \subseteq \mathcal{E} \times \mathcal{R} \times \mathcal{E}$. Note that this KG is widely used as a useful knowledge source for many natural language applications, including question answering and dialogue generation [229, 230, 231, 53]. However, the conventional mechanism to access facts in KGs is largely complex, which may hinder its broad applications, which we describe in the next paragraph.

Existing Knowledge Graph Retrieval The input of most natural language tasks is represented as a sequence of tokens: $\boldsymbol{x} = [w_1, w_2, \dots, w_{|\boldsymbol{x}|}]$. Suppose that, given the input $\boldsymbol{x}$, t^+ is a target triplet to retrieve[3]. Then, the objective of the conventional fact retrieval process for the KG $\mathcal{G}$ [247, 234] is, in many cases, formalized as the following three sequential tasks:

$$t^+ = \operatorname*{arg\,max}_{t \in \mathcal{G}} p_\theta(t|\texttt{e}, \boldsymbol{x}, \mathcal{G}) p_\phi(\texttt{e}|m, \boldsymbol{x}) p_\psi(m|\boldsymbol{x}), \tag{3.3}$$

where $p_\psi(m|\boldsymbol{x})$ is the model for mention detection with m as the detected entity mention within the input $\boldsymbol{x}$, $p_\phi(\texttt{e}|m, \boldsymbol{x})$ is the model for entity disambiguation, and $p_\theta(t|\texttt{e}, \boldsymbol{x}, \mathcal{G})$ is the model for relation classification, all of which are individually parameterized by ϕ, ψ, and θ, respectively.

However, there is a couple of limitations in such the three-step approaches. First, they are vulnerable to the accumulation of errors, since, for example, if the first two steps consisting of span detection and entity disambiguation are wrong and we are ending up with the incorrect entity irrelevant to the given query, we cannot find the relevant triplet in the final relation prediction stage. Second, due to their decomposed structures, three sub-modules are difficult to train in an end-to-end fashion, while requiring labels for training each sub-module. For example, to train $p_\psi(m|\boldsymbol{x})$ that aims to predict the mention boundary of the entity within the input text, they additionally require annotated pairs of the input text and its entity mentions: $\{(\boldsymbol{x}, m)\}$. Finally, certain modules are usually limited to predicting entities $\mathcal{E}$ and relations $\mathcal{R}$ specific to the particular KG schema, observed during training. Therefore, they are not directly applicable to unseen entities and relations, but also to different KGs.

3.2.3.2 Direct Knowledge Graph Retrieval

To tackle the aforementioned challenges of the existing fact retrieval approaches on KGs, we present the direct knowledge retrieval framework. In particular, our objective is simply formulated with the single sentence encoder model E_θ without introducing extra variables (e.g., m and $\texttt{e}$), as follows:

$$t^+ = \operatorname*{arg\,max}_{t \in \mathcal{G}} f(E_\theta(\boldsymbol{x}), E_\theta(t)), \tag{3.4}$$

where f is a scoring function that calculates the similarity between the input text representation $E_\theta(\boldsymbol{x})$ and triplet representation $E_\theta(t)$ (e.g., by using the dot product). Note that, in Equation 3.4, we use the

[3] For the sake of simplicity, we consider one triplet t^+ for each input; the retrieval target can be a set of triplets $\{t^+\}$.

sentence encoder E_θ to represent the triplet t. To do so, we first symbolize the triplet as a sequence of tokens: $t = [w_1, w_2, \dots, w_{|t|}]$, which is constructed by entity and relation tokens, and the separation token (i.e., a special token, [SEP]) between them. Then, we simply forward the triplet tokens to E_θ to obtain the triplet representation. While we use the single model for encoding both input queries and triplets, we might alternatively represent them with different encoders, which we leave as future work.

Training After formalizing the goal of our direct knowledge retrieval framework in Equation 3.4, the next step is to construct the training samples and the optimization objective to train the model (i.e., E_θ). According to Equation 3.4, the goal of our model is to minimize distances between the input text and its relevant triplets over an embedding space, while minimizing distances of irrelevant pairs. Therefore, following the existing dense retrieval work for documents [79], we use a contrastive loss as our objective to generate an effective representation space, formalized as follows:

$$\min_{\theta} -\log \frac{\texttt{exp}(f(E_\theta(\boldsymbol{x}), E_\theta(t^+)))}{\sum_{(\boldsymbol{x},t)\in\tau} \texttt{exp}(f(E_\theta(\boldsymbol{x}), E_\theta(t)))}, \tag{3.5}$$

where τ contains a set of pairs between the input text and all triplets in the same batch. In other words, $(\boldsymbol{x}, t+) \in \tau$ is the positive pair to maximize the similarity, whereas, others are negative pairs to minimize. Also, $\texttt{exp}(\cdot)$ is an exponential function.

Inference During the inference stage, given the input text $\boldsymbol{x}$, the model should return the relevant triplets, whose embeddings are closest to the input text embedding. Note that, since $E_\theta(\boldsymbol{x})$ and $E_\theta(t)$ in Equation 3.4 are decomposable, to efficiently do that, we represent and index all triplets in an offline manner. Note that, we use the FAISS library [237] for triplet indexing and similarity calculation, since it provides the extremely efficient search logic, also known to be applicable to billions of dense vectors; therefore, suitable for our fact retrieval from KGs. Moreover, to further reduce the search cost, we use the approximated neighborhood search algorithm, namely Hierarchical Navigable Small World Search with Scalar Quantizer. This mechanism not only quantizes the dense vectors to reduce the memory footprint, but also builds the hierarchical graph structures to efficiently find the nearest neighborhoods with few explorations. We term our **Di**rect **Fa**ct **R**etrieval method as **DiFaR**.

3.2.3.3 Reranking for Accurate Fact Retrieval

The fact retrieval framework outlined in Section 3.2.3.2 simplifies the conventional three subtasks used to access the knowledge into the single retrieval step. However, contrary to the document retrieval case, the fact is represented with the most compact triplet form, which consists of only two entities and one relation. Therefore, it might be suboptimal to rely on the similarity, calculated by the independently represented input text and triplets as in Equation 3.4. Also, it is critically important to find the correct triplet within the small k (e.g., $k = 1$) of the top-k retrieved triplets, since, considering the scenario of augmenting LMs with facts, forwarding several triplets to LMs yields huge computational costs.

To tackle such challenges, we propose to further calibrate the ranks of the retrieved triplets from our DiFaR framework. Specifically, we first obtain the k nearest facts in response to the input query over the embedding space, by using the direct retrieval mechanism defined in Section 3.2.3.2. Then, we use another LM, E_ϕ, that returns the similarity score of the pair of the input text and the retrieved triplet by encoding them simultaneously, unlike the fact retrieval in Equation 3.4. In other words, we first concatenate the token sequences of the input text and the triplet: $[\boldsymbol{x}, t]$, where $[\cdot]$ is the concatenation operation, and then forward it to $E_\phi([\boldsymbol{x}, t])$. By doing so, the reranking model E_ϕ can effectively consider token-level relationships between two inputs (i.e., input queries and triplets), which leads to accurate calibration of the ranks of retrieved triplets from DiFaR, especially for the top-k ranks with small k.

For training, similar to the objective of DiFaR defined in Section 3.2.3.2, we aim to maximize the similarities of positive pairs: $\{(\boldsymbol{x}, t^+)\}$, while minimizing the similarities of irrelevant pairs: $\{(\boldsymbol{x}, t)\} \setminus \{(\boldsymbol{x}, t^+)\}$, with a binary cross-entropy loss. However, contrary to the previous negative sampling strategy defined in Section 3.2.3.2 where we randomly sample the negative pairs, in this reranker training, we additionally manipulate them by using the initial retrieval results from DiFaR. The intuition here is that irrelevant triplets, included in the k nearest neighbors to the input query, are the most confusing examples, which are yet not filtered by the DiFaR model. Hereat, the goal of the reranking strategy is to further filter them by refining the ranks of the k retrieved triplets; therefore, to achieve this goal, we include them as the negative samples during reranker training. Formally, let $\tilde{\tau} = \{(\boldsymbol{x}, \tilde{t})\}$ is a set of pairs of the input query $\boldsymbol{x}$ and its k nearest facts retrieved from DiFaR. Then, the negative samples for the reranker are defined by excluding the positive pairs, formalized as follows: $\tilde{\tau} \setminus \{(\boldsymbol{x}, t^+)\}$. Note that constructing the negative samples with retrieval at every training iteration is costly; therefore, we create them at intervals of several epochs (e.g., ten), but also we use only a subset of triplets in KGs during retrieval. Our framework with the reranking strategy is referred to as **Di**rect **Fa**ct **R**etrieval with **R**eranking (**DiFaR^2**).

3.2.4 Experiments

We first explain datasets, models, metrics, and implementations.

3.2.4.1 Datasets

We validate our **Di**rect **Fa**ct **R**etrieval (**DiFaR**) on fact retrieval tasks, whose goal is to retrieve relevant triplets over KGs given a query. We use four datasets on question answering and dialogue tasks.

Question Answering The objective of KG-based QA tasks is to predict factual triplets in response to the given question, where predicted triplets are direct answers. For this task, we use three datasets: SimpleQuestions [251], WebQuestionsSP (WebQSP) [81, 33], and Mintaka [84]. SimpleQuestions and WebQSP are designed with the Freebase KG [82], and Mintaka is designed with the Wikidata KG [32].

Dialogue In addition to QA, we evaluate DiFaR on KG-based dialogue generation, whose one subtask is to retrieve relevant triplets on the KG that provides factual knowledge to respond to a user's conversation query. We use the OpenDialKG data [252], designed with Freebase.

Knowledge Graphs Following Diefenbach et al. [83] and Saffari et al. [119], we use Wikidata KG [32] for experiments on QA, and use their dataset processing settings. For OpenDialKG, we use Freebase.

3.2.4.2 Baselines and Our Models

We compare our DiFaR framework against other relevant baselines that involve subtasks, such as entity detection, disambiguation, and relation prediction. Note that most existing fact retrieval work either uses labeled entities in queries, or uses additional labels for training subcomponents; therefore, they are not comparable to DiFAR that uses only pairs of input texts and relevant triplets. For evaluations, we include models categorized as follows:

Table 3.8: Main results on the question answering domain for SimpleQuestions, WebQSP, and Mintaka datasets. We emphasize the best scores in bold, except for the incomparable model: Retrieval with Gold Entities, which uses labeled entities in inputs.

		SimpleQuestions			**WebQSP**			**Mintaka**		
Types	**Methods**	MRR	Hits@1	Hits@10	MRR	Hits@1	Hits@10	MRR	Hits@1	Hits@10
Unsupervised	Retrieval with Gold Entities	0.7213	0.5991	0.9486	0.5324	0.4355	0.7402	0.1626	0.0978	0.2969
	Retrieval with spaCy	0.3454	0.2917	0.4437	0.3530	0.2856	0.4863	0.0914	0.0585	0.1622
	Retrieval with GENRE	0.1662	0.1350	0.2234	0.3099	0.2498	0.4363	0.0935	0.0640	0.1540
	Retrieval with BLINK	0.5142	0.4276	0.6766	0.4853	0.3938	0.6694	0.1350	0.0850	0.2430
	Retrieval with ReFinED	0.4841	0.4047	0.6283	0.5008	0.4055	0.6953	0.1312	0.0831	0.2325
	Factoid QA by Retrieval	0.7835	0.6953	0.9304	0.3933	0.3089	0.5470	0.1350	0.0836	0.2344
	DiFaR (Ours)	0.7070	0.5872	0.9259	0.5196	0.4130	0.7352	0.1590	0.0895	0.3043
	DiFaR^2 (Ours)	**0.8361**	**0.7629**	**0.9470**	**0.5441**	**0.4321**	**0.7602**	**0.2077**	**0.1348**	**0.3595**
Supervised	Retrieval with Gold Entities	0.8007	0.7094	0.9477	0.6048	0.5079	0.7794	0.2705	0.1987	0.4070
	Retrieval with spaCy	0.3789	0.3380	0.4453	0.3963	0.3272	0.5162	0.1367	0.1019	0.2019
	Retrieval with GENRE	0.1921	0.1718	0.2255	0.3617	0.3014	0.4696	0.1346	0.1005	0.1964
	Retrieval with BLINK	0.5679	0.5008	0.6766	0.5483	0.4571	0.7052	0.2075	0.1530	0.3157
	Retrieval with ReFinED	0.5349	0.4765	0.6279	0.5707	0.4754	0.7377	0.2106	0.1562	0.3166
	Factoid QA by Retrieval	0.8590	0.8051	0.9293	0.5253	0.4546	0.6486	0.1548	0.1179	0.2179
	DiFaR (Ours)	0.7904	0.6986	0.9382	0.6102	0.5071	0.7927	0.3049	0.2138	0.4856
	DiFaR^2 (Ours)	**0.8992**	**0.8583**	**0.9576**	**0.7189**	**0.6528**	**0.8385**	**0.4189**	**0.3367**	**0.5847**

Retrieval with Entity Linking It predicts relations over candidate triplets associated with identified entities by the entity linking methods, namely **spaCy** [253], **GENRE** [254], **BLINK** [255, 256], and **ReFinED** [2] for Wikidata; **GrailQA** [257] for Freebase.

Factoid QA by Retrieval It retrieves entities and relations independently based on their similarities with the input query [239].

Our Models Our **Di**rect **K**nowl**e**dge **R**etrieval (**DiFaR**) directly retrieves the nearest triplets to the input text on the latent space. **DiFaR with Reranking (DiFaR^2)** is also ours, which includes a reranker to calibrate retrieved results.

Retrieval with Gold Entities It uses labeled entities in inputs and retrieves triplets based on their associated triplets. It is incomparable to others.

3.2.4.3 Evaluation Metrics

We measure the retrieval performances of models with standard ranking metrics, which are calculated by ranks of correctly retrieved triplets. In particular, we use **Hits@K** which measures whether retrieved Top-K triplets include a correct answer or not, and Mean Reciprocal Rank (**MRR**) which measures the rank of the first correct triplet for each input text and then computes the average of reciprocal ranks of all results. Following exiting document retrieval work [80, 258], we consider top-1000 retrieved triplets when calculating MRR, since considering ranks of all triplets in KGs are computationally prohibitive.

3.2.4.4 Implementation Details

We use a distilbert[4] as a retriever for all models, and a lightweight MiniLM model[5] as a reranker, both of which are pre-trained with the MSMARCO dataset [259]. During reranking, we sample top-100

[4] https://huggingface.co/sentence-transformers/msmarco-distilbert-base-v3

[5] https://huggingface.co/cross-encoder/ms-marco-MiniLM-L-6-v2

Table 3.9: Main results on the dialogue domain for the OpenDialKG dataset. We emphasize the best scores in bold except for Retrieval with Gold Entities, which uses labeled entities.

		OpenDialKG		
Types	**Methods**	MRR	Hits@1	Hits@10
Unsupervised	Retrieval with Gold Entities	0.2511	0.1560	0.4683
	Retrieval with GrailQA	0.2051	0.1271	0.3745
	Factoid QA by Retrieval	0.1977	0.0892	0.4231
	DiFaR (Ours)	0.2396	0.1395	0.4424
	DiFaR^{2} (Ours)	**0.2637**	**0.1603**	**0.4744**
Supervised	Retrieval with Gold Entities	0.2750	0.1495	0.5745
	Retrieval with GrailQA	0.2217	0.1198	0.4436
	Factoid QA by Retrieval	0.2042	0.1266	0.3587
	DiFaR (Ours)	0.2755	0.1405	0.5547
	DiFaR^{2} (Ours)	**0.4784**	**0.3535**	**0.7380**

triplets retrieved from DiFaR. We use off-the-shelf models for unsupervised settings, and further train them for supervised settings.

3.2.4.5 Experimental Results and Analyses

Main Results We first conduct experiments on question answering domains, and report the results in Table 3.8. As shown in Table 3.8, our DiFaR with Reranking (DiFaR^{2}) framework significantly outperforms all baselines on all datasets across both unsupervised and supervised experimental settings with large margins. Also, we further experiment on dialogue domain, and report results in Table 3.9. As shown in Table 3.9, similar to the results on QA domains, our DiFaR^{2} framework outperforms the relevant baselines substantially. These results on two different domains demonstrate that our DiFaR^{2} framework is highly effective in fact retrieval tasks.

To see the performance gains from our reranking strategy, we compare the performances between our model variants: DiFaR and DiFaR^{2}. As shown in Table 3.8 and Table 3.9, compared to DiFaR, DiFaR^{2} including the reranker brings huge performance improvements, especially on the challenging datasets: Mintaka and OpenDialKG. However, we consistently observe that our DiFaR itself can also show superior performances against all baselines except for the model of Factoid QA by Retrieval on the SimpleQuestions dataset. The inferior performance of our DiFaR on this SimpleQuestions dataset is because, its samples are automatically constructed from facts in KGs; therefore, it is extremely simple to extract entities and predict relations in response to the input query. On the other hand, our DiFaR framework sometimes outperforms the incomparable model: Retrieval with Gold Entities, which uses the labeled entities in the input queries. This is because this model is restricted to retrieve the facts that should be associated with entities in input queries; meanwhile, our DiFaR is not limited to query entities thanks to the direct retrieval scheme.

Analyses on Zero-Shot Generalization Our DiFaR can be generalizable to different datasets with the same KG, but also to ones with other KGs without any modifications. This is because it retrieves triplets based on their text-level similarities to input queries and does not leverage particular schema of entities and relations, unlike the existing entity linking methods. To demonstrate them, we perform experiments on zero-shot transfer learning, where we use the model, trained on the WebQSP dataset

Table 3.10: Zero-shot transfer learning results, where we use models trained on the WebQSP dataset with the Wikidata KG not only for SimpleQuestions and Mintaka datasets with the same KG, but also for the WebQSP dataset with the different Freebase KG. We use MRR as a metric.

	Wikidata		**Freebase**
Methods	SimpleQuestions	Mintaka	WebQSP
Retrieval with Gold Entities	0.7994	0.1950	0.6000
Retrieval with BLINK	0.5704	0.1617	N/A
Retrieval with ReFinED	0.5389	0.1591	N/A
Factoid QA by Retrieval	0.8014	0.1431	0.4239
DiFaR (Ours)	0.7812	0.2063	0.5913
DiFaR^{2} (Ours)	**0.8244**	**0.2769**	**0.6324**

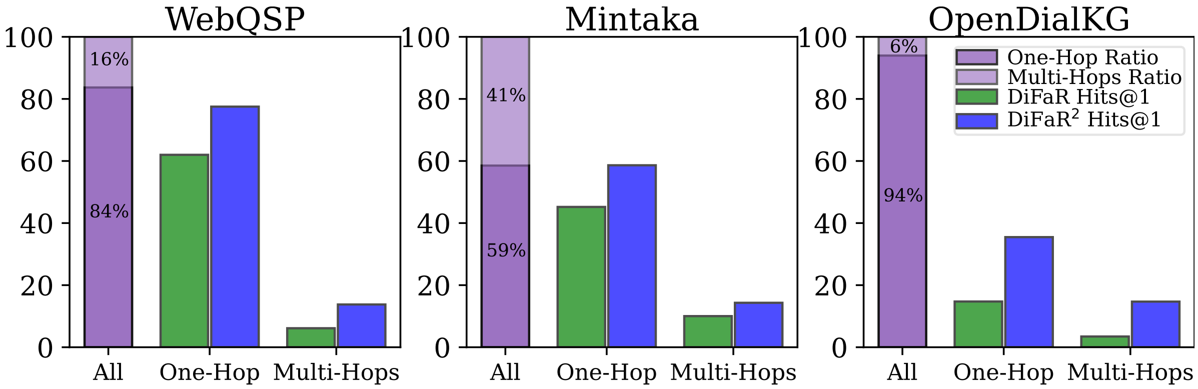


Figure 3.5: Breakdown results by single and multi-hops, where we report ratios of single and multi-hops samples on the left side of each subfigure, and Hits@1 of DiFaR and DiFaR^{2} across single and multi-hops on the middle and right. We exclude the SimpleQuestions dataset that consists of single-hop questions.

with the Wikidata KG, to different datasets with the same KG and also to ones with the different Freebase KG. As shown in Table 3.10, DiFaR is generalizable to different datasets and KGs; meanwhile, the pipeline methods involving entity linking are not generalizable to different KGs, and inferior to ours.

Analyses on Single- and Multi-Hops To see whether DiFaR can also perform challenging multi-hop retrieval that requires selecting triplets not directly associated with entities in input queries, we breakdown the performances by single- and multi-hop type queries. As shown in Figure 3.5, DiFaR can directly retrieve relevant triplets regardless of whether they are associated with entities in input queries (single-hop) or not (multi-hop), since it does not rely on entities in queries for retrieval. Also, we observe that our reranking strategy brings huge performance gains, especially on multi-hop type queries. However, due to the intrinsic complexity of multi-hop retrieval, its performances are relatively lower than performances in single-hop cases; thus, despite the fact that the majority of queries are answerable with single-hop retrieval and that our DiFaR can handle multi-hop queries, it is valuable to further extend the model for multi-hop, which we leave as future work. We also provide examples of facts retrieved by our DiFaR in Table 3.11. As shown in Table 3.11, since LMs, that is used for encoding both the question and the triplets for retrieval, might learn background knowledge about them during pre-trainnig, DiFaR can directly retrieve relevant triplets even for complex questions. For instance, in the first example, the LM already knows who was the us president in 1963, and directly retrieves whose religion.

Table 3.11: Retrieval examples for complex questions, on the challenging Mintaka dataset. We highlight the related phrases across the question and the triplet in yellow and green colors.

Question: What religion was the us president in 1963?
Retrieved Triplet: (Robert F. Kennedy, religion, Catholicism)
Answer: Catholicism

Question: Who commanded the allied invasion of western Europe at Normandy and was an American president?
Retrieved Triplet: (Normandy landings, participant, Dwight D. Eisenhower)
Answer: Dwight D. Eisenhower

Question: Which former Chicago Bull shooting guard was also selected to play on the 1992 US basketball team?
Retrieved Triplet: (1992 US men's basketball team, has part, Michael Jordan)
Answer: Michael Jordan

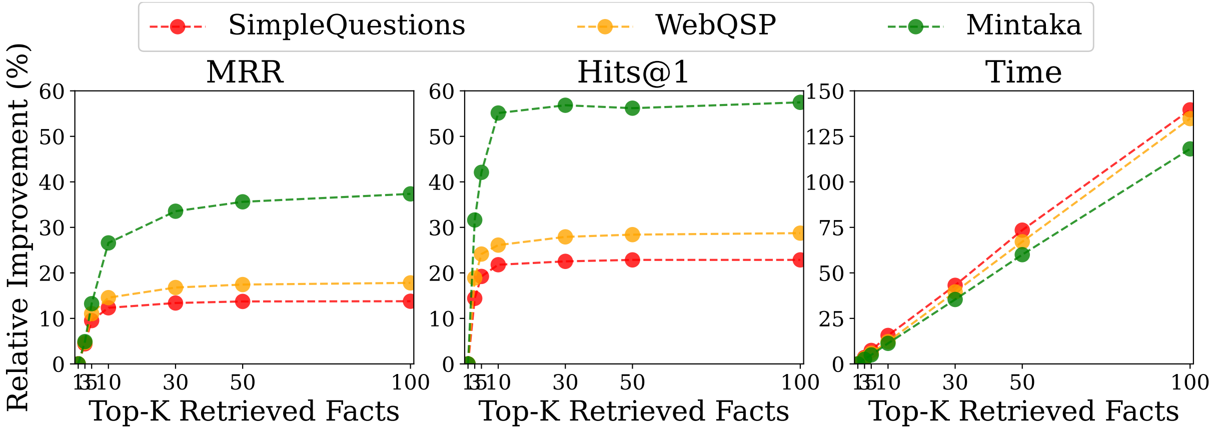


Figure 3.6: Performances and efficiencies of DiFaR^2 with varying K, where we change the number of Top-K retrieved triplets when leveraging the reranking strategy. We report results with the relative improvement (%) to our DiFaR without reranking. We report the time with average over 30 runs.

Analyses on Reranking with Varying K While we show huge performance improvements with our reranking strategy in Table 3.8 and Table 3.9, its performances and efficiencies depend on the number of retrieved Top-K triplets. Therefore, to further analyze it, we vary the number of K, and report the performances and efficiencies in Figure 3.6. As shown in Figure 3.6, the performances are rapidly increasing until Top-10 and saturated after it. Also, the time for reranking is linearly increasing when we increase the K values, and, in Top-10, the reranking mechanism takes only less than 20% time required for the initial retrieval. These results suggest that it might be beneficial to set the K value as around 10.

Sensitivity Analyses on Architectures To see different architectures of retrievers and rerankers make how many differences in performances, we perform sensitivity analyses by varying their backbones. We use available models in the huggingface library[6]. As shown in Table 3.12, we observe that the pre-trained backbones by the MSMARCO dataset [259] show superior performances compared to using the naive backbones, namely DistilBERT and MiniLM, on both retrievers and rerankers. Also, performance differences between models with the same pre-trained dataset (e.g., MSMARCO-TAS-B and MSMARCO-Distil) are marginal. These two results suggest that the knowledge required for document retrieval is also beneficial to fact retrieval, and that DiFaR frameworks are robust across different backbones.

Analyses on Entity Linking While our DiFaR framework is not explicitly trained to predict entity mentions in the input query and their ids in the KG, during the training of our DiFaR, it might learn

[6] https://huggingface.co/models

Table 3.12: Sensitivity analyses on architectures, where we change the backbones of retriever and reranker in our DiFaR^2. MSMARCO in the model name indicates it is pre-trained by the MSMARCO dataset, and we report results on the WebQSP dataset.

Types	**Models**	MRR	Hits@1	Hits@10
Retriever	DistilBERT	0.5983	0.4963	0.7810
	MSMARCO-TAS-B	0.6051	0.4963	0.7844
	MSMARCO-Distil	0.6102	0.5071	0.7927
Reranker	MiniLM	0.6675	0.5945	0.7927
	MSMARCO-TinyBERT	0.7068	0.6420	0.8177
	MSMARCO-MiniLM	0.7189	0.6528	0.8385

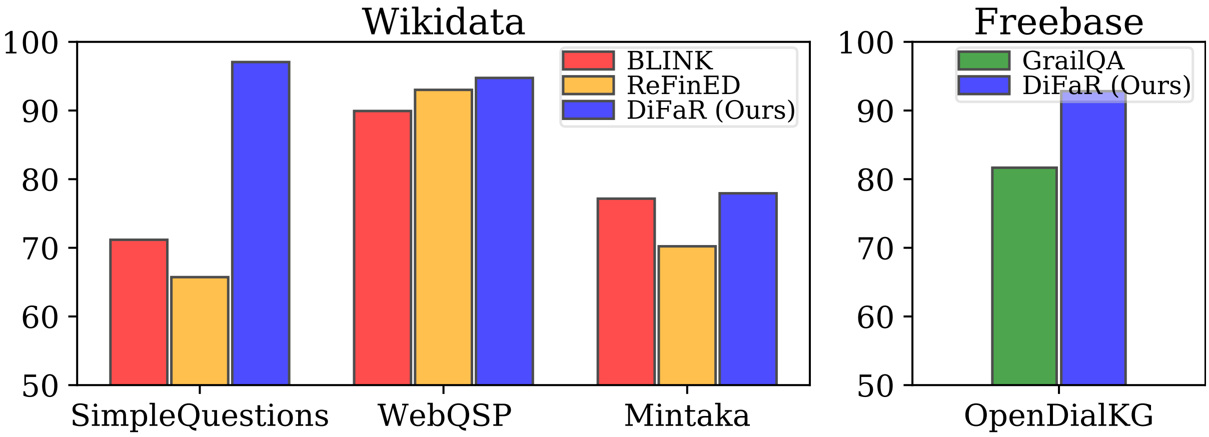


Figure 3.7: Entity linking results, where we measure the performances on benchmark datasets with Wikidata and Freebase KGs. Note that entity mentions of the SimpleQuestions dataset are not available; thus, we cannot fine-tune existing entity linkers, which additionally require mention labels, unlike ours.

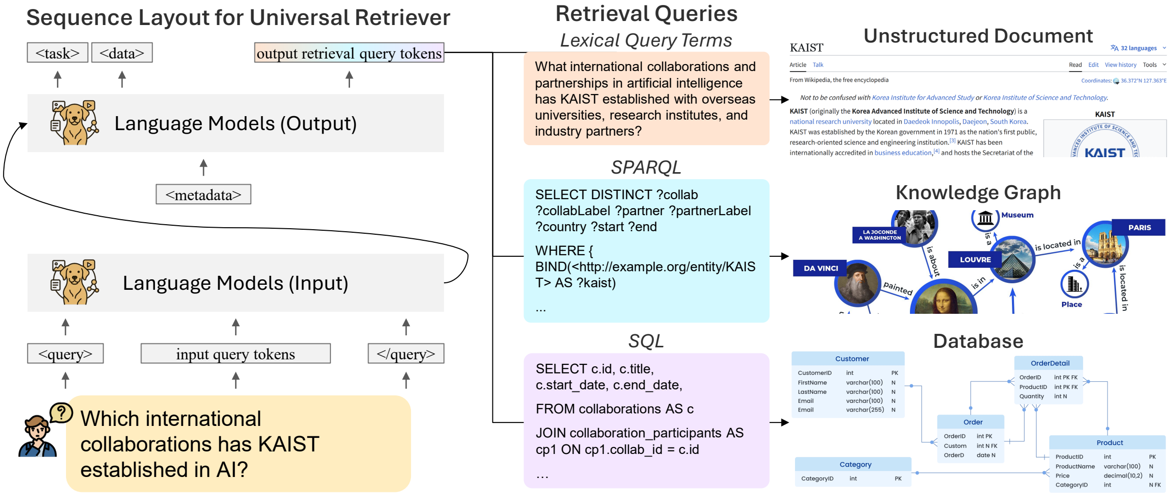
the knowledge on matching the input text to its entities. To demonstrate it, we measure entity linking performances by checking whether the retrieved triplets contain the labeled entities in the input query. As shown in Figure 3.7, our DiFaR surprisingly outperforms entity linking models. This might be because there are no accumulation of errors in entity linking steps, which are previously done with mention detection and entity disambiguation, thanks to direct retrieval with end-to-end learning; but also the fact in the triplet form has more beneficial information to retrieve contrary to the entity retrieval.

3.2.5 Summary

In this work, we focused on the limitations of the conventional fact retrieval pipeline, usually consisting of entity mention detection, entity disambiguation and relation classification, which not only requires additional labels for training each subcomponent but also is vulnerable to the error propagation across submodules. To this end, we proposed the extremely simple Direct Fact Retrieval (DiFaR) framework. During training, it requires only pairs of input texts and relevant triplets, while, in inference, it directly retrieves relevant triplets based on their representational similarities to the given query. Further, to calibrate the ranks of retrieved triplets, we proposed to use a reranker. We demonstrated that our DiFaR outperforms existing fact retrieval baselines despite its great simplicity, but also ours with the reranking strategy significantly improves the performances; for the first time, we revealed that fact retrieval can be easily yet effectively done, and we believe our work paves exciting new avenues for fact retrieval.

3.3 Universal Retrieval Across Heterogeneous Knowledge Bases

3.3.1 Motivation



Which international collaborations has KAIST established in AI?

What international collaborations and partnerships in artificial intelligence has KAIST established with overseas universities, research institutes, and industry partners?

```
SELECT DISTINCT ?collab
?collabLabel ?partner ?partnerLabel
?country ?start ?end

WHERE {
BIND(<http://example.org/entity/KAIS
T> AS ?kaist)

...
```

```
SELECT c.id, c.title,
c.start_date, c.end_date,

FROM collaborations AS c

JOIN collaboration_participants AS
cp1 ON cp1.collab_id = c.id

...
```

KAIST

32 languages

Article Talk Read Edit View history Tools

From Wikipedia, the free encyclopedia

Coordinates: 36.372°N 127.363°E

Not to be confused with Korea Institute for Advanced Study or Korea Institute of Science and Technology.

KAIST (originally the **Korea Advanced Institute of Science and Technology**) is a national research university located in Daedeok Innopolis, Daejeon, South Korea. KAIST was established by the Korean government in 1971 as the nation's first public, research-oriented science and engineering institution.[3] KAIST has been internationally accredited in business education,[4] and hosts the Secretariat of the

Customer		
CustomerID	int	PK
FirstName	varchar(100)	N
LastName	varchar(100)	N
Email	varchar(100)	N
Email	varchar(255)	N

OrderDetail	
OrderID	int PK FK
ProductID	int PK FK
Quantity	int N

Order	
OrderID	int PK
Custom	int N FK
OrderD	date N

Product		
ProductID	int	PK
ProductName	varchar(100)	N
Price	decimal(10,2)	N
CategoryID	int	N FK

Category		
CategoryID	int	PK

Figure 3.8: Overview of the proposed Universal Retriever: given a natural-language query from a user, it predicts task tokens, conditions on metadata, and generates source-specific retrieval queries (SQL, SPARQL, or lexical terms) for relational databases, knowledge graphs, and unstructured documents.

While prior retrieval methods address individual classes of knowledge (such as unstructured text corpora, multimodal documents, videos, or structured knowledge graphs), they assume that all queries can be addressed with a single homogeneous corpus [183, 138, 184, 185, 21, 17, 22]. However, real-world information needs rarely conform to a single modality or schema. A single user query may require SQL-style relational data, SPARQL-style graph relations, textual descriptions, or even multimodal evidence. As a result, retrieval might become a bottleneck not because relevant information is absent, but because existing retrievers lack the flexibility to select, access, and integrate the appropriate knowledge source.

However, heterogeneous knowledge bases differ not only in modality but also in schema, granularity, and access interface. For instance, relational databases require compositional operators over normalized tables [260, 6]; knowledge graphs expose entities and relations through symbolic triples [82, 32]; and documents are typically accessed through lexical or embedding-based ranking [183, 184]. In other words, since each knowledge source mandates a distinct query format, representation space, and retrieval mechanism, a single retriever (from existing literature) might not capture these structural differences. As a consequence, current contextualization approaches (even when paired with strong LLMs) struggle, since the retriever cannot serve as a universal access layer across heterogeneous knowledge sources, preventing the model from fully leveraging the breadth and diversity of all the available information.

This motivates a universal retrieval framework that unifies access to diverse knowledge bases while preserving their heterogeneous interfaces and data structures. Specifically, instead of forcing all knowledge into a single embedding space like DiFaR [21] (in Section 3.2), an approach that often introduces modality gaps and biases toward sources that resemble the query [18], we propose an any-to-any transformation framework (called Universal Retriever), where the model is designed to take any natural-language query and generate the appropriate retrieval request for any knowledge source, together with the corpus-specific metadata information (if available) needed for accurate query formulation. To be more specific, the proposed method (1) identifies the most relevant knowledge source for the given query by predicting the special tokens for it, (2) interleaves associated metadata (such as schemas for databases) to provide the structural scaffolding required for valid query formulation for retrieval, and (3) generates source-specific retrieval queries such as SQL for relational tables, SPARQL for knowledge graphs, or text-based lookups for embedding-based retrieval of unstructured documents, illustrated in Figure 3.8.

To validate the effectiveness of the universal any-to-any transformation capability of the proposed Universal Retriever, we conduct experiments across three representative and structurally diverse benchmarks: Spider for relational databases [260], KQA Pro for knowledge graphs [261], and BEIR for unstructured document retrieval [262]. Empirically, the Universal Retriever then demonstrates remarkably high source-selection accuracy, consistently produces structurally valid SQL and SPARQL queries, and yields substantial improvements in retrieval correctness compared to single-modality baselines. These results confirm that modeling retrieval as the any-to-any transformation is both feasible and effective, enabling a single retriever to operate robustly across heterogeneous knowledge bases.

3.3.2 Related Work

Information Retrieval Classical information retrieval methods have primarily focused on retrieving relevant items from the homogeneous corpus, most often unstructured text. For instance, early lexical methods such as TF–IDF and BM25 [138, 121] establish strong baselines for document ranking, while neural dense retrievers [79, 80] expand retrieval to embedding-based similarity over large text collections. Subsequent extensions incorporate non-text modalities (such as images and videos) with modality-specific encoders [171, 17]. Yet, despite their improvements, they remain designed around a single knowledge base, where the corpus is assumed to share a unified representation space and access interface. Although recent efforts attempt to bridge heterogeneous corpora: DiFaR [21] projects the facts (within knowledge graphs) into a unified embedding space to enable embedding-based retrieval, they still assume that all sources can be collapsed into a single representation space, leading to the inability to leverage source-specific constraints (such as symbolic relations or compositional operators). In other words, existing approaches are not equipped to handle knowledge sources that differ in schema (e.g., knowledge graphs or relational databases) and their corresponding structural constraints, simultaneously, fundamentally preventing them from providing universal, schema-aware access across heterogeneous knowledge bases.

Any-to-Any Transformation Large Language Models (LLMs) operate fundamentally as any-to-any transformation engines (capable of mapping any input sequence to any desired output format), a principle that has enabled a wide range of capabilities such as code generation, tool invocation, structured prediction, and symbolic reasoning [7, 8, 9, 10, 11]. Additionally, with this any-to-any transformation paradigm, prior works on schema-aware text-to-SQL generation and graph query synthesis demonstrate early forms of the capability of LLMs for retrieval query generation [263, 264, 265, 266]. However, these works typically study transformations within a single modality or for a specific target interface (e.g., only SQL generation). In contrast, the proposed setting of universal retrieval requires an LLM to extend this any-to-any capability across heterogeneous knowledge sources, each with its own schema, language, granularity, and access mechanism, enabling a single model to interface with them in a unified manner.

3.3.3 Approach

We begin by describing LLMs, used as a core building block for any-to-any universal retrieval.

Large Language Models LLMs can be viewed as conditional sequence transducers that map an arbitrary input token sequence $\boldsymbol{x} = [\boldsymbol{x_1}, \boldsymbol{x_2}, \ldots, \boldsymbol{x_n}]$ into an output token sequence $\boldsymbol{y} = [\boldsymbol{y_1}, \boldsymbol{y_2}, \ldots, \boldsymbol{y_m}]$ by modeling the distribution: $\boldsymbol{p_\theta(y \mid x) = \prod_{t=1}^{m} p_\theta(y_t \mid x,\ y_{<t})}$. Under this formulation, the model can generate both structured and unstructured artifacts, including natural language tokens and symbolic expressions (such as SQL queries or SPARQL statements), purely through token-level autoregression. In light of this, we exploit this property as the foundation of the proposed Universal Retriever, which treats retrieval as the any-to-any transformation problem: given a natural-language query, the model learns to (1) infer which knowledge base should be accessed, (2) incorporate source-specific metadata when necessary, and (3) synthesize a retrieval query tailored to the target corpus.

Predicting the Relevant Knowledge Source Given a natural-language query $\boldsymbol{q}$, the first process of the Universal Retriever is determining which knowledge base contains the information needed to answer it. We operationalize this through training the model to predict task tokens: special output tokens that represent target knowledge sources such as relational databases, knowledge graphs, or unstructured text corpora. In other words, the model learns to map a query to a specific task token indicating whether the answer is most likely found in SQL tables, graph-based relations, or unstructured documents.

Interleaving Metadata Information Knowledge sources such as relational databases and knowledge graphs require structural awareness for valid query formulation. For example, SQL generation requires information about available tables, their attributes, and their relational keys; SPARQL generation requires knowledge of entities, relations, and triple patterns. To equip the LLM with this structural scaffolding, we append metadata tokens (i.e., compact schema descriptions or graph summaries) to the model input. Notably, these metadata tokens are not generated by the Universal Retriever but supplied externally as part of the retrieval context (after predicting the task tokens), enabling the LLM to reason over schemas and produce syntactically correct, semantically grounded queries. This design preserves the heterogeneity of data sources while making them accessible through a unified token interface.

Generating Source-Specific Retrieval Queries Once the Universal Retriever (powered by an LLM) predicts the correct knowledge source and receives the corresponding structural metadata, it proceeds to generate a source-specific retrieval query tailored to that source's interface: for relational databases, the model outputs SQL statements with appropriate SELECT, JOIN, GROUP BY, and WHERE operators; for knowledge graphs, the output is the SPARQL query with triple patterns and variable bindings; for unstructured corpora, the model may emit a list of lexical query terms (that could potentially be enriched from original queries) for embedding-based retrieval. After that, the generated retrieval query is executed over the selected knowledge source (from the task tokens) using its native interface, thereby retrieving the corpus-specific evidence. We emphasize that our Universal Retriever accomplishes this entirely through autoregressive decoding (leveraging both the predicted task token and the appended metadata tokens as inputs for query formulation), which means this unified procedure allows it to function across structurally diverse knowledge bases without forcing them into a single retrieval mechanism.

3.3.4 Experiments

We validate the Universal Retriever on a suite of benchmarks designed to test its ability to operate across heterogeneous knowledge bases, each characterized by distinct schemas and retrieval mechanisms.

3.3.4.1 Data

We evaluate the Universal Retriever across three representative and structurally diverse benchmarks:

- **Relational Databases:** Spider [260] contains 200 databases (with 138 different domains) and 10,181 natural-language questions that require generating SQL queries involving multiple tables, joins, aggregations, and nested operations. This dataset is designed to test the capability of the model to synthesize SQL expressions (alongside the metadata tokens retrieved from the predicted task tokens).
- **Knowledge Graphs:** KQA Pro [261] consists of natural-language queries over a large-scale Freebase knowledge graph [82], requiring multi-hop reasoning and symbolic relation traversal. Notably, retrieval is performed through SPARQL, and queries should be constructed using appropriate triple patterns. This benchmark is designed to evaluate whether the model can produce valid SPARQL queries.
- **Unstructured Documents:** BEIR [262] is an extensive benchmark consisting of various unstructured text retrieval tasks, operating through lexical or embedding-based similarity search. Compared to other corpora, the model should emit lexicalized query expressions (rather than structured symbolic forms), which tests whether the model can fall back to text-based retrieval when appropriate.

3.3.4.2 Baselines and Our Model

We compare the proposed Universal Retriever against a simple yet representative baseline setup: **Single-Modality Retriever**, which assumes access to only one knowledge source at a time and therefore cannot select or interface with heterogeneous corpora. In contrast, our **Universal Retriever** is a single model designed to cover different knowledge sources, which infers the target knowledge base through task-token prediction, incorporates metadata tokens for schema-awareness, and generates retrieval queries.

3.3.4.3 Evaluation Metrics

To evaluate the effectiveness of the Universal Retriever, we measure performance along two dimensions. First, we assess **task-token prediction accuracy**, which quantifies whether the model correctly identifies the target knowledge source required for a given query. Second, we measure **retrieval-query correctness** by comparing the generated SQL, SPARQL, or lexicalized queries against the ground-truth queries (i.e., a generated query is considered correct if it is equivalent to the reference from benchmarks).

3.3.4.4 Experimental Results and Analyses

Task-Token Prediction Accuracy We first measure the capability of the model to determine which knowledge base should be accessed for a given natural-language query. Then, the proposed Universal Retriever achieves 99.00% task-token prediction accuracy, indicating that it can reliably infer which knowledge base contains the information needed for a given query. Notably, this high accuracy is crucial, as an incorrect source prediction would propagate errors downstream, preventing valid query generation regardless of the query formulation capability of the model. In addition,

Retrieval-Query Correctness We next assess whether the Universal Retriever can generate retrieval queries that are equivalent to the gold-standard reference queries (i.e., structurally valid and semantically correct) across different corpora. Then, we observe that compared to a single-modality retriever (which is limited to one corpus-specific interface and therefore cannot adapt to heterogeneous data sources), which achieves the performance of 33.33%, our Universal Retriever improves this to 56.93%, demonstrating that explicitly modeling retrieval as an any-to-any transformation leads to significantly more accurate query generation. Furthermore, to understand the upper bound of the query formulation capabilities of the Universal Retriever, we evaluate an oracle setting in which the model is provided with the correct task token during inference (i.e., perfect knowledge-source identification). Under this condition, the Universal Retriever achieves a correctness score of 63.33%, indicating that while the model is already highly effective at selecting knowledge sources, providing perfect routing signals removes the remaining ambiguity and allows the LLM to focus solely on generating precise, source-specific queries.

3.3.5 Summary

In this work, we introduced the Universal Retriever, a unified framework that treats retrieval as an any-to-any transformation problem and enables a single LLM to interface with heterogeneous knowledge bases in a schema-aware manner. In particular, by decomposing the retrieval process into three stages: predicting the relevant knowledge source through task tokens, conditioning on corpus-specific metadata information (when available), and generating source-specific retrieval queries, the proposed framework provides a principled mechanism for accessing relational databases, knowledge graphs, and unstructured text corpora within a single model. Empirically, the Universal Retriever achieves high source-selection accuracy and substantially improves retrieval-query correctness compared to single-modality approaches, demonstrating that modeling retrieval as an any-to-any transformation is both feasible and effective. These findings collectively highlight that a transformation-centric view of retrieval not only preserves the structural heterogeneity of diverse knowledge sources but also unlocks a scalable and generalizable pathway toward universally accessible retrieval systems, opening exciting avenues for future work.

4.1 Knowledge-Augmented Model Contextualization for Personalized Contextual Query Suggestion

4.1.1 Motivation

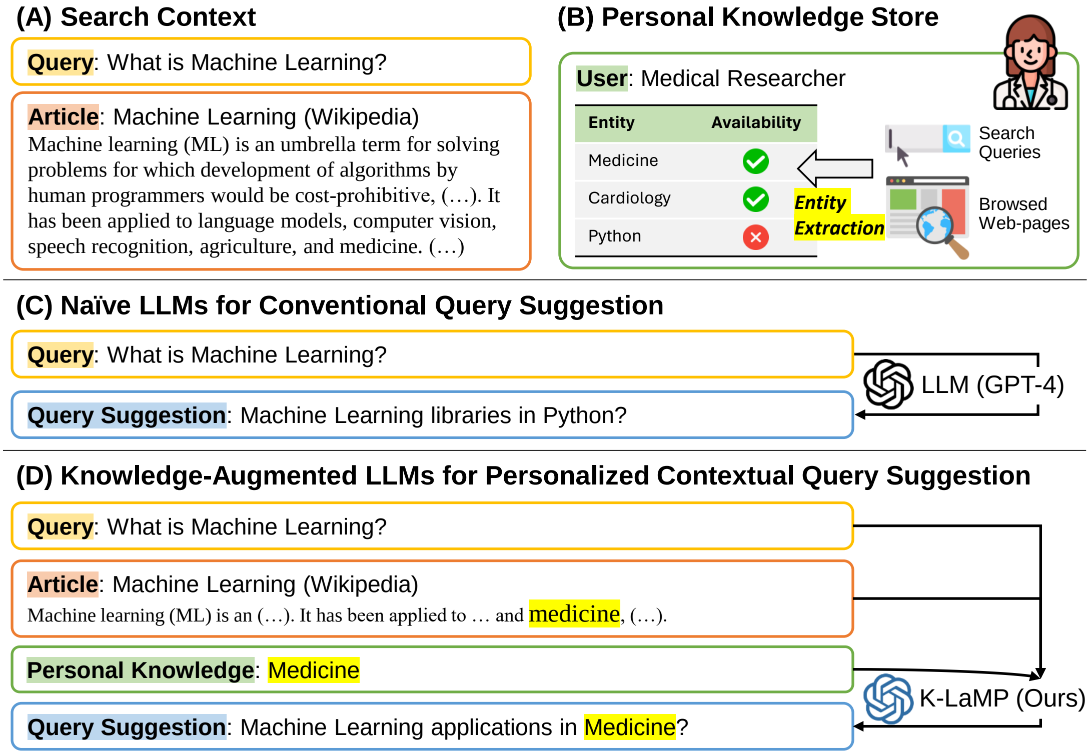

(A) Search Context
Query: What is Machine Learning?
Article: Machine Learning (Wikipedia)
Machine learning (ML) is an umbrella term for solving problems for which development of algorithms by human programmers would be cost-prohibitive, (…). It has been applied to language models, computer vision, speech recognition, agriculture, and medicine. (…)
(B) Personal Knowledge Store
User: Medical Researcher
Entity
Availability
Medicine ✓
Cardiology ✓
Python ✗
Entity Extraction
Search Queries
Browsed Web-pages
(C) Naïve LLMs for Conventional Query Suggestion
Query: What is Machine Learning?
Query Suggestion: Machine Learning libraries in Python?
LLM (GPT-4)
(D) Knowledge-Augmented LLMs for Personalized Contextual Query Suggestion
Query: What is Machine Learning?
Article: Machine Learning (Wikipedia)
Machine learning (ML) is an (…). It has been applied to … and medicine, (…).
Personal Knowledge: Medicine
Query Suggestion: Machine Learning applications in Medicine?
K-LaMP (Ours)

Figure 4.1: Illustration of the Knowledge-augmented large Language Model for Personalization (K-LaMP) framework for contextual query suggestion. (A) A user's search context includes a current query and a page being viewed. (B) The user has knowledge of medicine, extracted from past search activities. (C) Conventional query suggestion with naïve LLMs generates a query unrelated to the user's knowledge and the context. (D) K-LaMP suggests a query that is personally and contextually relevant.

Large Language Models (LLMs) [267, 123, 102, 268, 40, 113] are multi-billion parameter models trained on massive corpora, capable of internalizing general knowledge across diverse domains [28, 269]. This capability allows them to generate plausible, reasonable, and helpful outputs in response to user inputs that been leveraged with impressive results for a diverse range of natural language tasks, including question answering and dialogue generation, even without any task-specific training [123, 268, 113].

However, customizing LLMs to generate personalized responses, which take into account the individual preferences, needs, knowledge and context of users, with the goal of making them more meaningful and relevant to each user, remains challenging, due to the fact that re-training or fine-tuning LLMs for individual users is prohibitively expensive. Meanwhile, for some applications, such as query suggestion [270, 271], item recommendations [272, 273], snippet generation [274] or question answering [275], reflecting personal preferences, knowledge and needs of users in the model's outputs is essential.

Several recent studies [276, 277, 278, 279, 280, 281] have tackled the problem of LLM personalization through augmenting the user's input with relevant information, with in-context learning [267, 282, 283, 284]. For example, in order to suggest the next item that the user may interact with, they prepend a sequence of their past item interactions into the LLM's input [285, 281]. Due to the often large volume of users' historical information, more recent work [286, 287, 288, 289] proposes to inject only a fraction of the most relevant history by retrieving it from a complete interaction memory. Such retrieval-based personalization, while demonstrating success on several tasks including product recommendation and writing assistance [285, 289, 286, 290], remains ill-suited for more challenging personalization scenarios where a deeper understanding of users' personal *knowledge* is crucial. Meanwhile, some studies [291, 292] enable personalization through construction of deep user profiles that are then incorporated into LLM prompts. Nevertheless, such profile-based personalization captures user knowledge at the cost of privacy and scalability, requiring online modeling beyond the capabilities of existing logging infrastructure.

Thus, in this paper, we introduce a novel approach to personalizing the output of LLMs. Our method revolves around *an entity-centric light-weight personalization layer* that enables knowledge-augmentation of LLMs with contextual entities retrieved from a *personal knowledge store*. This knowledge store is derived from existing search logs that capture users' interactions with modern search engines. Specifically, this store is built over time by aggregating entities that appeared in queries that the user issued, or web-pages that they browsed, and is further enhanced by different views that capture the entities the user may be familiar or unfamiliar with, and those that may have recently lapsed from their memory.

This entity-centric personalization strategy has several advantages. First, contextually relevant retrievals from this entity-centric knowledge store encourage LLMs to generate outputs that are more deeply grounded in what users know and care about as compared with linearly stored past query logs. At the same time, it largely relies on already existing logging infrastructure, which means that it is more amenable to privacy, flexibility and scalability considerations than profile-based personalization, as it reduces data collection, modeling and update overheads. Also, thanks to its light-weight design, our approach offers easy integration with existing LLMs for other personalization tasks, both in search [274, 275] and beyond [286, 290]. Finally, our knowledge augmentation method is cost efficient since the knowledge injection employs entities as atoms. This results in minimal, succinct additions to the prompt, unlike other LLM contextualization approaches that operate over raw text [282, 283, 284]. We refer to our framework as **K**nowledge-augmented large **La**nguage **M**odels for **P**ersonalization, or K-LaMP.

While our method of personalizing LLM outputs is broadly applicable to problems in search (and beyond), it is especially relevant to tasks that require modeling the *knowledge* of users in addition to their interests. One such task is a new variant of query suggestion [293, 294, 295, 296], called *contextual* query suggestion. In this variant, a system must recommend queries to a user conditioned on a web-page they are currently reading, in addition to their historical query information. Thus, in this setting, knowing a user's domain of expertise and proficiency about a particular topic can lead to substantially different suggestions, as shown in Figure 4.1. We note that contextual query suggestion is different from existing *context-aware* query suggestion in the literature [297, 298, 299, 300], since the latter neither conditions recommendations on the body of the web-page being viewed by the user, nor explicitly captures the user's knowledge, instead focusing on surface-level relationships between queries and pages, or their titles.

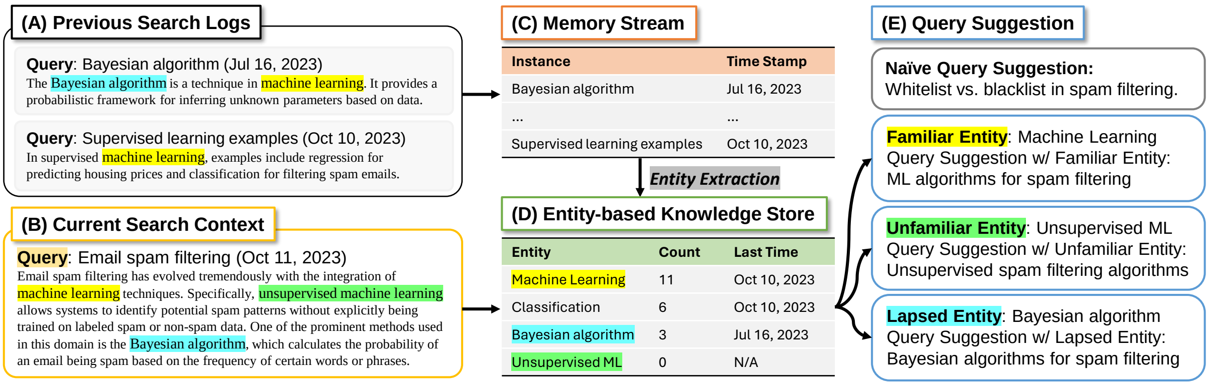
In our study, we validate the effectiveness of K-LaMP for contextual query suggestion, using real-world search logs from the Bing search engine [301], which is also used to construct our entity-centric knowledge store. On a battery of tests conducted via human evaluation, we find that K-LaMP substantially outperforms several LLM-powered (contextual) query suggestion baselines in generating recommendations that are better related and more useful to individuals, while maintaining high search result quality. Further analyses demonstrate that K-LaMP retrieves contextually relevant knowledge in a highly effective manner, and continues to become more performant as longer user interaction histories are processed and stored, neither of which other baselines are capable of doing.

4.1.2 Related Work

Large Language Models Language models [26, 37, 39, 27], which are pre-trained on unannotated text corpora using Transformer architectures [36] based on self-supervised learning objectives, have been shown to acquire knowledge from text corpora [28, 269, 42] and have been successfully used for various natural language tasks, such as question answering and dialogue generation tasks [108, 302]. Recently, Large Language Models (LLMs) [123, 268, 113], which are scaled-up versions of language models, have demonstrated the capability of handling diverse language tasks across various domains. In particular, LLMs have shown increased capacity for knowledge acquisition and retention thanks to their very large number of parameters [303, 304], as well as a remarkable ability to generalize across new domains with no need for additional task-specific fine-tuning and training data [46, 45]. Moreover, they are able to understand the context of given inputs and then generate contextually coherent responses, allowing users and system designers to easily customize LLM responses through prompt engineering [282, 283, 284]. For example, to generate factually correct answers in response to input questions, existing work [93, 15, 112] typically augments the internalized knowledge in LLMs with externally relevant factual knowledge related to questions. However, while they can effectively provide *generic responses* that may apply to a broad range of users, it remains challenging to generate *personalized responses* that capture the unique preferences, needs, and knowledge of individual users.

Large Language Models for Personalization In order to yield outputs that are customized to individual users, recent studies [305, 306] propose to personalize the generations of LLMs, with applications spanning various tasks and domains. These include product or content recommendations [285, 289, 307, 308], dialogue generations [288, 287], writing assistants [286, 290], and even robotic systems [309]. Specifically, early work [276, 277, 278, 279, 280, 281] proposes to incorporate the historical sequence of the user's interactions (e.g., recent purchase logs of items) into LLMs prompts, thereby allowing LLMs to generate outputs that are personalized (e.g., next item recommendation). While this simple, linear injection mechanism can effectively provide LLMs with relevant contextual information for personalization, it is limited by often very large interaction histories which exceed the capacity of LLM prompt windows. Also, not all of this history is relevant to every query. Based on this observation, recent work [286, 287, 288, 289] instead retrieves relevant content from an external memory [310] that stores the user's historical information. A few studies [287, 290] go a step further, processing the information in the interaction memory – for example, with summarization or key-word extraction – to gain higher-level insights from the user's history when contextualizing LLMs for personalization.

Unlike prior work in LLM output personalization that focuses largely on modeling the *interests* of users, our work additionally targets their *knowledge* over topics and domains of interest. Accordingly, rather than only leveraging the linearly stored interaction histories of users, we build a personal knowledge store consisting of entities mined from search queries and page visitations. This mechanism, which provides a lens through which user knowledge can be captured, has two additional advantages: it enables light-weight personalization by retrieval from the knowledge store without requiring explicit profiling of


(A) Previous Search Logs
Query: Bayesian algorithm (Jul 16, 2023)
The Bayesian algorithm is a technique in machine learning. It provides a probabilistic framework for inferring unknown parameters based on data.
Query: Supervised learning examples (Oct 10, 2023)
In supervised machine learning, examples include regression for predicting housing prices and classification for filtering spam emails.
(B) Current Search Context
Query: Email spam filtering (Oct 11, 2023)
Email spam filtering has evolved tremendously with the integration of machine learning techniques. Specifically, unsupervised machine learning allows systems to identify potential spam patterns without explicitly being trained on labeled spam or non-spam data. One of the prominent methods used in this domain is the Bayesian algorithm, which calculates the probability of an email being spam based on the frequency of certain words or phrases.
(C) Memory Stream
Entity Extraction
(D) Entity-based Knowledge Store
(E) Query Suggestion
Naïve Query Suggestion:
Whitelist vs. blacklist in spam filtering.
Familiar Entity: Machine Learning
Query Suggestion w/ Familiar Entity:
ML algorithms for spam filtering
Unfamiliar Entity: Unsupervised ML
Query Suggestion w/ Unfamiliar Entity:
Unsupervised spam filtering algorithms
Lapsed Entity: Bayesian algorithm
Query Suggestion w/ Lapsed Entity:
Bayesian algorithms for spam filtering

Instance	Time Stamp
Bayesian algorithm	Jul 16, 2023
...	...
Supervised learning examples	Oct 10, 2023

Entity	Count	Last Time
Machine Learning	11	Oct 10, 2023
Classification	6	Oct 10, 2023
Bayesian algorithm	3	Jul 16, 2023
Unsupervised ML	0	N/A

Figure 4.2: Overview of K-LaMP. The inputs to the system are (A) previous search logs and (B) the current search context of a user, which consist of a query and an associated web-page. (C) Each search record is stored in a memory stream along with a time stamp. (D) The entity-based knowledge store is constructed by aggregating entities extracted from the memory stream. (E) K-LaMP augmented with entities retrieved from the entity-based personal knowledge store, generates query suggestions that are compatible with the user's knowledge and interests.

users [291, 292]; the knowledge represented as entities is succinct, thereby leading to efficiency gains through reductions in input context length when compared with existing contextualization work [284].

Search Query Suggestion The goal of query suggestion is to recommend new queries of potential interest to users, based on current and previous queries in and across search sessions. This task is both highly practical and useful, having been shipped in web-scale search engines such as Google and Bing, as well as been widely applied to other tasks and domains, such as task-oriented search [311, 312] and recruitment platforms [313]. Early work on query suggestion has used frequency-based statistical (probabilistic) methods, which include Markov or LDA models [314, 297, 315, 316, 317]. More recently, neural network methods based on recurrent or attention-based architectures [293, 318, 294, 295, 298] have been leveraged to better model past query sequences and generalize to unseen and long-tail queries. Meanwhile, other studies have proposed to improve training strategies by performing either multi-task learning with a document ranker [299, 300, 296] or reinforcement learning [319]. Finally, other recent work [320, 321] uses pre-trained LMs [26, 322] to achieve superior performances with larger model capacity.

In comparison to prior work on query suggestion, we tackle the novel but practical task of *contextual* query suggestion, where recommendations are additionally conditioned on the web-page a user is currently viewing. This task is notably different from existing query suggestion work that leverages previously clicked pages [297, 298, 299, 300] only through surface-level association (such as relationships between past queries and page titles), because it requires contextualizing the full text of the page. This novel task is of particular interest to us because it exposes the need for personalized models that recommend queries based not only on what users are interested in, but also on what and how much they *know*.

4.1.3 Approach

In this section, we now introduce our approach to generating personalized outputs using a novel knowledge-augmentation method for LLMs and our entity-centric personal knowledge store. We also detail its application to the novel task of contextual query suggestion.

4.1.3.1 Problem Statement

We begin with preliminaries, introducing LLMs and the problem of contextual query suggestion.

Large Language Models Let us define an LLM as a model, parameterized by $\boldsymbol{\theta}$, that takes both an input sequence of tokens $\boldsymbol{x} = [\boldsymbol{x_1}, \boldsymbol{x_2}, ..., \boldsymbol{x_n}]$ and a supplemental sequence of context tokens $\boldsymbol{c} = [\boldsymbol{c_1}, \boldsymbol{c_2}, ..., \boldsymbol{c_k}]$ as a prompt, and then generates an output sequence of tokens $\boldsymbol{y} = [\boldsymbol{y_1}, \boldsymbol{y_2}, ..., \boldsymbol{y_m}]$. Then, formally, the inference of an LLM can be represented as: $\boldsymbol{y} = \texttt{LLM}_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{c})$. Here, $\boldsymbol{\theta}$ is the set of fixed model parameters typically pre-trained on massive text corpora; $\boldsymbol{x}$ is a task-dependant user-issued prompt or set of instructions; and $\boldsymbol{c}$ is some additional context provided by an auxiliary system that helps augment, ground, or otherwise improve the quality of the input, so that the LLM is able generate outputs $\boldsymbol{y}$ more effectively. This paper particularly focuses on the nature of $\boldsymbol{c}$ for the task of contextual query suggestion defined below, and with the use of entity-centric knowledge for more personalized outputs.

Contextual Query Suggestion Before formalizing Contextual Query Suggestion, we first define the task of conventional Query Suggestion. Let $\boldsymbol{q_j}$ be the most recent query issued by a user and $\boldsymbol{q_h} = [\boldsymbol{q_1}, \boldsymbol{q_2}, ..., \boldsymbol{q_{j-1}}]$ be a sequence of their historical queries. Then, a query suggestion model $\texttt{QS}$ aims to predict new queries $\boldsymbol{q_{j+1}}$ that an individual user with current query $\boldsymbol{q_j}$ and query history $\boldsymbol{q_h}$ might be likely to find useful. This process can be summarized as follows: $\boldsymbol{q_{j+1}} = \texttt{QS}_{\boldsymbol{\theta}}(\boldsymbol{q_j}, \boldsymbol{q_h})$.

Contextual query suggestion expands on this definition to incorporate a broader set of context $\boldsymbol{c} = [\boldsymbol{c_1}, \boldsymbol{c_2}, ..., \boldsymbol{c_k}]$ linearized as sequences of text. Specifically, let us first assume that $\boldsymbol{x}$ is an input query: $\boldsymbol{x} = \boldsymbol{q_j}$. Then, $\boldsymbol{q_h} \in \boldsymbol{c}$, meaning that the query history is one of the contextual signals capable of being leveraged for query suggestion. In this task, the text of a web-page $\boldsymbol{w}$ currently being consumed by the user is also included in $\boldsymbol{c}$: $\boldsymbol{w} \in \boldsymbol{c}$. Formally, this task can be summarized as follows: $\boldsymbol{q_{j+1}} = \texttt{QS}_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{c})$[1].

In this work, we attempt to solve the problem of contextual query suggestion by leveraging a knowledge-augmented model to yield more personalized outputs. Formally, for $\boldsymbol{q_{j+1}} = \texttt{QS}_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{c})$, we set $\boldsymbol{QS}$ to be an LLM (e.g., GPT-4 [123]) and include aggregated entity-centric knowledge from users' historical interactions in the context $\boldsymbol{c}$, in order to generate *better* recommendations $\boldsymbol{q_{j+1}}$, as measured by a set of personalization-focused metrics (see Section 4.1.4.3).

4.1.3.2 Knowledge Store for Output Personalization

We now discuss our knowledge-augmented LLM framework for personalization and describe two instantiations of this framework.

Recall that the supplemental context of an LLM $\boldsymbol{c}$ is provided by auxiliary sources or systems that help enrich the input prompt to the model. According to our goal of personalizing the LLM outputs, these auxiliary sources should ideally consist of data that captures the personal preferences, interests, and knowledge of users. Thus, if $\boldsymbol{\mathcal{K}}$ is a knowledge store that encapsulates these user-specific data, and $\boldsymbol{k} \in \boldsymbol{\mathcal{K}}$ is a contextually relevant subset linearized as text, then, for contextual query suggestion, the context $\boldsymbol{c}$ can be defined as follows: $\boldsymbol{c} = [\boldsymbol{q_h} \cdot \boldsymbol{w} \cdot \boldsymbol{k}]$, where $[\cdot]$ is the concatenation operation.

Given this general formulation, the important questions to answer are:

1. How is the personal knowledge store $\boldsymbol{\mathcal{K}}$ constructed? and
2. How are contextually relevant items $\boldsymbol{k}$ retrieved from the knowledge-store $\boldsymbol{\mathcal{K}}$.

We answer both of these questions below, focusing on construction first and retrieval next.

[1] Without the hard requirement of an input web document included in $\boldsymbol{c}$, this definition may be relaxed to capture prior work on *context-aware* query suggestion [297, 298, 299, 300].

Constructing Personal Knowledge Stores While a variety of sources may be used to construct a compendium of what a user cares about and knows, in this paper, we leverage their interaction histories with a web search engine. We argue that this is a natural choice given that users express goals, desires, interests, and depth of knowledge both explicitly and implicitly through issued queries, clicked web-pages, consumed content and behavioral patterns over time. It is also an especially relevant source of knowledge for contextual query suggestion, where generating helpful search suggestions from search histories is the eventual goal of the task. Given this source, we build two distinct instantiations of the knowledge store $\mathcal{K}$: a simple variant that linearly captures historical user queries and browsing patterns ($\boldsymbol{\mathcal{K}_s}$), and an entity-centric variant that aggregates users' personal interests and knowledge ($\boldsymbol{\mathcal{K}_e}$).

Personal Knowledge Store from Search History ($\boldsymbol{\mathcal{K}_s}$) The intuition behind this instantiation of the knowledge store $\boldsymbol{\mathcal{K}_s}$, is that users issue queries and click on web-pages that they are interested in or care about. And, when accumulated over time, these also start to construct a picture of what users know and how deeply do they know them. For example, if a user issues multiple queries over time that include "Machine Learning", "ML", "Optimization", "SGD", "Deep Learning" and clicks a number of web-pages resulting from these queries, we can assume that this user is at least familiar with the general concept of "Machine Learning".

In order to operationalize this intuition, we construct a time-stamped memory stream consisting of the queries issued by users and the web-pages associated with the results they clicked on (see Figure 4.2). Note that this is an extremely light-weight instantiation of the knowledge store, being only a partial view of user actions and interactions already logged by modern web-scale search engines. As a result, there are no privacy or scalability concerns beyond those already inherent in the search engine's logging system.

Personal Entity-centric Knowledge Store ($\boldsymbol{\mathcal{K}_e}$) While building a memory stream over users' search histories is simple, there are a few limitations that stem from its design. Firstly, because search queries and web-page visitations are stored and retrieved linearly, it is difficult to perform aggregations on the fly. Yet, such aggregation can be greatly beneficial for personalization. For example, knowing that a user clicked on web-pages associated with "Machine Learning" multiple times, while only clicking on a single web-page stemming from the query "Computational Biology" would tend to indicate a greater affinity for and knowledge of the machine learning subject. Other issues include the fact that individual web-pages visited by a user may contain mixtures of several different topics and domains, distracting LLMs in generating outputs consistent with the context, and the fact that including large amounts of text from lengthy web-pages renders LLM usage slow and expensive.

In order to address these concerns, we construct an entity-centric instance of the knowledge store $\boldsymbol{\mathcal{K}_e}$ (see Figure 4.2). Entities are useful atoms for capturing the interests and knowledge of users because they consist of the nouns (proper or otherwise) that describe the people, places, organizations, topics and domains that the users care and know about. Additionally, because they tend to be relatively short and easy to aggregate, and because entity recognition and linking [323, 255] are well-studied problems, the process of operationalizing the creation of this store is greatly simplified.

We employ an off-the-shelf state-of-the-art entity linking system [324] to tag and canonicalize the entities that appear in the search queries and associated web-pages visited by users. While individual occurrences of entities in the knowledge store are time-stamped, additional aggregation is done by counting the number of occurrences of entities in full user interaction histories.

While this entity-centric knowledge store instantiation $\boldsymbol{\mathcal{K}_e}$ may not be as minimalist as $\boldsymbol{\mathcal{K}_s}$, it is still relatively light-weight when compared with systems that personalize through the construction of deep profiles. The solution is scalable since the only external dependency is an existing entity linker, which can typically process thousands of tokens per second. Additionally, privacy concerns are also small because entity linking projects and aggregates mentions in text onto sub-graphs of public knowledge bases (e.g. Wikipedia). Aggregation of entity occurrences lends itself naturally to common privacy mitigation practices such as k-anonymization [325]. Additionally, records are amenable to easy removal upon request by simply eliminating associated entities from the store.

Contextual Retrieval from Personal Knowledge Stores We now turn to the question of retrieving contextually relevant items $\boldsymbol{k}$ from a knowledge-store $\boldsymbol{\mathcal{K}}$, conditioned on an input query $\boldsymbol{q_j}$ and a document $\boldsymbol{w}$ that the user is currently interacting with. A carefully considered retrieval step is essential in augmenting the capability of LLMs to produce personalized outputs, since it grounds generation in the historical interests and knowledge of users. In the following, we show how retrieval is performed for both instantiations of the knowledge store $\boldsymbol{\mathcal{K}_s}$ and $\boldsymbol{\mathcal{K}_e}$.

In the case of $\boldsymbol{\mathcal{K}_s}$ over users' search and browsing history, retrieval is done by finding and returning the most similar queries and previously visited web-pages to the current input $\boldsymbol{x}$. In practise, the queries are then elided from this result since they yield little benefit over the much longer text present in web-pages. To operationalize the retrieval step, we first represent all records in the knowledge-store $\boldsymbol{\mathcal{K}_s}$ using embeddings, then compute embedding-level similarities with the representation of current query $\boldsymbol{q_j}$ using Contriever [139]. The most similar records $\boldsymbol{k}$ are finally returned.

Meanwhile, for the entity-centric knowledge store $\boldsymbol{\mathcal{K}_e}$, retrieval is conditioned on the entities present in the current query $\boldsymbol{q_j}$ and the web-page $\boldsymbol{w}$, which are further matched against $\boldsymbol{\mathcal{K}_e}$. Given that entities are atomic units with associated counts and time-stamps, the matching and retrieval process can be operationalized in flexible ways (See Figure 4.2). We particularly explore three strategies for matching entities: familiar (entities frequently encountered by the user), unfamiliar (entities the user has encountered infrequently or not at all), and lapsed (entities that the user used to encounter previously but hasn't done so more recently). Specifically, for familiar entities, we sort the entities appearing in the search context $[\boldsymbol{x} \cdot \boldsymbol{w}]$ by frequency of occurrence in the knowledge store $\boldsymbol{\mathcal{K}_e}$, then sample 5 entities proportionately to their frequency. For unfamiliar entities, a similar process is used for sampling, except that entities are sorted inversely with respect to their occurrence in $\boldsymbol{\mathcal{K}_e}$. Finally, for lapsed entities, we start by filtering entities in $[\boldsymbol{x} \cdot \boldsymbol{w}]$ by time-stamp to retain only those that occur in $\boldsymbol{\mathcal{K}_e}$, but haven't been engaged with in the preceding two weeks. Then we sample from this filtered set of entities by frequency, much like we do with familiar entities.

4.1.4 Experiments

We first outline the datasets and models used in evaluation setups, as well as implementation details.

4.1.4.1 Data

We use real large-scale search logs from the Bing search engine [301]. Specifically, we sample three months of search logs, from May 01, 2023 to July 31, 2023. We then filter and sample this dataset to make it suitable for evaluating our task, which includes the following steps. First, because the task we are tackling is *contextual* query suggestion – i.e., recommendations are predicated on a current web-page the user is viewing – we filter out sessions that do not contain any clicked search results. We further filter the data to discard click events that lead to pages in domains other than Wikipedia or a curated set of 500 high-traffic news publishing sites. We do this because the entity linker [324] we use maps onto Wikipedia, and we want to maximize the chances of encountering linked entities. It is worth noting that K-LaMP itself is agnostic to the choice of the linker or its underlying knowledge graph, and our approach could readily be applied to a different domain, for example, using a linker over a product graph for shopping. Finally, we filter the remaining data to discard users who had fewer than 100 page visitations for three months, who we assume are infrequent search users. In addition, we perform and apply enterprise-level privacy checks and filters, such as using search queries requested from at least 50 individuals, to ensure that the data remains suitably anonymized.

The resulting data is still extremely large; therefore, we further randomly sample a subset of 1,000 users in order to get the benchmark set that forms the basis for all the evaluations we perform in this paper (Section 4.1.4.3). This final dataset, on average, contains **493** queries, **109** sessions, **177** clicked articles, and $\mathbf{3,053}$ encountered entities per user. For testing, we split the dataset and reserve the most recent 10 sessions of every user as prediction targets for contextual query suggestions and use all the earlier sessions to build search-and-browsing based ($\boldsymbol{\mathcal{K}_s}$) and entity-centric ($\boldsymbol{\mathcal{K}_e}$) personal knowledge stores for users, as described in Section 4.1.3.2.

4.1.4.2 Baselines and Our Model

We compare our approach to knowledge-augmented LLMs for output personalization against several relevant baselines that make query suggestions based on the search context of users. We note that, for the fairest comparison, all baselines and our model use LLMs (specifically GPT-4) to make query suggestions. Notably, we do not include prior query suggestion approaches based on older modeling techniques (e.g., RNNs or BART [293, 320]) in our experiments due to their limited capacity for understanding longer context and complex data inputs, particularly without dedicated training data (the framework we propose needs none). Also, these query suggestion techniques are not designed to handle longer contexts such as the web-page a user is currently viewing, thereby making a direct comparison trivially relevant at best.

The models evaluated in this work are listed as follows:

1. **Query Suggestion** – which uses a current query $\boldsymbol{q_j}$ and historical queries from $\boldsymbol{q_h}$ in the same session to suggest the next query $\boldsymbol{q_{j+1}}$;

2. **Contextual Query Suggestion** – which is similar to Query Suggestion, but further conditions the recommendation of the next query $\boldsymbol{q_{j+1}}$ on a web-page $\boldsymbol{w}$, clicked from the current query $\boldsymbol{q_j}$;

3. **Contextual Query Suggestion w/** $\boldsymbol{\mathcal{K}_s}$ – which includes retrievals from the knowledge store $\boldsymbol{\mathcal{K}_s}$ over users' historical search and browsing activities, as additional context to personalize the outputs of the LLM;

4. **K-LaMP** – which is our full model that augments LLMs with entity-centric knowledge from the knowledge store $\boldsymbol{\mathcal{K}_e}$ in order to perform contextual query suggestion.

4.1.4.3 Evaluation Setup

To evaluate the effectiveness of different query suggestion models on generating personalized outputs, a suitable evaluation metric should ideally not only capture whether the suggested queries are contextually relevant, but also whether they align well with the user's interests and knowledge. Given that contextual query suggestion is a novel problem we propose in this paper, no existing evaluation

Table 4.1: A manufactured example showing the type of data that was provided to human judges.

Types	**Texts**
Query	Tim Cook
Session	Apple, Tim Cook
Article	Tim Cook Leadership: A new profile examines how Apple CEO Tim Cook, with "cautious, collaborative and tactical" leadership, honed the Cupertino tech giant into the world's largest company. (. . .)
Trending entities	'GPT-4', 'OpenAI', 'Google Bard', 'Microsoft Copilot', 'Elon Musk', . . .
Personal summary	The user is interested in Apple products and technology ('Macbook', 'macOS', and 'Apple TV'). They have a keen interest in ML, with topics like 'Supervised Learning' and 'Optimization'. Additionally, they enjoy animation, showing interest in 'Studio Ghibli', 'Walt Disney', and 'Pixar'. Their preferences also extend to home entertainment ('DVD' and 'HDTV'). Lastly, they follow baseball ('MLB' and 'New York Yankees').
Personal entities	'Macbook', 'macOS', 'Machine Learning', 'Optimization', 'Supervised Learning', 'Apple TV', 'Animation', 'Studio Ghibli', 'DVD', 'Walt Disney', 'Pixar Animation Studios', 'Apple Inc.', 'Baseball', 'HDTV', 'Major League Baseball', 'New York Yankees', . . .

metrics are available for the task. In particular, metrics for evaluating conventional query suggestion are not applicable here because they do not account for the full *context* present in our task – namely the input document being consumed by the user. Therefore, we turn to human evaluation in order to measure and compare the different models on our experimental benchmark.

It is worth noting that human evaluation of any form of personalization is difficult, since the person performing the evaluation is rarely the person from whom the data originated. Short of flighting and A/B testing our system in a real-world setting – an engineering endeavor well beyond the scope of the scientific exploration in this paper – any evaluation on our task and dataset must be bounded by this constraint. Nevertheless, we attempt to provide annotators with as much information as possible in order to understand both the user's current search context and their personal interests and knowledge (see Table 4.1 for an example). In particular, to summarize the current search context for annotators, we show them the current and previous search queries of a user in a given session, the web-page the user clicked on after issuing the current search query, and a list of 20 trending entities which capture statistical surges in search volume across users. Additionally, in order to present an encapsulation of the personal interests and knowledge of users, we show annotators a list of the 30 most frequent entities from the user's personal entity-centric knowledge store, as well as a GPT-4 generated summary from these entities that states what topics or domains the user may know or care about much.

Presented with this data and recommended queries from the different baselines and our model (where the system names are obscured to annotators), a human judge is asked to evaluate the following three metrics on a 3-point Likert scale[2]:

1. **Validity** – whether an output query can be input into a search engine and be expected to yield relevant results;

2. **Relatedness** – whether the output query relates to the user's personal interests and knowledge;

3. **Usefulness** – whether the user is likely to click on the output query, given their historical interests and knowledge as well as their current search context.

[2] The 3-point Likert scale is composed of agree (2), neutral (1), and disagree (0).

Table 4.3: Main results on our contextual query suggestion task. The best results are marked in bold.

Types	Models	Validness (↑)	Relatedness (↑)	Usefulness (↑)	Ranking (↓)
Baselines	Query Suggestion	1.769	0.962	0.948	2.736
	Contextual Query Suggestion	**1.966**	1.267	1.245	2.415
	Contextual Query Suggestion w/ $\mathcal{K}_s$	1.822	1.192	1.166	2.654
Ours	**K-LaMP (Ours)**	**1.966**	**1.482**	**1.455**	**2.160**

Finally, we also ask the annotators for a fourth measure: 4. **Ranking** – where the outputs of the different systems are ranked according to the order in which they are likely to be clicked, based on their affinity to the user's interests, knowledge, and search context. Collectively, these four metrics capture how good the different query suggestions are, – both individually and in relation to one another – but also how well they align with the *personal* aspects of our evaluation task; namely, what users care about and know.

To perform evaluations with human judges, we recruit 12 annotators in India through a third-party vendor company [326]. They were provided with a guideline document, which includes instructions for the task, metrics and some annotated examples, and they were paid $11.98 per hour for the time they spent working on the task. Over several rounds of judgement and refinement, we obtain manual evaluation results for $\mathbf{1,309}$ sets of contextual query suggestion results from all four models listed in Section 4.1.4.2 (effectively a total of $\mathbf{5,236}$ annotations for individual query suggestions).

Table 4.2: Results of inter-annotator agreements on all query suggestion results evaluated by annotators.

Agreements	Metrics	Scores (↑)
Exact match	Validness	0.963
	Relatedness	0.850
	Usefulness	0.819
Cohen's kappa coefficient	Validness	0.606
	Relatedness	0.652
	Usefulness	0.622
Spearman's correlation coefficient	Ranking	0.654

Additionally, to validate the quality of annotations and to measure inter-annotator agreement, approximately **27%** of the data is annotated by two human judges. Specifically for Validity, Relatedness, and Usefulness, we measure an exact match score, which checks often annotators provide the same score on the 3-point likert scale, and Cohen's kappa coefficient [327] which additionally discounts for chance agreement. For Ranking, we report Spearman's correlation coefficient [328], which measures the correlation between two sets of ranked systems, averaged across pairs of users and data instances. As shown in Table 4.2, inter-annotator agreement is moderate to high, indicating that judges are in fact able to make reasonably informed decisions about personalized contextual query suggestion from the data we provide them with.

Finally, regarding implementation details, we use the GPT-4 [123] release from June 13, 2023, as the basis for query suggestion across all baselines and model variants, for a fair comparison. We set the hyperparameters of GPT-4 as temperature = $\mathbf{0.7}$ and top$_{\boldsymbol{p}}$ = $\mathbf{0.95}$. The entity linker used to construct instantiations of the knowledge store is NEMO [323, 324][3].

4.1.4.4 Experimental Results

We now present experimental results and report findings from various auxiliary studies and analyses.

Main Results Our main results are shown in Table 4.3. This confirms that our K-LaMP framework consistently and significantly outperforms all other baselines across Relatedness, Usefulness, and Ranking

[3] We eschew more recent LM-based entity linkers [255, 254] since they have restrictive input token limits that are often exceeded by inputs in our scenario.

Table 4.4: Results of different retrieval strategies on Retrieval Relevance to the user's search context.

Retrieval	Types	Retrieval Relevance (↑)
History-based Retrieval ($\mathcal{K}_s$)	Past Documents	0.299
Entity-centric Retrieval ($\mathcal{K}_e$)	Familiar Entities	**0.936**
	Unfamiliar Entities	**0.810**
	Lapsed Entities	**0.849**

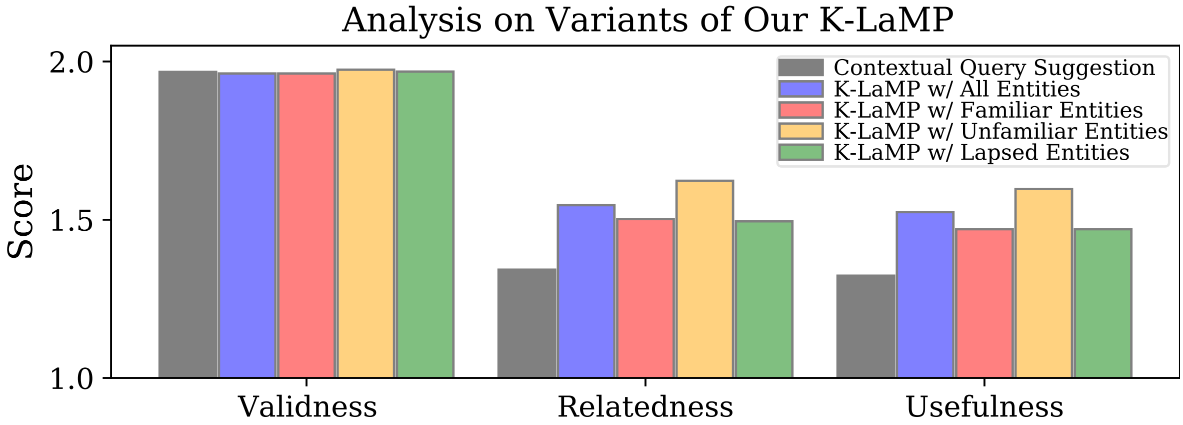


Figure 4.3: Results of variants of K-LaMP on knowledge retrieval strategy, and results without retrieval.

metrics. While it ties Contextual Query Suggestion on Validity, this finding is overall a positive albeit somewhat expected – since, intuitively, inclusion of personal context does not necessarily lead to queries that are more *valid* for search engine retrieval. Also, there are other interesting insights that can be gleaned. Contextual Query Suggestion with $\mathcal{K}_s$ does not outperform Contextual Query Suggestion. We hypothesize that this is because the information retrieved from the memory store ($\mathcal{K}_s$) has poor relevance to the current search context, leading to spurious augmentations that distract rather than help the LLM.

To investigate this hypothesis further, we conduct a supplementary evaluation, which asks human annotators to rate the information retrieved from knowledge stores for a particular search context (see Section 4.1.3.2 for details). Specifically, we report Retrieval Relevance from both instantiations of our knowledge stores in Table 4.4. This metric is the average score from a Yes/No question: whether the retrieved context is relevant to the current search context (1) or not (0). As shown in Table 4.4, the quality of retrievals from the entity-centric knowledge store is superior to those from the linear search history-based store. This is because we have far greater control with entities being the atomic units of the knowledge representation space, and are able to *exactly* match entities in the context against entities in the store, rather than rely on a fuzzy similarity-driven retrieval process with a dense retriever [139].

Ablation over Entity Matching Strategies Recall that K-LaMP relies on a combination of matching and retrieval of several different types of entities from its entity-centric knowledge store, namely: familiar, unfamiliar and lapsed entities (see Section 4.1.3.2). In order to individually measure the contribution of each strategy, we generate knowledge-augmented query suggestions on 313 search contexts using only one type of entity and ask human annotators to evaluate the results on Validity, Relatedness, and Usefulness. The comparative results are presented in Figure 4.3. Firstly, they reaffirm the fact that Validity is practically invariant to the choice of knowledge ingestion, since personal information does not affect whether a query is *valid* or not. The Relatedness and Usefulness metrics, however, are

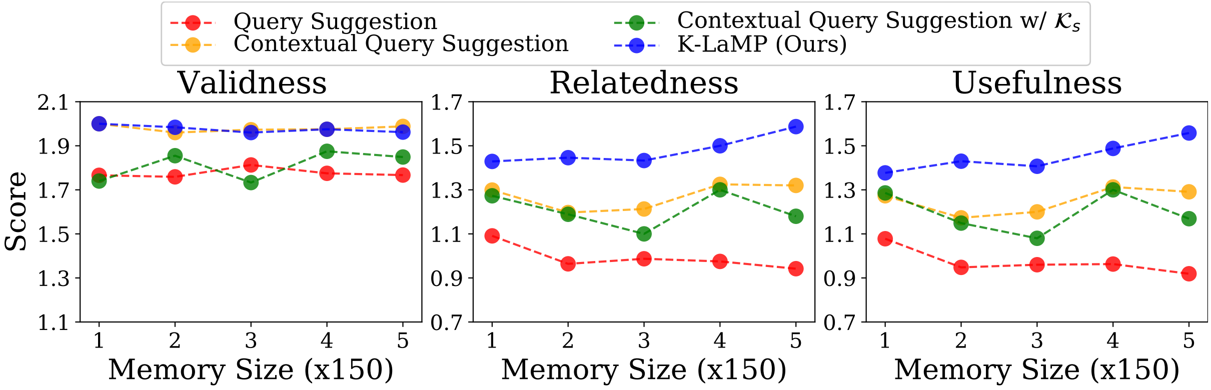


Figure 4.4: Results with different numbers of previous queries (i.e., different memory sizes).

Table 4.5: Results with different LLMs, namely GPT-3.5 and GPT-4.

Methods	LLMs	Validness	Relatedness	Usefulness
Query Suggestion	GPT-3.5	1.767	1.077	1.069
	GPT-4	1.747	1.080	1.060
Contextual Query Suggestion	GPT-3.5	1.967	1.177	1.202
	GPT-4	1.987	1.367	1.313
K-LaMP (Ours)	GPT-3.5	2.000	1.279	1.303
	GPT-4	1.983	**1.653**	**1.600**

clearly impacted by the choice of entity matching strategy in consistent ways. In particular, using only “unfamiliar” entities yields the highest scores across both metrics, even outperforming the full K-LaMP model. This seems to suggest that queries stemming from new (to the user) entities, which implicitly encourage exploration, are preferred over queries that revisit familiar ground.

While these results are true on this data, we note that real-world users may approach web search with different goals in mind: such as research, exploration, or revision, and it is difficult for judges to assess these goals from limited albeit rich data. As a result, we argue that having a comprehensive approach to capturing different views of the users’ knowledge so that it may be deepened, expanded, or revived as the use-case may demand, is a robust strategy.

Analysis over Interaction History Length A fundamental assumption in our setup for LLM output personalization is that we can learn about users as they interact with search engines. A natural follow-up question to this assumption is to ask how the performance of systems that rely on personal knowledge change as a function of the length of the interaction history. To answer this question, we conduct an analysis with varying history lengths and report results in Figure 4.4. From this, we observe again that Validity is not affected by the length of the interaction history, while Relatedness and Usefulness are. In particular, K-LaMP is the only model demonstrating consistent improvement with longer interaction histories, showcasing its ability to grow richer representations of personal interests and knowledge over time. A reasonable explanation for this increment is the aggregation that happens in K-LaMP’s entity-centric knowledge store, which contrasts with the linearly stored histories of the other approaches.

Analysis using different LLMs Finally, we conduct an auxiliary analysis to see how the quality of query recommendations from different systems change if an LLM other than GPT-4 is used. Specifically,

Table 4.6: Results with automatic evaluation metrics.

Types	Validness	Relatedness	Usefulness
Correlation w/ Human Evaluation	**0.445**	**0.397**	-0.016
Query Suggestion	1.784	1.189	0.882
Contextual Query Suggestion	1.891	1.340	0.831
Contextual Query Suggestion w/ $\mathcal{K}_s$	1.828	1.271	0.847
K-LaMP (Ours)	1.910	1.472	0.845

we use the June 13, 2023 version of GPT-3.5-Turbo as the LLM on 128 sets of query suggestions from two baselines (Query Suggestion and Contextual Query Suggestion) and our K-LaMP framework, and compare the results with GPT-4; these are shown in Table 4.5. Firstly, Query Suggestion is agnostic to the choice of LLMs, while Contextual Query Suggestion and K-LaMP are not. This is likely due to the fact that the latter two approaches must incorporate information from full web-pages as context and therefore benefit from the representational, reasoning and generative capabilities of the larger model. More relevant to the contributions in this paper, we find that, even with GPT-3.5-Turbo, K-LaMP shows comparable performance on the Usefulness metric with the second best model – Contextual Query Suggestion – despite the latter using GPT-4. This demonstrates the significant edge that an entity-centric representation of a user's personal interests and knowledge provides, for knowledge-augmented personalization of LLMs outputs, with high efficiency of deploying K-LaMP in real-world applications.

Automatic Evaluation Setup and Results While human evaluation is useful for measuring systems and gaining insights, especially on a new task like the one we introduce, the process is slow and expensive, and therefore not scalable to bigger datasets, or future extensions. To address these issues, we explore an initial set of automatic evaluation metrics mirroring the ones described in Section 4.1.4.3 that may be used in the absence of human judgement. Recall that even human evaluation for tasks that deal with personalization is non-trivial; therefore, automatically evaluating the outputs of a contextual query system while conditioning on complex personal preference and knowledge data is very difficult.

Nevertheless, we propose and experiment with the following automatic formulations:

1. **Validity** – we compute the similarity between the query suggestion output of a system and the top search result (title and snippet) returned from issuing that query to the web search engine (to see if the query yields reasonable search results);

2. **Relatedness** – we measure the similarity between the query suggestion and the set of contextual personal entities retrieved from the user's entity-centric knowledge store (to ensure that the query is grounded in the personal context of the user);

3. **Usefulness** – we calculate the similarity between the query suggestion and the real subsequent queries that the user ended up issuing (to compare recommendations against the true actions).

In each of these three metrics[4], similarity is computed by calculating the dot product of representations obtained from Contriever [139].

We validate these automatic evaluation metrics by ranking the systems on the test set, then computing Spearman's correlation against the ranking obtained by human judgement scores. As shown in

[4] We don't specify an automatic measure of Ranking, since this can be done trivially by scoring then sorting systems by one or more of the other automatic metrics.

Table 4.6, we find a moderate correlation on Validity and Relatedness, indicating that our proposed automatic metrics for these measures may be used as proxies in the absence of human labeling. However, there is no correlation between automatic and human Usefulness metrics. This is expected since (contextual) query recommendation is not expected to align perfectly with actual user behavior, which is the basis of our formulation for automatically computing Usefulness; users *should* be surprised and delighted by suggestions they would not have otherwise thought about.

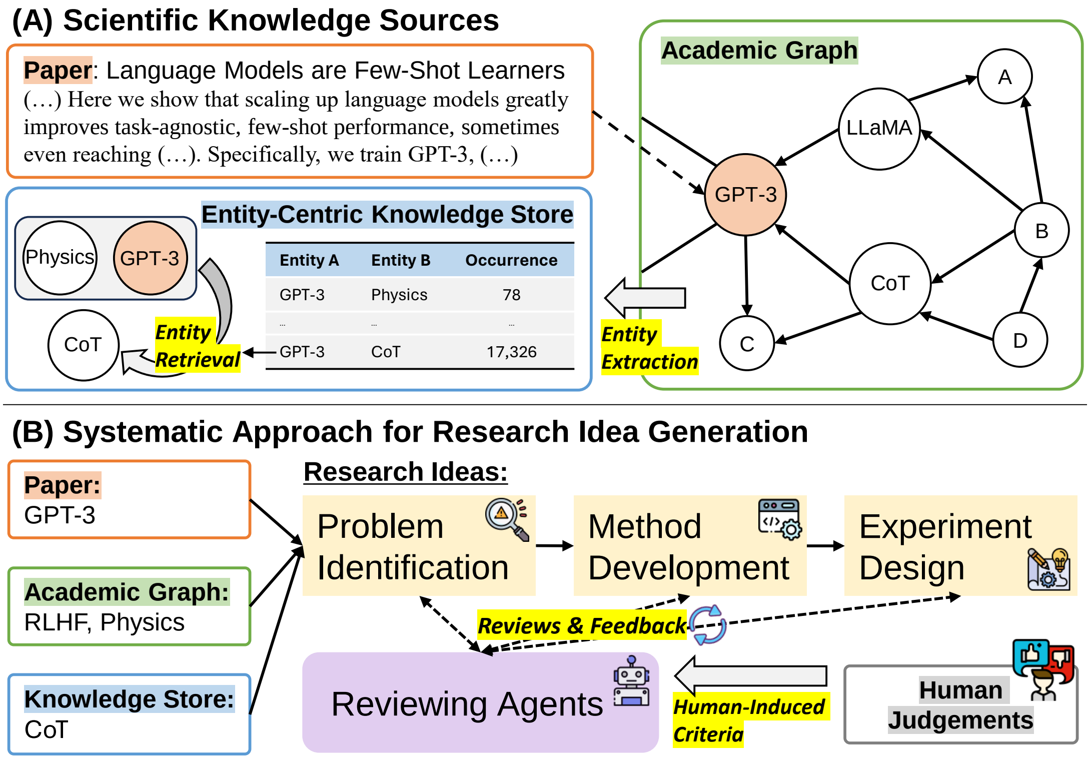
There are several ways to improve the automatic evaluation of contextual query suggestion. For example, we could use another LLM to perform a rubric-based evaluation of Validity, Relatedness and Usefulness, relying on its capacity to account for complex personal and preferential data. Or we could train parametrized versions of the automatic metrics we propose on manually labeled data with the goal of increasing correlation with human judgement. We leave these and other explorations to future work.

4.1.5 Summary

In this work, we proposed a knowledge-augmentation framework for LLM output personalization called K-LaMP, that leverages historical user interactions with a search engine. The core of the personal knowledge we used for LLM augmentation relies on a novel light-weight entity-centric personal knowledge store, constructed from the queries that users issue and the web-pages that they viewed as they search and browse the web. To stress-test our personalization framework, we focused on the novel task of contextual search query suggestion, which crucially requires modeling both the contextual interests and the knowledge of users. Through human evaluation on an extensive test set, we showed that our entity-centric knowledge-augmented LLM produces personalized query recommendations that are better related to users' intent, more useful, and consistently ranked above those produced by several other LLM-powered query suggestion models. Our findings show that entities are effective atomic units for the representation of personal knowledge, offering a robust middle-ground of performance, flexibility, privacy and scalability, when compared with other personalization approaches that rely either on deep profile building or simple linearization of a user's historical interactions. K-LaMP has the potential to impact both future research and product innovation. The use of personalized knowledge-augmentation for other search tasks such as snippet generation or question answering, the incorporation of other sources of data such as shopping or media-consumption histories, and the application to domains outside of search such as personal AI assistants, are all exciting avenues. At the same time, enhanced evaluation remains an important future goal, with improved automatic metrics and real-world deployment as potential directions for exploration.

4.2 ResearchAgent: Iterative Research Idea Generation over Scientific Literature with Model Contextualization

4.2.1 Motivation



(A) Scientific Knowledge Sources
Paper: Language Models are Few-Shot Learners
(…) Here we show that scaling up language models greatly improves task-agnostic, few-shot performance, sometimes even reaching (…). Specifically, we train GPT-3, (…)
Academic Graph
A
LLaMA
GPT-3
B
CoT
C
D
Entity-Centric Knowledge Store
Physics
GPT-3
CoT
Entity Retrieval
Entity A | Entity B | Occurrence
GPT-3 | Physics | 78
GPT-3 | CoT | 17,326
Entity Extraction
(B) Systematic Approach for Research Idea Generation
Paper: GPT-3
Academic Graph: RLHF, Physics
Knowledge Store: CoT
Research Ideas:
Problem Identification
Method Development
Experiment Design
Reviews & Feedback
Reviewing Agents
Human-Induced Criteria
Human Judgements

Figure 4.5: (A) The scientific knowledge used for research idea generation consists of a paper, its relationships over an academic graph, and entities within a knowledge store extracted from numerous papers. (B) Given them, the proposed research idea generation process involves problem identification, method development, and experiment design. Those are also iteratively refined by reviews and feedback from reviewing agents, aligned with criteria induced from human judgements.

Scientific research plays a crucial role in driving innovation, advancing knowledge, solving problems, expanding our understanding of the world, and ultimately improving the lives of people in tangible ways. This process usually consists of two key components: the formulation of new research ideas and the validation of these ideas through well-crafted experiments, which are typically conducted by human researchers [329, 330, 331]. However, this is a slow, effort-intensive process, which requires reading and synthesizing overwhelming amounts of knowledge over the vast corpus of rapidly growing scientific literature to formulate research ideas, as well as design and perform experimental validations of those ideas. For example, the number of academic papers published per year is more than 7 million [332]. Similarly, the process of testing a new pharmaceutical drug requires deep expertise, and is massively expensive and labor-intensive, often taking several years [333].

In the meantime, Large Language Models (LLMs) [268, 123, 10] have shown impressive capabilities in processing and generating text, even outperforming human experts across diverse specialized domains including math, physics, history, law, medicine, and ethics. They are able to process and analyze large volumes of data at speeds and scales far exceeding human capabilities, have internalized large swaths of human knowledge from being trained on virtually the entire web, and can identify patterns, trends, and correlations that may not be immediately apparent to human researchers (such as the usage of quantum mechanics in medical imaging or applying psychological insights in AI). This renders them ideally poised to become foundational tools to accelerate the two phases of the scientific research process: ideation of novel research opportunities, and scientific validation of those research hypotheses.

Recent papers in the domain of LLM-augmented scientific discovery have focused on the second phase. Specifically, they attempt [331, 334, 335] to mainly accelerate the experimental validation process, by writing code for machine-learning models, facilitating the exploration of chemical spaces, or advancing the simulation of molecular dynamics. In contrast, we leverage LLMs in the first phase of scientific research – specifically idea generation, whose key focus is conceptualizing novel research questions, methodologies, and experiments. To our knowledge, our work is the first to leverage and evaluate the capabilities of LLMs to act as mediators in scientific idea generation in an open-ended setting.

Given our goal to build an LLM-powered ResearchAgent, we draw inspiration from how human researchers position themselves to come up with novel research ideas. We draw distinctions between three key components of their workflow: a broad and deep understanding of related scientific literature, an encyclopedic view of concepts and how they relate to one another both within and across domains, and a community of colleagues on which to rely for feedback and constructive criticism.

We model each of these three aspects in our ResearchAgent. Specifically, in order to imbibe related work, the system begins with a core scientific paper and then explores a range of related papers through references and citation relationships. Further, to develop an encyclopedic view of related concepts, we build and then augment ResearchAgent with an entity-centric knowledge store derived from co-occurrences of key concepts in the scientific literature. This repository is aimed at capturing novel underlying relationships within and across domains, thereby increasing the chances of a cross-pollination of ideas [336]. Finally, to simulate robust feedback mechanisms, we instantiate a number of LLM-powered ReviewingAgents that help the ResearchAgent to iterate on research idea generation with constructive critiques. Crucially, these ReviewingAgents are prompted with evaluation criteria that are induced from real researchers' judgements, thus aligning them with actual scientific preferential standards. An illustration of our system is provided in Figure 4.5.

We validate the effectiveness of ResearchAgent for research idea generation based on scientific literature across multiple disciplines. Then, on a battery of tests conducted with both human- and model-based evaluations, we demonstrate that ResearchAgent outperforms strong LLM-powered baselines by large margins, generating more clear, relevant, and significant ideas that are especially novel. Furthermore, analyses show the efficacy of our comprehensive approach to modeling ResearchAgent: the entity-centric knowledge store and the iterative idea refinement steps help the system generate meaningfully better ideas compared with an instantiation that is purely based on prior related work.

These findings highlight the immense potential of AI-mediated research assistants like ResearchAgent to enhance the ideation process in scientific research. It can support researchers by identifying knowledge gaps, proposing problem statements, and suggesting potential methodologies in the research process. Also, it can assist in designing experiments and streamline the writing and refinement of papers by generating drafts and offering feedback on how to effectively frame contributions and cite relevant work.

4.2.2 Related Work

Large Language Models LLMs have shown impressive performances across various tasks [123, 10], including scientific fields such as mathematics, physics, medicine, and computer science [337, 338, 335, 331, 339]. For instance, GPT-4 can understand DNA sequences, design biomolecules, predict molecular behavior, and solve PDE problems [334]. However, LLMs have mainly been used for accelerating the experimental validation of already identified research ideas, but not for identifying new problems.

Hypothesis Generation The principle of hypothesis generation is based on literature-based discovery [340], which aims to discover relationships between concepts [341]. For instance, these concepts could be a specific disease and a compound not yet considered as a treatment for it. Early works on automatic hypothesis generation first build a corpus of discrete concepts, and then identify their relationships with machine learning approaches, e.g., using similarities between word (concept) vectors [342] or applying link prediction methods over a graph (where concepts are nodes) [343, 344]. Recent approaches are further powered by LLMs [3, 345, 4], leveraging their prior knowledge about scientific disciplines. Yet, all these approaches perform idea generation in a localized manner and are designed to identify potential relationships between two variables or generate sentence-level connections, which may be sub-optimal to capture the complexity and multifaceted nature of real-world problems (e.g., urban planning involves numerous interacting variables). Meanwhile, we do not artificially restrict the target research idea to be a predictive single concept or simple binary link, instead allowing the model to generate ideas in a more open-ended fashion. We note that there has been a recent surge of interest in idea generation: from Li et al. [346] that focus on evaluating whether LLMs can generate research ideas that are better than human ideas, to Lu et al. [347] that aim to automatically generate full research papers (including idea development, code writing, and experiment execution), to Li et al. [346] that enhance the idea generation process by organizing a sequential chain of literature, all of which build upon insights from our work.

Knowledge-Augmented LLMs The approach to augment LLMs with external knowledge makes them more accurate and relevant to target contexts. Much prior work aims at improving the factuality of LLM responses to queries by retrieving the relevant documents and injecting them into the LLM input [348, 349, 350]. In addition, given that entities or facts are atomic units for representing knowledge, recent studies augment LLMs with them [15, 351]. In contrast to these efforts, which use knowledge units piecemeal, we instead jointly leverage accumulated knowledge over massive troves of scientific papers. Also, Baek et al. [23] proposes to use entities for query suggestion, which yet has a different objective of narrowing the focus of LLMs to entities already present in their context. Instead, our approach retrieves and integrates entities outside the given context, enabling LLMs to explore other concepts.

Iterative Refinements with LLMs Similar to humans, LLMs do not always generate optimal outputs on their first attempt. To tackle this, drawing inspiration from humans who can iteratively refine their thoughts based on critiques from themselves and their peers, many recent studies have investigated the potential of LLMs to correct and refine their outputs, demonstrating that they indeed possess those capabilities [352, 353, 354, 355, 3, 345, 4]. Based on their findings, we extend this paradigm (and further test their capability) to our novel scenario of research idea generation.

4.2.3 Approach

We present ResearchAgent, a system that automatically proposes research ideas with LLMs.

4.2.3.1 LLM-Powered Research Idea Generation

We begin by formally introducing the new problem of research idea generation, followed by an explanation of how LLMs are utilized to tackle it.

Research Idea Generation The goal of the research idea generation task is to formulate new and valid research ideas, to enhance the overall efficiency of the first phase of scientific discovery. While we acknowledge that the real process by which humans conduct research is varied and complex to an extent well beyond the scope of this scientific study, we attempt to model simulacra in three systematic steps that would likely be maximally beneficial to a researcher seeking assistance from an AI system. These are namely, identifying novel research ideas, proposing methods to validate these ideas, and designing experiments to measure the success of these methods in relation to the ideas.

To accomplish the aforementioned steps, we utilize the existing literature (such as academic publications) as a primary source, which provides insights about existing knowledge along with gaps and unanswered questions[5]. Formally, let $\boldsymbol{\mathcal{L}}$ be the literature, and $\boldsymbol{o}$ be the ideas that consist of the problem $\boldsymbol{p}$, method $\boldsymbol{m}$, and experiment design $\boldsymbol{d}$, as follows: $\boldsymbol{o} = [\boldsymbol{p}, \boldsymbol{m}, \boldsymbol{d}]$ where each item consists of a sequence of tokens. Then, the idea generation model $\boldsymbol{f}$ can be represented as follows: $\boldsymbol{o} = \boldsymbol{f}(\boldsymbol{\mathcal{L}})$, which is further decomposed into three submodular steps: $\boldsymbol{p} = \boldsymbol{f}(\boldsymbol{\mathcal{L}})$ for identifying problems, $\boldsymbol{m} = \boldsymbol{f}(\boldsymbol{p}, \boldsymbol{\mathcal{L}})$ for developing methods, and $\boldsymbol{d} = \boldsymbol{f}(\boldsymbol{p}, \boldsymbol{m}, \boldsymbol{\mathcal{L}})$ for designing experiments. We operationalize $\boldsymbol{f}$ with LLMs, leveraging their capability to understand and generate academic text.

Large Language Models Before describing the LLM in the context of our problem setup, let us first provide its general definition, which takes an input sequence of tokens $\boldsymbol{x}$ and generates an output sequence of tokens $\boldsymbol{y}$, as follows: $\boldsymbol{y} = \texttt{LLM}_{\boldsymbol{\theta}}(\boldsymbol{\mathcal{T}}(\boldsymbol{x}))$. Here, the model parameters $\boldsymbol{\theta}$ are typically fixed after training, due to the high costs of further fine-tuning. In addition, the prompt template $\boldsymbol{\mathcal{T}}$ serves as a structured format that outlines the context (including the task descriptions and instructions) to direct the model in generating the desired outputs.

4.2.3.2 Knowledge-Augmented LLMs for Research Idea Generation

We now turn to our primary focus of automatically generating research ideas with LLMs. Recall that we aim to produce a complete idea consisting of the problem, method, and experiment design ($\boldsymbol{o} = [\boldsymbol{p}, \boldsymbol{m}, \boldsymbol{d}]$), while using the existing literature $\boldsymbol{\mathcal{L}}$ as a primary source of information. We operationalize this with LLMs by instantiating the aforementioned research idea generation function $\boldsymbol{f}$ with `LLM` coupled with the task-specific template. Formally, $\boldsymbol{p} = \texttt{LLM}(\boldsymbol{\mathcal{T}_p}(\boldsymbol{\mathcal{L}}))$ indicates the problem identification step, followed by $\boldsymbol{m} = \texttt{LLM}(\boldsymbol{\mathcal{T}_m}(\boldsymbol{p}, \boldsymbol{\mathcal{L}}))$ for method development and $\boldsymbol{d} = \texttt{LLM}(\boldsymbol{\mathcal{T}_e}(\boldsymbol{p}, \boldsymbol{m}, \boldsymbol{\mathcal{L}}))$ for experiment design, which constitutes the full idea: $\boldsymbol{o} = [\boldsymbol{p}, \boldsymbol{m}, \boldsymbol{d}]$.

Following this general formulation, the important question to answer is how the body of scientific literature is leveraged for actually generating research ideas with LLMs. Here, we outline three key desiderata that contribute to the success of human researchers ideating novel research ideas: a broad and deep understanding of related work, an encyclopedic perspective on the interconnectedness of concepts within and across scientific domains, and a community of peers who help iteratively improve ideas through constructive critiques. We describe our operationalization of these three desiderata using the prior literature and LLMs in what follows.

Citation Graph-based Literature Survey Due to the constraints on their input lengths and their reasoning abilities, particularly over very long contexts [357], it is not possible to incorporate all the existing publications from the literature $\boldsymbol{\mathcal{L}}$ into the LLM input. Instead, we need to find a meaningful

[5] We focus on the existing literature-based idea generation by following the paradigm that a *new idea* is more often than not just a new combination of old elements [356].

subset relevant to the problem at hand. To achieve this, we mirror the process followed by human researchers, who expand their knowledge of a paper by perusing other papers that either cite or are cited by it. Concretely, for the `LLM`, we initiate its literature review process by providing a core paper $\boldsymbol{l_0}$ from $\boldsymbol{\mathcal{L}}$ and then selectively incorporating subsequent papers $\{\boldsymbol{l_1, ..., l_n}\}$ that are directly connected based on a citation graph. This procedure makes the `LLM` input for idea generation more manageable and coherent. In addition, we operationalize the selection process of the core paper and its relevant citations with two design choices: 1) the core paper is selected based on its citation count (e.g., exceeding 100 over 3 months) typically indicating high impact; 2) its relevant papers (which may be potentially numerous) are further narrow-downed based on their similarities of abstracts with the core paper, ensuring a more focused and relevant set of related work.

However, despite the simplicity and intuitiveness of this idea generation approach, there exists one major limitation. This approach relies exclusively on a set of given papers (the core paper and its citations); however, since scientific knowledge is not confined to specific studies but rather accumulates across a wide range of publications (across various fields), we should ideally harness this extensive, interconnected, and relevant scientific knowledge in our method for research idea generation.

Entity-Centric Knowledge Augmentation In order to model an encyclopedic view of interconnected concepts, we must effectively design a framework to extract, store and effectively leverage the vast amount of knowledge in scientific literature $\boldsymbol{\mathcal{L}}$. In this work, we view entities as the atomic units of knowledge, which allows for ease of representation and accumulation over papers in a unified manner across different disciplines. For example, we can easily extract the term “database” whenever it appears in any paper, using existing off-the-shelf entity linking methods and then aggregate their linked occurrences into a knowledge store. Then, if the term “database” is prevalent within the realm of medical science but less so in hematology (which is a subdomain of medical science), the constructed knowledge store can capture the affinity between those two domains based on overlapping entities. This representational paradigm can then be used to suggest the term “database” when formulating the ideas about hematology. In other words, this approach enables providing novel and interdisciplinary insights by leveraging the interconnectedness of entities across various fields.

Formally, we design the knowledge store as a two-dimensional matrix $\boldsymbol{\mathcal{K}} \in \boldsymbol{\mathcal{R}^{m \times m}}$ where $\boldsymbol{m}$ is the total number of unique entities identified and $\boldsymbol{\mathcal{K}}$ is implemented in a sparse format. This knowledge store is constructed by extracting entities over all the available scientific articles in literature $\boldsymbol{\mathcal{L}}$[6], which not only counts the co-occurrences between entity pairs within individual papers but also quantifies the count for each entity. Our approach is versatile, thus, we can use any entity linker; in this paper we use one developed by Wu et al. [255]. This off-the-shelf system proves capable of extracting key scientific entities despite its lack of customized training for the scientific domain. Specifically, this linker tags and canonicalizes entities in a paper $\boldsymbol{l}$ from $\boldsymbol{\mathcal{L}}$, formalized as follows: $\boldsymbol{\mathcal{E}_l} = \texttt{EL}(\boldsymbol{l})$ where $\boldsymbol{\mathcal{E}_l}$ denotes a multiset of entities (allowing for repetitions) appearing in $\boldsymbol{l}$[7]. Upon extracting entities $\boldsymbol{\mathcal{E}}$, to store them into the knowledge store $\boldsymbol{\mathcal{K}}$, we consider all possible pairs of $\boldsymbol{\mathcal{E}}$ as follows: $\{\boldsymbol{e_i, e_j}\}_{\boldsymbol{(i,j) \in \mathcal{C}(|\mathcal{E}|, 2)}}$ where $\boldsymbol{e} \in \boldsymbol{\mathcal{E}}$.

Given this knowledge store $\boldsymbol{\mathcal{K}}$, our next goal is to enhance the previous vanilla research idea generation process implemented based on a group of interconnected papers, denoted as follows: $\boldsymbol{o} = \texttt{LLM}(\boldsymbol{\mathcal{T}}(\{\boldsymbol{l_0, l_1, ..., l_n}\}))$. We do this by augmenting the `LLM` with the relevant entities from $\boldsymbol{\mathcal{K}}$, which expand the context that LLMs consume with additional knowledge. Formally, let us define entities

[6] As extracting entities on all articles is computationally infeasible, we target papers appearing after May 01, 2023.

[7] Due to the extensive length of scientific publications, the target of entity extraction is restricted to titles and abstracts.

extracted from the group of interconnected papers, as follows: $\mathcal{E}_{\{l_0,\ldots,l_n\}} = \bigcup_{i=0}^{n} \texttt{EL}(l_i)$. Then, the probabilistic form of retrieving the top-k relevant external entities can be represented as follows:

$$\texttt{Ret}(\{l_0, ..., l_n\}\,; \mathcal{K}) = \underset{I\subset[m]:|I|=k}{\arg\max} \prod P(e_i|\mathcal{E}_{\{l_0,\ldots,l_n\}}), \tag{4.1}$$

where $[m] = \{1, ..., m\}$ and $e_i \notin \mathcal{E}_{\{l_0,\ldots,l_n\}}$. Also, for simplicity, by applying Bayes' rule and assuming that entities are independent, the retrieval operation (Equation 4.1) can be approximated as follows:

$$\underset{I\subset[m]:|I|=k}{\arg\max} \prod (\prod_{e_j\in\mathcal{E}_{\{l_0,\ldots,l_n\}}} P(e_j|e_i)) \times P(e_i), \tag{4.2}$$

where $P(e_j|e_i)$ and $P(e_i)$ can be derived from values in the two-dimensional matrix $\mathcal{K}$, suitably normalized. We note that the formulation in Equation 4.2 is only one instance of operationalizing retrieval; this could be replaced with other retrieval strategies – for example, embedding-based retrieval. Hereafter, the instantiation of research proposal generation augmented with relevant entity-centric knowledge is formalized as follows: $o = \texttt{LLM}(\mathcal{T}(\{l_0, ..., l_n\}\,, \texttt{Ret}(\{l_0, ..., l_n\}\,; \mathcal{K})))$[8]. We call this knowledge-augmented LLM-powered research idea generation approach *ResearchAgent*.

Iterative Research Idea Refinements We note that attempting to write a full research idea in one go may not be an effective strategy. Humans write drafts that are continually improved based on multiple rounds of reviews and feedback. Therefore, we lastly model a community of peers for iterative idea improvement by introducing a set of LLM-powered reviewing agents (called *ReviewingAgents*), which provide the ResearchAgent with reviews and feedback according to various criteria for improvement.

Specifically, similar to our approach to instantiate ResearchAgent with an LLM (`LLM`) and template ($\mathcal{T}$), ReviewingAgents are instantiated similarly but with different templates. Then, with ReviewingAgents, each of the generated research ideas (problem, method, and experiment design) is separately evaluated according to its own specific five criteria[9], which are provided in labels of Figure 4.7. Based on the reviews and feedback from ReviewingAgents, the ResearchAgent iteratively updates and refines its generation of research ideas.

Despite the proficiency of LLMs in the evaluation of machine-generated texts [358, 359], their judgments on research ideas may not be aligned with the judgments of real human researchers. However, there are no ground truth reference judgments available, and collecting them to align LLMs is expensive and often infeasible. Ideally, the judgments made by LLMs should be similar to the ones made by humans, and we aim to ensure this by automatically generating human preference-aligned evaluation criteria (used for automatic evaluations) with a few human annotations. Specifically, to obtain these human-aligned evaluation criteria, we first collect 10 pairs of research ideas and their associated scores for every evaluation criterion on a 5-point Likert scale, annotated by human researchers having at least 3 papers. After that, we prompt the LLM with these human-annotated pairs to induce detailed descriptions for evaluation criteria [360]. These criteria reflect the underlying human preferences[10] and are used as evaluation criteria by the ReviewingAgents.

[8] There may be additional knowledge sources (beyond the existing literature and entities) for research idea generation, and we leave exploring them as future work.

[9] We select the top five criteria which we consider as the most important, and leave exploring others as future work.

[10] We additionally ask five human annotators, who evaluate research ideas, to judge the quality of the induced criteria; two of them agree strongly, while the other three agree moderately.

4.2.4 Experiments

4.2.4.1 Data

The main source to generate research ideas is the scientific literature $\mathcal{L}$, which we obtain from the Semantic Scholar Academic Graph API[11]. From this, we select papers appearing after May 01, 2023, because LLMs that we use in our experiments are trained on data from the open web available before this point. This follows the procedure of existing literature-based hypothesis generation work [345]. Then, we select high-impact papers (that have more than 20 citations) as core papers, mirroring human researchers' tendency to leverage influential work, to ensure the high quality of generated ideas. The resulting data is still very large; thus, we further sample a subset of 300 papers as core papers to obtain a reasonably sized benchmark dataset. The average number of reference papers for each core paper is 87; the abstract of each paper has 2.17 entities on average. The distribution of disciplines for all papers is in Figure 4.6.

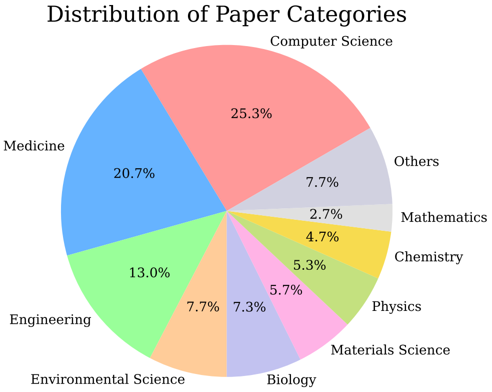


Figure 4.6: Visualization of the distribution of disciplines for all core papers, selected for research idea generation.

4.2.4.2 Baselines and Our Model

As we target the novel task of research idea generation involving the generation of problems, methods, and experimental designs, there are no baselines for direct comparison. Thus, we mainly compare our full ResearchAgent model against its ablated variants, outlined as follows:

1. **Naive ResearchAgent** – which uses only a core paper to generate research ideas.
2. **ResearchAgent w/o Entity Retrieval** – which uses the core paper and its relevant references without considering entities.
3. **ResearchAgent** – which is our full model that uses the relevant references and entities along with the core paper, to augment LLMs.

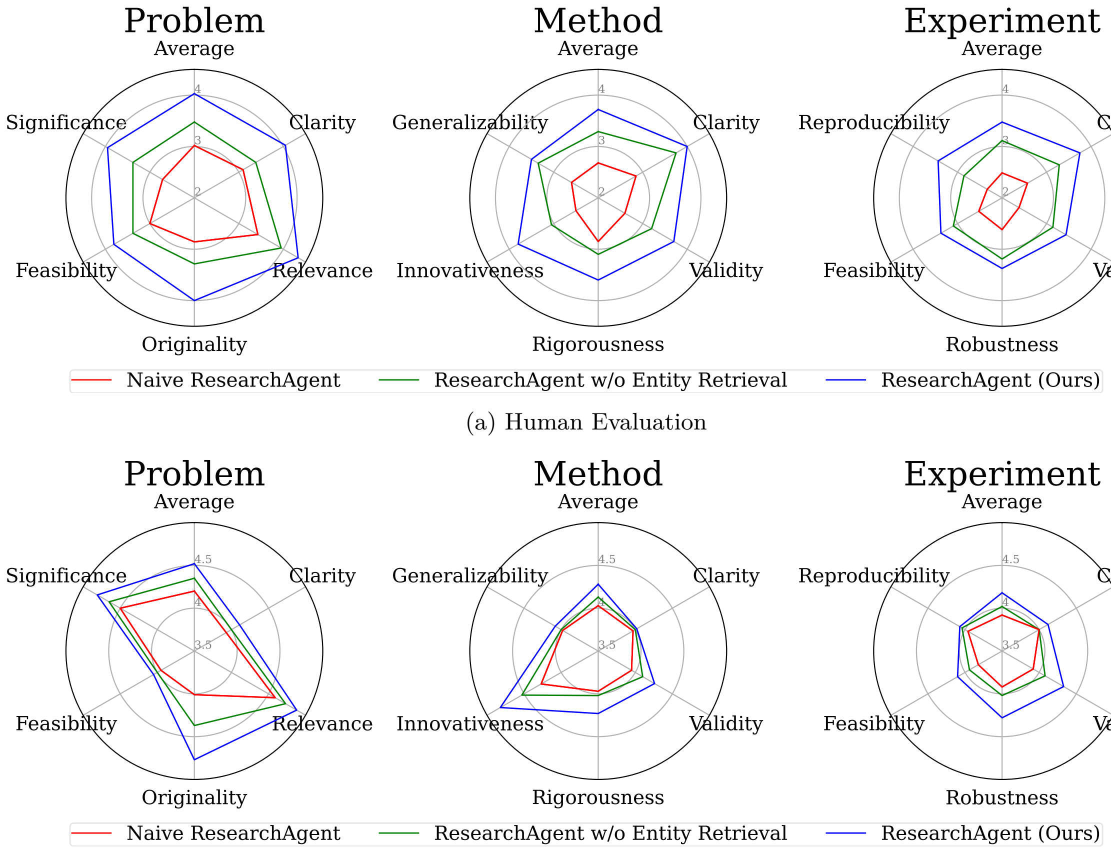
In addition to this set of core baselines, we also compare our approach against existing hypothesis generation work from prior literature in Table 4.9.

4.2.4.3 Evaluation Setup

Given our formulation of idea generation (Sec 4.2.3.1), there are no ground-truth answers to measure the quality of the generated ideas. Yet, exhaustively listing pairs of core papers and reference research ideas is suboptimal, since there may exist a large number of valid research ideas for each core paper, and this process requires much time, effort and expertise on the part of human researchers. Thus, we use a combination of model-based automatic evaluation and manual human evaluation to validate different models on our experimental benchmark.

Model-based Evaluation Following the recent trends in using LLMs to judge the quality of output texts (especially in the setting of reference-free evaluations) [358, 359, 176], we use GPT-4 to judge the

[11] https://www.semanticscholar.org/product/api



(a) Human Evaluation

(b) Model-based Evaluation

Figure 4.7: Main results on our research idea generation task with human- (top) and model-based (bottom) evaluations, where we report the score of each idea (problem, method, or experiment design) based on its own five criteria and their average score.

quality of research ideas. Note that each of the problem, method, and experiment design is evaluated with five different criteria (See labels of Figure 4.7 for criteria). We ask the LLM-based evaluation model to either rate the generated idea on a 5-point Likert scale for each criterion or perform pairwise comparisons between two ideas from different models.

Human Evaluation Similar to model-based evaluations, we perform human evaluations that involve assigning a score for each criterion and conducting pairwise comparisons between two ideas. As the generated ideas are knowledge-intensive, we carefully select annotators who are well-versed in the field and provide them with ideas that are highly relevant to their field of expertise[12]. Specifically, we choose ten expert researchers who have authored at least three papers and ask them to judge only the ideas that are generated based on their own papers.

4.2.4.4 Implementation Details

We mainly use the GPT-4 [123] release from Nov 06, 2023, as the basis for all models, which is, notably, reported to be trained with data up to Apr 2023 (meanwhile, the papers used for idea generation appear after May 2023). To extract entities and build the entity-centric knowledge store, we use the

[12] We also experiment with human evaluation using non-domain-experts, but this proves to be suboptimal; therefore, we focus on experts for reliable judgments of generated ideas.

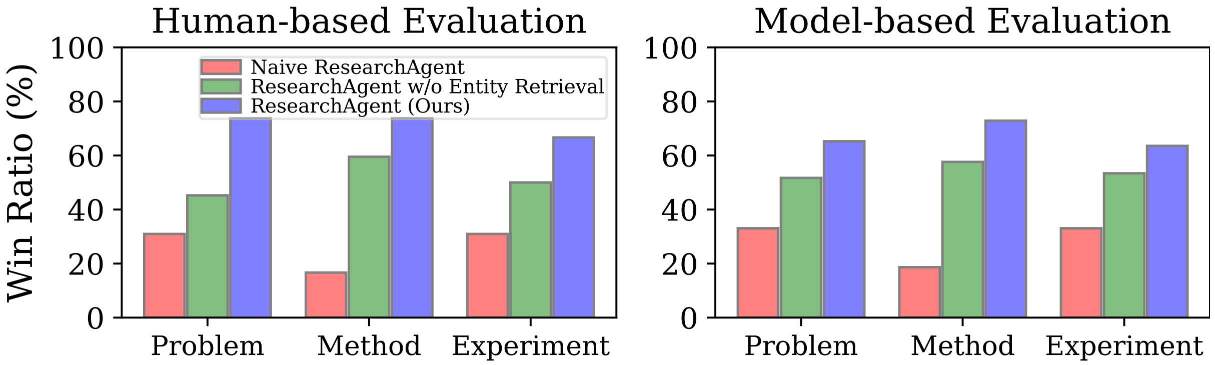


Figure 4.8: Results of pairwise comparisons between ideas from two of any different approaches.

off-the-shelf BLINK entity linker [255], with papers from May 01, 2023, to Dec 31, 2023 (available from Semantic Scholar API) along with their references, which number 50,091 in total.

4.2.4.5 Experimental Results and Analyses

Main Results Our main results on scoring with human and model-based evaluations are provided in Figure 4.7. These demonstrate that our full ResearchAgent outperforms all baselines by large margins on every metric across problems, methods, and experiment designs (constituting the complete research ideas). Particularly, the full ResearchAgent augmented with relevant entities exhibits strong gains on metrics related to creativity (such as Originality for problems and Innovativeness for methods) since entities may offer novel concepts and views that may not be observable in the group of citation-based papers alone. In addition, the results of pairwise comparisons between models with both human and model-based evaluations – shown in Figure 4.8 – demonstrate that the full ResearchAgent shows the highest win ratio over its baselines.

Analysis on Inter-Annotator Agreements To see the quality and reliability of human annotations, we measure the inter-annotator agreements, where 20% of the generated ideas are evaluated by two human judges, and report results in Table 4.7. Specifically, for the scoring, we first rank scores from each annotator and measure Spearman's correlation coefficient [328] between the ranked scores of two annotators. For the pairwise comparison between two judges, we measure Cohen's kappa coefficient [327]. Table 4.7 shows that the inter-annotator agreement is high, confirming the reliability of our assessments about the quality of generated research ideas. Also, while agreement scores for experimental designs are slightly lower than other aspects, this does not necessarily indicate a shortcoming in the quality of experimental designs produced by ResearchAgent, as shown in Figures 4.7 and 4.8. Instead, we view this as the inherent subjectivity and variability in how such designs are perceived and evaluated by different annotators (i.e., the nature of the variability itself makes achieving high agreement challenging).

Table 4.7: Results of agreements between two human annotation results and between human and model evaluation results.

Categories	**Metrics**	**Problem**	**Method**	**Experiment**
Human and Human	Scoring	0.83	0.76	0.67
	Pairwise	0.62	0.62	0.41
Human and Model	Scoring	0.64	0.58	0.49
	Pairwise	0.71	0.62	0.52

Analysis on Human-Model Agreements Similar to what we did for the aforementioned inter-annotator agreements, we measure agreements between human-based and model-based evaluations, to

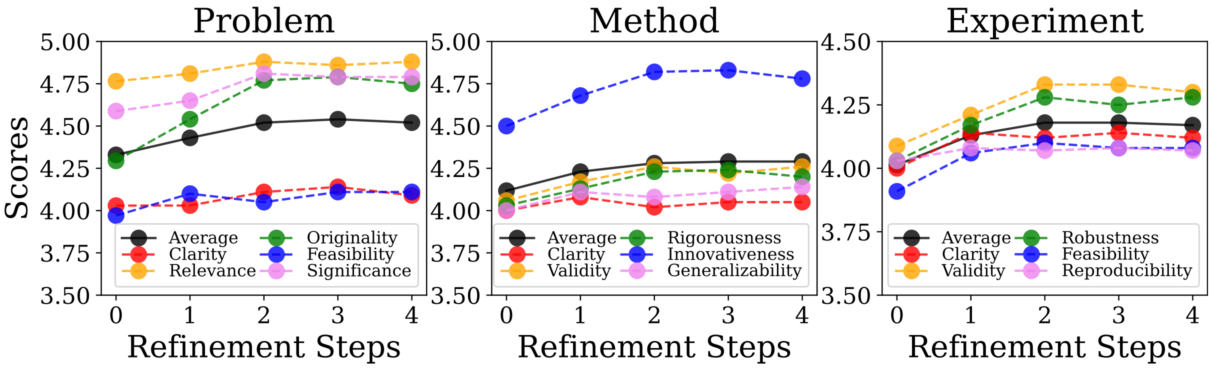


Figure 4.9: Results with varying the number of refinement steps for iterative idea generation.

Table 4.8: Results of ablation study on references and entities.

Methods	Problem	Method	Experiment
ResearchAgent	**4.52**	**4.28**	**4.18**
- w/o Entities	4.35	4.13	4.02
- w/ Random Entities	4.41	4.19	4.13
- w/o References	4.26	4.08	3.97
- w/ Random References	4.35	4.16	4.02
- w/o Entities & References	4.20	4.03	3.92

ensure the reliability of model-based evaluations. As shown in Table 4.7, we further confirm that agreements between humans and models are high, indicating that model-based evaluations are a reasonable proxy to judge research idea generation.

Analysis of Refinement Steps To see the effectiveness of iterative refinements of research ideas with ReviewingAgents, in Figure 4.9, we report the averaged scores on the generated ideas as a function of refinement steps. We first observe initial improvements in the quality of ideas with increased refinement steps. Yet, the performance becomes saturated after three iterations, which may indicate diminishing returns for subsequent iterations, similar to the pattern observed in agent-based refinement work [361].

Ablation on Knowledge Sources Recall that the full ResearchAgent is augmented with two different knowledge sources, namely relevant references and entities. To see their individual contribution, we perform an ablation study by either excluding one of the knowledge sources or replacing it with random elements. As shown in Table 4.8, each knowledge source contributes to performance improvement, and the relevant references are especially helpful. We also note that providing random elements is more helpful than providing no elements at all; we hypothesize that this may be due to the LLM's capability to filter out noise while still gaining incidental value from random inputs.

Analysis on Human Alignment for Evaluation Recall that to align judgments from model-based evaluations with actual human preferences, we generated the evaluation criteria based on human evaluation results and used them as the criteria for model-based evaluations. Figure 4.10 demonstrates the efficacy of this strategy, presenting the score distribution of human evaluation compared with the distributions of model-based evaluations with and without human alignment. We find that the score

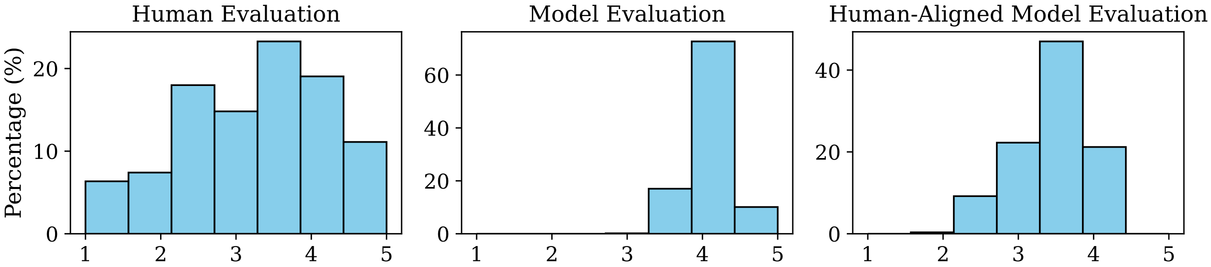


Figure 4.10: Distributions of model-based evaluation results with and without the human-induced score criteria alignment (middle and right), as well as human evaluation results (left).

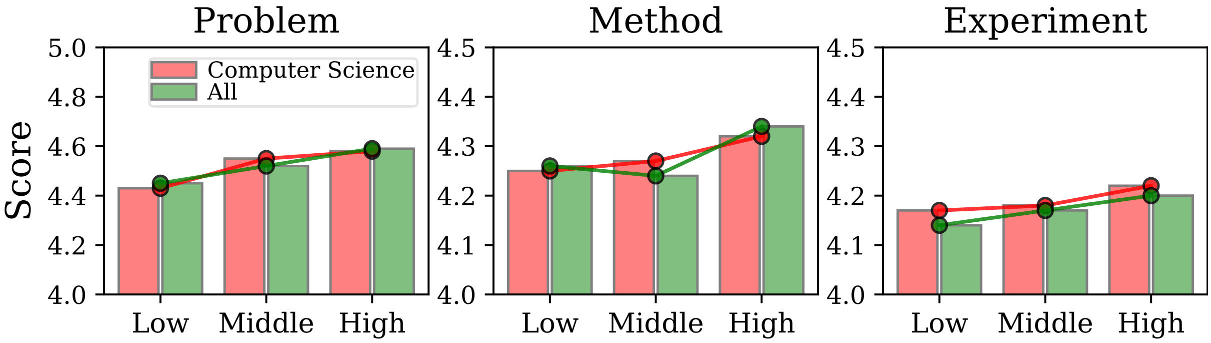


Figure 4.11: Results with bucketing papers based on citations.

Table 4.9: Comparisons of ResearchAgent with hypothesis generation methods [3, 4].

Methods	Clarity	Relevance	Originality	Feasibility	Significance
SciMON	4.04	4.37	4.56	3.98	4.15
Hypothesis Proposer	3.97	4.14	4.07	4.01	4.11
ResearchAgent	**4.11**	**4.88**	**4.77**	**4.05**	**4.81**

distribution of model-based evaluations without alignment is skewed and different from the score distribution of human judgments. Meanwhile, after aligning the model-based evaluations with human-induced score criteria, the calibrated distribution more closely resembles the distribution of humans.

Correlation on Citation Counts We further investigate whether a high-impact paper (when used as a core paper) leads to high-quality research ideas. To measure this, we categorize papers by their citation count (as a proxy for impact), and visualize the average score of each bucket (with model-based evaluations) in Figure 4.11. We find that ideas from high-impact papers tend to be of higher quality, likely due to their ability to identify research gaps, propose feasible methods, and connect with other works. Additionally, based on the paper distribution (See Figure 4.6) and for the ease of manual quality check, evaluation criteria for model-based evaluations are induced mainly with computer science papers. To see whether those criteria are applicable to diverse fields, we also compare a correlation between scores of computer science papers and all papers in Figure 4.11. From this, we observe that the scores increase when the citation increases for both domains, which may support the generalizability of human-preference-induced evaluation criteria.

Comparisons to Hypothesis Generation Recall that existing methods for hypothesis generation focus on predicting links between variables or generating hypotheses based on these links, which dif-

Table 4.10: Results with different, open and proprietary LLMs.

LLMs	Models	Problem	Method	Experiment
GPT-4.0	Naive ResearchAgent	4.20	4.03	3.92
	ResearchAgent (Ours)	4.52	4.28	4.18
GPT-3.5	Naive ResearchAgent	3.56	3.56	3.63
	ResearchAgent (Ours)	3.58	3.58	3.60
Llama3 (8B)	Naive ResearchAgent	3.76	3.69	3.54
	ResearchAgent (Ours)	4.18	4.03	3.95
Mixtral (8x7B)	Naive ResearchAgent	3.31	3.27	3.20
	ResearchAgent (Ours)	3.28	3.35	3.31
Qwen1.5 (32B)	Naive ResearchAgent	3.64	3.74	3.66
	ResearchAgent (Ours)	4.02	3.97	3.94

fers from our experimental setup of generating open-ended research ideas (problems, methods, and experiments). Nevertheless, to understand how the quality of the generated research ideas from prior works [3, 4] differs from our ResearchAgent, we perform comparisons. As shown in Table 4.9, we observe that ResearchAgent is capable of generating superior research hypotheses, due to the utilization of broad and deep knowledge across domains as well as the iterative review and refinement procedures.

Analysis using Different LLMs To assess how ResearchAgent's performance changes with different LLMs [362, 363], we conduct an auxiliary analysis, as shown in Table 4.10. These results show a significant performance drop with less capable models. Moreover, the performance differences between the Naive ResearchAgent without knowledge augmentation and the full ResearchAgent become marginal (for Mixtral and GPT-3.5), which indicates that they might struggle with capturing complex concepts between scientific papers. This can likely be attributed to the emergent abilities of LLMs for complex reasoning (but not in smaller LMs) [364], although other subtle issues may also be contributing factors.

4.2.5 Summary

In this work, we introduced ResearchAgent, a system designed to assist researchers by generating research ideas, which encompass problem identification, method development, and experiment design. Inspired by the human process of ideation, our approach conducts broad and deep literature reviews, integrates knowledge across domains to foster idea cross-pollination, and employs a community of reviewing agents to iteratively refine the generated ideas. Our evaluations, both human and model-based, demonstrated that ResearchAgent produces ideas that are more creative, valid, and clear compared to baselines. While this initial foray shows promising results, multiple challenges remain to operationalize ResearchAgent in real-world research settings. Practical considerations include scaling the knowledge store to encompass diverse research domains, and keeping it current with the latest publications, through which the system can become adaptable even to emerging fields.

4.2.6 Extension: Paper2Code for Research Idea Execution

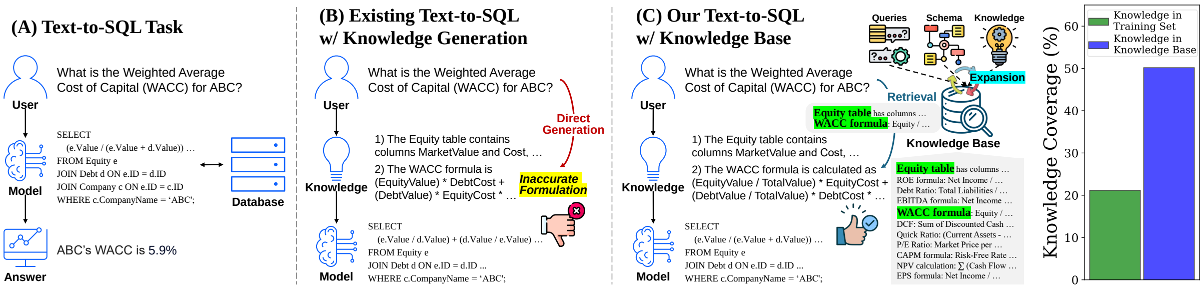
While ResearchAgent focuses on the ideation stage of scientific research (formulating problems, proposing methods, defining experiments, and iteratively refining ideas with feedback from reviewing agents), it does not support another research cycle: validation, where ideas must be instantiated, implemented, and empirically validated. To bridge this gap, we propose PaperCoder [365], a multi-agent system designed to translate scientific ideas (or papers) into executable, modular code repositories, which can enable researchers to rapidly materialize the research ideas into executable prototypes, especially in machine learning domains, along with ResearchAgent to conceptualize new ideas. We believe that the two frameworks of ResearchAgent and PaperCoder illustrate a broader vision of AI-mediated research, in which LLM-driven agents support both the generation and the operationalization of scientific ideas.

4.2.7 Extension: Chain of Retrieval for AI-Mediated Research

Although ResearchAgent supports the ideation phase by synthesizing insights from multiple papers and concepts, and PaperCoder supports the validation phase by operationalizing ideas into executable prototypes, high-quality ideation and validation often depend on how well the system can retrieve and contextualize relevant scientific literature. However, traditional retrieval methods for scientific papers rely on abstract-level similarity and overlook the rich, multi-aspect structure of full papers. To address this limitation and to ultimately build a comprehensive AI-mediated research system capable of grounding and validating scientific ideas with the full context of relevant papers, we introduce Chain of Retrieval (CoR) [366], a multi-aspect, iterative retrieval method. Specifically, it decomposes a query paper into multiple aspect-specific views, retrieves candidate papers across them, and iteratively expands the search by promoting top results as new queries, ultimately constructing a tree-structured retrieval process whose hierarchical evidence is aggregated in a post-order manner. We believe that this capability is crucial not only for ideation (to ground ideas in a comprehensive scientific context) but also for validation (to surface implementation-relevant papers); thus, integrating CoR into AI-mediated research therefore strengthens the entire workflow, ensuring that it can generate, refine, and execute ideas while remaining anchored in the broader landscape of scientific knowledge.

4.3 Knowledge-Augmented Model Contextualization for Text-to-SQL with Knowledge Base Construction

4.3.1 Motivation



What is the Weighted Average Cost of Capital (WACC) for ABC?

```
SELECT
   (e.Value / (e.Value + d.Value)) …
FROM Equity e
JOIN Debt d ON e.ID = d.ID
JOIN Company c ON e.ID = c.ID
WHERE c.CompanyName = 'ABC';
```

ABC's WACC is 5.9%

What is the Weighted Average Cost of Capital (WACC) for ABC?

1) The Equity table contains columns MarketValue and Cost, …

2) The WACC formula is (EquityValue) * DebtCost + (DebtValue) * EquityCost * …

```
SELECT
   (e.Value / d.Value) + (d.Value / e.Value) …
FROM Equity e
JOIN Debt d ON e.ID = d.ID ...
WHERE c.CompanyName = 'ABC';
```

What is the Weighted Average Cost of Capital (WACC) for ABC?

Equity table has columns …
WACC formula: Equity / …

1) The Equity table contains columns MarketValue and Cost, …

2) The WACC formula is calculated as (EquityValue / TotalValue) * EquityCost + (DebtValue / TotalValue) * DebtCost * …

```
SELECT
   (e.Value / (e.Value + d.Value)) …
FROM Equity e
JOIN Debt d ON e.ID = d.ID ...
WHERE c.CompanyName = 'ABC';
```

Equity table has columns …
ROE formula: Net Income / …
Debt Ratio: Total Liabilities / …
EBITDA formula: Net Income …
WACC formula: Equity / …
DCF: Sum of Discounted Cash …
Quick Ratio: (Current Assets - …
P/E Ratio: Market Price per …
CAPM formula: Risk-Free Rate …
NPV calculation: ∑ (Cash Flow …
EPS formula: Net Income / …

Figure 4.12: (A) Text-to-SQL aims to translate a user query into a SQL statement executable over a database, to access the desired information. (B) Existing Text-to-SQL with Knowledge Generation approaches generate the knowledge relevant to the user query and formulate the SQL statement with this generated knowledge. (C) Our Text-to-SQL with Knowledge Base Construction approach builds the repository of the knowledge and reuses the knowledge within it across multiple queries and databases. (Right:) The knowledge in the training set of the text-to-SQL benchmark [6] covers 21% of the knowledge required for test-time queries, and our constructed knowledge base further covers 50% of them.

Text-to-SQL aims to transform natural language queries from users into Structured Query Language (SQL) statements, to interact with and retrieve the information from databases [367, 368, 369, 370], as illustrated in Figure 4.12 (A). This task has recently gained much attention since it allows non-experts to access and manipulate database information without needing to understand complex database languages. In the meantime, Large Language Models (LLMs) have shown impressive capabilities in processing and generating text and code, which have been further extended for text-to-SQL [263, 264].

Despite their huge successes, transforming user queries into SQL statements may still be challenging due to the need for specific domain knowledge and an understanding of the underlying database schemas, which poses a significant hurdle even for the most advanced LLMs to achieve high accuracy across diverse datasets [6]. For example, consider a scenario where the user asks for the query: "What is the WACC for Company X?". To translate this into an SQL statement, the text-to-SQL model should understand the concept and calculation of Weighted Average Cost of Capital (WACC), which involves multiple factors including the cost of equity, cost of debt, and the respective proportions of each in the capital structure. In addition, the model needs to comprehend the specific schema of the financial database, where relevant data is distributed across multiple tables such as 'Equity', 'Debt', and 'Capital Structure'.

To tackle the aforementioned limitations due to the lack of the domain-specific knowledge for SQL generation, recent studies have proposed collecting and annotating explicit knowledge, which is then leveraged for SQL generation [371, 6]. However, while these approaches substantially improve the performance of existing text-to-SQL models, they rely on extensive human annotations, which may be suboptimal (and nearly impractical) to conduct for all queries considering a diverse source of domain-specific knowledge from numerous databases. To address this issue, recent work proposes generating a few pieces of knowledge for each query based on the query itself and its relevant database schema [372] (see Figure 4.12 (B)). However, although this method demonstrates promise in automatic knowledge generation, certain knowledge required for one query can be directly reused or provide insights for multiple queries within the same database, as shown in Figure 4.12 (Right). Also, this knowledge can be generalizable to other queries for different databases.

Motivated by these observations, this work proposes an automatic approach to build a knowledge base, designed to serve as a comprehensive repository of domain-specific knowledge for text-to-SQL and capable of providing knowledge for multiple queries with the same database and even across the different databases. To construct this knowledge base, we generate knowledge entries based on available samples and their associated database schemas through LLM prompting, and then compile all of them together. During this prompting process, we provide LLMs with relevant examples to contextualize and guide the generation of useful knowledge in the right format that is further grounded in the database schema. Then, once constructed, the knowledge base allows for the retrieval of relevant knowledge for the given test-time query, which is then used alongside the query to formulate the SQL statement. Note that while ideally the knowledge base would cover all possible queries, it may not always do so. Nevertheless, the existing knowledge in it could still offer valuable insights for generating the required knowledge for new queries. Thus, by leveraging similar knowledge from the knowledge base, we further prompt LLMs to produce the most suitable knowledge for the query at inference time. We call our method Knowledge-Augmented Text-to-SQL (KAT-SQL), depicted in Figure 4.12 (C).

We experimentally validate KAT-SQL on two different text-to-SQL scenarios, involving both the overlapping and non-overlapping databases between training and test phases, showing that the proposed knowledge base construction-based text-to-SQL approach surpasses the existing (knowledge-augmented) text-to-SQL baselines. We also assess the generalizability of our knowledge base constructed from one dataset by applying it to different datasets that lack any annotated knowledge, demonstrating that it is versatile and can effectively improve SQL generation for even unseen databases from other datasets.

4.3.2 Related Work

LLM-Powered Text-to-SQL LLMs have shown remarkable performances across a wide range of tasks [123, 10, 373], including text-to-SQL, due to their strong capability in understanding natural language and generating structured code [263, 264]. Specifically, various studies have developed and advanced the prompting techniques for text-to-SQL, for example, using Chain-of-Thought (CoT) [374, 375, 376], investigating sophisticated prompt design strategies [377], and aggregating LLM-generated outputs from multiple prompts [378, 379] akin to self-consistency [380]. In addition, another line of study proposes decomposing the text-to-SQL problem into multiple subtasks, and feeding the solutions of subtasks (from multiple models or agents) into the LLM to derive the final SQL statement [381, 382, 383]. The knowledge internalized in LLMs might however not be sufficient to handle diverse queries, which oftentimes requires grounding in the database schemas or additional domain-specific information for specialized domains, which gives rise to the need for leveraging external knowledge for text-to-SQL.

Knowledge-Augmented Text-to-SQL There are a few recent studies that propose augmenting text-to-SQL models with explicit knowledge. Specifically, Dou et al. [371] collect formulaic knowledge (e.g., Trade Balance = Exports − Imports) available from public resources such as finance reports and store the collected knowledge into a knowledge bank with proper human-involved post-processing. The text-to-SQL model then retrieves relevant knowledge for any given query from the knowledge bank and uses it to convert the query into the SQL statement. In addition, Li et al. [6] release a large-scale benchmark dataset for the text-to-SQL task, where each question is associated with specific knowledge that is manually annotated by humans. Manual annotation is however costly and time consuming, requiring effort and expertise on the part of domain-experts. To address this challenge, more recent work proposes automatically generating the knowledge based on the question and database schema, and utilizing this knowledge for text-to-SQL [372]. In our work, instead of generating only a few pieces of knowledge for each question, we propose to construct a comprehensive knowledge base. This provides a repository of reusable knowledge that can be leveraged across multiple queries, which can be further adapted to various databases over different domains in a scalable way, in contrast to existing work.

Data Generation with LLMs The recent advent of LLMs has revolutionized the field of data generation, as they can produce vast amounts of high-quality samples without costly human annotation. Specifically, several efforts around LLM-based synthetic data generation, such as Self-Instruct [384], Alpaca [385], Evol-Instruct [386], Orca [387], and InstructLab [388], propose generating a large number of samples from LLMs by prompting them. Also, motivated by the capabilities of LLMs in generating synthetic data and memorizing factual knowledge, some other work aims to populate an encyclopedic knowledge base like Wikidata [32] with LLMs [389, 390, 391]. Most of the knowledge in such encyclopedic knowledge bases is however unsuitable for text-to-SQL since it is neither relevant to formulate SQL statements from user queries nor aware of database schemas necessary for the query conversion. Thus, our approach stands apart as the first to automatically construct a text-to-SQL knowledge base.

4.3.3 Approach

In this section, we present Knowledge-Augmented Text-to-SQL (KAT-SQL), an approach that automatically constructs a knowledge base and utilizes the relevant knowledge from it for text-to-SQL.

4.3.3.1 Problem Statement

We begin with formally explaining text-to-SQL and the knowledge augmentation technique for it.

Text-to-SQL Text-to-SQL aims to translate a natural language query from a user into a syntactically correct and semantically precise SQL statement. Formally, let $\boldsymbol{q}$ be the user query (consisting of a sequence of tokens) and $\boldsymbol{\mathcal{D}}$ be the database schema containing multiple tables and columns. Then, the SQL generation model $\boldsymbol{f}$ can be represented as follows: $\boldsymbol{s} = \boldsymbol{f}(\boldsymbol{q}, \boldsymbol{\mathcal{D}})$ where $\boldsymbol{s}$ is the SQL statement (consisting of a sequence of tokens) that attempts to retrieve the information requested by $\boldsymbol{q}$ over $\boldsymbol{\mathcal{D}}$.

Here, we operationalize $\boldsymbol{f}$ with LLMs, to harness their capability in understanding the semantics of $\boldsymbol{q}$ and generating the corresponding SQL code $\boldsymbol{s}$, as follows: $\boldsymbol{s} = \texttt{LLM}_{\boldsymbol{\theta}}(\boldsymbol{\mathcal{T}}(\boldsymbol{q}, \boldsymbol{\mathcal{D}}))$ where $\boldsymbol{\theta}$ is the model parameters and $\boldsymbol{\mathcal{T}}$ is the prompt template. Typically, the model parameters $\boldsymbol{\theta}$ remain fixed due to the high costs associated with further fine-tuning and sometimes their limited accessibility. Also, the prompt template $\boldsymbol{\mathcal{T}}$ serves as a structured format that outlines the context, which includes task descriptions and instructions as well as few-shot examples, to guide the model in generating accurate SQL codes.

Notably, while there have been great successes in advancing the LLM itself and optimizing its usage for text-to-SQL, such as using advanced prompting techniques or breaking down the task into multiple subtasks [374, 375, 376, 381, 382, 383], these improvements alone may not be sufficient to fully handle queries that require the deep domain knowledge or precise understanding of complex database schemas. In other words, the internal parametric knowledge of LLMs, while robust, may not fully encompass the diverse range of query variations and database structures, especially when these databases have distinct schemas or certain specialized terminology.

Knowledge-Augmented Text-to-SQL To tackle the aforementioned limitations, we focus on augmenting text-to-SQL with the knowledge relevant to the query, providing valuable insights into the domain-specific terminology and complex database schemas. If we denote this knowledge as $\boldsymbol{k}$, then the previous text-to-SQL process is redefined to incorporate it, as follows: $\boldsymbol{s} = \mathtt{LLM}_{\boldsymbol{\theta}}(\boldsymbol{\mathcal{T}}(\boldsymbol{q}, \boldsymbol{k}, \boldsymbol{\mathcal{D}}))$.

While there have been few studies that explore this knowledge-augmented text-to-SQL paradigm, there are still a couple of challenges. Specifically, Dou et al. [371] and Li et al. [6] propose collecting and annotating the explicit knowledge required to convert queries into SQL statements. Yet, to operationalize, this annotation-based approach can be costly and time-consuming, especially when dealing with a large number of diverse queries. On the other hand, Hong et al. [372] propose an automatic generation of knowledge, based on the question and its associated database schema. However, this method is still limiting as it generates only a few pieces of knowledge for each query without leveraging the potential for reuse. In contrast, since much of the knowledge used for one query can be applicable to multiple similar queries (See Figure 4.12, Right), we aim to design a more effective approach for knowledge augmentation.

4.3.3.2 Knowledge Base Construction

To address the aforementioned limitations of existing approaches in knowledge augmentation for text-to-SQL, we propose a novel approach to automatically construct a comprehensive and reusable knowledge base. Ideally, this can serve as a foundational resource, encapsulating diverse domain information and offering insights into various database schemas, to enhance the understanding of queries and their associated database structures.

Formally, we design this knowledge base $\boldsymbol{\mathcal{K}}$ as a collection of knowledge entries, each represented as a concise sentence, denoted as follows: $\boldsymbol{k} \in \boldsymbol{\mathcal{K}}$. For instance, in the medical domain, one knowledge entry might be “Abnormal white blood cell count refers to WBC $\leqslant$ 3.5 or WBC $\geqslant$ 9.0”, which describes the abnormal range of white blood cell counts and its corresponding column name “WBC” in the database schema, applicable to queries related to abnormal white blood cells. The next question to answer is then how to construct this knowledge base based on the available resources.

In this work, we start with collecting all the existing knowledge entries from the publicly available dataset [6], which includes the knowledge and its related pair of query and database schema. Yet, while this initial collection can serve as the foundational layer of our knowledge base, it may not capture the full scope of the required information. To address this gap, we propose an automatic knowledge base expansion technique that leverages LLMs, which possess domain-specific knowledge and the ability to comprehend the given context (including instructions, codes, and database structures) by generating additional knowledge entries. Specifically, given the query and its associated database schema from the available datasets, we prompt LLMs (along with a prompting template $\boldsymbol{\mathcal{T}}$ for knowledge generation) to produce the knowledge, formulated as follows: $\boldsymbol{k} = \mathtt{LLM}(\boldsymbol{\mathcal{T}}(\boldsymbol{q}, \boldsymbol{\mathcal{D}}))$, and then store this knowledge $\boldsymbol{k}$ into the knowledge base $\boldsymbol{\mathcal{K}}$. In addition, as it may be more accurate and reliable to provide the LLM with relevant examples (which can help it understand the context, nuances, and expectations of the desired output), we further prepend the small number of relevant examples into the prompt of `LLM`. It is worth noting that these examples are comprised of the triplets of the user queries, their associated database schemas, and the knowledge they are derived from, and that those triplets come from the existing dataset (used to construct the initial knowledge base). Also, we select only those highly relevant to the query based on its embedding-level cosine similarities with samples from the existing dataset, calculated by MPNet [1]. This process can ultimately enable the LLM to generate more precise and contextually appropriate knowledge for text-to-SQL.

Algorithm 1 Knowledge-Augmented Text-to-SQL

```
Require: Dataset $\boldsymbol{D}$ containing query-schema pairs $(\boldsymbol{q}, \boldsymbol{\mathcal{D}})$; LLM model LLM; Prompt templates $\boldsymbol{\mathcal{T}}$
Ensure: SQL statement $\boldsymbol{s}$ for a given query $\boldsymbol{q}$
Phase 1: Knowledge Base Construction
$\boldsymbol{\mathcal{K}} \leftarrow \{\} \cup \boldsymbol{D}$                                    ▻ Initialize knowledge base
for all $(\boldsymbol{q}, \boldsymbol{\mathcal{D}}) \in \boldsymbol{D}$ do
    $\boldsymbol{\mathcal{E}} \leftarrow$ Retrieve top-$\boldsymbol{k}$ relevant examples from $\boldsymbol{D}$
    $\boldsymbol{k}_{\text{new}} \leftarrow \texttt{LLM}(\boldsymbol{\mathcal{T}}_{\text{gen}}(\boldsymbol{q}, \boldsymbol{\mathcal{D}}, \boldsymbol{\mathcal{E}}))$                  ▻ Generate knowledge
    $\boldsymbol{\mathcal{K}} \leftarrow \boldsymbol{\mathcal{K}} \cup \boldsymbol{k}_{\text{new}}$                                  ▻ Store knowledge
end for
Phase 2: Knowledge-Augmented SQL Generation
function KAT-SQL($\boldsymbol{q}$, $\boldsymbol{\mathcal{D}}$, $\boldsymbol{\mathcal{K}}$)
    $\{\boldsymbol{k_i}\}_{\boldsymbol{i=1}}^{\boldsymbol{j}} \leftarrow$ Retrieve top-$\boldsymbol{j}$ knowledge from $\boldsymbol{\mathcal{K}}$
    $\boldsymbol{k}' \leftarrow \texttt{LLM}(\boldsymbol{\mathcal{T}}_{\text{ref}}(\boldsymbol{q}, \{\boldsymbol{k_i}\}_{\boldsymbol{i=1}}^{\boldsymbol{j}}, \boldsymbol{\mathcal{D}}))$             ▻ Refine knowledge
    $\boldsymbol{s} \leftarrow \texttt{LLM}(\boldsymbol{\mathcal{T}}_{\text{text-to-SQL}}(\boldsymbol{q}, \boldsymbol{k}', \boldsymbol{\mathcal{D}}))$               ▻ Generate SQL
    return $\boldsymbol{s}$
end function
```

In addition to this relevant example-based knowledge generation approach, to further enrich the diversity and comprehensiveness of the knowledge base, we implement a simple yet effective strategy that involves sampling and permutation of few-shot examples provided to the LLM. Specifically, for the given query and its associated database schema, instead of generating their corresponding knowledge only once, we iteratively sample a different set of relevant examples (provided to contextualize the LLM) multiple times and further permute their order. This can allow the LLM to explore different contextual nuances and generate a wider range of knowledge entries, with the goal of ultimately increasing the robustness and applicability of the knowledge base for a broader range of queries.

4.3.3.3 Text-to-SQL with Knowledge Base

Based on the LLM-powered knowledge base construction process, we now have the knowledge base $\boldsymbol{\mathcal{K}}$. Hereafter, the next question to answer is then how to use this knowledge base for text-to-SQL.

Given the extensive nature of $\boldsymbol{\mathcal{K}}$, containing a large number of entries, it is crucial to identify and retrieve the most pertinent entries for the query $\boldsymbol{q}$. Formally, this process can be represented as follows: $\{\boldsymbol{k_i}\}_{\boldsymbol{i=1}}^{\boldsymbol{j}} = \texttt{Retriever}(\boldsymbol{q}, \boldsymbol{\mathcal{K}})$. Also, this can be operationalized by calculating the embedding-level similarities between the query and all the knowledge entries in the knowledge base, then selecting the top-$\boldsymbol{j}$ similar entries $\{\boldsymbol{k_i}\}_{\boldsymbol{i=1}}^{\boldsymbol{j}}$, where embeddings are obtained from a sentence embedding model [79, 1]. Moreover, to further enhance the retrieval accuracy, we train this embedding model with contrastive learning, which maximizes the similarity between the query and its relevant knowledge while minimizing the similarities of others, denoted as follows: $-\log \frac{\exp(\text{sim}(\boldsymbol{q},\boldsymbol{k}^+)/\boldsymbol{\tau})}{\exp(\text{sim}(\boldsymbol{q},\boldsymbol{k}^+)/\boldsymbol{\tau})+\sum_{\boldsymbol{k}^-}\exp(\text{sim}(\boldsymbol{q},\boldsymbol{k}^-)/\boldsymbol{\tau})}$, where $\text{sim}(\boldsymbol{q}, \boldsymbol{k})$ denotes the similarity measure between query $\boldsymbol{q}$ and knowledge $\boldsymbol{k}$, $\boldsymbol{\tau}$ is the temperature parameter, $\boldsymbol{k}^+$ is the relevant knowledge, and $\boldsymbol{k}^-$ represents the set of irrelevant knowledge.

Note that while the retrieved knowledge entries from $\boldsymbol{\mathcal{K}}$ are relevant to the given query and can assist in SQL statement formulation, they may require additional refinement to perfectly align with the query's specific needs. For instance, if the user query pertains to abnormal data conditions, but the retrieved knowledge primarily focuses on normal data, a direct application of this knowledge could lead to inaccurate SQL generation. To address this issue, we further prompt the LLM to generate the knowledge tailored to the given query by considering its relevant knowledge entries and database schema, as follows: $\boldsymbol{k}' = \texttt{LLM}(\boldsymbol{\mathcal{T}}(\boldsymbol{q}, \{\boldsymbol{k_i}\}_{\boldsymbol{i=1}}^{\boldsymbol{j}}, \boldsymbol{\mathcal{D}}))$, where $\{\boldsymbol{k_i}\}_{\boldsymbol{i=1}}^{\boldsymbol{j}}$ is the knowledge retrieved from $\boldsymbol{\mathcal{K}}$. This refined knowledge $\boldsymbol{k}'$ is subsequently used as input, along with the user query and its associated database schema, to guide the text-to-SQL LLM in generating a more accurate and contextually appropriate SQL statement: $\boldsymbol{s} = \texttt{LLM}(\boldsymbol{\mathcal{T}}(\boldsymbol{q}, \boldsymbol{k}', \boldsymbol{\mathcal{D}}))$. Please see Algorithm 1 for our overall approach.

4.3.4 Experiments

4.3.4.1 Datasets and Tasks

Datasets To validate the efficacy of KAT-SQL, we first use two widely used text-to-SQL benchmark datasets, namely BIRD [6] and Spider [260]. Specifically, BIRD is a recently released large-scale text-to-SQL dataset, built on top of 95 distinct databases spanning 37 domains. Additionally, each query in this dataset is associated with knowledge that is manually annotated by humans, providing a useful prior for formulating SQL statements. Spider is another benchmark dataset, built upon 200 databases across 138 domains. Unlike BIRD, samples in Spider do not have annotated knowledge for text-to-SQL. Lastly, we consider a challenging real-world text-to-SQL data, namely CSTINSIGHT, which is designed with actual customer queries over a data lakehouse with 34 tables without human-annotated knowledge.

Tasks and Scenarios We evaluate our KAT-SQL on three realistic text-to-SQL tasks. First of all, we consider the scenario where the prior information about some samples and their associated knowledge for each database is available, meaning that the databases used in training samples overlap with those in test samples (Overlap). We note that this setting is practical, since annotating a few pairs of questions and their corresponding knowledge for each database in advance is feasible. In addition to this, we test KAT-SQL with the existing benchmark setup, which is more challenging since it assumes there are no overlaps between databases during the training and test phases (Non-Overlap). In other words, no samples from the test-time databases are available beforehand, which means the model should be able to generalize to test-time queries based on the schemas of test-time databases as well as the samples and their associated knowledge from the different (training-time) databases. Lastly, we validate KAT-SQL on the most challenging scenario, where there are no overlaps between the databases used during training and testing, but also no knowledge is available for both training and test samples. This setup aims to test the model's ability to generalize (in the absence of any prior knowledge about the dataset), allowing us to evaluate how well our knowledge base constructed with one dataset performs on different datasets. Notably, since the Spider and CSTINSIGHT datasets have no available knowledge for all queries, we use them for the most challenging last scenario; meanwhile, we use BIRD for the first two scenarios.

4.3.4.2 Baselines and Our Model

We compare our KAT-SQL approach against relevant baselines that target our primary objective of improving knowledge-augmented text-to-SQL systems, which vary in their usage of knowledge. We note that for the fairest comparison, we fix the LLM as the same for all methods, explained as follows:

1. **No Knowledge** – which uses only the queries themselves to formulate the SQL statements without any additional knowledge.
2. **DELLM** – which generates the knowledge based on the query and its relevant database structures, and use this synthesized knowledge for text-to-SQL [372].

Table 4.11: Main results on text-to-SQL across multiple scenarios, with the best results in bold.

	BIRD (Overlap)		**BIRD (Non-Overlap)**		**Spider**		**CSTINSIGHT**	
Methods	EX	VES	EX	VES	EX	VES	EX	VES
No Knowledge	23.76	28.81	20.66	16.72	70.99	37.53	4.76	5.28
DELLM	34.70	33.15	24.64	19.27	72.44	42.90	11.90	12.02
KAT-SQL (Ours)	**41.18**	**41.33**	**41.07**	**31.14**	**74.56**	**47.20**	**14.29**	**14.50**
Oracle Knowledge	54.67	49.71	49.41	37.93	N/A	N/A	N/A	N/A

3. **KAT-SQL** – which is our model, building the knowledge base and utilizing the knowledge from it (with retrieval) for text-to-SQL.
4. **Oracle Knowledge** – which uses oracle knowledge annotated by humans, along with the queries to generate the SQL statements. This approach serves as an upper bound and is not directly comparable to other models due to its reliance on manually curated knowledge that is typically unavailable.

4.3.4.3 Evaluation Metrics

Following the standard evaluation protocols from prior work [6, 372], we use the following two metrics: 1) Execution Accuracy (EX), which measures the ratio of generated SQL code that has the same execution results with ground-truth SQL code; 2) Valid Efficiency Score (VES), which considers the efficiency of generated SQLs by weighting them based on their relative efficiency improvement over ground-truth SQLs further multiplied by execution accuracy.

4.3.4.4 Implementation Details

We mainly use Llama-3 70B [373] as the basis for text-to-SQL generation and knowledge generation across all baselines and our model variants for most experiments, for a fair comparison, while we also experiment with other LLMs in an analysis (Table 4.16) to see the robustness of KAT-SQL. For the hyperparameters, except for the temperature (which we set as 0.0 for reproducibility), we use its default values. In addition, for the retriever, we use MPNet [1], which is based on dense retrieval; we train it with a batch size of 128 and a number of training epochs of 30.

4.3.4.5 Experimental Results and Analyses

Main Results We provide main results in Table 4.11, which confirms that our KAT-SQL approach consistently outperforms all baselines by large margins. Specifically, while we observe some performance improvement of the knowledge-augmented text-to-SQL approach (namely DELLM, which generates a few pieces of knowledge for each query) over the baseline without knowledge augmentation, KAT-SQL achieves even greater gains, demonstrating the effectiveness of our knowledge base construction-based text-to-SQL paradigm. However, the performance of the (incomparable) model with the oracle knowledge (annotated by human experts) remains superior to all other approaches, which suggests potential future opportunities for developing a more advanced pipeline for knowledge generation.

Analysis on Knowledge Base To further understand the coverage and relevance of the knowledge within our knowledge base, we compare each piece of knowledge required for test-time queries with all the available entries in the knowledge base, as a function of the number of knowledge generation steps during

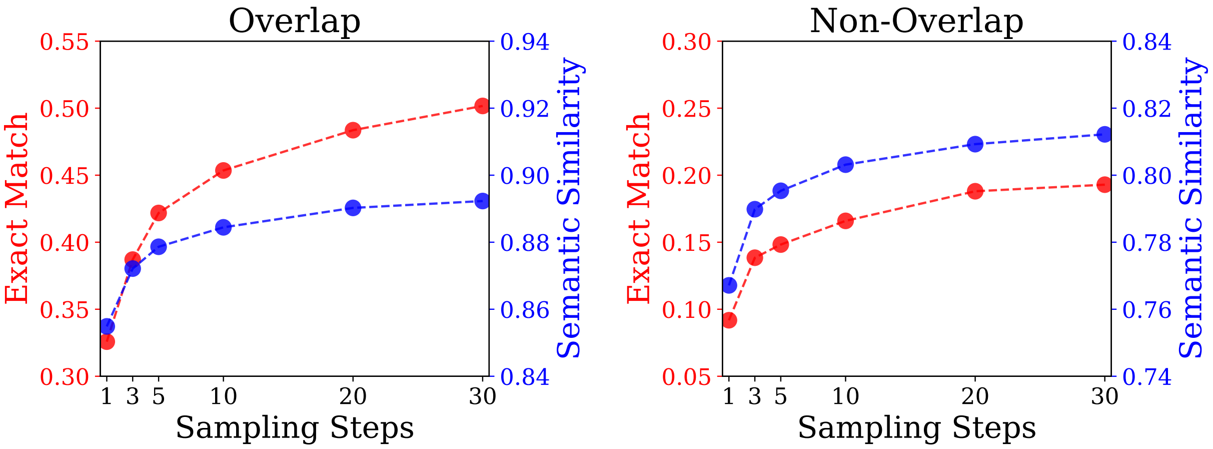


Figure 4.13: Results for coverage and relevance of the knowledge entries in the constructed knowledge base against the gold knowledge, with different numbers of knowledge generation steps.

Table 4.12: Results for knowledge generation with and without the use of the Knowledge Base (KB), while varying the prompt construction with and without the relevant few-shot examples.

		Overlap		Non-Overlap	
KB	**Few-Shot**	**EM**	**SS**	**EM**	**SS**
w/o KB	Random	10.96	68.77	7.88	66.77
	Retrieval	20.21	73.62	9.24	68.78
w/ KB	Random	11.13	69.14	7.93	66.80
	Retrieval	**24.97**	**77.87**	**12.94**	**71.24**

knowledge base construction. For evaluation, we use two metrics: Exact Match, which identifies whether the knowledge base contains an entry that precisely matches the knowledge required for a given query, and Semantic Similarity, which assesses how closely related the most similar entry (in the knowledge base) is to the required knowledge based on the embedding-level similarity. As shown in Figure 4.13, we observe that, under the Overlap setting, half of the knowledge entries needed for test-time queries are available in the knowledge base while the Semantic Similarity is around 90%, which demonstrates substantial coverage by our knowledge base. In addition, for the challenging setup where training and test databases are distinct, we still observe that 20% of the test-time knowledge entries are available in the knowledge base and that the Semantic Similarity exceeds 80%, showing the utility of our knowledge base. Finally, as we increase the number of knowledge generation steps for each instance during knowledge base construction, we observe a corresponding improvement in both coverage and relevance of our knowledge base, which supports the effectiveness of our expansion strategy to enrich its diversity.

Analysis on Knowledge Generation Recall that we further refine the retrieved knowledge to make it more suitable for each query, in addition to constructing the knowledge base and retrieving the relevant knowledge. Thus, to see how relevant the generated knowledge is to the human-annotated gold knowledge with regards to the use of our knowledge base, we report comparison results according to Exact Match and Semantic Similarity (SS) in Table 4.12. We observe that when we retrieve the relevant knowledge from the knowledge base and then use it for knowledge generation, there are performance gains over the case where we do not leverage it, which indicates that the retrieved knowledge is helpful in formulating

Table 4.15: Retrieval results with different scenarios and models.

Settings	Models	MRR	Top@3	Top@10
Overlap	BERT	0.5506	0.6621	0.8911
	TAS-B	0.5630	0.6943	0.9035
	TAS-B*	0.8288	0.9143	0.9765
Non-Overlap	BERT	0.2148	0.2692	0.4231
	TAS-B	0.2364	0.3846	0.4615
	TAS-B*	0.7565	0.8347	0.9210

the necessary knowledge for test-time queries. We also provide few-shot examples to guide the knowledge generation model in generating useful knowledge in the right format, and when we select them based on their similarities with the given query, we observe further gains in the quality of the generated knowledge.

Table 4.13: Text-to-SQL results without using any knowledge, based on the retrieved knowledge, and based on the refined knowledge from the retrieved knowledge (our KAT-SQL).

Settings	Models	EX
Overlap	KAT-SQL (Ours)	41.18
	w/o Generation	38.94
	w/o Retrieval & Generation	23.76
Non-Overlap	KAT-SQL (Ours)	41.07
	w/o Generation	38.42
	w/o Retrieval & Generation	20.66

Beyond evaluating the quality of the generated knowledge by comparing it to the human-annotated gold knowledge, we also examine the impact of knowledge generation on downstream text-to-SQL performance with and without the incorporation of generated knowledge. As shown in Table 4.13, compared to the results without the knowledge retrieval and generation on both Overlap and Non-Overlap settings, there are substantial improvements when we incorporate the retrieved knowledge from our knowledge base into the text-to-SQL generation process. Furthermore, instead of directly using the retrieved knowledge, refining this retrieved knowledge yields additional improvements, underscoring the importance of not only retrieving relevant knowledge but also tailoring it to better align with the specific needs of test-time queries.

Table 4.14: Breakdown text-to-SQL results into overlapping and non-overlapping domain settings between training (knowledge base construction) and test (text-to-SQL evaluation) databases.

Models	Overlap	Non-Overlap
No Knowledge	22.85	16.20
DELLM	27.20	19.43
KAT-SQL (Ours)	**49.37**	**24.19**

Generalization Analysis to Different Domains To see whether our knowledge base can be generalizable to databases of different domains (that are not overlapped with those for knowledge base construction), we breakdown the performance based on whether test databases share domains with training databases or belong to different domains (according to 37 domains categorized from Li et al. [6]). As shown in Table 4.14, our KAT-SQL achieves substantially higher performance when test databases overlap with training domains compared to those from unseen domains; however, even in the latter case, KAT-SQL still outperforms existing baselines. These results indicate that, while the lack of domain overlaps degrades the performance, our knowledge base still provides meaningful benefits for unseen domains, demonstrating its generalizability.

Table 4.17: Results of KAT-SQL with the state-of-the-art text-to-SQL model on the BIRD leaderboard.

Models	EX
ChatGPT	24.05
ChatGPT + CoT	25.88
ExSL + granite-20b-code	51.69
ExSL + granite-20b-code w/ KAT-SQL (Ours)	57.56
ExSL + granite-20b-code w/ Oracle Knowledge	65.38

Retrieval Analysis We also analyze the accuracy of knowledge retrieval from our knowledge base by reporting its retrieval performance in Table 4.15 according to Mean Reciprocal Rank (MRR) and Top@K Accuracy. We observe that the retrieval accuracy on the Overlap setting is higher than that on the Non-Overlap setting, due to the less availability of relevant knowledge required for test-time queries in the Non-Overlap setting. Yet, when we replace the knowledge base constructed from our approach with the Oracle knowledge base (*), which includes all the necessary knowledge for test-time queries, the MRR on both settings reaches around 80%, indicating the importance of expanding the coverage of the knowledge base for accurate knowledge retrieval. The table also compares the performance of different basis models for retrieval – BERT [26] and TAS-B [35] – with the latter being fine-tuned for retrieval. It can be seen that the extra training of the model on retrieval tasks aids in achieving superior performance for retrieving the knowledge for text-to-SQL.

Analysis with Different LLMs To evaluate how robust our KAT-SQL approach is across different LLMs, we conduct the additional analysis instantiating the text-to-SQL and knowledge generation models with other recent LLMs such as Granite 34B [392] and Mixtral 8x7B [363]; results are shown in Table 4.16. From this, we observe that KAT-SQL consistently outperforms all baselines regardless of the choice of LLMs, which demonstrates the effectiveness and versatility of our proposed KAT-SQL approach.

Table 4.16: Text to SQL results with different LLMs.

LLMs	Methods	Overlap	Non-Overlap
Llama	No Knowledge	23.76	20.66
	DELLM	34.70	24.64
	KAT-SQL	41.18	41.07
	Oracle Knowledge	54.67	49.41
Granite	No Knowledge	25.83	17.75
	DELLM	34.04	20.21
	KAT-SQL	39.28	35.83
	Oracle Knowledge	46.56	38.32
Mixtral	No Knowledge	11.75	10.58
	DELLM	27.17	11.29
	KAT-SQL	29.31	20.30
	Oracle Knowledge	37.26	30.88

Finally, we augment the state-of-the-art text-to-SQL model (in the setting without oracle knowledge) on the BIRD leaderboard [6], namely ExSL + granite-20b-code, using the knowledge generated from our proposed knowledge base construction-based approach. As shown in Table 4.17, we observe that the text-to-SQL model combined with our KAT-SQL approach establishes the new state-of-the-art performance, highlighting the value of our method as a powerful tool for text-to-SQL.

Analysis on Efficiency While our primary focus is on improving the text-to-SQL accuracy through knowledge base construction and augmentation, we also consider the efficiency of our approach. Note that the construction of the knowledge base is performed offline and does not affect real-time query processing; therefore, the extra computational overhead comes from retrieving relevant knowledge and generating the SQL statement for the given query. In this regard, our retrieval process accounts for only

Table 4.18: Case study of knowledge and SQL generation results from three queries.

Example	Query	Generated Knowledge and SQL
Example 1	Which position has a lower minimum salary, Account Representative or Trainee?	**Generated Knowledge:** 1) lower minimum salary refers to `MIN(minsalary)` 2) Account Representative and Trainee are position titles **Generated SQL:** `SELECT positiontitle FROM position WHERE positiontitle = ’Account Representative’ OR positiontitle = ’Trainee’ ORDER BY minsalary ASC LIMIT 1`
Example 2	Among the employees working at the office in New York, how many of them have a good job performance?	**Generated Knowledge:** 1) New York refers to `state = ’NY’` 2) good job performance refers to `performance = ’Good’` **Generated SQL:** `SELECT COUNT(*) FROM employee AS T1 INNER JOIN location AS T2 ON T1.locationID = T2.locationID WHERE T2.state = ’NY’ AND T1.performance = ’Good’`
Example 3	For all patients born in 1982, state if their albumin is within normal range.	**Generated Knowledge:** 1) albumin is within normal range refers to `ALB between 3.5 and 5.5` **Generated SQL:** `SELECT CASE WHEN T2.ALB >= 3.5 AND T2.ALB <= 5.5 THEN ’normal’ ELSE ’abnormal’ END FROM Patient AS T1 INNER JOIN Laboratory AS T2 ON T1.ID = T2.ID WHERE STRFTIME(’%Y’, T1.Birthday) = ’1982’`

Table 4.19: Examples of original and its (similar) constructed knowledge within the knowledge base.

Example	Original Knowledge	Constructed Similar Knowledge
Example 1	albumin is within normal range refers to `ALB between 3.5 and 5.5`	1) albumin is outside the normal range refers to `ALB less than 3.5 or greater than 5.5` 2) glucose is within normal range refers to `GLU between 70 and 100 mg/dL` 3) Hemoglobin (Hb) is considered normal for males if levels range from `13.5 to 17.5 g/dL`
Example 2	Eligible free rate for K-12 = Free Meal Count (K-12) / Enrollment (K-12)	1) Eligible reduced-price rate for K-12 = Reduced-Price Meal Count (K-12) / Enrollment (K-12) 2) Eligible free meal rate for students aged 5-17 = Free Meal Count (Ages 5-17) / Enrollment (Ages 5-17) 3) Difference between K-12 and ages 5-17 enrollment = `Enrollment (K-12) - Enrollment (Ages 5-17)`
Example 3	Slovakia can be represented as `Country = ’SVK’`	1) France can be represented as `Country = ’FRA’` 2) Brazil can be represented as `Country = ’BRA’` 3) Monaco can be represented as `Country = ’MCO’`

2% of the overall generation time, thanks to efficient search algorithm [237], making its impact negligible. Also, although incorporating knowledge into the text-to-SQL pipeline increases the prompt length by 30%, this overhead aligns with other knowledge augmentation methods (such as DELLM) and does not introduce additional latency specific to our approach. Overall, each query is processed under 5 seconds.

Examples We provide examples for the knowledge generation and text-to-SQL in Table 4.18 as well as the entries in the knowledge base in Table 4.19.

4.3.5 Summary

In this work, we proposed a novel knowledge base construction-based text-to-SQL approach called KAT-SQL, based on the motivation that one piece of knowledge can be reused across multiple queries and databases. Our approach involves the creation of the knowledge base from which relevant knowledge is retrieved and utilized to generate SQL statements from queries. Through extensive evaluations on multiple datasets with two different scenarios, we showed that KAT-SQL outperforms relevant baselines. In addition, our detailed analyses highlight the effectiveness of each component in the knowledge generation and retrieval processes, but also the high coverage and relevance of the entries in the base.

Chapter 5. Conclusion and Future Work

5.1 Summary of Contributions

This dissertation investigated contextualization (i.e., the process of retrieving, structuring, verifying, and integrating knowledge) as a foundational principle for advancing the capabilities of modern foundation models beyond parameter scaling. In particular, while scaling laws have driven much of the recent progress in large models, the limitations of their internalized knowledge (which could be incomplete and outdated), along with the inherent statelessness of these models, impose a ceiling on reliability. Accordingly, this thesis argued that the effectiveness of a model is ultimately governed not by its architectural improvements or sizes but by the quality of the context supplied to it at inference time. The dissertation presented three interconnected research directions that collectively advance this central thesis, as follows:

First, we proposed several knowledge-augmented contextualization methods [15, 16, 17, 18] that allow models to effectively incorporate structured, unstructured, and multimodal sources without modifying their parameters. Across tasks involving knowledge graphs, text, and video, these approaches showed that even closed-source models can operate over richer and more targeted knowledge when equipped with appropriately constructed context. Furthermore, we identified failure modes of contextualization methods: retrieval errors and grounding errors, and introduced Knowledge-Augmented Language Model Verification (KALMV) and its streaming variant [19, 20], which detects and rectifies these errors through iterative verification, substantially reducing hallucinations and improving factual consistency.

Second, we developed retrieval techniques that broaden the knowledge space for contextualization. Specifically, we explored multimodal document representation learning for retrieval [22], temporal and visual retrieval from videos [17], and relational fact retrieval from knowledge graphs [21]. Additionally, these techniques further demonstrate how heterogeneous knowledge (such as text, images, tables, videos, and graph facts) can be transformed into representations, which are compatible with retrieval for the conventional text corpus. Extending this idea further, I proposed a universal retrieval framework capable of generating source-specific queries while maintaining a unified interface across modalities and schemas.

Third, we demonstrated how contextualization enables real-world applications requiring domain-specific grounding, which includes (1) personalized query suggestion via lightweight user-specific knowledge stores [23]; (2) scientific idea generation and refinement through the integration of papers, citation graphs, and concept-level knowledge [24]; and (3) natural-language access to enterprise databases via knowledge-base construction for text-to-SQL generation [25]. Across these domains, contextualization consistently enhanced accuracy, relevance, and domain suitability, showing its practical value.

5.2 Future Research Directions

While this dissertation establishes contextualization as a foundational principle for advancing the reliability, adaptability, and applicability of foundation models, several important avenues remain open. Below, we outline promising future directions that could further expand the impact of contextualization and support the development of more capable AI agents and systems.

Conflict- and Reliability-Aware Contextualization As models increasingly rely on heterogeneous and independently constructed knowledge sources, both knowledge conflicts (arising across sources, over time, or across perspectives) and varying levels of source reliability pose new challenges; however, most of the current retrieval and augmentation strategies primarily optimize for relevance and do not explicitly model contradictions, uncertainty, or the trustworthiness of evidence. Therefore, future work may develop conflict- and reliability-aware contextualization mechanisms that (1) detect and represent disagreements and reliability signals across retrieved evidence, (2) perform perspective alignment that accounts for the provenance and trustworthiness of each source, and (3) enable models to reason about uncertainty, competing hypotheses, and source credibility. Such mechanisms would be particularly crucial in domains like scientific discovery or law, where evidence could be heterogeneous and often ambiguous.

Memory-Enhanced Agentic Systems for Lifelong Contextualization As agents (powered by foundation models) become increasingly autonomous and long-running, their behavior will depend not only on immediate retrieval but also on their ability to build, maintain, and refine long-term memory. However, extending contextualization into a continual setting involves additional challenges: (1) constructing persistent knowledge stores (potentially through graph-structures that better capture the entities, relations, temporal updates, and provenance of accumulated knowledge); (2) maintaining temporal coherence, recency, and forgetting strategies; and (3) enabling models to update beliefs and contextual knowledge over time while avoiding catastrophic drift. In addition, future research may unify retrieval, memory organization, and verification into a single agentic framework, enabling agents to contextualize themselves as they interact with the world, accumulate new information, and refine past knowledge.

Verification Beyond Knowledge-Grounded Contextualization One noteworthy observation in this dissertation is that verification tends to be easier than generation within the domain of knowledge-grounded contextualization (Section 2.2), since verification often takes the form of checking consistency between a query, retrieved evidence, and generated output: an operation that reduces to a discrimination problem where relevance (to the query) and grounding (in the output) can be evaluated with explicit knowledge. Yet, this structural advantage does not generalize to settings in which truth is not explicitly observable. For example, domains such as evaluating open-ended research ideas (Section 4.2) may lack a single authoritative ground truth and require reasoning over interpretations that vary across jurisdictions, framings, and disciplinary perspectives. In other words, in such settings, verifier (and reward) models should reason over multiple plausible interpretations rather than checking consistency, and it makes verification far more ambiguous and subjective. This gap suggests that while contextualization benefits from the verifiable structure, advancing verification mechanisms capable of operating under uncertainty, plurality, and contested notions of correctness should be an interesting direction for future work.

Bibliography

[1] Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. Mpnet: Masked and permuted pre-training for language understanding. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

[2] Tom Ayoola, Shubhi Tyagi, Joseph Fisher, Christos Christodoulopoulos, and Andrea Pierleoni. Refined: An efficient zero-shot-capable approach to end-to-end entity linking. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Industry Track, NAACL 2022, Hybrid: Seattle, Washington, USA + Online, July 10-15, 2022*, pages 209–220. Association for Computational Linguistics, 2022.

[3] Qingyun Wang, Doug Downey, Heng Ji, and Tom Hope. Learning to generate novel scientific directions with contextualized literature-based discovery. *arXiv preprint arXiv:2305.14259*, 2023.

[4] Zonglin Yang, Xinya Du, Junxian Li, Jie Zheng, Soujanya Poria, and Erik Cambria. Large language models for automated open-domain scientific hypotheses discovery. In *Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024*, pages 13545–13565. Association for Computational Linguistics, 2024.

[5] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

[6] Jinyang Li, Binyuan Hui, Ge Qu, Jiaxi Yang, Binhua Li, Bowen Li, Bailin Wang, Bowen Qin, Ruiying Geng, Nan Huo, Xuanhe Zhou, Chenhao Ma, Guoliang Li, Kevin Chen-Chuan Chang, Fei Huang, Reynold Cheng, and Yongbin Li. Can LLM already serve as A database interface? A big bench for large-scale database grounded text-to-sqls. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023.

[7] OpenAI. Introducing chatgpt. https://openai.com/blog/chatgpt, 2022.

[8] OpenAI. GPT-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[9] OpenAI. GPT-4V(ision) system card. https://openai.com/index/gpt-4v-system-card/, 2023.

[10] Gemini Team. Gemini: A family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

[11] Gemini Team. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025.

[12] Llama Team. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

[13] An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jianxin Yang, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Xuejing Liu, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, Zhifang Guo, and Zhihao Fan. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*, 2024.

[14] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Ming-Hsuan Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025.

[15] Jinheon Baek, Alham Fikri Aji, and Amir Saffari. Knowledge-augmented language model prompting for zero-shot knowledge graph question answering. In *Proceedings of the 1st Workshop on Natural Language Reasoning and Structured Explanations (NLRSE)*, Toronto, Canada, June 2023. Association for Computational Linguistics.

[16] Soyeong Jeong, Jinheon Baek, Sukmin Cho, Sung Ju Hwang, and Jong Park. Adaptive-rag: Learning to adapt retrieval-augmented large language models through question complexity. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), NAACL 2024, Mexico City, Mexico, June 16-21, 2024*, pages 7036–7050. Association for Computational Linguistics, 2024.

[17] Soyeong Jeong, Kangsan Kim, Jinheon Baek, and Sung Ju Hwang. Videorag: Retrieval-augmented generation over video corpus. In *Findings of the Association for Computational Linguistics, ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, pages 21278–21298. Association for Computational Linguistics, 2025.

[18] Woongyeong Yeo, Kangsan Kim, Soyeong Jeong, Jinheon Baek, and Sung Ju Hwang. Universalrag: Retrieval-augmented generation over multiple corpora with diverse modalities and granularities. *arXiv preprint arXiv:2504.20734*, 2025.

[19] Jinheon Baek, Soyeong Jeong, Minki Kang, Jong C. Park, and Sung Ju Hwang. Knowledge-augmented language model verification. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 1720–1736. Association for Computational Linguistics, 2023.

[20] Joonho Ko, Jinheon Baek, and Sung Ju Hwang. Efficient real-time refinement of language model text generation. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, Suzhou, China, November 2025. Association for Computational Linguistics.

[21] Jinheon Baek, Alham Fikri Aji, Jens Lehmann, and Sung Ju Hwang. Direct fact retrieval from knowledge graphs without entity linking. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 10038–10055. Association for Computational Linguistics, 2023.

[22] Jaewoo Lee, Joonho Ko, Jinheon Baek, Soyeong Jeong, and Sung Ju Hwang. Unified multimodal interleaved document representation for retrieval. In *The 1st Workshop on Vector Databases*, 2025.

[23] Jinheon Baek, Nirupama Chandrasekaran, Silviu Cucerzan, Allen Herring, and Sujay Kumar Jauhar. Knowledge-augmented large language models for personalized contextual query suggestion. In *Proceedings of the ACM on Web Conference 2024, WWW 2024, Singapore, May 13-17, 2024*, pages 3355–3366. ACM, 2024.

[24] Jinheon Baek, Sujay Kumar Jauhar, Silviu Cucerzan, and Sung Ju Hwang. Researchagent: Iterative research idea generation over scientific literature with large language models. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL 2025 - Volume 1: Long Papers, Albuquerque, New Mexico, USA, April 29 - May 4, 2025*, pages 6709–6738. Association for Computational Linguistics, 2025.

[25] Jinheon Baek, Horst Samulowitz, Oktie Hassanzadeh, Dharmashankar Subramanian, Sola Shirai, Alfio Gliozzo, and Debarun Bhattacharjya. Knowledge base construction for knowledge-augmented text-to-sql. In *Findings of the Association for Computational Linguistics, ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, pages 26569–26583. Association for Computational Linguistics, 2025.

[26] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics, 2019.

[27] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67, 2020.

[28] Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander H. Miller. Language models as knowledge bases? 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 2019, Hong Kong, China, November 3-7, 2019*, pages 2463–2473. Association for Computational Linguistics, 2019.

[29] Adam Roberts, Colin Raffel, and Noam Shazeer. How much knowledge can you pack into the parameters of a language model? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 5418–5426. Association for Computational Linguistics, 2020.

[30] Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Comput. Surv.*, 55(9):195:1–195:35, 2023.

[31] Anna Rohrbach, Lisa Anne Hendricks, Kaylee Burns, Trevor Darrell, and Kate Saenko. Object hallucination in image captioning. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 4035–4045. Association for Computational Linguistics, 2018.

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

[33] Wen-tau Yih, Matthew Richardson, Christopher Meek, Ming-Wei Chang, and Jina Suh. The value of semantic parse labeling for knowledge base question answering. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 2: Short Papers*. The Association for Computer Linguistics, 2016.

[34] Hannah Bast, Björn Buchhold, and Elmar Haussmann. Semantic search on text and knowledge bases. *Found. Trends Inf. Retr.*, 10(2-3):119–271, 2016.

[35] Sebastian Hofstätter, Sheng-Chieh Lin, Jheng-Hong Yang, Jimmy Lin, and Allan Hanbury. Efficiently teaching an effective dense retriever with balanced topic aware sampling. In *SIGIR '21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, Canada, July 11-15, 2021*, pages 113–122. ACM, 2021.

[36] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008, 2017.

[37] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[38] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 5754–5764, 2019.

[39] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018.

[40] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin

Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. Palm: Scaling language modeling with pathways. *J. Mach. Learn. Res.*, 24:240:1–240:113, 2023.

[41] Saleh Soltan, Shankar Ananthakrishnan, Jack FitzGerald, Rahul Gupta, Wael Hamza, Haidar Khan, Charith Peris, Stephen Rawls, Andy Rosenbaum, Anna Rumshisky, Chandana Satya Prakash, Mukund Sridhar, Fabian Triefenbach, Apurv Verma, Gökhan Tür, and Prem Natarajan. Alexatm 20b: Few-shot learning using a large-scale multilingual seq2seq model. *arXiv preprint arXiv:2208.01448*, 2022.

[42] Mujeen Sung, Jinhyuk Lee, Sean S. Yi, Minji Jeon, Sungdong Kim, and Jaewoo Kang. Can language models be biomedical knowledge bases? In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 4723–4734. Association for Computational Linguistics, 2021.

[43] Ohad Rubin, Jonathan Herzig, and Jonathan Berant. Learning to retrieve prompts for in-context learning. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL 2022, Seattle, WA, United States, July 10-15, 2022*, pages 2655–2671. Association for Computational Linguistics, 2022.

[44] Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. What makes good in-context examples for gpt-3? In *Proceedings of Deep Learning Inside Out: The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures, DeeLIO@ACL 2022, Dublin, Ireland and Online, May 27, 2022*, pages 100–114. Association for Computational Linguistics, 2022.

[45] Victor Sanh, Albert Webson, Colin Raffel, Stephen H. Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Arun Raja, Manan Dey, M Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal V. Nayak, Debajyoti Datta, Jonathan Chang, Mike Tian-Jian Jiang, Han Wang, Matteo Manica, Sheng Shen, Zheng Xin Yong, Harshit Pandey, Rachel Bawden, Thomas Wang, Trishala Neeraj, Jos Rozen, Abheesht Sharma, Andrea Santilli, Thibault Févry, Jason Alan Fries, Ryan Teehan, Teven Le Scao, Stella Biderman, Leo Gao, Thomas Wolf, and Alexander M. Rush. Multitask prompted training enables zero-shot task generalization. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022.

[46] Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. Finetuned language models are zero-shot learners. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022.

[47] Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Pro-*

cessing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021, pages 3045–3059. Association for Computational Linguistics, 2021.

[48] Xiang Chen, Lei Li, Ningyu Zhang, Xiaozhuan Liang, Shumin Deng, Chuanqi Tan, Fei Huang, Luo Si, and Huajun Chen. Decoupling knowledge from memorization: Retrieval-augmented prompt learning. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022.

[49] Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. Retrieval augmented language model pre-training. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 3929–3938. PMLR, 2020.

[50] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

[51] Fabian Galetzka, Jewgeni Rose, David Schlangen, and Jens Lehmann. Space efficient context encoding for non-task-oriented dialogue generation with graph attention transformer. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 7028–7041. Association for Computational Linguistics, 2021.

[52] Md. Rashad Al Hasan Rony, Ricardo Usbeck, and Jens Lehmann. Dialokg: Knowledge-structure aware task-oriented dialogue generation. In *Findings of the Association for Computational Linguistics: NAACL 2022, Seattle, WA, United States, July 10-15, 2022*, pages 2557–2571. Association for Computational Linguistics, 2022.

[53] Minki Kang, Jin Myung Kwak, Jinheon Baek, and Sung Ju Hwang. Knowledge-consistent dialogue generation with knowledge graphs. In *ICML 2022 Workshop on Knowledge Retrieval and Language Models*, 2022.

[54] Gautier Izacard, Patrick Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. Atlas: Few-shot learning with retrieval augmented language models. *J. Mach. Learn. Res.*, 24:251:1–251:43, 2023.

[55] Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022.

[56] Jiacheng Liu, Alisa Liu, Ximing Lu, Sean Welleck, Peter West, Ronan Le Bras, Yejin Choi, and Hannaneh Hajishirzi. Generated knowledge prompting for commonsense reasoning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long*

Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022, pages 3154–3169. Association for Computational Linguistics, 2022.

[57] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022.

[58] Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023.

[59] Angeliki Lazaridou, Elena Gribovskaya, Wojciech Stokowiec, and Nikolai Grigorev. Internet-augmented language models through few-shot prompting for open-domain question answering. *arXiv preprint arXiv:2203.05115*, 2022.

[60] Nilesh Chakraborty, Denis Lukovnikov, Gaurav Maheshwari, Priyansh Trivedi, Jens Lehmann, and Asja Fischer. Introduction to neural network based approaches for question answering over knowledge graphs. *arXiv preprint arXiv:1907.09361*, 2019.

[61] Bin Fu, Yunqi Qiu, Chengguang Tang, Yang Li, Haiyang Yu, and Jian Sun. A survey on complex question answering over knowledge base: Recent advances and challenges. *arXiv preprint arXiv:2007.13069*, 2020.

[62] Wen-tau Yih, Ming-Wei Chang, Xiaodong He, and Jianfeng Gao. Semantic parsing via staged query graph generation: Question answering with knowledge base. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers*, pages 1321–1331. The Association for Computer Linguistics, 2015.

[63] Junwei Bao, Nan Duan, Zhao Yan, Ming Zhou, and Tiejun Zhao. Constraint-based question answering with knowledge graph. In *COLING 2016, 26th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, December 11-16, 2016, Osaka, Japan*, pages 2503–2514. ACL, 2016.

[64] Kangqi Luo, Fengli Lin, Xusheng Luo, and Kenny Q. Zhu. Knowledge base question answering via encoding of complex query graphs. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 2185–2194. Association for Computational Linguistics, 2018.

[65] Haitian Sun, Bhuwan Dhingra, Manzil Zaheer, Kathryn Mazaitis, Ruslan Salakhutdinov, and William W. Cohen. Open domain question answering using early fusion of knowledge bases and text. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 4231–4242. Association for Computational Linguistics, 2018.

[66] Apoorv Saxena, Aditay Tripathi, and Partha P. Talukdar. Improving multi-hop question answering over knowledge graphs using knowledge base embeddings. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 4498–4507. Association for Computational Linguistics, 2020.

[67] Michihiro Yasunaga, Hongyu Ren, Antoine Bosselut, Percy Liang, and Jure Leskovec. QA-GNN: reasoning with language models and knowledge graphs for question answering. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 535–546. Association for Computational Linguistics, 2021.

[68] William W. Cohen, Haitian Sun, R. Alex Hofer, and Matthew Siegler. Scalable neural methods for reasoning with a symbolic knowledge base. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020.

[69] Amir Saffari, Armin Oliya, Priyanka Sen, and Tom Ayoola. End-to-end entity resolution and question answering using differentiable knowledge graphs. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 4193–4200. Association for Computational Linguistics, 2021.

[70] Priyanka Sen, Armin Oliya, and Amir Saffari. Expanding end-to-end question answering on differentiable knowledge graphs with intersection. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 8805–8812. Association for Computational Linguistics, 2021.

[71] Yucheng Zhou, Xiubo Geng, Tao Shen, Wenqiang Zhang, and Daxin Jiang. Improving zero-shot cross-lingual transfer for multilingual question answering over knowledge graph. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 5822–5834. Association for Computational Linguistics, 2021.

[72] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for NLP. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 2790–2799. PMLR, 2019.

[73] Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 3045–3059. Association for Computational Linguistics, 2021.

[74] Ledell Wu, Fabio Petroni, Martin Josifoski, Sebastian Riedel, and Luke Zettlemoyer. Scalable zero-shot entity linking with dense entity retrieval. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 6397–6407. Association for Computational Linguistics, 2020.

[75] Belinda Z. Li, Sewon Min, Srinivasan Iyer, Yashar Mehdad, and Wen-tau Yih. Efficient one-pass end-to-end entity linking for questions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 6433–6441. Association for Computational Linguistics, 2020.

[76] Barlas Oguz, Xilun Chen, Vladimir Karpukhin, Stan Peshterliev, Dmytro Okhonko, Michael Sejr Schlichtkrull, Sonal Gupta, Yashar Mehdad, and Scott Yih. Unik-qa: Unified representations of structured and unstructured knowledge for open-domain question answering. In *Findings of the Association for Computational Linguistics: NAACL 2022, Seattle, WA, United States, July 10-15, 2022*, pages 1535–1546. Association for Computational Linguistics, 2022.

[77] Kaixin Ma, Hao Cheng, Xiaodong Liu, Eric Nyberg, and Jianfeng Gao. Open domain question answering with A unified knowledge interface. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 1605–1620. Association for Computational Linguistics, 2022.

[78] Jimmy Lin, Rodrigo Nogueira, and Andrew Yates. *Pretrained Transformers for Text Ranking: BERT and Beyond*. Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers, 2021.

[79] Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 6769–6781. Association for Computational Linguistics, 2020.

[80] Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul N. Bennett, Junaid Ahmed, and Arnold Overwijk. Approximate nearest neighbor negative contrastive learning for dense text retrieval. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021.

[81] Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. Semantic parsing on freebase from question-answer pairs. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, EMNLP 2013, 18-21 October 2013, Grand Hyatt Seattle, Seattle, Washington, USA, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1533–1544. ACL, 2013.

[82] Kurt D. Bollacker, Colin Evans, Praveen K. Paritosh, Tim Sturge, and Jamie Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD 2008, Vancouver, BC, Canada, June 10-12, 2008*, pages 1247–1250. ACM, 2008.

[83] Dennis Diefenbach, Thomas Pellissier Tanon, Kamal Deep Singh, and Pierre Maret. Question answering benchmarks for wikidata. In *Proceedings of the ISWC 2017 Posters & Demonstrations and Industry Tracks co-located with 16th International Semantic Web Conference (ISWC 2017), Vienna, Austria, October 23rd - to - 25th, 2017*, volume 1963 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2017.

[84] Priyanka Sen, Alham Fikri Aji, and Amir Saffari. Mintaka: A complex, natural, and multilingual dataset for end-to-end question answering. In *Proceedings of the 29th International Conference*

on Computational Linguistics, COLING 2022, Gyeongju, Republic of Korea, October 12-17, 2022, pages 1604–1619. International Committee on Computational Linguistics, 2022.

[85] Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona T. Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. OPT: open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*, 2022.

[86] Jun Yin, Xin Jiang, Zhengdong Lu, Lifeng Shang, Hang Li, and Xiaoming Li. Neural generative question answering. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016*, pages 2972–2978. IJCAI/AAAI Press, 2016.

[87] Vaibhav Mavi, Anubhav Jangra, and Adam Jatowt. A survey on multi-hop question answering and generation. *arXiv preprint arXiv:2204.09140*, 2022.

[88] Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 8086–8098. Association for Computational Linguistics, 2022.

[89] Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. Calibrate before use: Improving few-shot performance of language models. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 12697–12706. PMLR, 2021.

[90] Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Y. Zhao, Yanping Huang, Andrew M. Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. Scaling instruction-finetuned language models. *J. Mach. Learn. Res.*, 25:70:1–70:53, 2024.

[91] Yejin Bang, Samuel Cahyawijaya, Nayeon Lee, Wenliang Dai, Dan Su, Bryan Wilie, Holy Lovenia, Ziwei Ji, Tiezheng Yu, Willy Chung, Quyet V. Do, Yan Xu, and Pascale Fung. A multitask, multilingual, multimodal evaluation of chatgpt on reasoning, hallucination, and interactivity. In *Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics, IJCNLP 2023 -Volume 1: Long Papers, Nusa Dua, Bali, November 1 - 4, 2023*, pages 675–718. Association for Computational Linguistics, 2023.

[92] Shen Zheng, Jie Huang, and Kevin Chen-Chuan Chang. Why does chatgpt fall short in answering questions faithfully? *arXiv preprint arXiv:2304.10513*, 2023.

[93] Angeliki Lazaridou, Elena Gribovskaya, Wojciech Stokowiec, and Nikolai Grigorev. Internet-augmented language models through few-shot prompting for open-domain question answering. *arXiv preprint arXiv:2203.05115*, 2022.

[94] Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 9802–9822. Association for Computational Linguistics, 2023.

[95] Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. Self-refine: Iterative refinement with self-feedback. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023.

[96] Luyu Gao, Zhuyun Dai, Panupong Pasupat, Anthony Chen, Arun Tejasvi Chaganty, Yicheng Fan, Vincent Y. Zhao, Ni Lao, Hongrae Lee, Da-Cheng Juan, and Kelvin Guu. RARR: researching and revising what language models say, using language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 16477–16508. Association for Computational Linguistics, 2023.

[97] Zhengbao Jiang, Frank F. Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. Active retrieval augmented generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 7969–7992. Association for Computational Linguistics, 2023.

[98] Zhibin Gou, Zhihong Shao, Yeyun Gong, Yelong Shen, Yujiu Yang, Nan Duan, and Weizhu Chen. CRITIC: large language models can self-correct with tool-interactive critiquing. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024.

[99] Shicheng Xu, Liang Pang, Huawei Shen, Xueqi Cheng, and Tat-Seng Chua. Search-in-the-chain: Towards the accurate, credible and traceable content generation for complex knowledge-intensive tasks. *arXiv preprint arXiv:2304.14732*, 2023.

[100] Jiazhan Feng, Chongyang Tao, Xiubo Geng, Tao Shen, Can Xu, Guodong Long, Dongyan Zhao, and Daxin Jiang. Knowledge refinement via interaction between search engines and large language models. *arXiv preprint arXiv:2305.07402*, 2023.

[101] Baolin Peng, Michel Galley, Pengcheng He, Hao Cheng, Yujia Xie, Yu Hu, Qiuyuan Huang, Lars Liden, Zhou Yu, Weizhu Chen, and Jianfeng Gao. Check your facts and try again: Improving large language models with external knowledge and automated feedback. *arXiv preprint arXiv:2302.12813*, 2023.

[102] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurélien Rodriguez, Armand

Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

[103] Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. Finetuned language models are zero-shot learners. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022.

[104] Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. ERNIE: enhanced language representation with informative entities. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 1441–1451. Association for Computational Linguistics, 2019.

[105] Ikuya Yamada, Akari Asai, Hiroyuki Shindo, Hideaki Takeda, and Yuji Matsumoto. LUKE: deep contextualized entity representations with entity-aware self-attention. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 6442–6454. Association for Computational Linguistics, 2020.

[106] Yujia Qin, Yankai Lin, Ryuichi Takanobu, Zhiyuan Liu, Peng Li, Heng Ji, Minlie Huang, Maosong Sun, and Jie Zhou. ERICA: improving entity and relation understanding for pre-trained language models via contrastive learning. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 3350–3363. Association for Computational Linguistics, 2021.

[107] Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George van den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, Diego de Las Casas, Aurelia Guy, Jacob Menick, Roman Ring, Tom Hennigan, Saffron Huang, Loren Maggiore, Chris Jones, Albin Cassirer, Andy Brock, Michela Paganini, Geoffrey Irving, Oriol Vinyals, Simon Osindero, Karen Simonyan, Jack W. Rae, Erich Elsen, and Laurent Sifre. Improving language models by retrieving from trillions of tokens. In *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pages 2206–2240. PMLR, 2022.

[108] Minki Kang, Jinheon Baek, and Sung Ju Hwang. KALA: knowledge-augmented language model adaptation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL 2022, Seattle, WA, United States, July 10-15, 2022*, pages 5144–5167. Association for Computational Linguistics, 2022.

[109] Daliang Li, Ankit Singh Rawat, Manzil Zaheer, Xin Wang, Michal Lukasik, Andreas Veit, Felix X. Yu, and Sanjiv Kumar. Large language models with controllable working memory. In *Findings of the Association for Computational Linguistics: ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 1774–1793. Association for Computational Linguistics, 2023.

[110] Gautier Izacard, Patrick Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. Atlas: Few-shot learning with retrieval augmented language models. *J. Mach. Learn. Res.*, 24:251:1–251:43, 2023.

[111] Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 10014–10037. Association for Computational Linguistics, 2023.

[112] Weijia Shi, Sewon Min, Michihiro Yasunaga, Minjoon Seo, Richard James, Mike Lewis, Luke Zettlemoyer, and Wen-tau Yih. REPLUG: retrieval-augmented black-box 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), NAACL 2024, Mexico City, Mexico, June 16-21, 2024*, pages 8371–8384. Association for Computational Linguistics, 2024.

[113] Rohan Anil, Andrew M. Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, Eric Chu, Jonathan H. Clark, Laurent El Shafey, Yanping Huang, Kathy Meier-Hellstern, Gaurav Mishra, Erica Moreira, Mark Omernick, Kevin Robinson, Sebastian Ruder, Yi Tay, Kefan Xiao, Yuanzhong Xu, Yujing Zhang, Gustavo Hernández Ábrego, Junwhan Ahn, Jacob Austin, Paul Barham, Jan A. Botha, James Bradbury, Siddhartha Brahma, Kevin Brooks, Michele Catasta, Yong Cheng, Colin Cherry, Christopher A. Choquette-Choo, Aakanksha Chowdhery, Clément Crepy, Shachi Dave, Mostafa Dehghani, Sunipa Dev, Jacob Devlin, Mark Díaz, Nan Du, Ethan Dyer, Vladimir Feinberg, Fangxiaoyu Feng, Vlad Fienber, Markus Freitag, Xavier Garcia, Sebastian Gehrmann, Lucas Gonzalez, and et al. Palm 2 technical report. *arXiv preprint arXiv:2305.10403*, 2023.

[114] Chunting Zhou, Junxian He, Xuezhe Ma, Taylor Berg-Kirkpatrick, and Graham Neubig. Prompt consistency for zero-shot task generalization. In *Findings of the Association for Computational Linguistics: EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 2613–2626. Association for Computational Linguistics, 2022.

[115] Angela Fan, Mike Lewis, and Yann N. Dauphin. Hierarchical neural story generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 889–898. Association for Computational Linguistics, 2018.

[116] Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. Latent retrieval for weakly supervised open domain question answering. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 6086–6096. Association for Computational Linguistics, 2019.

[117] Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur P. Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. Natural questions: a benchmark for question answering research. *Trans. Assoc. Comput. Linguistics*, 7:452–466, 2019.

[118] Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language*

Processing, Brussels, Belgium, October 31 - November 4, 2018, pages 2369–2380. Association for Computational Linguistics, 2018.

[119] Amir Saffari, Armin Oliya, Priyanka Sen, and Tom Ayoola. End-to-end entity resolution and question answering using differentiable knowledge graphs. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 4193–4200. Association for Computational Linguistics, 2021.

[120] Kurt Shuster, Spencer Poff, Moya Chen, Douwe Kiela, and Jason Weston. Retrieval augmentation reduces hallucination in conversation. In *Findings of the Association for Computational Linguistics: EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 16-20 November, 2021*, pages 3784–3803. Association for Computational Linguistics, 2021.

[121] Stephen E. Robertson, Steve Walker, Susan Jones, Micheline Hancock-Beaulieu, and Mike Gatford. Okapi at TREC-3. In *Proceedings of The Third Text REtrieval Conference, TREC 1994, Gaithersburg, Maryland, USA, November 2-4, 1994*, volume 500-225 of *NIST Special Publication*, pages 109–126. National Institute of Standards and Technology (NIST), 1994.

[122] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019.* OpenReview.net, 2019.

[123] OpenAI. GPT-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[124] Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, and Chunyuan Li. Llava-onevision: Easy visual task transfer. *Trans. Mach. Learn. Res.*, 2025, 2025.

[125] Wenliang Dai, Nayeon Lee, Boxin Wang, Zhuoling Yang, Zihan Liu, Jon Barker, Tuomas Rintamaki, Mohammad Shoeybi, Bryan Catanzaro, and Wei Ping. NVLM: open frontier-class multimodal llms. *arXiv preprint arXiv:2409.11402*, 2024.

[126] Ori Ram, Yoav Levine, Itay Dalmedigos, Dor Muhlgay, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. In-context retrieval-augmented language models. *Trans. Assoc. Comput. Linguistics*, 11:1316–1331, 2023.

[127] Cheng Niu, Yuanhao Wu, Juno Zhu, Siliang Xu, Kashun Shum, Randy Zhong, Juntong Song, and Tong Zhang. Ragtruth: A hallucination corpus for developing trustworthy retrieval-augmented language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 10862–10878. Association for Computational Linguistics, 2024.

[128] Orlando Ayala and Patrice Béchard. Reducing hallucination in structured outputs via retrieval-augmented generation. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Industry Track, NAACL 2024, Mexico City, Mexico, June 16-21, 2024*, pages 228–238. Association for Computational Linguistics, 2024.

[129] Shi Yu, Chaoyue Tang, Bokai Xu, Junbo Cui, Junhao Ran, Yukun Yan, Zhenghao Liu, Shuo Wang, Xu Han, Zhiyuan Liu, and Maosong Sun. Visrag: Vision-based retrieval-augmented generation on multi-modality documents. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025.

[130] Monica Riedler and Stefan Langer. Beyond text: Optimizing RAG with multimodal inputs for industrial applications. *arXiv preprint arXiv:2410.21943*, 2024.

[131] Yongdong Luo, Xiawu Zheng, Xiao Yang, Guilin Li, Haojia Lin, Jinfa Huang, Jiayi Ji, Fei Chao, Jiebo Luo, and Rongrong Ji. Video-rag: Visually-aligned retrieval-augmented long video comprehension. *arXiv preprint arXiv:2411.13093*, 2024.

[132] Ziyu Ma, Chenhui Gou, Hengcan Shi, Bin Sun, Shutao Li, Hamid Rezatofighi, and Jianfei Cai. Drvideo: Document retrieval based long video understanding. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2025, Nashville, TN, USA, June 11-15, 2025*, pages 18936–18946. Computer Vision Foundation / IEEE, 2025.

[133] Md. Adnan Arefeen, Biplob Debnath, Md. Yusuf Sarwar Uddin, and Srimat Chakradhar. irag: Advancing RAG for videos with an incremental approach. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management, CIKM 2024, Boise, ID, USA, October 21-25, 2024*, pages 4341–4348. ACM, 2024.

[134] Lu Zhang, Tiancheng Zhao, Heting Ying, Yibo Ma, and Kyusong Lee. Omagent: A multi-modal agent framework for complex video understanding with task divide-and-conquer. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, FL, USA, November 12-16, 2024*, pages 10031–10045. Association for Computational Linguistics, 2024.

[135] Valeria Bolotova-Baranova, Vladislav Blinov, Sofya Filippova, Falk Scholer, and Mark Sanderson. Wikihowqa: A comprehensive benchmark for multi-document non-factoid question answering. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 5291–5314. Association for Computational Linguistics, 2023.

[136] Antoine Miech, Dimitri Zhukov, Jean-Baptiste Alayrac, Makarand Tapaswi, Ivan Laptev, and Josef Sivic. Howto100m: Learning a text-video embedding by watching hundred million narrated video clips. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, pages 2630–2640. IEEE, 2019.

[137] Penghao Zhao, Hailin Zhang, Qinhan Yu, Zhengren Wang, Yunteng Geng, Fangcheng Fu, Ling Yang, Wentao Zhang, and Bin Cui. Retrieval-augmented generation for ai-generated content: A survey. *arXiv preprint arXiv:2402.19473*, 2024.

[138] Karen Spärck Jones. A statistical interpretation of term specificity and its application in retrieval. *J. Documentation*, 60(5):493–502, 2004.

[139] Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. Unsupervised dense information retrieval with contrastive learning. *Trans. Mach. Learn. Res.*, 2022, 2022.

[140] Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. Self-rag: Learning to retrieve, generate, and critique through self-reflection. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024.

[141] Taeho Hwang, Soyeong Jeong, Sukmin Cho, SeungYoon Han, and Jong Park. DSLR: Document refinement with sentence-level re-ranking and reconstruction to enhance retrieval-augmented generation. In *Proceedings of the 3rd Workshop on Knowledge Augmented Methods for NLP*, Bangkok, Thailand, August 2024. Association for Computational Linguistics.

[142] Qinyuan Cheng, Xiaonan Li, Shimin Li, Qin Zhu, Zhangyue Yin, Yunfan Shao, Linyang Li, Tianxiang Sun, Hang Yan, and Xipeng Qiu. Unified active retrieval for retrieval augmented generation. In *Findings of the Association for Computational Linguistics: EMNLP 2024, Miami, Florida, USA, November 12-16, 2024*, pages 17153–17166. Association for Computational Linguistics, 2024.

[143] Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 10014–10037. Association for Computational Linguistics, 2023.

[144] Jaewoo Lee, Joonho Ko, Jinheon Baek, Soyeong Jeong, and Sung Ju Hwang. Unified multi-modal interleaved document representation for information retrieval. *arXiv preprint arXiv:2410.02729*, 2024.

[145] Soyeong Jeong, Jinheon Baek, Sukmin Cho, Sung Ju Hwang, and Jong C. Park. Database-augmented query representation for information retrieval. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, Suzhou, China, November 2025. Association for Computational Linguistics.

[146] Manuel Faysse, Hugues Sibille, Tony Wu, Bilel Omrani, Gautier Viaud, Céline Hudelot, and Pierre Colombo. Colpali: Efficient document retrieval with vision language models. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025.

[147] Wenhu Chen, Hexiang Hu, Xi Chen, Pat Verga, and William W. Cohen. Murag: Multimodal retrieval-augmented generator for open question answering over images and text. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 5558–5570. Association for Computational Linguistics, 2022.

[148] Weizhe Lin and Bill Byrne. Retrieval augmented visual question answering with outside knowledge. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 11238–11254. Association for Computational Linguistics, 2022.

[149] Yucan Guo, Zixuan Li, Xiaolong Jin, Yantao Liu, Yutao Zeng, Wenxuan Liu, Xiang Li, Pan Yang, Long Bai, Jiafeng Guo, and Xueqi Cheng. Retrieval-augmented code generation for universal information extraction. In *Natural Language Processing and Chinese Computing - 13th National*

CCF Conference, NLPCC 2024, Hangzhou, China, November 1-3, 2024, Proceedings, Part II, volume 15360 of *Lecture Notes in Computer Science*, pages 30–42. Springer, 2024.

[150] Feifei Pan, Mustafa Canim, Michael R. Glass, Alfio Gliozzo, and James A. Hendler. End-to-end table question answering via retrieval-augmented generation. *arXiv preprint arXiv:2203.16714*, 2022.

[151] Asim Biswal, Liana Patel, Siddarth Jha, Amog Kamsetty, Shu Liu, Joseph E. Gonzalez, Carlos Guestrin, and Matei Zaharia. Text2sql is not enough: Unifying AI and databases with TAG. *arXiv preprint arXiv:2408.14717*, 2024.

[152] Yi Yuan, Haohe Liu, Xubo Liu, Qiushi Huang, Mark D. Plumbley, and Wenwu Wang. Retrieval-augmented text-to-audio generation. In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2024, Seoul, Republic of Korea, April 14-19, 2024*, pages 581–585. IEEE, 2024.

[153] Sukmin Cho, Sangjin Choi, Taeho Hwang, Jeongyeon Seo, Soyeong Jeong, Huije Lee, Hoyun Song, Jong C. Park, and Youngjin Kwon. Lossless acceleration of large language models with hierarchical drafting based on temporal locality in speculative decoding. In *Findings of the Association for Computational Linguistics: NAACL 2025, Albuquerque, New Mexico, USA, April 29 - May 4, 2025*, pages 3895–3911. Association for Computational Linguistics, 2025.

[154] Hoyun Song, Huije Lee, Jisu Shin, Sukmin Cho, Changgeon Ko, and Jong C. Park. Does rationale quality matter? enhancing mental disorder detection via selective reasoning distillation. In *Findings of the Association for Computational Linguistics, ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, pages 21738–21756. Association for Computational Linguistics, 2025.

[155] Sheng-Chieh Lin, Chankyu Lee, Mohammad Shoeybi, Jimmy Lin, Bryan Catanzaro, and Wei Ping. Mm-embed: Universal multimodal retrieval with multimodal LLMS. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025.* OpenReview.net, 2025.

[156] Florian Bordes, Richard Yuanzhe Pang, Anurag Ajay, Alexander C. Li, Adrien Bardes, Suzanne Petryk, Oscar Mañas, Zhiqiu Lin, Anas Mahmoud, Bargav Jayaraman, Mark Ibrahim, Melissa Hall, Yunyang Xiong, Jonathan Lebensold, Candace Ross, Srihari Jayakumar, Chuan Guo, Diane Bouchacourt, Haider Al-Tahan, Karthik Padthe, Vasu Sharma, Hu Xu, Xiaoqing Ellen Tan, Megan Richards, Samuel Lavoie, Pietro Astolfi, Reyhane Askari Hemmat, Jun Chen, Kushal Tirumala, Rim Assouel, Mazda Moayeri, Arjang Talattof, Kamalika Chaudhuri, Zechun Liu, Xilun Chen, Quentin Garrido, Karen Ullrich, Aishwarya Agrawal, Kate Saenko, Asli Celikyilmaz, and Vikas Chandra. An introduction to vision-language modeling. *arXiv preprint arXiv:2405.17247*, 2024.

[157] Beier Zhu and Hanwang Zhang. Debiasing vision-language models for vision tasks: a survey. *Frontiers Comput. Sci.*, 19(1):191321, 2025.

[158] DeepSeek-AI, Qihao Zhu, Daya Guo, Zhihong Shao, Dejian Yang, Peiyi Wang, Runxin Xu, Y. Wu, Yukun Li, Huazuo Gao, Shirong Ma, Wangding Zeng, Xiao Bi, Zihui Gu, Hanwei Xu, Damai Dai, Kai Dong, Liyue Zhang, Yishi Piao, Zhibin Gou, Zhenda Xie, Zhewen Hao, Bingxuan Wang, Junxiao Song, Deli Chen, Xin Xie, Kang Guan, Yuxiang You, Aixin Liu, Qiushi Du, Wenjun Gao, Xuan Lu, Qinyu Chen, Yaohui Wang, Chengqi Deng, Jiashi Li, Chenggang Zhao, Chong Ruan,

Fuli Luo, and Wenfeng Liang. Deepseek-coder-v2: Breaking the barrier of closed-source models in code intelligence. *arXiv preprint arXiv:2406.11931*, 2024.

[159] Binyuan Hui, Jian Yang, Zeyu Cui, Jiaxi Yang, Dayiheng Liu, Lei Zhang, Tianyu Liu, Jiajun Zhang, Bowen Yu, Kai Dang, An Yang, Rui Men, Fei Huang, Xingzhang Ren, Xuancheng Ren, Jingren Zhou, and Junyang Lin. Qwen2.5-coder technical report. *arXiv preprint arXiv:2409.12186*, 2024.

[160] Yunlong Tang, Jing Bi, Siting Xu, Luchuan Song, Susan Liang, Teng Wang, Daoan Zhang, Jie An, Jingyang Lin, Rongyi Zhu, Ali Vosoughi, Chao Huang, Zeliang Zhang, Feng Zheng, Jianguo Zhang, Ping Luo, Jiebo Luo, and Chenliang Xu. Video understanding with large language models: A survey. *arXiv preprint arXiv:2312.17432*, 2023.

[161] Han Wang, Yongjie Ye, Yanjie Wang, Yuxiang Nie, and Can Huang. Elysium: Exploring object-level perception in videos via MLLM. In *Computer Vision - ECCV 2024 - 18th European Conference, Milan, Italy, September 29-October 4, 2024, Proceedings, Part XXII*, volume 15080 of *Lecture Notes in Computer Science*, pages 166–185. Springer, 2024.

[162] Muhammad Maaz, Hanoona Abdul Rasheed, Salman Khan, and Fahad Khan. Video-chatgpt: Towards detailed video understanding via large vision and language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 12585–12602. Association for Computational Linguistics, 2024.

[163] Chaoyi Zhang, Kevin Lin, Zhengyuan Yang, Jianfeng Wang, Linjie Li, Chung-Ching Lin, Zicheng Liu, and Lijuan Wang. Mm-narrator: Narrating long-form videos with multimodal in-context learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*, pages 13647–13657. IEEE, 2024.

[164] Bo He, Hengduo Li, Young Kyun Jang, Menglin Jia, Xuefei Cao, Ashish Shah, Abhinav Shrivastava, and Ser-Nam Lim. MA-LMM: memory-augmented large multimodal model for long-term video understanding. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*, pages 13504–13514. IEEE, 2024.

[165] Junke Wang, Dongdong Chen, Chong Luo, Bo He, Lu Yuan, Zuxuan Wu, and Yu-Gang Jiang. Omnivid: A generative framework for universal video understanding. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*, pages 18209–18220. IEEE, 2024.

[166] Eui Jun Hwang, Sukmin Cho, Junmyeong Lee, and Jong C. Park. An efficient gloss-free sign language translation using spatial configurations and motion dynamics with llms. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL 2025 - Volume 1: Long Papers, Albuquerque, New Mexico, USA, April 29 - May 4, 2025*, pages 3901–3920. Association for Computational Linguistics, 2025.

[167] Joya Chen, Zhaoyang Lv, Shiwei Wu, Kevin Qinghong Lin, Chenan Song, Difei Gao, Jia-Wei Liu, Ziteng Gao, Dongxing Mao, and Mike Zheng Shou. Videollm-online: Online video large

language model for streaming video. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*, pages 18407–18418. IEEE, 2024.

[168] Kangsan Kim, Geon Park, Youngwan Lee, Woongyeong Yeo, and Sung Ju Hwang. Videoicl: Confidence-based iterative in-context learning for out-of-distribution video understanding. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2025, Nashville, TN, USA, June 11-15, 2025*, pages 3295–3305. Computer Vision Foundation / IEEE, 2025.

[169] Hu Xu, Gargi Ghosh, Po-Yao Huang, Dmytro Okhonko, Armen Aghajanyan, Florian Metze, Luke Zettlemoyer, and Christoph Feichtenhofer. Videoclip: Contrastive pre-training for zero-shot video-text understanding. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 6787–6800. Association for Computational Linguistics, 2021.

[170] Yuanhan Zhang, Bo Li, haotian Liu, Yong jae Lee, Liangke Gui, Di Fu, Jiashi Feng, Ziwei Liu, and Chunyuan Li. Llava-next: A strong zero-shot video understanding model, April 2024.

[171] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR, 2021.

[172] Michihiro Yasunaga, Armen Aghajanyan, Weijia Shi, Richard James, Jure Leskovec, Percy Liang, Mike Lewis, Luke Zettlemoyer, and Wen-Tau Yih. Retrieval-augmented multimodal language modeling. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 39755–39769. PMLR, 2023.

[173] Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, Barcelona, Spain, July 2004. Association for Computational Linguistics.

[174] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, July 6-12, 2002, Philadelphia, PA, USA*, pages 311–318. ACL, 2002.

[175] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with BERT. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020.* OpenReview.net, 2020.

[176] Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. G-eval: NLG evaluation using gpt-4 with better human alignment. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 2511–2522. Association for Computational Linguistics, 2023.

[177] Zhe Chen, Weiyun Wang, Yue Cao, Yangzhou Liu, Zhangwei Gao, Erfei Cui, Jinguo Zhu, Shenglong Ye, Hao Tian, Zhaoyang Liu, Lixin Gu, Xuehui Wang, Qingyun Li, Yimin Ren, Zixuan Chen, Jiapeng Luo, Jiahao Wang, Tan Jiang, Bo Wang, Conghui He, Botian Shi, Xingcheng Zhang, Han Lv, Yi Wang, Wenqi Shao, Pei Chu, Zhongying Tu, Tong He, Zhiyong Wu, Huipeng Deng, Jiaye

Ge, Kai Chen, Min Dou, Lewei Lu, Xizhou Zhu, Tong Lu, Dahua Lin, Yu Qiao, Jifeng Dai, and Wenhai Wang. Expanding performance boundaries of open-source multimodal models with model, data, and test-time scaling. *arXiv preprint arXiv:2412.05271*, 2024.

[178] Yi Wang, Kunchang Li, Xinhao Li, Jiashuo Yu, Yinan He, Guo Chen, Baoqi Pei, Rongkun Zheng, Zun Wang, Yansong Shi, Tianxiang Jiang, Songze Li, Jilan Xu, Hongjie Zhang, Yifei Huang, Yu Qiao, Yali Wang, and Limin Wang. Internvideo2: Scaling foundation models for multimodal video understanding. In *Computer Vision - ECCV 2024 - 18th European Conference, Milan, Italy, September 29-October 4, 2024, Proceedings, Part LXXXV*, volume 15143 of *Lecture Notes in Computer Science*, pages 396–416. Springer, 2024.

[179] Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. Robust speech recognition via large-scale weak supervision. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 28492–28518. PMLR, 2023.

[180] Sanket Shah, Anand Mishra, Naganand Yadati, and Partha Pratim Talukdar. KVQA: knowledge-aware visual question answering. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 8876–8884. AAAI Press, 2019.

[181] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

[182] Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. Retrieval augmented language model pre-training. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 3929–3938. PMLR, 2020.

[183] Stephen E. Robertson, Steve Walker, Susan Jones, Micheline Hancock-Beaulieu, and Mike Gatford. Okapi at TREC-3. In *Proceedings of The Third Text REtrieval Conference, TREC 1994, Gaithersburg, Maryland, USA, November 2-4, 1994*, volume 500-225 of *NIST Special Publication*, pages 109–126. National Institute of Standards and Technology (NIST), 1994.

[184] Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 6769–6781. Association for Computational Linguistics, 2020.

[185] Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul N. Bennett, Junaid Ahmed, and Arnold Overwijk. Approximate nearest neighbor negative contrastive learning for dense text retrieval. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021.

[186] Lei Li, Yuqi Wang, Runxin Xu, Peiyi Wang, Xiachong Feng, Lingpeng Kong, and Qi Liu. Multimodal arxiv: A dataset for improving scientific comprehension of large vision-language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 14369–14387. Association for Computational Linguistics, 2024.

[187] Kuicai Dong, Derrick-Goh-Xin Deik, Yi Lee, Hao Zhang, Xiangyang Li, Cong Zhang, and Yong Liu. Mc-indexing: Effective long document retrieval via multi-view content-aware indexing. In *Findings of the Association for Computational Linguistics: EMNLP 2024, Miami, Florida, USA, November 12-16, 2024*, pages 2673–2691. Association for Computational Linguistics, 2024.

[188] Ziyan Jiang, Xueguang Ma, and Wenhu Chen. Longrag: Enhancing retrieval-augmented generation with long-context llms. *arXiv preprint arXiv:2406.15319*, 2024.

[189] Manuel Faysse, Hugues Sibille, Tony Wu, Bilel Omrani, Gautier Viaud, Céline Hudelot, and Pierre Colombo. Colpali: Efficient document retrieval with vision language models. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025.

[190] Xueguang Ma, Sheng-Chieh Lin, Minghan Li, Wenhu Chen, and Jimmy Lin. Unifying multimodal retrieval via document screenshot embedding. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, FL, USA, November 12-16, 2024*, pages 6492–6505. Association for Computational Linguistics, 2024.

[191] Yutao Zhu, Huaying Yuan, Shuting Wang, Jiongnan Liu, Wenhan Liu, Chenlong Deng, Zhicheng Dou, and Ji-Rong Wen. Large language models for information retrieval: A survey. *arXiv preprint arXiv:2308.07107*, 2023.

[192] Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Qianyu Guo, Meng Wang, and Haofen Wang. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*, 2023.

[193] Ori Ram, Yoav Levine, Itay Dalmedigos, Dor Muhlgay, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. In-context retrieval-augmented language models. *Trans. Assoc. Comput. Linguistics*, 11:1316–1331, 2023.

[194] Weijia Shi, Sewon Min, Michihiro Yasunaga, Minjoon Seo, Richard James, Mike Lewis, Luke Zettlemoyer, and Wen-tau Yih. REPLUG: retrieval-augmented black-box 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), NAACL 2024, Mexico City, Mexico, June 16-21, 2024*, pages 8371–8384. Association for Computational Linguistics, 2024.

[195] Soyeong Jeong, Jinheon Baek, Sukmin Cho, Sung Ju Hwang, and Jong Park. Adaptive-rag: Learning to adapt retrieval-augmented large language models through question complexity. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), NAACL 2024, Mexico City, Mexico, June 16-21, 2024*, pages 7036–7050. Association for Computational Linguistics, 2024.

[196] Zheyuan Liu, Cristian Rodriguez Opazo, Damien Teney, and Stephen Gould. Image retrieval on real-life images with pre-trained vision-and-language models. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, pages 2105–2114. IEEE, 2021.

[197] Soyeong Jeong, Jinheon Baek, Sukmin Cho, Sung Ju Hwang, and Jong C. Park. Database-augmented query representation for information retrieval. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, Suzhou, China, November 2025. Association for Computational Linguistics.

[198] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR, 2021.

[199] Zilin Xiao, Ming Gong, Paola Cascante-Bonilla, Xingyao Zhang, Jie Wu, and Vicente Ordonez. Grounding language models for visual entity recognition. In *Computer Vision - ECCV 2024 - 18th European Conference, Milan, Italy, September 29-October 4, 2024, Proceedings, Part XI*, volume 15069 of *Lecture Notes in Computer Science*, pages 393–411. Springer, 2024.

[200] Jonathan Herzig, Thomas Müller, Syrine Krichene, and Julian Martin Eisenschlos. Open domain question answering over tables via dense retrieval. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 512–519. Association for Computational Linguistics, 2021.

[201] Peter Baile Chen, Yi Zhang, and Dan Roth. Is table retrieval a solved problem? exploring join-aware multi-table retrieval. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 2687–2699. Association for Computational Linguistics, 2024.

[202] Junjie Zhou, Zheng Liu, Shitao Xiao, Bo Zhao, and Yongping Xiong. VISTA: visualized text embedding for universal multi-modal retrieval. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 3185–3200. Association for Computational Linguistics, 2024.

[203] Xinwei Long, Jiali Zeng, Fandong Meng, Zhiyuan Ma, Kaiyan Zhang, Bowen Zhou, and Jie Zhou. Generative multi-modal knowledge retrieval with large language models. In *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2014, February 20-27, 2024, Vancouver, Canada*, pages 18733–18741. AAAI Press, 2024.

[204] Paul Lerner, Olivier Ferret, and Camille Guinaudeau. Cross-modal retrieval for knowledge-based visual question answering. In *Advances in Information Retrieval - 46th European Conference on Information Retrieval, ECIR 2024, Glasgow, UK, March 24-28, 2024, Proceedings, Part I*, volume 14608 of *Lecture Notes in Computer Science*, pages 421–438. Springer, 2024.

[205] Averi Nowak, Francesco Piccinno, and Yasemin Altun. Multimodal chart retrieval: A comparison of text, table and image based approaches. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), NAACL 2024, Mexico City, Mexico, June 16-21, 2024*, pages 5488–5505. Association for Computational Linguistics, 2024.

[206] Davide Caffagni, Federico Cocchi, Nicholas Moratelli, Sara Sarto, Marcella Cornia, Lorenzo Baraldi, and Rita Cucchiara. Wiki-llava: Hierarchical retrieval-augmented generation for multimodal llms. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024 - Workshops, Seattle, WA, USA, June 17-18, 2024*, pages 1818–1826. IEEE, 2024.

[207] Jaemin Cho, Debanjan Mahata, Ozan Irsoy, Yujie He, and Mohit Bansal. M3docrag: Multimodal retrieval is what you need for multi-page multi-document understanding. *arXiv preprint arXiv:2411.04952*, 2024.

[208] Yihao Ding, Kaixuan Ren, Jiabin Huang, Siwen Luo, and Soyeon Caren Han. PDF-MVQA: A dataset for multimodal information retrieval in pdf-based visual question answering. *arXiv preprint arXiv:2404.12720*, 2024.

[209] Ziyan Jiang, Rui Meng, Xinyi Yang, Semih Yavuz, Yingbo Zhou, and Wenhu Chen. Vlm2vec: Training vision-language models for massive multimodal embedding tasks. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025.

[210] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023*, pages 11941–11952. IEEE, 2023.

[211] Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven C. H. Hoi. Instructblip: Towards general-purpose vision-language models with instruction tuning. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023.

[212] Pan Zhang, Xiaoyi Dong, Bin Wang, Yuhang Cao, Chao Xu, Linke Ouyang, Zhiyuan Zhao, Shuangrui Ding, Songyang Zhang, Haodong Duan, Wenwei Zhang, Hang Yan, Xinyue Zhang, Wei Li, Jingwen Li, Kai Chen, Conghui He, Xingcheng Zhang, Yu Qiao, Dahua Lin, and Jiaqi Wang. Internlm-xcomposer: A vision-language large model for advanced text-image comprehension and composition. *arXiv preprint arXiv:2309.15112*, 2023.

[213] Feng Li, Renrui Zhang, Hao Zhang, Yuanhan Zhang, Bo Li, Wei Li, Zejun Ma, and Chunyuan Li. Llava-next-interleave: Tackling multi-image, video, and 3d in large multimodal models. *arXiv preprint arXiv:2407.07895*, 2024.

[214] Thomas Mensink, Jasper R. R. Uijlings, Lluís Castrejón, Arushi Goel, Felipe Cadar, Howard Zhou, Fei Sha, André Araújo, and Vittorio Ferrari. Encyclopedic VQA: visual questions about detailed properties of fine-grained categories. In *IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023*, pages 3090–3101. IEEE, 2023.

[215] Yang Chen, Hexiang Hu, Yi Luan, Haitian Sun, Soravit Changpinyo, Alan Ritter, and Ming-Wei Chang. Can pre-trained vision and language models answer visual information-seeking questions? In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 14948–14968. Association for Computational Linguistics, 2023.

[216] Paul Lerner, Olivier Ferret, Camille Guinaudeau, Hervé Le Borgne, Romaric Besançon, José G. Moreno, and Jesús Lovón-Melgarejo. Viquae, a dataset for knowledge-based visual question answering about named entities. In *SIGIR '22: The 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, Madrid, Spain, July 11 - 15, 2022*, pages 3108–3120. ACM, 2022.

[217] Sunjun Kweon, Yeonsu Kwon, Seonhee Cho, Yohan Jo, and Edward Choi. Open-wikitable : Dataset for open domain question answering with complex reasoning over table. In *Findings of the Association for Computational Linguistics: ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 8285–8297. Association for Computational Linguistics, 2023.

[218] Victor Zhong, Caiming Xiong, and Richard Socher. Seq2sql: Generating structured queries from natural language using reinforcement learning. *arXiv preprint arXiv:1709.00103*, 2017.

[219] Panupong Pasupat and Percy Liang. Compositional semantic parsing on semi-structured tables. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers*, pages 1470–1480. The Association for Computer Linguistics, 2015.

[220] Junnan Li, Dongxu Li, Caiming Xiong, and Steven C. H. Hoi. BLIP: bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pages 12888–12900. PMLR, 2022.

[221] Cong Wei, Yang Chen, Haonan Chen, Hexiang Hu, Ge Zhang, Jie Fu, Alan Ritter, and Wenhu Chen. Uniir: Training and benchmarking universal multimodal information retrievers. In *Computer Vision - ECCV 2024 - 18th European Conference, Milan, Italy, September 29-October 4, 2024, Proceedings, Part LXXXVII*, volume 15145 of *Lecture Notes in Computer Science*, pages 387–404. Springer, 2024.

[222] Kexin Wang, Nils Reimers, and Iryna Gurevych. DAPR: A benchmark on document-aware passage retrieval. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 4313–4330. Association for Computational Linguistics, 2024.

[223] Ting Jiang, Minghui Song, Zihan Zhang, Haizhen Huang, Weiwei Deng, Feng Sun, Qi Zhang, Deqing Wang, and Fuzhen Zhuang. E5-V: universal embeddings with multimodal large language models. *arXiv preprint arXiv:2407.12580*, 2024.

[224] Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, and Chunyuan Li. Llava-onevision: Easy visual task transfer. *Trans. Mach. Learn. Res.*, 2025, 2025.

[225] Jialiang Xu, Michael Moor, and Jure Leskovec. Reverse image retrieval cues parametric memory in multimodal llms. *arXiv preprint arXiv:2405.18740*, 2024.

[226] Ziyan Jiang, Xueguang Ma, and Wenhu Chen. Longrag: Enhancing retrieval-augmented generation with long-context llms. *arXiv preprint arXiv:2406.15319*, 2024.

[227] Jinhyuk Lee, Anthony Chen, Zhuyun Dai, Dheeru Dua, Devendra Singh Sachan, Michael Boratko, Yi Luan, Sébastien M. R. Arnold, Vincent Perot, Siddharth Dalmia, Hexiang Hu, Xudong Lin, Panupong Pasupat, Aida Amini, Jeremy R. Cole, Sebastian Riedel, Iftekhar Naim, Ming-Wei Chang, and Kelvin Guu. Can long-context language models subsume retrieval, rag, sql, and more? *arXiv preprint arXiv:2406.13121*, 2024.

[228] Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N. Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick van Kleef, Sören Auer, and Christian Bizer. Dbpedia - A large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web*, 6(2): 167–195, 2015.

[229] Barlas Oguz, Xilun Chen, Vladimir Karpukhin, Stan Peshterliev, Dmytro Okhonko, Michael Sejr Schlichtkrull, Sonal Gupta, Yashar Mehdad, and Scott Yih. Unik-qa: Unified representations of structured and unstructured knowledge for open-domain question answering. In *Findings of the Association for Computational Linguistics: NAACL 2022, Seattle, WA, United States, July 10-15, 2022*, pages 1535–1546. Association for Computational Linguistics, 2022.

[230] Kaixin Ma, Hao Cheng, Xiaodong Liu, Eric Nyberg, and Jianfeng Gao. Open domain question answering with A unified knowledge interface. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 1605–1620. Association for Computational Linguistics, 2022.

[231] Fabian Galetzka, Jewgeni Rose, David Schlangen, and Jens Lehmann. Space efficient context encoding for non-task-oriented dialogue generation with graph attention transformer. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 7028–7041. Association for Computational Linguistics, 2021.

[232] Bin Fu, Yunqi Qiu, Chengguang Tang, Yang Li, Haiyang Yu, and Jian Sun. A survey on complex question answering over knowledge base: Recent advances and challenges. *arXiv preprint arXiv:2007.13069*, 2020.

[233] Yunshi Lan, Gaole He, Jinhao Jiang, Jing Jiang, Wayne Xin Zhao, and Ji-Rong Wen. A survey on complex knowledge base question answering: Methods, challenges and solutions. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI 2021, Virtual Event / Montreal, Canada, 19-27 August 2021*, pages 4483–4491. ijcai.org, 2021.

[234] Zhiguo Wang, Patrick Ng, Ramesh Nallapati, and Bing Xiang. Retrieval, re-ranking and multi-task learning for knowledge-base question answering. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, EACL 2021, Online, April 19 - 23, 2021*, pages 347–357. Association for Computational Linguistics, 2021.

[235] Kuldeep Singh, Ioanna Lytra, Arun Sethupat Radhakrishna, Saeedeh Shekarpour, Maria-Esther Vidal, and Jens Lehmann. No one is perfect: Analysing the performance of question answering components over the dbpedia knowledge graph. *J. Web Semant.*, 65:100594, 2020.

[236] Namgi Han, Goran Topic, Hiroshi Noji, Hiroya Takamura, and Yusuke Miyao. An empirical analysis of existing systems and datasets toward general simple question answering. In *Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020, Barcelona, Spain (Online), December 8-13, 2020*, pages 5321–5334. International Committee on Computational Linguistics, 2020.

[237] Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with gpus. *IEEE Trans. Big Data*, 7(3):535–547, 2021.

[238] Phillip Schneider, Tim Schopf, Juraj Vladika, Michael Galkin, Elena Simperl, and Florian Matthes. A decade of knowledge graphs in natural language processing: A survey. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing, AACL/IJCNLP 2022 - Volume 1: Long Papers, Online Only, November 20-23, 2022*, pages 601–614. Association for Computational Linguistics, 2022.

[239] Denis Lukovnikov, Asja Fischer, Jens Lehmann, and Sören Auer. Neural network-based question answering over knowledge graphs on word and character level. In *Proceedings of the 26th International Conference on World Wide Web, WWW 2017, Perth, Australia, April 3-7, 2017*, pages 1211–1220. ACM, 2017.

[240] Nilesh Chakraborty, Denis Lukovnikov, Gaurav Maheshwari, Priyansh Trivedi, Jens Lehmann, and Asja Fischer. Introduction to neural network-based question answering over knowledge graphs. *WIREs Data Mining Knowl. Discov.*, 11(3), 2021.

[241] Stephen E. Robertson and Hugo Zaragoza. The probabilistic relevance framework: BM25 and beyond. *Found. Trends Inf. Retr.*, 3(4):333–389, 2009.

[242] Rodrigo Nogueira, Wei Yang, Jimmy Lin, and Kyunghyun Cho. Document expansion by query prediction. *arXiv preprint arXiv:1904.08375*, 2019.

[243] Soyeong Jeong, Jinheon Baek, ChaeHun Park, and Jong Park. Unsupervised document expansion for information retrieval with stochastic text generation. In *Proceedings of the Second Workshop on Scholarly Document Processing*, Online, June 2021. Association for Computational Linguistics.

[244] Yingqi Qu, Yuchen Ding, Jing Liu, Kai Liu, Ruiyang Ren, Wayne Xin Zhao, Daxiang Dong, Hua Wu, and Haifeng Wang. Rocketqa: An optimized training approach to dense passage retrieval for open-domain question answering. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 5835–5847. Association for Computational Linguistics, 2021.

[245] Li Dong and Mirella Lapata. Language to logical form with neural attention. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. The Association for Computer Linguistics, 2016.

[246] Percy Liang. Lambda dependency-based compositional semantics. *arXiv preprint arXiv:1309.4408*, 2013.

[247] Antoine Bordes, Sumit Chopra, and Jason Weston. Question answering with subgraph embeddings. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 615–620. ACL, 2014.

[248] Yanchao Hao, Yuanzhe Zhang, Kang Liu, Shizhu He, Zhanyi Liu, Hua Wu, and Jun Zhao. An end-to-end model for question answering over knowledge base with cross-attention combining global knowledge. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 221–231. Association for Computational Linguistics, 2017.

[249] Salman Mohammed, Peng Shi, and Jimmy Lin. Strong baselines for simple question answering over knowledge graphs with and without neural networks. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 2 (Short Papers)*, pages 291–296. Association for Computational Linguistics, 2018.

[250] Yu Chen, Lingfei Wu, and Mohammed J. Zaki. Bidirectional attentive memory networks for question answering over knowledge bases. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 2913–2923. Association for Computational Linguistics, 2019.

[251] Antoine Bordes, Nicolas Usunier, Sumit Chopra, and Jason Weston. Large-scale simple question answering with memory networks. *arXiv preprint arXiv:1506.02075*, 2015.

[252] Seungwhan Moon, Pararth Shah, Anuj Kumar, and Rajen Subba. Opendialkg: Explainable conversational reasoning with attention-based walks over knowledge graphs. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28-August 2, 2019, Volume 1: Long Papers*, pages 845–854. Association for Computational Linguistics, 2019.

[253] Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. spacy: Industrial-strength natural language processing in python. 2020.

[254] Nicola De Cao, Gautier Izacard, Sebastian Riedel, and Fabio Petroni. Autoregressive entity retrieval. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021.

[255] Ledell Wu, Fabio Petroni, Martin Josifoski, Sebastian Riedel, and Luke Zettlemoyer. Scalable zero-shot entity linking with dense entity retrieval. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 6397–6407. Association for Computational Linguistics, 2020.

[256] Belinda Z. Li, Sewon Min, Srinivasan Iyer, Yashar Mehdad, and Wen-tau Yih. Efficient one-pass end-to-end entity linking for questions. In *Proceedings of the 2020 Conference on Empirical Methods*

in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020, pages 6433–6441. Association for Computational Linguistics, 2020.

[257] Yu Gu, Sue Kase, Michelle Vanni, Brian M. Sadler, Percy Liang, Xifeng Yan, and Yu Su. Beyond I.I.D.: three levels of generalization for question answering on knowledge bases. In *WWW '21: The Web Conference 2021, Virtual Event / Ljubljana, Slovenia, April 19-23, 2021*, pages 3477–3488. ACM / IW3C2, 2021.

[258] Soyeong Jeong, Jinheon Baek, Sukmin Cho, Sung Ju Hwang, and Jong C. Park. Augmenting document representations for dense retrieval with interpolation and perturbation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 442–452. Association for Computational Linguistics, 2022.

[259] Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. MS MARCO: A human generated machine reading comprehension dataset. In *Proceedings of the Workshop on Cognitive Computation: Integrating neural and symbolic approaches 2016 co-located with the 30th Annual Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain, December 9, 2016*, volume 1773 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2016.

[260] Tao Yu, Rui Zhang, Kai Yang, Michihiro Yasunaga, Dongxu Wang, Zifan Li, James Ma, Irene Li, Qingning Yao, Shanelle Roman, Zilin Zhang, and Dragomir R. Radev. Spider: A large-scale human-labeled dataset for complex and cross-domain semantic parsing and text-to-sql task. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 3911–3921. Association for Computational Linguistics, 2018.

[261] Shulin Cao, Jiaxin Shi, Liangming Pan, Lunyiu Nie, Yutong Xiang, Lei Hou, Juanzi Li, Bin He, and Hanwang Zhang. KQA pro: A dataset with explicit compositional programs for complex question answering over knowledge base. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 6101–6119. Association for Computational Linguistics, 2022.

[262] Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. BEIR: A heterogeneous benchmark for zero-shot evaluation of information retrieval models. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual*, 2021.

[263] Nitarshan Rajkumar, Raymond Li, and Dzmitry Bahdanau. Evaluating the text-to-sql capabilities of large language models. *arXiv preprint arXiv:2204.00498*, 2022.

[264] Dawei Gao, Haibin Wang, Yaliang Li, Xiuyu Sun, Yichen Qian, Bolin Ding, and Jingren Zhou. Text-to-sql empowered by large language models: A benchmark evaluation. *Proc. VLDB Endow.*, 17(5):1132–1145, 2024.

[265] Andrew Drozdov, Nathanael Schärli, Ekin Akyürek, Nathan Scales, Xinying Song, Xinyun Chen, Olivier Bousquet, and Denny Zhou. Compositional semantic parsing with large language models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023.

[266] Phillip Schneider, Manuel Klettner, Kristiina Jokinen, Elena Simperl, and Florian Matthes. Evaluating large language models in semantic parsing for conversational question answering over knowledge graphs. In *Proceedings of the 16th International Conference on Agents and Artificial Intelligence, ICAART 2024, Volume 3, Rome, Italy, February 24-26, 2024*, pages 807–814. SCITEPRESS, 2024.

[267] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

[268] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurélien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

[269] Adam Roberts, Colin Raffel, and Noam Shazeer. How much knowledge can you pack into the parameters of a language model? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 5418–5426. Association for Computational Linguistics, 2020.

[270] Alessandro Sordoni, Yoshua Bengio, Hossein Vahabi, Christina Lioma, Jakob Grue Simonsen, and Jian-Yun Nie. A hierarchical recurrent encoder-decoder for generative context-aware query suggestion. In *Proceedings of the 24th ACM International Conference on Information and Knowledge Management, CIKM 2015, Melbourne, VIC, Australia, October 19 - 23, 2015*, pages 553–562. ACM, 2015.

[271] Mostafa Dehghani, Sascha Rothe, Enrique Alfonseca, and Pascal Fleury. Learning to attend, copy, and generate for session-based query suggestion. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, CIKM 2017, Singapore, November 06 - 10, 2017*, pages 1747–1756. ACM, 2017.

[272] Likang Wu, Zhi Zheng, Zhaopeng Qiu, Hao Wang, Hongchao Gu, Tingjia Shen, Chuan Qin, Chen Zhu, Hengshu Zhu, Qi Liu, Hui Xiong, and Enhong Chen. A survey on large language models for recommendation. *World Wide Web (WWW)*, 27(5):60, 2024.

[273] Zihuai Zhao, Wenqi Fan, Jiatong Li, Yunqing Liu, Xiaowei Mei, Yiqi Wang, Zhen Wen, Fei Wang, Xiangyu Zhao, Jiliang Tang, and Qing Li. Recommender systems in the era of large language models (llms). *IEEE Trans. Knowl. Data Eng.*, 36(11):6889–6907, 2024.

[274] Wei-Fan Chen, Shahbaz Syed, Benno Stein, Matthias Hagen, and Martin Potthast. Abstractive snippet generation. In *WWW '20: The Web Conference 2020, Taipei, Taiwan, April 20-24, 2020*, pages 1309–1319. ACM / IW3C2, 2020.

[275] Chao-Chun Hsu, Eric Lind, Luca Soldaini, and Alessandro Moschitti. Answer generation for retrieval-based question answering systems. In *Findings of the Association for Computational Linguistics: ACL/IJCNLP 2021, Online Event, August 1-6, 2021*, volume ACL/IJCNLP 2021 of *Findings of ACL*, pages 4276–4282. Association for Computational Linguistics, 2021.

[276] Shijie Geng, Shuchang Liu, Zuohui Fu, Yingqiang Ge, and Yongfeng Zhang. Recommendation as language processing (RLP): A unified pretrain, personalized prompt & predict paradigm (P5). In *RecSys '22: Sixteenth ACM Conference on Recommender Systems, Seattle, WA, USA, September 18 - 23, 2022*, pages 299–315. ACM, 2022.

[277] Wang-Cheng Kang, Jianmo Ni, Nikhil Mehta, Maheswaran Sathiamoorthy, Lichan Hong, Ed H. Chi, and Derek Zhiyuan Cheng. Do llms understand user preferences? evaluating llms on user rating prediction. *arXiv preprint arXiv:2305.06474*, 2023.

[278] Jianchao Ji, Zelong Li, Shuyuan Xu, Wenyue Hua, Yingqiang Ge, Juntao Tan, and Yongfeng Zhang. Genrec: Large language model for generative recommendation. In *Advances in Information Retrieval - 46th European Conference on Information Retrieval, ECIR 2024, Glasgow, UK, March 24-28, 2024, Proceedings, Part III*, volume 14610 of *Lecture Notes in Computer Science*, pages 494–502. Springer, 2024.

[279] Hanjia Lyu, Song Jiang, Hanqing Zeng, Yinglong Xia, Qifan Wang, Si Zhang, Ren Chen, Christopher Leung, Jiajie Tang, and Jiebo Luo. Llm-rec: Personalized recommendation via prompting large language models. In *Findings of the Association for Computational Linguistics: NAACL 2024, Mexico City, Mexico, June 16-21, 2024*, pages 583–612. Association for Computational Linguistics, 2024.

[280] Sunhao Dai, Ninglu Shao, Haiyuan Zhao, Weijie Yu, Zihua Si, Chen Xu, Zhongxiang Sun, Xiao Zhang, and Jun Xu. Uncovering chatgpt's capabilities in recommender systems. In *Proceedings of the 17th ACM Conference on Recommender Systems, RecSys 2023, Singapore, Singapore, September 18-22, 2023*, pages 1126–1132. ACM, 2023.

[281] Junling Liu, Chao Liu, Renjie Lv, Kang Zhou, and Yan Zhang. Is chatgpt a good recommender? A preliminary study. *arXiv preprint arXiv:2304.10149*, 2023.

[282] Andrew K. Lampinen, Ishita Dasgupta, Stephanie C. Y. Chan, Kory W. Mathewson, Michael Henry Tessler, Antonia Creswell, James L. McClelland, Jane Wang, and Felix Hill. Can language models learn from explanations in context? In *Findings of the Association for Computational Linguistics: EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 537–563. Association for Computational Linguistics, 2022.

[283] Zhenmei Shi, Junyi Wei, Zhuoyan Xu, and Yingyu Liang. Why larger language models do in-context learning differently? In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net, 2024.

[284] Ori Ram, Yoav Levine, Itay Dalmedigos, Dor Muhlgay, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. In-context retrieval-augmented language models. *Trans. Assoc. Comput. Linguistics*, 11:1316–1331, 2023.

[285] Zheng Chen. PALR: personalization aware llms for recommendation. *arXiv preprint arXiv:2305.07622*, 2023.

[286] Alireza Salemi, Sheshera Mysore, Michael Bendersky, and Hamed Zamani. Lamp: When large language models meet personalization. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 7370–7392. Association for Computational Linguistics, 2024.

[287] Kai Zhang, Fubang Zhao, Yangyang Kang, and Xiaozhong Liu. Memory-augmented LLM personalization with short- and long-term memory coordination. *arXiv preprint arXiv:2309.11696*, 2023.

[288] Kurt Shuster, Jing Xu, Mojtaba Komeili, Da Ju, Eric Michael Smith, Stephen Roller, Megan Ung, Moya Chen, Kushal Arora, Joshua Lane, Morteza Behrooz, William Ngan, Spencer Poff, Naman Goyal, Arthur Szlam, Y-Lan Boureau, Melanie Kambadur, and Jason Weston. Blenderbot 3: a deployed conversational agent that continually learns to responsibly engage. *arXiv preprint arXiv:2208.03188*, 2022.

[289] Yancheng Wang, Ziyan Jiang, Zheng Chen, Fan Yang, Yingxue Zhou, Eunah Cho, Xing Fan, Yanbin Lu, Xiaojiang Huang, and Yingzhen Yang. Recmind: Large language model powered agent for recommendation. In *Findings of the Association for Computational Linguistics: NAACL 2024, Mexico City, Mexico, June 16-21, 2024*, pages 4351–4364. Association for Computational Linguistics, 2024.

[290] Cheng Li, Mingyang Zhang, Qiaozhu Mei, Yaqing Wang, Spurthi Amba Hombaiah, Yi Liang, and Michael Bendersky. Teach llms to personalize - an approach inspired by writing education. *arXiv preprint arXiv:2308.07968*, 2023.

[291] Saizheng Zhang, Emily Dinan, Jack Urbanek, Arthur Szlam, Douwe Kiela, and Jason Weston. Personalizing dialogue agents: I have a dog, do you have pets too? In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 2204–2213. Association for Computational Linguistics, 2018.

[292] Mojtaba Komeili, Kurt Shuster, and Jason Weston. Internet-augmented dialogue generation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 8460–8478. Association for Computational Linguistics, 2022.

[293] Alessandro Sordoni, Yoshua Bengio, Hossein Vahabi, Christina Lioma, Jakob Grue Simonsen, and Jian-Yun Nie. A hierarchical recurrent encoder-decoder for generative context-aware query suggestion. In *Proceedings of the 24th ACM International Conference on Information and Knowledge*

Management, CIKM 2015, Melbourne, VIC, Australia, October 19 - 23, 2015, pages 553–562. ACM, 2015.

[294] Wanyu Chen, Fei Cai, Honghui Chen, and Maarten de Rijke. Attention-based hierarchical neural query suggestion. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, SIGIR 2018, Ann Arbor, MI, USA, July 08-12, 2018*, pages 1093–1096. ACM, 2018.

[295] Mostafa Dehghani, Sascha Rothe, Enrique Alfonseca, and Pascal Fleury. Learning to attend, copy, and generate for session-based query suggestion. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, CIKM 2017, Singapore, November 06 - 10, 2017*, pages 1747–1756. ACM, 2017.

[296] Wasi Uddin Ahmad, Kai-Wei Chang, and Hongning Wang. Multi-task learning for document ranking and query suggestion. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018.

[297] Eugene Kharitonov, Craig Macdonald, Pavel Serdyukov, and Iadh Ounis. Intent models for contextualising and diversifying query suggestions. In *22nd ACM International Conference on Information and Knowledge Management, CIKM'13, San Francisco, CA, USA, October 27 - November 1, 2013*, pages 2303–2308. ACM, 2013.

[298] Jyun-Yu Jiang and Wei Wang. RIN: reformulation inference network for context-aware query suggestion. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management, CIKM 2018, Torino, Italy, October 22-26, 2018*, pages 197–206. ACM, 2018.

[299] Wasi Uddin Ahmad, Kai-Wei Chang, and Hongning Wang. Context attentive document ranking and query suggestion. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2019, Paris, France, July 21-25, 2019*, pages 385–394. ACM, 2019.

[300] Qiannan Cheng, Zhaochun Ren, Yujie Lin, Pengjie Ren, Zhumin Chen, Xiangyuan Liu, and Maarten de Rijke. Long short-term session search: Joint personalized reranking and next query prediction. In *WWW '21: The Web Conference 2021, Virtual Event / Ljubljana, Slovenia, April 19-23, 2021*, pages 239–248. ACM / IW3C2, 2021.

[301] Microsoft. Bing search engine, 2023. URL https://www.bing.com/.

[302] Romal Thoppilan, Daniel De Freitas, Jamie Hall, Noam Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du, YaGuang Li, Hongrae Lee, Huaixiu Steven Zheng, Amin Ghafouri, Marcelo Menegali, Yanping Huang, Maxim Krikun, Dmitry Lepikhin, James Qin, Dehao Chen, Yuanzhong Xu, Zhifeng Chen, Adam Roberts, Maarten Bosma, Yanqi Zhou, Chung-Ching Chang, Igor Krivokon, Will Rusch, Marc Pickett, Kathleen S. Meier-Hellstern, Meredith Ringel Morris, Tulsee Doshi, Renelito Delos Santos, Toju Duke, Johnny Soraker, Ben Zevenbergen, Vinodkumar Prabhakaran, Mark Diaz, Ben Hutchinson, Kristen Olson, Alejandra Molina, Erin Hoffman-John, Josh Lee, Lora Aroyo, Ravi Rajakumar, Alena Butryna, Matthew Lamm, Viktoriya Kuzmina, Joe Fenton, Aaron Cohen, Rachel Bernstein, Ray Kurzweil,

Blaise Agüera y Arcas, Claire Cui, Marian Croak, Ed H. Chi, and Quoc Le. Lamda: Language models for dialog applications. *arXiv preprint arXiv:2201.08239*, 2022.

[303] Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 9802–9822. Association for Computational Linguistics, 2023.

[304] Wenhao Yu, Dan Iter, Shuohang Wang, Yichong Xu, Mingxuan Ju, Soumya Sanyal, Chenguang Zhu, Michael Zeng, and Meng Jiang. Generate rather than retrieve: Large language models are strong context generators. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023.

[305] Jin Chen, Zheng Liu, Xu Huang, Chenwang Wu, Qi Liu, Gangwei Jiang, Yuanhao Pu, Yuxuan Lei, Xiaolong Chen, Xingmei Wang, Kai Zheng, Defu Lian, and Enhong Chen. When large language models meet personalization: perspectives of challenges and opportunities. *World Wide Web (WWW)*, 27(4):42, 2024.

[306] Lei Li, Yongfeng Zhang, Dugang Liu, and Li Chen. Large language models for generative recommendation: A survey and visionary discussions. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation, LREC/COLING 2024, 20-25 May, 2024, Torino, Italy*, pages 10146–10159. ELRA and ICCL, 2024.

[307] Luke Friedman, Sameer Ahuja, David Allen, Zhenning Tan, Hakim Sidahmed, Changbo Long, Jun Xie, Gabriel Schubiner, Ajay Patel, Harsh Lara, Brian Chu, Zexi Chen, and Manoj Tiwari. Leveraging large language models in conversational recommender systems. *arXiv preprint arXiv:2305.07961*, 2023.

[308] Zhankui He, Zhouhang Xie, Rahul Jha, Harald Steck, Dawen Liang, Yesu Feng, Bodhisattwa Prasad Majumder, Nathan Kallus, and Julian J. McAuley. Large language models as zero-shot conversational recommenders. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management, CIKM 2023, Birmingham, United Kingdom, October 21-25, 2023*, pages 720–730. ACM, 2023.

[309] Jimmy Wu, Rika Antonova, Adam Kan, Marion Lepert, Andy Zeng, Shuran Song, Jeannette Bohg, Szymon Rusinkiewicz, and Thomas A. Funkhouser. Tidybot: personalized robot assistance with large language models. *Auton. Robots*, 47(8):1087–1102, 2023.

[310] Joon Sung Park, Joseph C. O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology, UIST 2023, San Francisco, CA, USA, 29 October 2023- 1 November 2023*, pages 2:1–2:22. ACM, 2023.

[311] Darío Garigliotti and Krisztian Balog. Generating query suggestions to support task-based search. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, Shinjuku, Tokyo, Japan, August 7-11, 2017*, pages 1153–1156. ACM, 2017.

[312] Heng Ding, Shuo Zhang, Darío Garigliotti, and Krisztian Balog. Generating high-quality query suggestion candidates for task-based search. In *Advances in Information Retrieval - 40th European Conference on IR Research, ECIR 2018, Grenoble, France, March 26-29, 2018, Proceedings*, volume 10772 of *Lecture Notes in Computer Science*, pages 625–631. Springer, 2018.

[313] Zile Zhou, Xiao Zhou, Mingzhe Li, Yang Song, Tao Zhang, and Rui Yan. Personalized query suggestion with searching dynamic flow for online recruitment. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management, Atlanta, GA, USA, October 17-21, 2022*, pages 2773–2783. ACM, 2022.

[314] Qiaozhu Mei, Dengyong Zhou, and Kenneth Ward Church. Query suggestion using hitting time. In *Proceedings of the 17th ACM Conference on Information and Knowledge Management, CIKM 2008, Napa Valley, California, USA, October 26-30, 2008*, pages 469–478. ACM, 2008.

[315] Di Jiang, Kenneth Wai-Ting Leung, Jan Vosecky, and Wilfred Ng. Personalized query suggestion with diversity awareness. In *IEEE 30th International Conference on Data Engineering, Chicago, ICDE 2014, IL, USA, March 31 - April 4, 2014*, pages 400–411. IEEE Computer Society, 2014.

[316] Wanyu Chen, Fei Cai, Honghui Chen, and Maarten de Rijke. Personalized query suggestion diversification. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, Shinjuku, Tokyo, Japan, August 7-11, 2017*, pages 817–820. ACM, 2017.

[317] Thanh Vu, Alistair Willis, Udo Kruschwitz, and Dawei Song. Personalised query suggestion for intranet search with temporal user profiling. In *Proceedings of the 2017 Conference on Conference Human Information Interaction and Retrieval, CHIIR 2017, Oslo, Norway, March 7-11, 2017*, pages 265–268. ACM, 2017.

[318] Jianling Zhong, Weiwei Guo, Huiji Gao, and Bo Long. Personalized query suggestions. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25-30, 2020*, pages 1645–1648. ACM, 2020.

[319] Praveen Kumar Bodigutla. High quality related search query suggestions using deep reinforcement learning. *arXiv preprint arXiv:2108.04452*, 2021.

[320] Agnès Mustar, Sylvain Lamprier, and Benjamin Piwowarski. Using BERT and BART for query suggestion. In *Proceedings of the First Joint Conference of the Information Retrieval Communities in Europe (CIRCLE 2020), Samatan, Gers, France, July 6-9, 2020*, volume 2621 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2020.

[321] Agnès Mustar, Sylvain Lamprier, and Benjamin Piwowarski. On the study of transformers for query suggestion. *ACM Trans. Inf. Syst.*, 40(1):18:1–18:27, 2022.

[322] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 7871–7880. Association for Computational Linguistics, 2020.

[323] Silviu Cucerzan. Large-scale named entity disambiguation based on wikipedia data. In *EMNLP-CoNLL 2007, Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, June 28-30, 2007, Prague, Czech Republic*, pages 708–716. ACL, 2007.

[324] Silviu Cucerzan and Avirup Sil. The MSR systems for entity linking and temporal slot filling at TAC 2013. In *Proceedings of the Sixth Text Analysis Conference, TAC 2013, Gaithersburg, Maryland, USA, November 18-19, 2013*. NIST, 2013.

[325] Pierangela Samarati and Latanya Sweeney. Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression. 1998.

[326] KarmaHub. Karmahub, 2023. URL https://www.mykarmahub.com/.

[327] Jacob Cohen. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20:37 – 46, 1960.

[328] C. Spearman. The proof and measurement of association between two things. *The American Journal of Psychology*, 15(1):72–101, 1904.

[329] Tom Hope, Doug Downey, Daniel S. Weld, Oren Etzioni, and Eric Horvitz. A computational inflection for scientific discovery. *Commun. ACM*, 66(8):62–73, 2023.

[330] Hanchen Wang, Tianfan Fu, Yuanqi Du, Wenhao Gao, Kexin Huang, Ziming Liu, Payal Chandak, Shengchao Liu, Peter Van Katwyk, Andreea Deac, Anima Anandkumar, Karianne Bergen, Carla P. Gomes, Shirley Ho, Pushmeet Kohli, Joan Lasenby, Jure Leskovec, Tie-Yan Liu, Arjun K. Manrai, Debora S. Marks, Bharath Ramsundar, Le Song, Jimeng Sun, Jian Tang, Petar Velickovic, Max Welling, Linfeng Zhang, Connor W. Coley, Yoshua Bengio, and Marinka Zitnik. Scientific discovery in the age of artificial intelligence. *Nat.*, 620(7972):47–60, 2023.

[331] Qian Huang, Jian Vora, Percy Liang, and Jure Leskovec. Benchmarking large language models as AI research agents. *arXiv preprint arXiv:2310.03302*, 2023.

[332] Michael Fire and Carlos Guestrin. Over-optimization of academic publishing metrics: observing goodhart's law in action. *GigaScience*, 8(6):giz053, 2019.

[333] Jessica Vamathevan, Dominic Clark, Paul Czodrowski, Ian Dunham, Edgardo Ferran, George Lee, Bin Li, Anant Madabhushi, Parantu K. Shah, Michaela Spitzer, and Shanrong Zhao. Applications of machine learning in drug discovery and development. *Nature Reviews Drug Discovery*, 18:463 – 477, 2019.

[334] Microsoft Research AI4Science and Microsoft Quantum. The impact of large language models on scientific discovery: a preliminary study using gpt-4. *arXiv preprint arXiv:2311.07361*, 2023.

[335] Andres M. Bran, Sam Cox, Oliver Schilter, Carlo Baldassari, Andrew D. White, and Philippe Schwaller. Augmenting large language models with chemistry tools. *Nat. Mac. Intell.*, 6(5):525–535, 2024.

[336] Jan Philip Wahle, Terry Ruas, Mohamed Abdalla, Bela Gipp, and Saif M. Mohammad. We are who we cite: Bridges of influence between natural language processing and other academic fields. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing,*

EMNLP 2023, Singapore, December 6-10, 2023, pages 12896–12913. Association for Computational Linguistics, 2023.

[337] Jason Portenoy, Marissa Radensky, Jevin D. West, Eric Horvitz, Daniel S. Weld, and Tom Hope. Bursting scientific filter bubbles: Boosting innovation via novel author discovery. In *CHI '22: CHI Conference on Human Factors in Computing Systems, New Orleans, LA, USA, 29 April 2022 - 5 May 2022*, pages 309:1–309:13. ACM, 2022.

[338] Bernardino Romera-Paredes, Mohammadamin Barekatain, Alexander Novikov, Matej Balog, M. Pawan Kumar, Emilien Dupont, Francisco J. R. Ruiz, Jordan S. Ellenberg, Pengming Wang, Omar Fawzi, Pushmeet Kohli, and Alhussein Fawzi. Mathematical discoveries from program search with large language models. *Nat.*, 625(7995):468–475, 2024.

[339] Yiren Liu, Si Chen, Haocong Cheng, Mengxia Yu, Xiao Ran, Andrew Mo, Yiliu Tang, and Yun Huang. How AI processing delays foster creativity: Exploring research question co-creation with an llm-based agent. In *Proceedings of the CHI Conference on Human Factors in Computing Systems, CHI 2024, Honolulu, HI, USA, May 11-16, 2024*, pages 17:1–17:25. ACM, 2024.

[340] Don R. Swanson. Undiscovered public knowledge. *The Library Quarterly*, 56:103 – 118, 1986.

[341] Sam Henry and Bridget T. McInnes. Literature based discovery: Models, methods, and trends. *J. Biomed. Informatics*, 74:20–32, 2017.

[342] Vahe Tshitoyan, John Dagdelen, Leigh Weston, Alexander Dunn, Ziqin Rong, Olga Kononova, Kristin A. Persson, Gerbrand Ceder, and Anubhav Jain. Unsupervised word embeddings capture latent knowledge from materials science literature. *Nat.*, 571(7763):95–98, 2019.

[343] Justin Sybrandt, Ilya Tyagin, Michael Shtutman, and Ilya Safro. AGATHA: automatic graph mining and transformer based hypothesis generation approach. In *CIKM '20: The 29th ACM International Conference on Information and Knowledge Management, Virtual Event, Ireland, October 19-23, 2020*, pages 2757–2764. ACM, 2020.

[344] Rahul Nadkarni, David Wadden, Iz Beltagy, Noah A. Smith, Hannaneh Hajishirzi, and Tom Hope. Scientific language models for biomedical knowledge base completion: An empirical study. In *3rd Conference on Automated Knowledge Base Construction, AKBC 2021, Virtual, October 4-8, 2021*, 2021.

[345] Biqing Qi, Kaiyan Zhang, Haoxiang Li, Kai Tian, Sihang Zeng, Zhang-Ren Chen, and Bowen Zhou. Large language models are zero shot hypothesis proposers. *arXiv preprint arXiv:2311.05965*, 2023.

[346] Long Li, Weiwen Xu, Jiayan Guo, Ruochen Zhao, Xingxuan Li, Yuqian Yuan, Boqiang Zhang, Yuming Jiang, Yifei Xin, Ronghao Dang, Deli Zhao, Yu Rong, Tian Feng, and Lidong Bing. Chain of ideas: Revolutionizing research via novel idea development with LLM agents. *arXiv preprint arXiv:2410.13185*, 2024.

[347] Chris Lu, Cong Lu, Robert Tjarko Lange, Jakob N. Foerster, Jeff Clune, and David Ha. The AI scientist: Towards fully automated open-ended scientific discovery. *arXiv preprint arXiv:2408.06292*, 2024.

[348] Angeliki Lazaridou, Elena Gribovskaya, Wojciech Stokowiec, and Nikolai Grigorev. Internet-augmented language models through few-shot prompting for open-domain question answering. *arXiv preprint arXiv:2203.05115*, 2022.

[349] Ori Ram, Yoav Levine, Itay Dalmedigos, Dor Muhlgay, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. In-context retrieval-augmented language models. *Trans. Assoc. Comput. Linguistics*, 11:1316–1331, 2023.

[350] Weijia Shi, Sewon Min, Michihiro Yasunaga, Minjoon Seo, Richard James, Mike Lewis, Luke Zettlemoyer, and Wen-tau Yih. REPLUG: retrieval-augmented black-box 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), NAACL 2024, Mexico City, Mexico, June 16-21, 2024*, pages 8371–8384. Association for Computational Linguistics, 2024.

[351] Yike Wu, Nan Hu, Sheng Bi, Guilin Qi, Jie Ren, Anhuan Xie, and Wei Song. Retrieve-rewrite-answer: A kg-to-text enhanced llms framework for knowledge graph question answering. *arXiv preprint arXiv:2309.11206*, 2023.

[352] Sean Welleck, Ximing Lu, Peter West, Faeze Brahman, Tianxiao Shen, Daniel Khashabi, and Yejin Choi. Generating sequences by learning to self-correct. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023.

[353] Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. Self-refine: Iterative refinement with self-feedback. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023.

[354] Kumar Shridhar, Koustuv Sinha, Andrew Cohen, Tianlu Wang, Ping Yu, Ramakanth Pasunuru, Mrinmaya Sachan, Jason Weston, and Asli Celikyilmaz. The ART of LLM refinement: Ask, refine, and trust. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), NAACL 2024, Mexico City, Mexico, June 16-21, 2024*, pages 5872–5883. Association for Computational Linguistics, 2024.

[355] Deep Ganguli, Amanda Askell, Nicholas Schiefer, Thomas I. Liao, Kamile Lukosiute, Anna Chen, Anna Goldie, Azalia Mirhoseini, Catherine Olsson, Danny Hernandez, Dawn Drain, Dustin Li, Eli Tran-Johnson, Ethan Perez, Jackson Kernion, Jamie Kerr, Jared Mueller, Joshua Landau, Kamal Ndousse, Karina Nguyen, Liane Lovitt, Michael Sellitto, Nelson Elhage, Noemí Mercado, Nova DasSarma, Oliver Rausch, Robert Lasenby, Robin Larson, Sam Ringer, Sandipan Kundu, Saurav Kadavath, Scott Johnston, Shauna Kravec, Sheer El Showk, Tamera Lanham, Timothy Telleen-Lawton, Tom Henighan, Tristan Hume, Yuntao Bai, Zac Hatfield-Dodds, Ben Mann, Dario Amodei, Nicholas Joseph, Sam McCandlish, Tom Brown, Christopher Olah, Jack Clark, Samuel R. Bowman, and Jared Kaplan. The capacity for moral self-correction in large language models. *arXiv preprint arXiv:2302.07459*, 2023.

[356] J. Young. *A Technique for Producing Ideas*. McGraw Hill LLC, 2003. ISBN 9780071426251.

[357] Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. Lost in the middle: How language models use long contexts. *Trans. Assoc. Comput. Linguistics*, 12:157–173, 2024.

[358] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging llm-as-a-judge with mt-bench and chatbot arena. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023.

[359] Jinlan Fu, See-Kiong Ng, Zhengbao Jiang, and Pengfei Liu. Gptscore: Evaluate as you desire. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), NAACL 2024, Mexico City, Mexico, June 16-21, 2024*, pages 6556–6576. Association for Computational Linguistics, 2024.

[360] Ying-Chun Lin, Jennifer Neville, Jack W. Stokes, Longqi Yang, Tara Safavi, Mengting Wan, Scott Counts, Siddharth Suri, Reid Andersen, Xiaofeng Xu, Deepak Gupta, Sujay Kumar Jauhar, Xia Song, Georg Buscher, Saurabh Tiwary, Brent J. Hecht, and Jaime Teevan. Interpretable user satisfaction estimation for conversational systems with large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 11100–11115. Association for Computational Linguistics, 2024.

[361] Yilun Du, Shuang Li, Antonio Torralba, Joshua B. Tenenbaum, and Igor Mordatch. Improving factuality and reasoning in language models through multiagent debate. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net, 2024.

[362] Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu, Keming Lu, Jianxin Ma, Rui Men, Xingzhang Ren, Xuancheng Ren, Chuanqi Tan, Sinan Tan, Jianhong Tu, Peng Wang, Shijie Wang, Wei Wang, Shengguang Wu, Benfeng Xu, Jin Xu, An Yang, Hao Yang, Jian Yang, Shusheng Yang, Yang Yao, Bowen Yu, Hongyi Yuan, Zheng Yuan, Jianwei Zhang, Xingxuan Zhang, Yichang Zhang, Zhenru Zhang, Chang Zhou, Jingren Zhou, Xiaohuan Zhou, and Tianhang Zhu. Qwen technical report. *arXiv preprint arXiv:2309.16609*, 2023.

[363] Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mixtral of experts. *arXiv preprint arXiv:2401.04088*, 2024.

[364] Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol

Vinyals, Percy Liang, Jeff Dean, and William Fedus. Emergent abilities of large language models. *Trans. Mach. Learn. Res.*, 2022, 2022.

[365] Minju Seo, Jinheon Baek, Seongyun Lee, and Sung Ju Hwang. Paper2code: Automating code generation from scientific papers in machine learning. *arXiv preprint arXiv:2504.17192*, 2025.

[366] Sangwoo Park, Jinheon Baek, Soyeong Jeong, and Sung Ju Hwang. Chain of retrieval: Multi-aspect iterative search expansion and post-order search aggregation for full paper retrieval. *arXiv preprint arXiv:2507.10057*, 2025.

[367] John M. Zelle and Raymond J. Mooney. Learning to parse database queries using inductive logic programming. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence and Eighth Innovative Applications of Artificial Intelligence Conference, AAAI 96, IAAI 96, Portland, Oregon, USA, August 4-8, 1996, Volume 2*, pages 1050–1055. AAAI Press / The MIT Press, 1996.

[368] Xiaojun Xu, Chang Liu, and Dawn Song. Sqlnet: Generating structured queries from natural language without reinforcement learning. *arXiv preprint arXiv:1711.04436*, 2017.

[369] Navid Yaghmazadeh, Yuepeng Wang, Isil Dillig, and Thomas Dillig. Sqlizer: query synthesis from natural language. *Proc. ACM Program. Lang.*, 1(OOPSLA):63:1–63:26, 2017.

[370] Ruichu Cai, Boyan Xu, Zhenjie Zhang, Xiaoyan Yang, Zijian Li, and Zhihao Liang. An encoder-decoder framework translating natural language to database queries. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden*, pages 3977–3983. ijcai.org, 2018.

[371] Longxu Dou, Yan Gao, Xuqi Liu, Mingyang Pan, Dingzirui Wang, Wanxiang Che, Dechen Zhan, Min-Yen Kan, and Jian-Guang Lou. Towards knowledge-intensive text-to-sql semantic parsing with formulaic knowledge. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 5240–5253. Association for Computational Linguistics, 2022.

[372] Zijin Hong, Zheng Yuan, Hao Chen, Qinggang Zhang, Feiran Huang, and Xiao Huang. Knowledge-to-sql: Enhancing SQL generation with data expert LLM. In *Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024*, pages 10997–11008. Association for Computational Linguistics, 2024.

[373] Llama Team. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

[374] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022.

[375] Xiping Liu and Zhao Tan. Divide and prompt: Chain of thought prompting for text-to-sql. *arXiv preprint arXiv:2304.11556*, 2023.

[376] Chang-Yu Tai, Ziru Chen, Tianshu Zhang, Xiang Deng, and Huan Sun. Exploring chain of thought style prompting for text-to-sql. In *Proceedings of the 2023 Conference on Empirical Methods in*

Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023, pages 5376–5393. Association for Computational Linguistics, 2023.

[377] Shuaichen Chang and Eric Fosler-Lussier. How to prompt llms for text-to-sql: A study in zero-shot, single-domain, and cross-domain settings. *arXiv preprint arXiv:2305.11853*, 2023.

[378] Dongjun Lee, Choongwon Park, Jaehyuk Kim, and Heesoo Park. MCS-SQL: leveraging multiple prompts and multiple-choice selection for text-to-sql generation. In *Proceedings of the 31st International Conference on Computational Linguistics, COLING 2025, Abu Dhabi, UAE, January 19-24, 2025*, pages 337–353. Association for Computational Linguistics, 2025.

[379] Xuemei Dong, Chao Zhang, Yuhang Ge, Yuren Mao, Yunjun Gao, Lu Chen, Jinshu Lin, and Dongfang Lou. C3: zero-shot text-to-sql with chatgpt. *arXiv preprint arXiv:2307.07306*, 2023.

[380] Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023.

[381] Zihui Gu, Ju Fan, Nan Tang, Lei Cao, Bowen Jia, Sam Madden, and Xiaoyong Du. Few-shot text-to-sql translation using structure and content prompt learning. *Proc. ACM Manag. Data*, 1 (2):147:1–147:28, 2023.

[382] Mohammadreza Pourreza and Davood Rafiei. DIN-SQL: decomposed in-context learning of text-to-sql with self-correction. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023.

[383] Bing Wang, Changyu Ren, Jian Yang, Xinnian Liang, Jiaqi Bai, Linzheng Chai, Zhao Yan, Qian-Wen Zhang, Di Yin, Xing Sun, and Zhoujun Li. MAC-SQL: A multi-agent collaborative framework for text-to-sql. In *Proceedings of the 31st International Conference on Computational Linguistics, COLING 2025, Abu Dhabi, UAE, January 19-24, 2025*, pages 540–557. Association for Computational Linguistics, 2025.

[384] Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 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), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 13484–13508. Association for Computational Linguistics, 2023.

[385] Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Stanford alpaca: An instruction-following llama model, 2023.

[386] Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, Qingwei Lin, and Daxin Jiang. Wizardlm: Empowering large pre-trained language models to follow complex instructions. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024.

[387] Subhabrata Mukherjee, Arindam Mitra, Ganesh Jawahar, Sahaj Agarwal, Hamid Palangi, and Ahmed Awadallah. Orca: Progressive learning from complex explanation traces of GPT-4. *arXiv preprint arXiv:2306.02707*, 2023.

[388] Shivchander Sudalairaj, Abhishek Bhandwaldar, Aldo Pareja, Kai Xu, David D. Cox, and Akash Srivastava. LAB: large-scale alignment for chatbots. *arXiv preprint arXiv:2403.01081*, 2024.

[389] Dimitrios Alivanistos, Selene Baez Santamaría, Michael Cochez, Jan-Christoph Kalo, Emile van Krieken, and Thiviyan Thanapalasingam. Prompting as probing: Using language models for knowledge base construction. In *Proceedings of the Semantic Web Challenge on Knowledge Base Construction from Pre-trained Language Models 2022 co-located with the 21st International Semantic Web Conference (ISWC2022), Virtual Event, Hanghzou, China, October 2022*, volume 3274 of *CEUR Workshop Proceedings*, pages 11–34. CEUR-WS.org, 2022.

[390] Anmol Nayak and Hariprasad Timmapathini. LLM2KB: constructing knowledge bases using instruction tuned context aware large language models. In *Joint proceedings of the 1st workshop on Knowledge Base Construction from Pre-Trained Language Models (KBC-LM) and the 2nd challenge on Language Models for Knowledge Base Construction (LM-KBC) co-located with the 22nd International Semantic Web Conference (ISWC 2023), Athens, Greece, November 6, 2023*, volume 3577 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2023.

[391] Blerta Veseli, Simon Razniewski, Jan-Christoph Kalo, and Gerhard Weikum. Evaluating the knowledge base completion potential of GPT. In *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, pages 6432–6443. Association for Computational Linguistics, 2023.

[392] Mayank Mishra, Matt Stallone, Gaoyuan Zhang, Yikang Shen, Aditya Prasad, Adriana Meza Soria, Michele Merler, Parameswaran Selvam, Saptha Surendran, Shivdeep Singh, Manish Sethi, Xuan-Hong Dang, Pengyuan Li, Kun-Lung Wu, Syed Zawad, Andrew Coleman, Matthew White, Mark Lewis, Raju Pavuluri, Yan Koyfman, Boris Lublinsky, Maximilien de Bayser, Ibrahim Abdelaziz, Kinjal Basu, Mayank Agarwal, Yi Zhou, Chris Johnson, Aanchal Goyal, Hima Patel, Syed Yousaf Shah, Petros Zerfos, Heiko Ludwig, Asim Munawar, Maxwell Crouse, Pavan Kapanipathi, Shweta Salaria, Bob Calio, Sophia Wen, Seetharami Seelam, Brian Belgodere, Carlos A. Fonseca, Amith Singhee, Nirmit Desai, David D. Cox, Ruchir Puri, and Rameswar Panda. Granite code models: A family of open foundation models for code intelligence. *arXiv preprint arXiv:2405.04324*, 2024.

Acknowledgment

I would first like to express my deepest gratitude to my advisor, Prof. Sung Ju Hwang, for his unwavering guidance, encouragement, and intellectual generosity throughout my Ph.D. journey. His keen ability to identify fundamental research questions, combined with his passion for rigorous and impactful research, has shaped my way of thinking as a researcher. Under his mentorship, I learned not only how to conduct independent and principled research, but also how to pursue long-term research visions with persistence and integrity. Without his continuous support and trust, this dissertation would not have been possible, and I feel truly fortunate to have had the opportunity to be advised by him.

I am also sincerely grateful to my thesis committee members, Prof. Jinwoo Shin, Prof. Minjoon Seo, Prof. Yejin Choi, and Dr. Sujay Kumar Jauhar, for their insightful feedback, constructive criticism, and thoughtful questions. Their diverse perspectives and deep expertise greatly strengthened this dissertation and helped refine both its technical depth and broader research implications. I truly appreciate the time and effort they devoted to evaluating my work and guiding me throughout this process.

I am extremely fortunate to be surrounded by a group of brilliant people at the Machine Learning & Artificial Intelligence (MLAI) lab, who made my Ph.D. journey both productive and enjoyable. I am especially grateful to Dong Bok Lee, who took great care of me when I first joined the lab and helped me adapt to both the research environment and lab culture. I would also like to thank Minki Kang and Seanie Lee, who began their graduate studies around the same time as I did and became a constant source of support and encouragement throughout the early stages of my Ph.D. journey. Jaehyeong Jo, Dongki Kim, Sangwon Jang, Sunil Hwang, and Jaehong Yoon deserve special thanks for sharing many enjoyable moments beyond research, which made my time in the lab more memorable and balanced. Beyond these, I am also deeply thankful to the many senior and fellow colleagues at the MLAI lab with whom I had the privilege of collaborating and exchanging ideas. I would like to especially thank Yumin Choi, Minju Seo, Kangsan Kim, Woongyeong Yeo, Jaewoo Lee, Simon Aytes, Joonho Ko, Jongha Kim, Wonyoung Jeong, Hayeon Lee, Geon Park, Seul Lee, Jiongdao Jin, Jin Myung Kwak, as well as two different Sangwoo Park. Working with them continuously contributed to my growth as a researcher. Furthermore, beyond individual collaborations, the collective intellectual atmosphere and mutual support within the lab played an important role in shaping my research mindset and sustaining my motivation throughout the long Ph.D. journey. Whenever I needed help (whether in research discussions, technical challenges, or everyday matters), there was always someone willing to lend their time and effort generously.

I cannot describe my Ph.D. journey without the time I spent and the mentorship I received during my internships, where I had the privilege of working with outstanding mentors and collaborators. One of the most memorable moments was collaborating with Sujay Kumar Jauhar and Silviu Cucerzan at Microsoft Research, an experience that left a lasting impression on me. I was deeply inspired by their guidance and the depth of research discussions we had, which substantially influenced my perspective on impactful and principled research. I am also thankful that, on my many visits to Seattle and whenever we happened to be in the same place, they made time to meet me and welcomed me so warmly. I am equally grateful for my time at Amazon, where working with Alham Fikri Aji, Amir Saffari, and Jens Lehmann played an important role in strengthening my research foundations, especially during the early stages of my Ph.D.; in particular, I sincerely thank Alham for his continued support and for extending a collaboration opportunity even after the internship concluded. I also sincerely appreciate my experience

at IBM Research, where I had the great pleasure of working with Horst Samulowitz, Oktie Hassanzadeh, Dharmashankar Subramanian, Sola Shirai, Alfio Gliozzo, and Debarun Bhattacharjya, who entrusted me with a high degree of research freedom and encouraged independent exploration, allowing me to grow with confidence as a researcher. Finally, I cannot conclude this acknowledgement without highlighting my time at Google, which marked the final chapter of my internship experiences and a particularly rewarding period of my Ph.D., which was made especially meaningful by the support and collaboration of Sun Jae Lee, Prakhar Gupta, Geunseob (GS) Oh, Siddharth Dalmia, and Prateek Kolhar. Working with Sun Jae Lee and Geunseob (GS) Oh was especially meaningful to me, as they were the first Korean collaborators I had the opportunity to work with during many internships, which added a unique sense of familiarity and personal resonance. Overall, all my internships were instrumental in shaping my research trajectory, broadening my perspective, and reinforcing my motivation to pursue ambitious and meaningful research.

Beyond my advisors, lab colleagues, and internship mentors, I have been greatly fortunate to meet and work with many mentors, collaborators, and friends, whose encouragement and openness profoundly influenced different stages of my research journey. I would like to first thank the members of the DMIS lab at Korea University (my undergraduate university), where I was given my very first opportunity to engage in research. I am especially thankful to Prof. Jaewoo Kang for opening the door to research for me and for his thoughtful guidance, as well as to Hyunjae Kim, Yookyung Koh, Donghyeon Park, and Buru Chang, who supported and mentored me during my earliest steps as a researcher. I am also grateful for my collaborations at KAIST with Seungone Kim and Sukmin Cho, whose discussions and joint work were both stimulating and enjoyable, and to Prof. Jong C. Park and Prof. James Thorne for their insightful feedback and collaborative spirit, which greatly enriched our work together. In addition, I would like to thank Peerat Limkonchotiwat, whom I met during my time at Amazon and who later became a friend and collaborator. I am sincerely thankful to him for extending collaboration opportunities, as well as to his colleagues Patomporn Payoungkhamdee and Prof. Sarana Nutanong for their openness and productive collaboration. These collaborations enriched my research experience, expanded my professional network, and reminded me of the importance of trust, generosity, and shared curiosity in scientific collaboration.

Beyond direct research collaborations, I would also like to acknowledge the mentors and programs that supported my academic development and encouraged me to pursue and persist in a path of intellectual growth and exploration over the years. I am very grateful to the Presidential Science Scholarship (for Graduate Study) for its generous support, which provided both financial assistance and strong encouragement during my graduate studies. I also remain deeply grateful to the Samsung Dream Scholarship Foundation for its continued support and genuine encouragement, which always made me feel welcomed and supported throughout my journey, both academically and personally. In addition, I am thankful to the Microsoft Student Partner program for offering me early opportunities to engage with computer science in a global and collaborative environment; through this experience, I began to envision myself contributing to research and development at organizations like Microsoft, and it was particularly meaningful to later (somewhat) realize this aspiration during my Ph.D. through my internship at Microsoft Research. I would also like to express my appreciation to the SW Maestro program, which played an important role in motivating me to pursue artificial intelligence as my primary research field. Finally, I am grateful to Prof. Heejo Lee for his mentorship through the KUICS and Inc0gnito programs, where his continued guidance and thoughtful questions helped me reflect on my long-term goals, and to Kwangjae Jang for encouraging me (during my high school years) to pursue computer science and engineering.

Beyond my academic and professional life, I owe much of my wonderful time to friends outside the lab and the research community as well. While the number of people who have supported me along the way is far too many to enumerate, I am sincerely thankful to all of my friends who helped me relieve stress, maintain balance, and find joy throughout the long and sometimes demanding journey of graduate school. In particular, I am especially grateful to many friends I met through various communities at Korea University, including the Honam community, the Department of Computer Science and Engineering, and the WoonWha educational volunteering club, as well as those I met through the Samsung Dream Scholarship Foundation and the Microsoft Student Partner program, all of whom enriched my life in different yet equally meaningful ways and reminded me that there is a world beyond papers and deadlines.

Last but not least, I would like to express my sincere thanks to my parents and younger sister (my father Cheonseok Baek, my mother Miran Jeong, and my sister Jina Baek) for their unconditional love, patience, and unwavering support throughout every stage of my journey. Their constant belief in me, even during moments of uncertainty, has been a source of strength that sustained me through the most challenging periods of my life. And finally, at the very heart of this journey, there is Soyeong Jeong, my closest friend, collaborator, and fiancée, who stood by me through both my lowest and highest moments, sharing every challenge and joy along the way, and whose presence, trust, and partnership made this journey not only possible but profoundly meaningful, one that I could never have walked alone.

Curriculum Vitae

Name	:	Jinheon Baek
Date of Birth	:	July 16, 1997

Educations

2022. 03. – 2026. 02.	Ph.D. in Artificial Intelligence, KAIST
2020. 03. – 2022. 02.	M.S. in Artificial Intelligence, KAIST
2016. 03. – 2020. 02.	B.S. in Computer Science and Engineering, Korea University B.E. in Software Technology and Enterprise Program, Korea University

Experiences

2024. 09. – 2024. 11.	Research Intern, Google DeepMind
2024. 05. – 2024. 08.	Research Intern, IBM Research
2023. 06. – 2023. 09.	Research Intern, Microsoft Research
2022. 08. – 2022. 11.	Applied Scientist II Intern, Amazon

Honors & Awards

1.	Awarded the Presidential Science Scholarship for Graduate Study	2024-2026
2.	Quoted in Nature (2025) for Expert Commentary on AI-Generated Research	2025
3.	Received the Best Paper Award at NAACL 2025 (The BiGGen Bench)	2025
4.	Selected as the 6th Most Influential Paper at NAACL 2025 (ResearchAgent)	2025
5.	Selected as the 7th Most Influential Paper at NAACL 2024 (Adaptive-RAG)	2025
6.	Received the 3rd Place Poster Presentation Award at Citadel PhD Summit	2025
7.	Selected as One of the Great Reviewers for ACL ARR (Multiple Times)	2024-2025
8.	Awarded the Travel Grant from KAIST-Google Partnership Program for WWW	2024
9.	Received the Best Poster Presentation Award at Samsung AI Forum	2023
10.	Received the Best Paper Award at NLRSE Workshop in ACL	2023
11.	Awarded the ICML Travel Grant for ICML	2023
12.	Awarded the Google Travel Grant for NeurIPS	2022
13.	Selected as One of the Top Reviewers (Top 10%) of NeurIPS	2022
14.	Selected as One of the Highlighted Reviewers (Top 10%) of ICLR	2022
15.	Selected as One of the Best Reviewers (Top 10%) of ICML	2021
16.	Received the Best Paper Award at CKAIA	2020

Publications (∗: Equal Contribution; †: Corresponding Author)

1. Universal Retrieval: Unifying Data Access Across Heterogeneous Knowledge Bases
Jinheon Baek, Soyeong Jeong, Sangwoo Park, Minki Kang, Woongyeong Yeo, and Sung Ju Hwang
in preparation

2. Multimodal Prompt Optimization: Why Not Leverage Multiple Modalities for MLLMs
Yumin Choi*, Dongki Kim*, Jinheon Baek, and Sung Ju Hwang
arXiv preprint

3. Rethinking Reward Models for Multi-Domain Test-Time Scaling
Dong Bok Lee, Seanie Lee, Sangwoo Park, Minki Kang, Jinheon Baek, Dongki Kim, Dominik Wagner, Jiongdao Jin, Heejun Lee, Tobias Bocklet, Jinyu Wang, Jingjing Fu, Sung Ju Hwang, Jiang Bian, and Lei Song
arXiv preprint

4. Paper2Code: Automating Code Generation from Scientific Papers in Machine Learning
Minju Seo, Jinheon Baek, Seongyun Lee, and Sung Ju Hwang
arXiv preprint

5. UniversalRAG: Retrieval-Augmented Generation over Corpora of Diverse Modalities and Granularities
Woongyeong Yeo*, Kangsan Kim*, Soyeong Jeong, Jinheon Baek, and Sung Ju Hwang
arXiv preprint

6. Chain of Retrieval: Multi-Aspect Iterative Search Expansion and Post-Order Search Aggregation for Full Paper Retrieval
Sangwoo Park, Jinheon Baek, Soyeong Jeong, and Sung Ju Hwang
arXiv preprint

7. System Prompt Optimization with Meta-Learning
Yumin Choi*, Jinheon Baek*, and Sung Ju Hwang
NeurIPS, 2025

8. Database-Augmented Query Representation for Information Retrieval
Soyeong Jeong, Jinheon Baek, Sukmin Cho, Sung Ju Hwang, and Jong C. Park
EMNLP, 2025 **(Oral)**

9. Sketch-of-Thought: Efficient LLM Reasoning with Adaptive Cognitive-Inspired Sketching
Simon A. Aytes, Jinheon Baek, and Sung Ju Hwang
EMNLP, 2025

10. Efficient Real-time Refinement of Language Model Text Generation
Joonho Ko, Jinheon Baek, and Sung Ju Hwang
EMNLP, 2025

11. CaMMT: Benchmarking Culturally Aware Multimodal Machine Translation
Emilio Villa-Cueva, Sholpan Bolatzhanova, Diana Turmakhan, Kareem Elzeky, ..., Jinheon Baek, ..., Soyeong Jeong, ..., Injy Hamed, Atnafu Lambebo Tonja, and Thamar Solorio
EMNLP Findings, 2025

12. Unified Multimodal Interleaved Document Representation for Retrieval
Jaewoo Lee*, Joonho Ko*, Jinheon Baek*, Soyeong Jeong, and Sung Ju Hwang
VecDB Workshop @ ICML, 2025

13. Efficient Long Context Language Model Retrieval with Compression
Minju Seo, Jinheon Baek, Seongyun Lee, and Sung Ju Hwang
ACL, 2025

14. Revisiting In-Context Learning with Long Context Language Models
Jinheon Baek, Sun Jae Lee, Prakhar Gupta, Geunseob (GS) Oh, Siddharth Dalmia, and Prateek Kolhar
ACL Findings, 2025

15. Knowledge Base Construction for Knowledge-Augmented Text-to-SQL
Jinheon Baek, Horst Samulowitz, Oktie Hassanzadeh,
Dharmashankar Subramanian, Sola Shirai, Alfio Gliozzo, and Debarun Bhattacharjya
ACL Findings, 2025

16. VideoRAG: Retrieval-Augmented Generation over Video Corpus
Soyeong Jeong*, Kangsan Kim*, Jinheon Baek*, and Sung Ju Hwang
ACL Findings, 2025

17. Towards Better Understanding of Program-of-Thought Reasoning
in Cross-Lingual and Multilingual Environments
Patomporn Payoungkhamdee, Pume Tuchinda, Jinheon Baek, Samuel Cahyawijaya, Can Udomcharoenchaikit, Potsawee Manakul, Peerat Limkonchotiwat, Ekapol Chuangsuwanich, and Sarana Nutanong
ACL Findings, 2025

18. ResearchAgent: Iterative Research Idea Generation
over Scientific Literature with Large Language Models
Jinheon Baek, Sujay Kumar Jauhar, Silviu Cucerzan, and Sung Ju Hwang
NAACL, 2025 **(Oral)**

19. The BiGGen Bench: A Principled Benchmark
for Fine-grained Evaluation of Language Models with Language Models
Seungone Kim, Juyoung Suk, Ji Yong Cho, Shayne Longpre, Chaeeun Kim,
Dongkeun Yoon, Guijin Son, Yejin Cho, Sheikh Shafayat, Jinheon Baek, ...,
Graham Neubig, Moontae Lee, Kyungjae Lee, and Minjoon Seo
NAACL, 2025 **(Best Paper)**

20. CVQA: Culturally-diverse Multilingual Visual Question Answering Benchmark
David Romero, Chenyang Lyu, Haryo Akbarianto Wibowo, Teresa Lynn, Injy Hamed,
Aditya Nanda Kishore, ..., Jinheon Baek, ..., Soyeong Jeong, ..., Thamar Solorio, and Alham Fikri Aji
NeurIPS, 2024 **(Oral)**

21. Retrieval-Augmented Data Augmentation for Low-Resource Domain Tasks
Minju Seo*, Jinheon Baek*, James Thorne, and Sung Ju Hwang
AFM Workshop @ NeurIPS, 2024

22. An Empirical Study of Multilingual Reasoning Distillation for Question Answering
Patomporn Payoungkhamdee, Peerat Limkonchotiwat, Jinheon Baek,
Potsawee Manakul, Can Udomcharoenchaikit, Ekapol Chuangsuwanich, and Sarana Nutanong
EMNLP, 2024

23. Rethinking Code Refinement: Learning to Judge Code Efficiency
Minju Seo, Jinheon Baek, and Sung Ju Hwang
EMNLP Findings, 2024

24. Adaptive-RAG: Learning to Adapt Retrieval-Augmented
    Large Language Models through Question Complexity
    Soyeong Jeong, Jinheon Baek, Sukmin Cho, Sung Ju Hwang, and Jong C. Park
    NAACL, 2024

25. Knowledge-Augmented Large Language Models for Personalized Contextual Query Suggestion
    Jinheon Baek, Nirupama Chandrasekaran, Silviu Cucerzan, Allen Herring, and Sujay Kumar Jauhar
    WWW, 2024

26. Knowledge-Augmented Reasoning Distillation
    for Small Language Models in Knowledge-Intensive Tasks
    Minki Kang, Seanie Lee, Jinheon Baek, Kenji Kawaguchi, and Sung Ju Hwang
    NeurIPS, 2023

27. Knowledge-Augmented Language Model Verification
    Jinheon Baek, Soyeong Jeong, Minki Kang, Jong C. Park, and Sung Ju Hwang
    EMNLP, 2023

28. Test-Time Self-Adaptive Small Language Models for Question Answering
    Soyeong Jeong, Jinheon Baek, Sukmin Cho, Sung Ju Hwang, and Jong C. Park
    EMNLP Findings, 2023

29. Direct Fact Retrieval from Knowledge Graphs without Entity Linking
    Jinheon Baek†, Alham Fikri Aji, Jens Lehmann, and Sung Ju Hwang
    ACL, 2023

30. Phrase Retrieval for Open-Domain Conversational Question Answering
    with Conversational Dependency Modeling via Contrastive Learning
    Soyeong Jeong, Jinheon Baek, Sung Ju Hwang, and Jong C. Park
    ACL Findings, 2023

31. Knowledge-Augmented Language Model Prompting
    for Zero-Shot Knowledge Graph Question Answering
    Jinheon Baek†, Alham Fikri Aji, and Amir Saffari
    NLRSE Workshop @ ACL, 2023 **(Best Paper)**
    MATCHING Workshop @ ACL, 2023 **(Oral)**

32. Personalized Subgraph Federated Learning
    Jinheon Baek*, Wonyong Jeong*, Jiongdao Jin, Jaehong Yoon, and Sung Ju Hwang
    ICML, 2023

33. Realistic Conversational Question Answering with Answer Selection
    based on Calibrated Confidence and Uncertainty Measurement
    Soyeong Jeong, Jinheon Baek, Sung Ju Hwang, and Jong C. Park
    EACL, 2023

34. Graph Self-supervised Learning with Accurate Discrepancy Learning
    Dongki Kim*, Jinheon Baek*, and Sung Ju Hwang
    NeurIPS, 2022

35. Object Detection in Aerial Images with Uncertainty-Aware Graph Network
    Jongha Kim, Jinheon Baek, and Sung Ju Hwang
    VOLI Workshop @ ECCV, 2022

36. Knowledge Graph-Augmented Language Models for Knowledge-Grounded Dialogue Generation
Minki Kang*, Jin Myung Kwak*, Jinheon Baek*, and Sung Ju Hwang
KRLM Workshop @ ICML, 2022

37. KALA: Knowledge-Augmented Language Model Adaptation
Minki Kang*, Jinheon Baek*, and Sung Ju Hwang
NAACL, 2022 **(Oral)**

38. Augmenting Document Representations for Dense Retrieval with Interpolation and Perturbation
Soyeong Jeong, Jinheon Baek, Sukmin Cho, Sung Ju Hwang, and Jong C. Park
ACL, 2022 **(Oral)**

39. Edge Representation Learning with Hypergraphs
Jaehyeong Jo*, Jinheon Baek*, Seul Lee*, Dongki Kim, Minki Kang, and Sung Ju Hwang
NeurIPS, 2021

40. Task-Adaptive Neural Network Retrieval with Meta-Contrastive Learning
Wonyong Jeong*, Hayeon Lee*, Geon Park*, Eunyoung Hyung, Jinheon Baek, and Sung Ju Hwang
NeurIPS, 2021 **(Spotlight)**

41. Unsupervised Document Expansion for Information Retrieval with Stochastic Text Generation
Soyeong Jeong, Jinheon Baek, ChaeHun Park, and Jong C. Park
SDP Workshop @ NAACL, 2021 **(Oral)**

42. Accurate Learning of Graph Representations with Graph Multiset Pooling
Jinheon Baek*, Minki Kang*, and Sung Ju Hwang
ICLR, 2021

43. Exploring The Spatial Reasoning Ability of Neural Models in Human IQ Tests
Hyunjae Kim*, Yookyung Koh*, Jinheon Baek, and Jaewoo Kang
Neural Networks, 2021

44. Learning to Extrapolate Knowledge: Transductive Few-shot Out-of-Graph Link Prediction
Jinheon Baek, Dong Bok Lee, and Sung Ju Hwang
NeurIPS, 2020

Academic Activities

1.	Reviewer of NeurIPS	2021-2025
2.	Reviewer of ICML	2021-2025
3.	Reviewer of ICLR	2022-2026
4.	Reviewer of ACL ARR	2023-2025
5.	Reviewer of WWW	2024-2026
6.	Reviewer of AAAI	2025-2026
7.	Reviewer of TMLR	2022-2025
8.	Reviewer of COLM	2024-2025
9.	Reviewer of LoG	2022-2025
10.	Reviewer of TPAMI	2023, 2025

11.	Reviewer of ACM Computing Surveys	2025
12.	Reviewer of Pattern Recognition	2025
13.	Reviewer of Information Processing and Management	2025
14.	Reviewer of International Journal of Human-Computer Interaction	2025
15.	Reviewer of Expert Systems with Applications	2025
16.	Reviewer of Internet of Things	2024
17.	Reviewer of TKDE	2023-2024
18.	Reviewer of SMCA	2024
19.	Reviewer of T-IFS	2024
20.	Reviewer of EACL	2023
21.	Reviewer of TNNLS	2023
22.	Reviewer of TETCI	2023
23.	Reviewer of Big Data Research	2023
24.	Reviewer of AISD Workshop	2025
25.	Reviewer of NLRSE Workshop	2023-2024