Thesis n° 10 926

# Text-Graph Encoders and Retrieval-Augmented Generation

Presented on 5th December 2025

School of Engineering
L'IDIAP Laboratory
Doctoral program in Electrical Engineering

for the award of the degree of Docteur ès Sciences (PhD)

by


## Andrei Catalin COMAN



Accepted on the jury's recommendation

Prof. J.-Ph. Thiran, jury president
Prof. D. Gatica-Perez, Dr J. Henderson, thesis directors
Dr P. Minervini, examiner
Prof. M. Cochez, examiner
Prof. A. Alahi, examiner



École
polytechnique
fédérale
de Lausanne


2025

*"Change is the only constant."*
— Heraclitus

To all who shared their kindness and encouragement throughout this journey.

# Acknowledgements

Reaching this point has been a long journey made possible by the guidance, patience, and kindness of many people to whom I am deeply grateful.

I would like to begin by expressing my heartfelt gratitude to my supervisor, James Henderson, who has been an exceptional mentor throughout this thesis. His curiosity, clarity of thought, and patience have shaped my way of thinking about research and have guided me through every difficult turn. I am immensely thankful for his constant encouragement, his openness to new ideas, and his trust in my independence. Working with Jamie has been both a privilege and an education in itself, and I could not have imagined a better supervisor to learn from.

I would also like to thank my thesis committee for their time, careful evaluation, and constructive feedback during the review and defence of this thesis: Prof. Jean-Philippe Thiran, Prof. Daniel Gatica-Perez, Prof. Alexandre Alahi, Prof. Michael Cochez, and Dr. Pasquale Minervini. I am grateful to each of them for the care and attention with which they read the thesis and for the stimulating discussions during the viva. It was a privilege to have such a committee. A special thanks goes to my co-supervisor, Daniel, for his steady presence, kindness, and encouragement to pursue new academic experiences.

I gratefully acknowledge the Idiap Research Institute and École Polytechnique Fédérale de Lausanne (EPFL) for providing a stimulating and supportive environment in which to grow as a researcher. I would also like to thank the Swiss National Science Foundation (SNSF) for supporting my work through its funding, and Switzerland for being such an inspiring place to live and conduct research.

I am also thankful to Amazon for hosting me during my internships and for the opportunity to work in such a dynamic and collaborative environment. I am especially grateful to my manager, Adrià de Gispert, and to my mentors, Gianni Barlacchi and Ionuț-Teodor Sorodoc, for their guidance and encouragement. I would also like to thank everyone in the team for being so welcoming, supportive, and generous with their time. I have learned a great deal from each of them, and I will look back on this period as a defining experience in my professional growth.

I would like to thank my parents, Daniel and Emilia, for their constant support throughout this academic path. If I am in the position to write these lines today, it is only thanks to the many sacrifices they have made to allow me to pursue my studies. I have immense respect and gratitude for them, and I consider myself fortunate to have always been able to count on their guidance and affection. I would also like to thank my sister, Laura, who, no matter how much she grows up, will always be my little sister and the heart of our family.

I would also like to thank my extended Swiss family, Anne, Michel, Isabelle, Benoît, Mathilde, Pauline, Evan, Lily, Arvine, and Sherlock, for their warmth, generosity, and constant support. Their kindness and sense of togetherness have made me feel part of their family and truly at home in Switzerland. Through them, I have learned the importance of a cohesive and caring family, and I am deeply grateful for all the moments we have shared.

I would like to thank my friends in Italy, Mahed, Giacomo, Stefano, Lorenzo, Andrea, and Paolo, for their friendship and support throughout all these years. I am also especially grateful to Elia, who, even from afar, has always been present, genuinely caring, and a source of calm and perspective. I am confident that the bonds we have built over time are strong and will remain so for many years to come, no matter where life takes each of us.

I would like to thank all the wonderful people at Idiap, who have been much more than colleagues to me. Over the years, this group has become like a second family, one that made every stage of this journey more meaningful and enjoyable. I will refrain from listing names, as there are too many who have made these years special, and I would not want to overlook anyone. I am grateful for all the memorable times together, for the support during busy periods, and for the laughter and camaraderie that filled our time at work and beyond. The kindness, curiosity, and enthusiasm that define the Idiap community have shaped both my work and my perspective on research. I feel truly fortunate to have shared these years with such an inspiring and caring group of people.

A very special thanks goes to Fabio, my housemate and dear friend, who has been like the brother I never knew I needed. It is difficult to express how much your presence has meant to me over these years. Your kindness, sincerity, and determination have been a constant source of inspiration, and you have truly embodied the spirit of *"volere è potere"* and of *"kindness: free to give and invaluable to receive."* I am deeply grateful for your friendship and for the many moments, lessons, and memories we have shared along the way.

Above all, I would like to thank my dearest Chloé, whose unconditional love and support have been the pillar of this journey. You have been by my side through every challenge, making even the hardest moments brighter with your patience, warmth, and belief in me. I am deeply grateful for all the experiences we have shared and for the joy you bring into my life every day. I can't wait to discover all that life has in store for us and to keep building it together.

This thesis marks the end of one chapter and the beginning of another. I carry forward not only what I have learned, but also the kindness, patience, and inspiration of all those who have walked beside me. Thank you for being part of this journey.

# Abstract

Modern information-seeking systems increasingly rely on Large Language Models (LLMs) paired with external knowledge sources to generate accurate, context-aware responses. However, current Retrieval-Augmented Generation (RAG) pipelines face limitations in conversational question answering, including limited retrieval coverage and high reader (generator) latency, which affect answer quality and system efficiency. Evaluating responses grounded in retrieved evidence also presents challenges. Reward Models (RMs), typically trained on general preference data, are not equipped to distinguish between plausible responses based on parametric knowledge alone and those grounded in non-parametric evidence retrieved at inference time, as their training lacks retrieval-aware supervision and evidence-conditioned evaluation signals. Although the shift from parametric to non-parametric knowledge has enabled models to access more up-to-date and domain-specific information, most systems continue to rely solely on unstructured (text) inputs. This unstructured evidence often contains implicit, incomplete, or scattered information, making it difficult for models to piece together accurate responses. A promising direction is to incorporate structured non-parametric knowledge, such as knowledge graphs, which make relational information explicit and easier to leverage during inference. This requires architectures capable of jointly encoding unstructured and structured sources in a unified and effective manner.

This thesis explores methods to enhance RAG and text-graph encoders through two complementary lines of investigation. The first part focuses on improving the effectiveness and efficiency of retriever–reader pipelines, by introducing a lightweight reranking component between the retriever and reader, coupled with targeted fine-tuning steps that increase retrieval coverage and enable the reader to operate on fewer but more relevant passages. Additionally, this part proposes a methodology to adapt reward models for evaluating responses in the RAG setting. By repurposing existing question answering datasets into contextual preference pairs that reflect RAG-specific criteria, it enables the training of evaluation models that prioritise grounded, evidence-based answers over those relying solely on parametric knowledge. The second part investigates how attention-based models can be extended to incorporate explicit relational structure. It introduces a mechanism for jointly encoding unstructured (text) and structured (graph) information within a unified attention framework, enabling the model to draw on both types of input when building representations. This facilitates targeted and context-aware aggregation of information across multiple spans of text. This part further explores how these models can operate over knowledge graphs in inductive settings, where entities and relations encountered at inference time have not been seen during training. It shows that incorporating explicit relational information can reduce the reliance on text encoders to predict links between entities from unstructured input alone.

This thesis aims to show the potential of bringing together retrieval-based architectures and structured knowledge representations for building LLM-based information-seeking systems that can ground their generation in retrieved evidence, whether unstructured or structured, and access both types of information seamlessly via the attention mechanism.

**Keywords:** retrieval-augmented generation, conversational question answering, link prediction, retriever, reranker, reader, reward, text, graph, unstructured, structured, attention.

# Sommario

I moderni sistemi di ricerca di informazioni si affidano sempre più ai Modelli Linguistici di Grandi Dimensioni (Large Language Models, LLMs) abbinati a fonti di conoscenza esterne per generare risposte accurate e adatte al contesto informativo. Tuttavia, le attuali pipeline di Generazione Aumentata con Recupero (Retrieval-Augmented Generation, RAG) presentano limitazioni nel quadro delle conversazioni di tipo domanda e risposta, tra cui limitata copertura del recuperatore (retriever) e alta latenza del generatore (reader), che incidono sulla qualità delle risposte e sull'efficienza del sistema. La valutazione di risposte basate su elementi informativi ottenuti in seguito alla fase di recupero comporta ulteriori sfide. I Modelli di Ricompensa (Reward Models, RMs), tipicamente addestrati solamente su dati di preferenza generici, non sono in grado di distinguere tra risposte plausibili basate unicamente su conoscenza parametrica e risposte basate su elementi informativi non parametrici recuperati in fase di inferenza, poiché durante l'addestramento non ricevono una supervisione progettata per tenere conto del processo di recupero, né dispongono di segnali di valutazione costruiti in funzione degli elementi informativi disponibili. Sebbene il passaggio dalla conoscenza parametrica a quella non parametrica abbia permesso ai modelli di accedere a informazioni più aggiornate e specifiche di dominio, la maggior parte dei sistemi continua a basarsi esclusivamente su input non strutturati (testo). Questi elementi informativi non strutturati spesso contengono informazioni implicite, incomplete o frammentarie, rendendo difficile per i modelli comporre risposte accurate. Una direzione promettente consiste nell'integrare conoscenza non parametrica strutturata, come i grafi di conoscenza (Knowledge Graphs, KG), che rendono esplicite le relazioni e più facili da sfruttare in fase di inferenza. Ciò richiede architetture capaci di rappresentare congiuntamente fonti non strutturate e strutturate in maniera unificata ed efficace.

Questa tesi esplora metodi per migliorare RAG e codificatori testo–grafo su due linee di ricerca complementari. La prima parte si concentra sul miglioramento dell'efficacia e dell'efficienza delle pipeline recuperatore–generatore (retriever–reader), introducendo un modulo leggero di riordinamento (reranking, reranker) tra recuperatore e generatore, unito a fasi di messa a punto (fine-tuning) mirate ad aumentare la copertura del recuperatore e consentire al generatore di operare su un numero inferiore ma più rilevante di passaggi. Inoltre, questa parte propone una metodologia per adattare i RM alla valutazione di risposte nel contesto RAG. Riutilizzando dataset esistenti di domanda e risposta per creare coppie di preferenza contestualizzate secondo criteri specifici RAG, viene reso possibile l'addestramento di modelli di valutazione che privilegiano risposte basate su elementi informativi recuperati rispetto a quelle basate unicamente su conoscenza parametrica. La seconda parte indaga come i modelli basati sull'attenzione possano essere estesi per incorporare esplicitamente le strutture relazionali. Viene introdotto un meccanismo per rappresentare congiuntamente informazioni non strutturate (testo) e strutturate (grafo) all'interno di un unico quadro di riferimento di attenzione, consentendo al modello di attingere a entrambi i tipi di input nella costruzione delle rappresentazioni. Questo approccio consente di combinare e integrare in modo preciso e contestualmente pertinente le informazioni provenienti da più segmenti di testo. Questa parte mostra inoltre che l'integrazione di informazioni relazionali esplicite può ridurre la dipendenza dai codificatori di testo nel prevedere collegamenti (link prediction) tra entità sulla base di input esclusivamente non strutturati.

Questa tesi mira a mostrare il potenziale della convergenza tra architetture basate sul recupero e rappresentazioni di conoscenza strutturata per costruire sistemi LLM di ricerca di informazioni in grado di basare la generazione su elementi informativi recuperati, siano essi non strutturati o strutturati, e di accedere a entrambi i tipi di informazione in modo integrato tramite il meccanismo di attenzione.

**Parole chiave:** generazione aumentata con recupero (Retrieval-Augmented Generation, RAG), domanda e risposta conversazionale (conversational question answering), predizione di collegamenti (link prediction), recuperatore (retriever), riordinatore (reranker), generatore (reader), modelli di ricompensa (Reward Models, RMs), testo, grafo, non strutturato, strutturato, attenzione (attention).

# Contents

# 1 Motivation

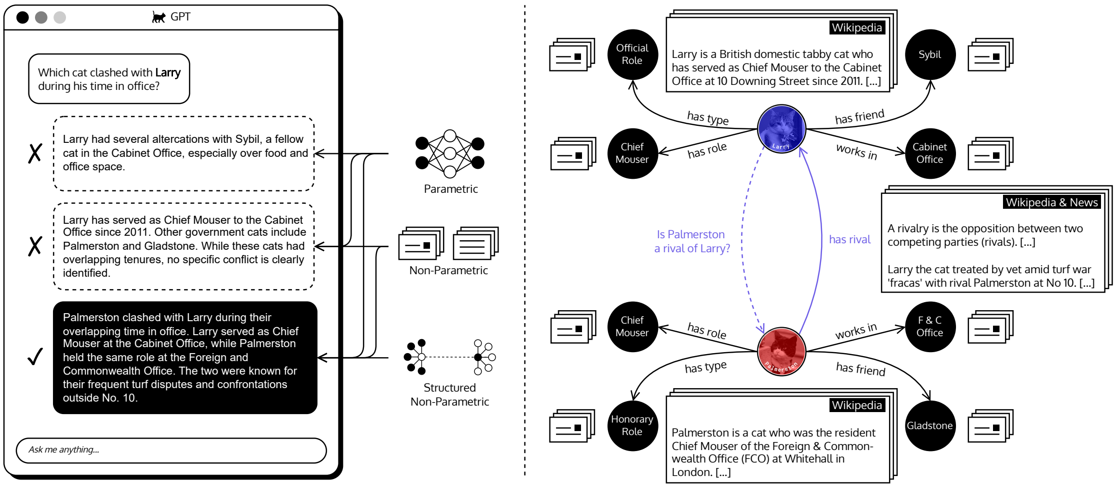



Figure 1.1: Answering a relational question with increasing levels of system capability. The question *"Which cat clashed with Larry during his time in office?"* serves as a running example to illustrate three distinct approaches. **Parametric:** A language model relying solely on internal knowledge fails to answer correctly, suggesting Sybil based on superficial association rather than grounded evidence. **Non-Parametric:** Retrieved unstructured text introduces relevant facts about Larry and Palmerston, such as their overlapping tenures, but the connection between them remains unstated, so the system cannot identify the correct answer. **Structured Non-Parametric:** Structured data, represented as a knowledge graph, enables the system to infer the unstated rivalry between Larry and Palmerston via link prediction. This allows it to produce the correct answer.

## 1.1 Unstructured and Structured Sources for Effective Information-Seeking Systems

We live in an age where automated systems are becoming central to how people access and interact with information. In particular, information-seeking conversations, where a user engages in a natural back and forth dialogue with a system (Oddy, 1977; Croft and Thompson, 1987; Belkin et al., 1995), asking follow-up questions, clarifying their intent, and expecting responses that are both contextually relevant and grounded in external evidence, have moved beyond speculative research into practical, everyday use. They are now embedded in tools that support open-ended exploration across a wide range of domains (Perplexity AI, 2025; OpenAI, 2025; Gemini Team, 2025b).

A prominent task within this broader setting is Open-Domain Conversational Question Answering (ODConvQA) (Qu et al., 2020a), which has gained increasing attention with the rise of Large Language Models (LLMs) capable of generating fluent responses conditioned on user input (Ouyang et al., 2022; Anthropic, 2023; Touvron et al., 2023; Gemini Team, 2025a; DeepSeek-AI et al., 2024). These models are often paired with retrieval components that provide access to external sources of knowledge, enabling them to respond not only based on what they have been trained on, but also on information retrieved at inference time (Fan et al., 2024). A key paradigm enabling this capability is Retrieval-Augmented Generation (RAG) (Lewis et al., 2020b), which allows LLMs to incorporate up-to-date and domain-specific knowledge, and to support open-ended question answering beyond the limitations of static search.

Figure 1.1 illustrates how seemingly straightforward questions can expose fundamental challenges in building effective information-seeking systems. Parametric models often rely on superficial associations and lack the means to verify or contextualise their responses. Introducing retrieval offers access to external evidence. However, when that evidence is drawn from unstructured text, it can still fall short, as relevant facts may be scattered, implicit, or only loosely connected. Structured sources help to surface these latent relationships, although they also introduce challenges related to representation and integration. Beyond retrieving and organising information, effective systems must also align their responses with the user's intent and the supporting evidence.

## 1.2 Research Questions and Contributions

Following the challenges outlined in Figure 1.1, this thesis focuses on how systems retrieve relevant content, align responses with both the retrieved evidence and the user's intent, and combine unstructured text with structured information. These challenges are explored through four research questions, grouped into two complementary lines of investigation: **Retrieval-Augmented Generation** and **Text-Graph Encoders**. An overview is shown in Figure 1.2.

### 1.2.1 Research Question 1: *What limits the effectiveness of retrieval-based conversational question answering pipelines, and how can these be addressed?*

Chapter 4 examines why widely adopted retriever-reader pipelines such as Dense Passage Retrieval (DPR) retriever (Karpukhin et al., 2020) and Fusion-in-Decoder (FiD) reader (Izacard and Grave, 2021b) underperform in ODConvQA. It identifies key issues including sensitivity to noisy inputs, limited retrieval coverage, and high reader latency. The chapter introduces Retriever-Reranker-Reader Fine-tuning (*R3FINE*), a simple strategy that combines fast reranking with targeted fine-tuning of the retriever and reader, delivering stronger QA performance with significantly lower inference time.

### 1.2.2 Research Question 2: *How can reward models be extended to evaluate responses in the RAG setting?*

Chapter 5 addresses the mismatch between conventional Reward Models (RMs) (Bai et al., 2022a), which are typically trained on general preference data, and the distinct evaluation challenges of the RAG setting, where responses must be evaluated with respect to retrieved evidence rather than parametric knowledge alone. It introduces *RAGferee*, a methodology that repurposes QA datasets into preference pairs designed to train contextual RMs. These models focus on RAG-specific dimensions such as faithfulness, completeness, conciseness, appropriate refusals, and relevance to the user query. Despite being trained on a small, carefully curated dataset, the *RAGferee* RMs demonstrate significantly stronger performance than larger RMs trained on extensive general corpora, highlighting the value of supervision that prioritises contextual relevance (based on non-parametric knowledge) over stylistic features (driven by parametric knowledge).

### 1.2.3 Research Question 3: *How can attention-based models learn to pool information from unstructured inputs using explicit structured relational information?*

Chapter 7 examines the challenge of aggregating information from unstructured inputs in settings where relevant content is distributed across multiple spans of text. Existing approaches rely heavily on hand-crafted pooling heuristics, which are typically fixed, not data-driven, and difficult to transfer across tasks or models. This chapter introduces a Graph-Assisted Declarative Pooling method (*GADePo*) that augments attention-based models with special aggregation tokens and explicit graph relations. By integrating structural relations directly into the attention computation, *GADePo* enables pooling to be guided by domain knowledge while still being learned from data, demonstrating how attention mechanisms can serve as a unifying interface between unstructured input and explicit structured relational information.

### 1.2.4 Research Question 4: *How can models make use of textual and relational information to operate over structured knowledge in inductive settings?*

Chapter 8 addresses the challenge of operating over structured knowledge in inductive settings, where both entities (nodes) and relations (edges) may be unseen in the training data. It introduces Fast-and-Frugal Text-Graph Transformers (*FnF-TG*), a model designed to integrate unstructured textual descriptions and structured relational data through attention mechanisms for inductive link prediction. *FnF-TG* encodes text-attributed ego-graphs and constructs embeddings for both entities and relations from their textual descriptions, allowing it to handle unseen elements without relying on fixed vocabularies or representations.

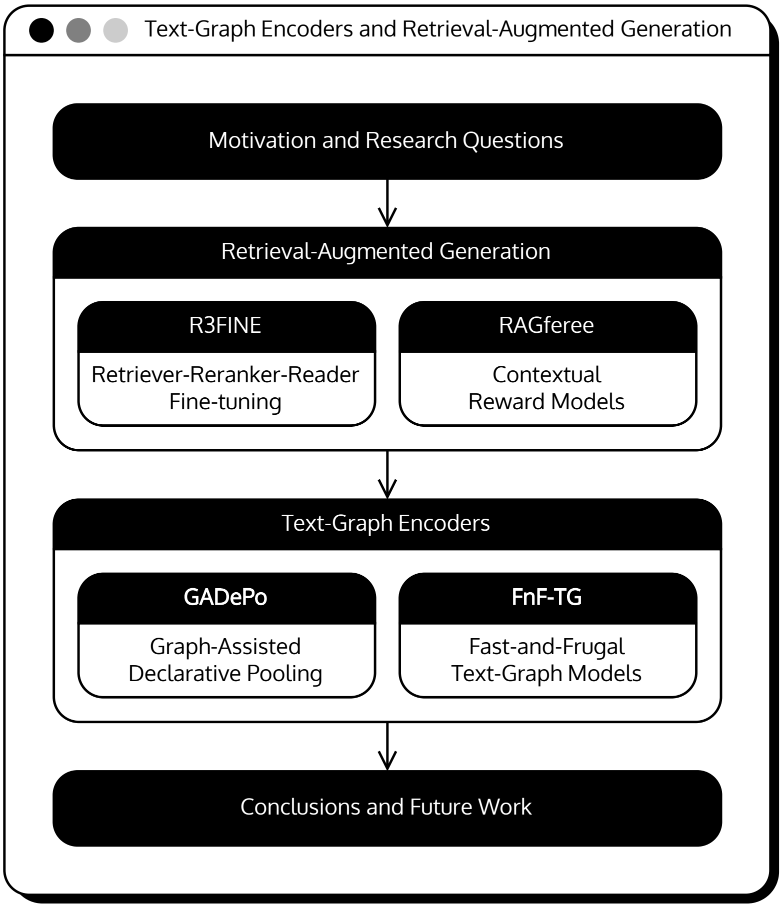



Figure 1.2: Schematic overview of the thesis. The work is organised into two complementary lines of investigation: **Retrieval-Augmented Generation**, which examines the limitations of retrieval-based QA pipelines and the extension of Reward Models to better evaluate responses in the RAG setting (Chapter 4 and Chapter 5); and **Text-Graph Encoders**, which explores how attention-based models can learn to pool information from unstructured inputs using explicit structured relational information, and how models can operate over structured knowledge in inductive settings using both textual and relational information (Chapter 7 and Chapter 8).

# 2 Associated Publications

## 2.1 Thesis Publications

This thesis is based on the following publications:

- **Part I: Retrieval-Augmented Generation**
  - **Chapter 4:** Coman, A., Barlacchi, G., and de Gispert, A. (2023). Strong and efficient baselines for open domain conversational question answering. In Bouamor, H., Pino, J., and Bali, K., editors, *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 6305–6314, Singapore. Association for Computational Linguistics
  - **Chapter 5:** Coman, A. C., Sorodoc, I.-T., Ribeiro, L. F. R., Byrne, B., Henderson, J., and de Gispert, A. (2025a). Ragferee: Building contextual reward models for retrieval-augmented generation
- **Part II: Text-Graph Encoders**
  - **Chapter 7:** Coman, A., Theodoropoulos, C., Moens, M.-F., and Henderson, J. (2024). GADePo: Graph-assisted declarative pooling transformers for document-level relation extraction. In Yu, W., Shi, W., Yasunaga, M., Jiang, M., Zhu, C., Hajishirzi, H., Zettlemoyer, L., and Zhang, Z., editors, *Proceedings of the 3rd Workshop on Knowledge Augmented Methods for NLP*, pages 1–14, Bangkok, Thailand. Association for Computational Linguistics

    Henderson, J., Mohammadshahi, A., Coman, A., and Miculicich, L. (2023). Transformers as graph-to-graph models. In Elazar, Y., Ettinger, A., Kassner, N., Ruder, S., and A. Smith, N., editors, *Proceedings of the Big Picture Workshop*, pages 93–107, Singapore. Association for Computational Linguistics
  - **Chapter 8:** Coman, A. C., Theodoropoulos, C., Moens, M.-F., and Henderson, J. (2025b). Fast-and-frugal text-graph transformers are effective link predictors. In Che, W., Nabende, J., Shutova, E., and Pilehvar, M. T., editors, *Findings of the Association for Computational Linguistics: ACL 2025*, pages 11828–11841, Vienna, Austria. Association for Computational Linguistics

## 2.2 Other Publications

Outside the scope of this thesis, contributions were also made to the following publications:

- Theodoropoulos, C., Henderson, J., Coman, A. C., and Moens, M.-F. (2021). Imposing relation structure in language-model embeddings using contrastive learning. In Bisazza, A. and Abend, O., editors, *Proceedings of the 25th Conference on Computational Natural Language Learning*, pages 337–348, Online. Association for Computational Linguistics
- Theodoropoulos, C., Coman, A. C., Henderson, J., and Moens, M.-F. (2024). Enhancing biomedical knowledge discovery for diseases: An open-source framework applied on rett syndrome and alzheimer's disease. *IEEE Access*, 12:180652–180673
- Fehr, F. J., Baia, A. E., Chang, X., Coman, A. C., Hajal, K. E., Zein, D. E., Kumar, S., Gomez, J. P. Z., Cavallaro, A., Teney, D., and Henderson, J. (2025). Fine-tuning pretrained models with NVIB for improved generalisation. In *Workshop on Spurious Correlation and Shortcut Learning: Foundations and Solutions*
- Theodoropoulos, C., Coman, A. C., Henderson, J., and Moens, M.-F. (2025). Reduction of supervision for biomedical knowledge discovery. *BMC Bioinformatics*, 26

# Part I
# Retrieval-Augmented Generation

# 3 Retrieve, Augment, and Generate

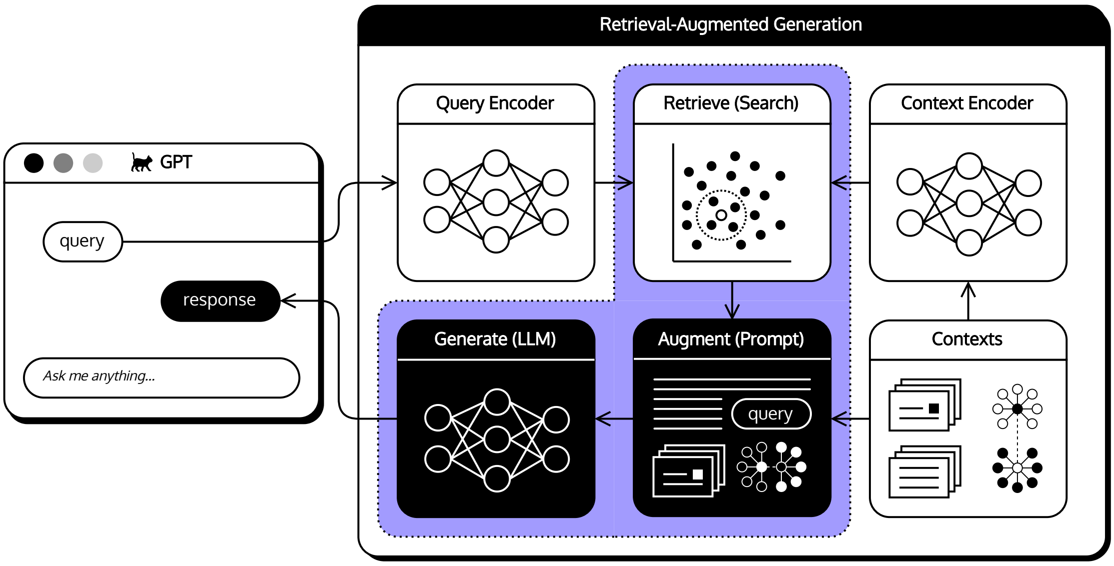



Figure 3.1: Diagram of the Retrieval-Augmented Generation (RAG) architecture. In the **retrieve** component, the query and contexts are encoded using the *Query Encoder* and *Context Encoder* components shown in white, and relevant contexts are selected from an external corpus via the ***Retrieve*** *(Search)* component, also shown in white. The ***Augment*** *(Prompt)* component, shown in black, combines the query with the retrieved contexts to form a single input. The ***Generate*** *(LLM)* component, also shown in black, uses a generative model that conditions on this input to produce the final output/response.

Modern language models have achieved impressive results across a wide range of natural language processing tasks by storing world knowledge implicitly within their parameters (Petroni et al., 2019; Roberts et al., 2020; Raffel et al., 2020; Safavi and Koutra, 2021; Yu et al., 2024). While effective in many settings, this parametric approach has inherent limitations when models are required to produce specific, up-to-date, or rare factual content. In such cases, the inability to revise or inspect the internal knowledge representation can lead to outputs that are factually inaccurate or unverifiable, a phenomenon commonly referred to as hallucination (Tonmoy et al., 2024). These challenges motivate hybrid architectures that supplement parametric knowledge with access to explicit, external sources of information (Guu et al., 2020; Karpukhin et al., 2020; Petroni et al., 2020; Borgeaud et al., 2022; Rashiti et al., 2024).

This chapter provides background on the Retrieval-Augmented Generation (RAG) architecture (Figure 3.1). This pipeline enables language models to dynamically incorporate retrieved evidence into the generation process by decomposing the workflow into three components: **retrieval** of relevant contexts from a non-parametric memory based on the query, **augmentation** of the input with retrieved contexts, and **generation** of a final output/response, conditioned on both the query and supporting evidence.

More formally, let $q \in Q$ denote the input query, and let $r = (r_1, r_2, \ldots, r_N)$ represent the target response sequence. The system also has access to a corpus $C$ of external contexts. The objective is to model the conditional distribution $p(r \mid q)$, where the generation is guided by the selection of relevant contexts $c \in C$, retrieved at inference time.

## 3.1 Retrieval

The retrieval component typically employs a dual encoder architecture in which the query and external contexts are encoded independently. As illustrated in Figure 3.1 and Figure 3.2, the query $q$ is passed through a *Query Encoder* function $f_{\text{Query Encoder}}$ to obtain a dense representation $\mathbf{q} = f_{\text{query}}(q)$. Simultaneously, each context $c \in C$ from the external corpus is encoded using a separate *Context Encoder* function $f_{\text{Context Encoder}}$, producing $\mathbf{c} = f_{\text{context}}(c)$.

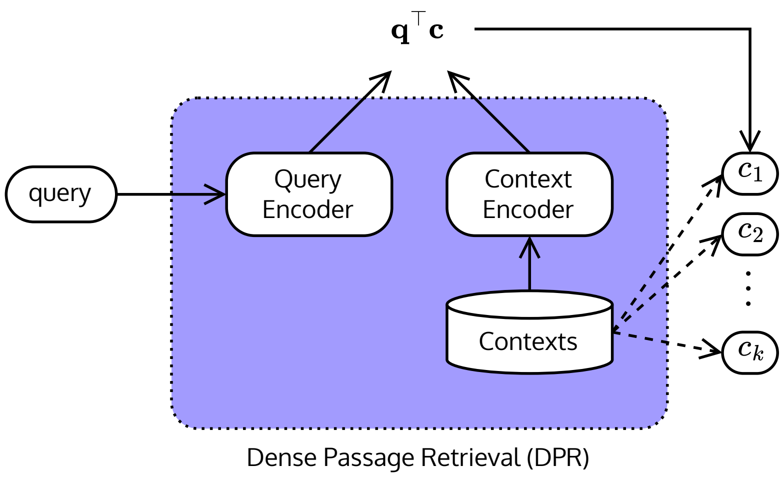



Figure 3.2: Dense Passage Retrieval (DPR) (Karpukhin et al., 2020): Dual encoder architecture for retrieval. The query and candidate contexts are independently encoded into dense representations using separate encoders. The inner product between the query and context embeddings is used to rank relevant contexts.

A relevance score between the query and each context is computed via the inner product of their dense representations:

$$p(c \mid q) \propto \mathbf{q}^\top \mathbf{c}.$$

The top $k$ most relevant contexts, denoted $\{c_1, c_2, \ldots, c_k\} \subset C$, are selected for the next stage in the pipeline. Since this retrieval must be performed potentially over millions (or even billions) of contexts, the process is implemented via Maximum Inner Product Search (MIPS), often relying on approximate nearest neighbour methods to ensure efficiency at scale (Johnson et al., 2021).

A typical realisation of this retrieval mechanism is Dense Passage Retrieval (DPR) (Karpukhin et al., 2020), shown in Figure 3.2. DPR uses two independently parameterised BERT-based (Devlin et al., 2019) encoders for queries and contexts, trained using a contrastive objective that brings relevant query–context pairs closer together while pushing apart the non-relevant ones.

## 3.2 Augmentation and Generation

Once the top $k$ relevant contexts $\{c_1, c_2, \dots, c_k\} \subset C$ have been retrieved, the next stage augments the query $q$ with the supporting evidence to form the input to the generative model. Each retrieved context $c_i$ is paired with the original query $q$ to create a set of concatenated inputs $\{(q, c_1), (q, c_2), \dots, (q, c_k)\}$. Since the original ground truth context is unknown at inference time, the generation process treats the contexts as latent variables and marginalises over them to model the conditional distribution over the full response $r = (r_1, r_2, \dots, r_N)$, as follows:

$$p(r \mid q) = \sum_{i=1}^{k} p(c_i, q) \cdot p(r \mid q, c_i),$$

where $p(c_i, q)$ comes from the retriever. The generative model then produces the response autoregressively, conditioning on the query $q$ and each context $c_i$, as follows:

$$p(r \mid q, c_i) = \prod_{j=1}^{N} p(r_j \mid q, c_i, r_{<j}).$$

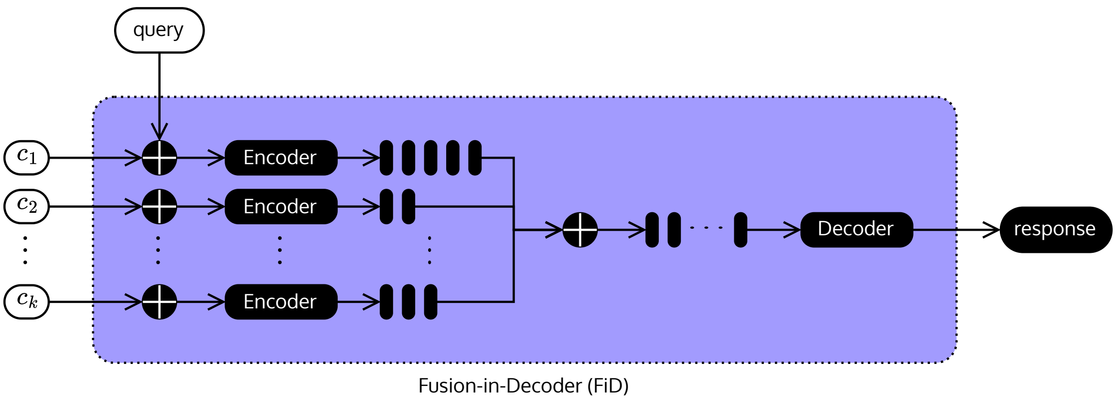



Figure 3.3: Fusion-in-Decoder (FiD) (Izacard and Grave, 2021b): *Encoder–Decoder* architecture for generation. Each augmented query–context pair is independently encoded via the *Encoder*. The *Decoder* then jointly attends to all encoded representations to generate the output/response sequence.

A common realisation of the generation component is Fusion-in-Decoder (FiD) (Izacard and Grave, 2021b), shown in Figure 3.3. FiD is typically implemented using an encoder-decoder architecture such as T5 (Raffel et al., 2020) or BART (Lewis et al., 2020a), where each input pair $(q, c_i)$ is independently encoded via the *Encoder* function $f_{\text{Encoder}}$, and the resulting representations are fused inside a shared *Decoder* function $f_{\text{Decoder}}$ via cross-attention. At each decoding step, the *Decoder* generates token $r_j$ based on the query $q$, the retrieved contexts $\{c_1, c_2, \ldots, c_k\} \subset C$, and the previously generated tokens $r_{<j}$, as follows:

$$p(r_j \mid q, \{c_1, c_2, \ldots, c_k\}, r_{<j}) = f_{\text{Decoder}}(q, \{c_1, c_2, \ldots, c_k\}, r_{<j}).$$

This formulation enables the model to condition on multiple retrieved contexts simultaneously while generating a single output sequence. Since the *Decoder* attends to all contexts representations jointly and generates tokens autoregressively, the number of decoding steps depends only on the length of the output response $r$, not on the number of retrieved contexts $k$.

# 4 *R3FINE*: Retriever-Reranker-Reader Fine-tuning

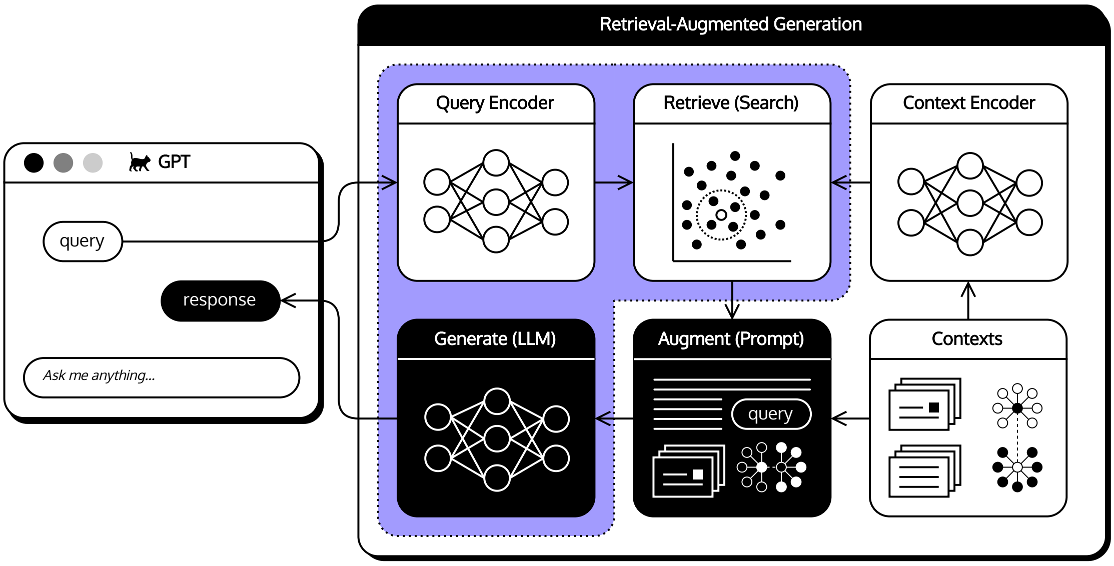



Figure 4.1: This chapter focuses on the *Query Encoder*, *Retrieve (Search)*, and *Generate (LLM)* components of the RAG pipeline from Chapter 3. It examines how limitations in the retriever and reader (generator) affect overall performance on Open Domain Conversational Question Answering (ODConvQA) tasks. In particular, it highlights the retriever's limited coverage and the reader's lack of robustness to noisy inputs. The chapter addresses these issues by introducing a lightweight reranking component and applying targeted fine-tuning to both retriever and reader. In the context of the Larry and Palmerston example from Figure 1.1, this approach could in principle help surface more relevant contextual evidence, including their overlapping tenures and documented confrontations, thereby enabling the system to infer relationships such as their rivalry, which might not be surfaced by the standard RAG pipeline.

## 4.1 Research Question

*What limits the effectiveness of retrieval-based conversational question answering pipelines, and how can these be addressed?*

## 4.2 Publication

Coman, A., Barlacchi, G., and de Gispert, A. (2023). Strong and efficient baselines for open domain conversational question answering. In Bouamor, H., Pino, J., and Bali, K., editors, *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 6305–6314, Singapore. Association for Computational Linguistics

## 4.3 Summary

Unlike the Open Domain Question Answering (ODQA) setting, the conversational (ODConvQA) domain has received limited attention when it comes to re-evaluating baseline pipelines for both effectiveness and efficiency. Following their introduction in Chapter 3, this chapter examines the commonly used pipeline (Figure 4.1) of Dense Passage Retrieval (DPR) retriever (Karpukhin et al., 2020) and Fusion-in-Decoder (FiD) reader (Izacard and Grave, 2021b), and shows that it significantly underperforms on ODConvQA tasks due to several key limitations. Some of these include retriever's limited coverage, reader's lack of robustness to noisy inputs, and the overall latency of the system when processing large numbers of passages. To address these limitations, the chapter proposes and evaluates strong yet simple and efficient baselines by introducing a fast reranking component between the retriever and the reader, and by performing targeted fine-tuning of the two. Experiments on two ODConvQA datasets, namely TopiOCQA (Adlakha et al., 2022) and OR-QuAC (Qu et al., 2020b), show that these modifications outperform existing approaches while reducing reader's latency by 60%. These findings offer practical insights into how retrieval-based pipelines can be made both stronger and more efficient for ODConvQA, without increasing model complexity.

## 4.4 Introduction

In an automated information-seeking conversation scenario between two parties, the human questioner asks a series of questions and expects to receive a relevant response from the answering system (Oddy, 1977; Croft and Thompson, 1987; Belkin et al., 1995). A widely adopted setup, models the answerer using two neural components: Dense Passage Retrieval (DPR) (Karpukhin et al., 2020) as the retriever, and Fusion-in-Decoder (FiD) (Izacard and Grave, 2021b) as the reader. This combination has proven effective in overcoming several shortcomings of earlier sparse and extractive methods, such as their reliance on lexical heuristics and their limitation to extracting spans from text (Chen et al., 2017; Yang et al., 2019; Lee et al., 2019; Das et al., 2019; Guu et al., 2020; Lewis et al., 2020b; Shen et al., 2023). Promising advances in this space have explored improved training strategies (Guu et al., 2020; Balachandran et al., 2021; Qu et al., 2021), the incorporation of rerankers (Hu et al., 2019; Mao et al., 2021; Barlacchi et al., 2022; Iyer et al., 2021; Glass et al., 2022), question rewriting (Vakulenko et al., 2021; Del Tredici et al., 2021), knowledge distillation from reader to retriever (Izacard and Grave, 2021a), memory-efficient architectures (Izacard et al., 2020; Del Tredici et al., 2022), and the integration of structured information (Min et al., 2019; Yu et al., 2022).

Unlike the Open Domain Question Answering (ODQA) setting, the conversational variant (ODConvQA) has received comparatively little attention in terms of re-evaluating baseline pipelines for both effectiveness and efficiency. In this chapter, we focus on the typical DPR retriever and FiD reader (DPR+FiD) pipeline, and show its limitations when applied to the ODConvQA setting. Despite its popularity, we find that this baseline significantly underperforms when fine-tuned on downstream tasks. We show that simple improvements in the training, architecture, and inference setups of the DPR+FiD pipeline, provide a strong and efficient baseline that outperforms existing approaches on two common ODConvQA datasets: TopiOCQA (Adlakha et al., 2022) and ORConvQA (OR-QUAC) (Qu et al., 2020b).

We point out several limitations of the pipeline, such as:

1) *reader's susceptibility to noisy input*, 2) *retriever's reduced coverage*, 3) *retriever's lack of cross semantic encoding between the conversation and the retrieved passages*, and 4) *reader's latency is heavily impacted by the number of input passages*. To mitigate these, we propose and evaluate a simple and effective approach by including a fast reranking component between the retriever and the reader, and by performing targeted fine-tuning steps. The proposed Retriever-Reranker-Reader Fine-tuning (*R3FINE*) strategy leads to baseline models with a better latency/performance trade-off. These baselines, which are simple and easy to replicate, serve as a reference point for comparing new and more complex models, and for determining their effectiveness.

Our **contributions** are as follows:

- We identify and address several limitations of the typical pipeline used in ODConvQA.
- We propose the *R3FINE* strategy, which outperforms existing approaches on two common datasets and reduces pipeline's latency by 60%.
- We provide new and valuable insights for creating simple and efficient baselines, which serve as a reference point for future comparison with more complex approaches.

## 4.5 End-to-End Baselines for ODConvQA

Figure 4.2 presents an instantiation of the RAG pipeline described in Chapter 3, adapted for the ODConvQA setting. It consists of a DPR-based retriever, an intermediate reranker, and an FiD-based reader.

A conversation history $c_1, c_2, \ldots, c_N$ is input to the DPR retriever, which applies $f_{\text{Conversation Encoder}}$ to compute a dense representation $\mathbf{c} = f_{\text{Conversation Encoder}}(c_1, c_2, \ldots, c_N)$. Next, a collection of passages $p_1, p_2, \ldots, p_N$, defined as text blocks from an external knowledge source (e.g., Wikipedia) and serving as basic retrieval units, is encoded into dense representations $\mathbf{p}_i = f_{\text{Passage Encoder}}(p_i)$. Dense retrieval is then performed using Maximum Inner Product Search (MIPS), which ranks passages based on the similarity between the conversation embedding $\mathbf{c}$ and each passage embedding $\mathbf{p}_i$, computed as $\mathbf{c}^\top \mathbf{p}_i$.

The top-$k$ passages retrieved by the DPR retriever are each paired with the conversation history to form input pairs $(c, p_1), (c, p_2), \ldots, (c, p_k)$, which are independently encoded by the FiD reader's *Encoder* via $f_{\text{Encoder}}(c, p_i)$. The resulting representations are concatenated as $f_{\text{Encoder}}(c, p_1) \oplus f_{\text{Encoder}}(c, p_2) \oplus \cdots \oplus f_{\text{Encoder}}(c, p_k)$ and passed to the FiD's *Decoder* component, which generates the final answer $\mathbf{a} = f_{\text{Decoder}}(f_{\text{Encoder}}(c, p_1) \oplus \cdots \oplus f_{\text{Encoder}}(c, p_k))$.

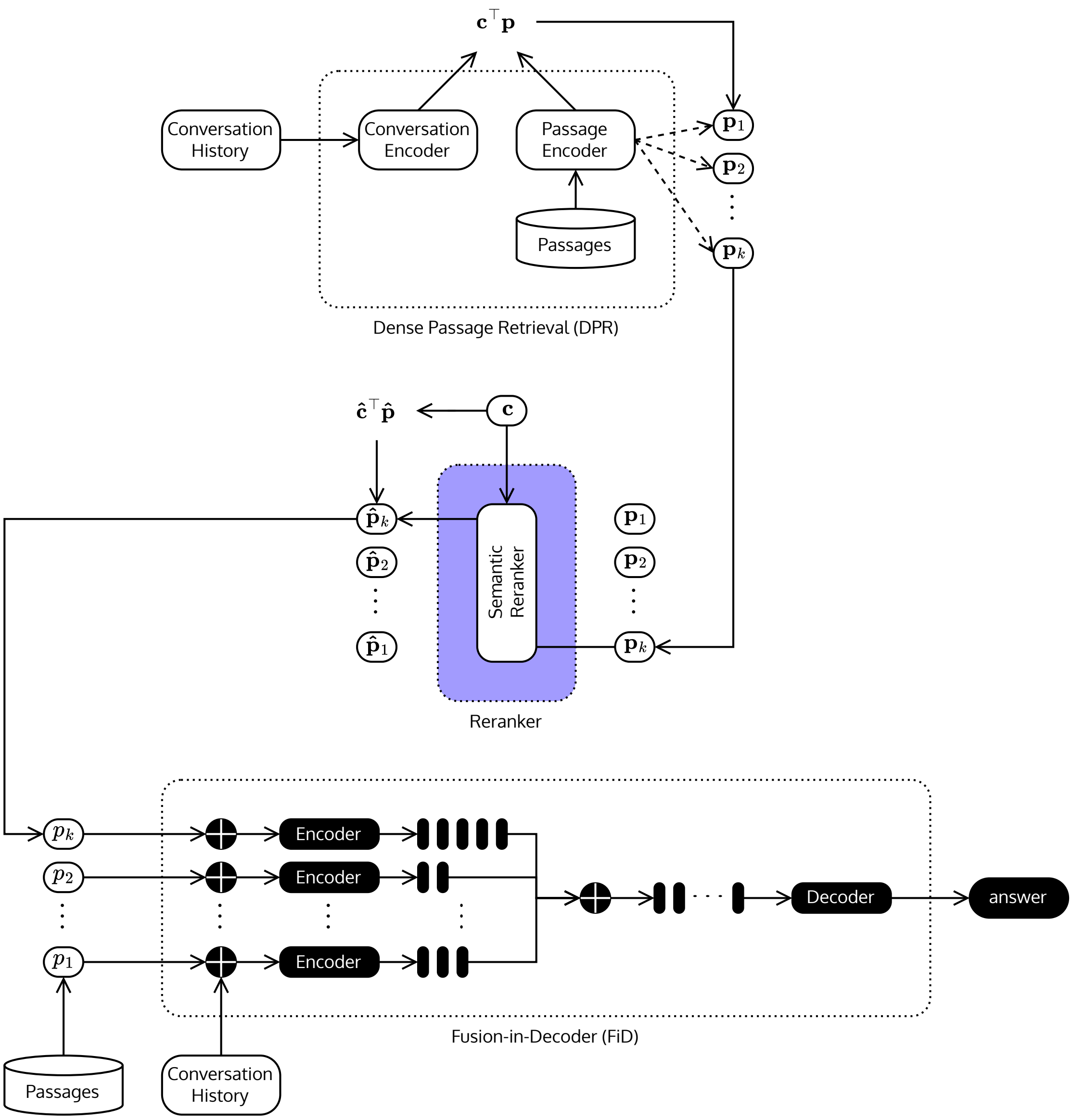



Figure 4.2: The Retriever-Reranker-Reader Fine-tuning (*R3FINE*) pipeline. A DPR retriever encodes the conversation history to retrieve relevant passages from an external knowledge source. Retrieved passages are then jointly encoded with the conversation history by a *Semantic Reranker* (SR), which re-estimates their relevance. Finally, the top-ranked passages are passed to the FiD reader, which generates the final answer.

## 4.6 Experimental Setup

We focus on two main datasets. TOPIOCQA (Adlakha et al., 2022) is a large-scale open-domain information-seeking conversational dataset, characterised by the presence of topic switching, which poses a notable challenge. OR-QUAC (Qu et al., 2020b) builds upon context independent question rewrites from the CANARD dataset (Elgohary et al., 2019), and adapts them to the open-domain setting defined by the QuAC dataset (Choi et al., 2018). Further details regarding the datasets are provided in Appendix A.

We make use of the DPR and FiD models provided with the TOPIOCQA dataset. At the time of writing, only the training and development sets are available for TOPIOCQA. We apply its DPR module to both datasets. However, since OR-QUAC is of extractive type, in contrast to the generative nature of TOPIOCQA, we train the FiD reader from scratch using the same configuration as for TOPIOCQA.

We report performance using two standard metrics in QA: Exact Match (EM) and F1 score. EM measures the percentage of predictions that match any one of the ground-truth answers exactly, making it a strict indicator of correctness. In contrast, F1 provides a softer evaluation by computing the harmonic mean of precision and recall over word overlaps between the predicted and ground-truth answers.

## 4.7 Strong Baseline Models

We outline a number of limitations of the DPR+FiD pipeline, along with suggestions on how to mitigate them. While some of those interconnect at different levels the various efforts made in the ODQA domain (Balachandran et al., 2021; Yu et al., 2022), our goal is to offer a perspective on the ODConvQA setting.

### 4.7.1 Limitations and Bottlenecks

**Reader's susceptibility to noisy input.** Previous work has shown that the performance of the FiD reader improves significantly as the number of retrieved passages increases (Izacard and Grave, 2021b). While this observation is confirmed in our findings, Table 4.1 also offers a complementary perspective. Specifically, we observe that when the reader is provided with the relevant (i.e., gold) passage, its performance deteriorates as the number of retrieved passages increases. This indicates a trade-off in the reader's input: when the gold passage is present, a smaller and more focused list is preferable, whereas in its absence, including more passages increases the likelihood of useful information being available. This analysis is limited to the TOPIOCQA dataset as it is the only one that provides gold passage annotations for the development set.

| | TopiOCQA | | | |
|---|---|---|---|---|
| | w/o gold | | w/ gold | |
| top-$k$ | EM | F1 | EM | F1 |
| 1 | 19.3 | 37.6 | 38.3 | 65.5 |
| 5 | 27.0 | 49.6 | 36.3 | 62.5 |
| 10 | 29.8 | 52.4 | 35.8 | 61.5 |
| 20 | 31.3 | 54.0 | 36.2 | 60.8 |
| 50 | 33.0 | 55.1 | 35.9 | 59.5 |

Table 4.1: FiD reader performance (Exact Match and F1 scores) on the TopiOCQA development set, with and without the gold passage (w/ gold and w/o gold) in the top-$k$ limit.

**Retriever's reduced coverage.** Typical solutions impose a hard top-$k$ limit on the number of passages returned by the DPR retriever, implicitly assuming that relevant content is captured within this restricted set. Table 4.1 shows that retrieval coverage is critical for effective downstream performance by the reader. To address this limitation, we propose introducing a simple and efficient Transformer-based (Vaswani et al., 2017) reranker after the retriever. This component, shown in Figure 4.2 and described in the next paragraph, is designed to reconsider a larger pool of passages returned by the DPR retriever and to provide the FiD reader with a reduced and improved list of passages. Since this module operates at the semantic level, specifically over the induced representations produced by the DPR's *Passage Encoder* function $f_{\text{Passage Encoder}}$, we refer to it as the *Semantic Reranker* (SR). Table 4.2 shows the potential coverage margins and the retrieval results obtained after the introduction of the *Semantic Reranker* module, when a larger number of passages (50 vs. 1000) is considered.

| | TopiOCQA | | OR-QuAC | |
|---|---|---|---|---|
| top-$k$ | w/o SR | w/ SR | w/o SR | w/ SR |
| 1 | 24.66 | 42.64 | 29.27 | 56.64 |
| 5 | 51.87 | 68.62 | 51.63 | 70.82 |
| 10 | 62.45 | 75.62 | 57.14 | 72.27 |
| 20 | 70.21 | 80.75 | 61.37 | 72.80 |
| 50 | 77.41 | 84.69 | 65.22 | 73.53 |
| 500 | 89.58 | 90.73 | 72.97 | 74.34 |
| 1000 | 91.49 | 91.49 | 74.55 | 74.55 |

Table 4.2: Development set retrieval coverage before and after the introduction of the *Semantic Reranker* component (w/o SR and w/ SR) when a larger number of passages (50 vs. 1000) is considered. For TopiOCQA, we report the presence of the gold passage within the top-$k$ limit. For OR-QuAC we report the presence of the gold answer within the top-$k$ limit. Extra retrieval results on the TopiOCQA's training set and the OR-QuAC's test set are reported in Appendix A.

**Retriever's lack of cross-semantic encoding between the conversation and the retrieved passages.** The DPR retriever uses a dual encoder architecture in which the conversation history and the passages are encoded independently by two separate functions: $f_{\text{Conversation Encoder}}$ and $f_{\text{Passage Encoder}}$. This design, while efficient for retrieval, inherently prevents the model from capturing cross-attentive/joint semantic interactions between the dense representations of the conversation history and the retrieved passages during encoding. As a result, the retriever is unable to exploit semantic relationships among passages or assess their relevance in the broader conversational context. To address this limitation, we introduce a *Semantic Reranker*, implemented as a Transformer encoder. This module jointly processes the dense representations of the conversation history $\mathbf{c}$ and the retrieved passages $\mathbf{p}_1, \ldots, \mathbf{p}_N$, enabling cross-attention among them:

$$\hat{\mathbf{c}}, \hat{\mathbf{p}}_1, \ldots, \hat{\mathbf{p}}_N = \textit{Semantic Reranker}(\mathbf{c}, \mathbf{p}_1, \ldots, \mathbf{p}_N).$$

Each input representation attends to the others, allowing for richer contextualisation. Ranking is then performed over the transformed sequence $\hat{\mathbf{c}}, \hat{\mathbf{p}}_1, \ldots, \hat{\mathbf{p}}_N$, based on the inner product $\hat{\mathbf{c}}^\top \hat{\mathbf{p}}_i$.

**Reader's latency is heavily impacted by the number of input passages.** Figure 4.3 shows that reader's latency can be significantly reduced by decreasing the number of input passages. However, a trivial limitation to top-$k$ considerably degrades the performance of the module, thus leading to an inevitable trade-off. The task of the *Semantic Reranker* involves pushing relevant passages into the top-$k$ list, and allowing for a low $k$ value to be set.

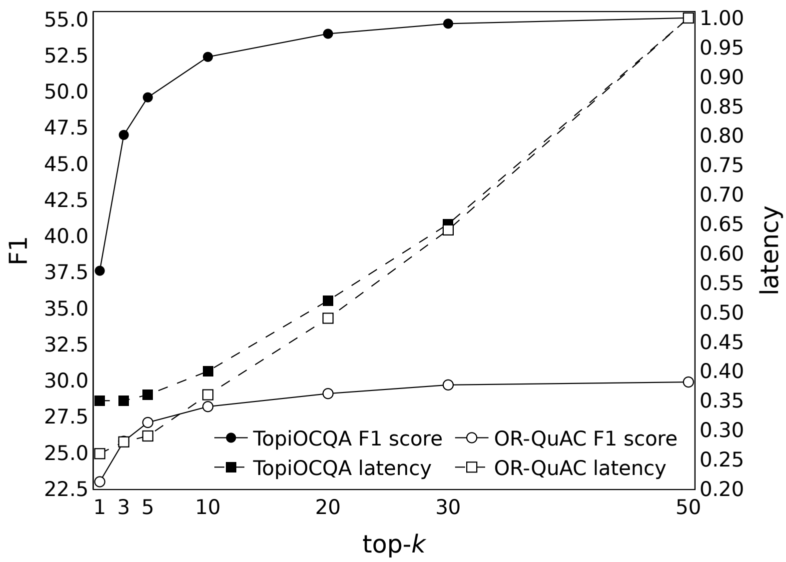



Figure 4.3: F1 score and relative latency of the FiD reader on the TopiOCQA development set and OR-QuAC test set, evaluated with varying numbers of top-$k$ input passages. Latency is reported relative to the top-50 setting.

### 4.7.2 The *R3FINE* Strategy

Building on the findings above, this work introduces the Retriever–Reranker–Reader finetuning (*R3FINE*) strategy for constructing strong and efficient baselines in the ODConvQA setting. The strategy begins by increasing the number of passages returned by the DPR retriever from the initial 50 to 1000. A *Semantic Reranker* component, implemented as a single Transformer encoder layer, is then incorporated. The *Semantic Reranker* is trained and fine-tuned alongside the *Conversation Encoder*, while the *Passage Encoder* remains frozen. Training follows the same contrastive objective used in the DPR retriever, encouraging high similarity between matching conversation–passage pairs while pushing apart non-matching ones. Finally, based on the observation from Table 4.1 that fewer but more relevant passages improve FiD reader performance, an additional fine-tuning step is applied to the reader using the top-10 passages returned by the *Semantic Reranker*.

## 4.8 Experiments and Results

Table 4.3 and Table 4.4 present a comparison between our *R3FINE* strategy and prior baselines, including traditional sparse retrievers such as BM25 (Robertson and Zaragoza, 2009) and extractive readers like the DPR Reader (Karpukhin et al., 2020). *R3FINE* achieves an F1 score of 59.0 on TopiOCQA and 32.9 on OR-QuAC, surpassing the best results reported in the original papers by 3.9 and 3.0 points, respectively. Notably, these substantial gains are obtained through simple yet effective modifications to the training, architecture, and inference setup of the DPR+FiD pipeline, without introducing new, more complex models.

| **TopiOCQA** | | |
|---|---|---|
| **Model** | **EM** | **F1** |
| BM25 + DPR Reader | 13.6 | 25.0 |
| BM25 + FiD | 24.1 | 37.2 |
| DPR Retriever + DPR Reader | 21.0 | 43.4 |
| DPR Retriever + FiD | 33.0 | 55.3 |
| (Ours) DPR Retriever + FiD | 33.0 | 55.1 |
| + R3FINE (top-10) | **35.8** | **59.0** |

Table 4.3: Performance (Exact Match and F1 scores) comparison on the TopiOCQA development set between our approach and baseline combinations of sparse (BM25) vs. dense (DPR Retriever) retrievers, and extractive (DPR Reader) vs. generative (FiD) readers.

| OR-QuAC | | |
|---|---|---|
| **Model** | **EM** | **F1** |
| DrQA (Chen et al., 2017) | - | 6.3 |
| BERTserini (Yang et al., 2019) | - | 26.0 |
| ORConvQA (Qu et al., 2020b) | - | 29.4 |
| (Ours) DPR Retriever + FiD | **22.0** | 29.9 |
| + R3FINE (top-10) | 21.6 | **32.9** |

Table 4.4: Performance (Exact Match and F1 scores) comparison on the OR-QuAC test set between our approach and baseline models.

To further support our *R3FINE* strategy, Table 4.5 presents an ablation study quantifying its impact on the DPR+FiD pipeline. We observe that incorporating the *Semantic Reranker* (w/ SR) consistently improves performance over the DPR+FiD baseline (w/o SR). Notably, it enables a 5-fold reduction in input size by processing only the top 10 passages, while maintaining comparable or superior results. In addition, a further fine-tuning step of the FiD reader (w/ SR + FT) leads to improvements of 1.7 and 2.9 F1 points on TopiOCQA and OR-QuAC, respectively. Additional experiments in Appendix A include reranker configuration ablations and extended reader fine-tuning results.

| | TopiOCQA | | | | | | OR-QuAC | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | w/o SR | | w/ SR | | w/ SR + FT | | w/o SR | | w/ SR | | w/ SR + FT | |
| **top-$k$** | **EM** | **F1** | **EM** | **F1** | **EM** | **F1** | **EM** | **F1** | **EM** | **F1** | **EM** | **F1** |
| 1 | 19.3 | 37.6 | 28.1 | 50.4 | 30.7 | 52.4 | 14.7 | 23.0 | 13.9 | 25.7 | 16.6 | 28.9 |
| 10 | 29.8 | 52.4 | 33.2 | 57.3 | **35.8** | **59.0** | 19.0 | 28.2 | 19.4 | 30.0 | 21.6 | **32.9** |
| 50 | <u>33.0</u> | <u>55.1</u> | 33.9 | 56.2 | 35.2 | 56.1 | <u>22.0</u> | <u>29.9</u> | 22.1 | 30.2 | **23.5** | 32.2 |

Table 4.5: FiD reader performance (Exact Match and F1 scores) on the TopiOCQA development set and OR-QuAC test set before and after the introduction of the *Semantic Reranker* (w/o SR and w/ SR), together with the results obtained after a further reader fine-tuning step with top-10 output by the SR (w/ SR + FT). <u>Underlined</u> values indicate the results obtained by the DPR+FiD pipeline. **Bold** values indicate the results obtained by *R3FINE* after the introduction of the *Semantic Reranker* together with targeted fine-tuning steps.

Finally, Figure 4.3 shows that reducing the number of input passages from top-50 to top-10 results in an average latency reduction of 60% for the FiD reader across both datasets. We conducted latency measurements to assess the overhead introduced by the *Semantic Reranker* and its associated parameters, with full details provided in Appendix A. As the *Semantic Reranker* consists of a single Transformer encoder layer, its parameter count is negligible compared to both DPR and FiD. In practice, it accounts for only 0.34% of FiD's overall latency, adding approximately 2.4 ms per example on top of the 710 ms taken by FiD. It is important to note that this impact is only considered in relation to FiD, as the retrieval phase remains constant regardless of the inclusion of the *Semantic Reranker*.

## 4.9 Conclusions

In this chapter, we identified several limitations of the typical Dense Passage Retrieval (DPR) retriever and Fusion-in-Decoder (FiD) reader pipeline when applied in an ODConvQA setting. We proposed and evaluated an improved approach by including a fast reranking component between these two modules and by performing targeted fine-tuning steps. The proposed *R3FINE* strategy lead to a better latency/performance trade-off. The new baseline has proven to be both strong and efficient when compared to previous baselines, thus making it suitable for future comparisons of new approaches.

These improvements become clearer in the context of the Larry and Palmerston example from Figure 1.1. By increasing the chances of surfacing passages that contain evidence such as their overlapping tenures and documented confrontations, the *R3FINE* strategy may help the system capture relations like their rivalry.

## 4.10 Limitations

The study presented in this chapter aimed to identify and address various limitations of the commonly used ODConvQA pipeline. While our approach may not be technically groundbreaking, the work's novelty lies in the presented findings to design strong and efficient baselines for ODConvQA. It should be noted that further research is needed to compare the performance of the proposed *R3FINE* strategy with other rerankers on non-conversational QA datasets, which would provide valuable insights into how effective the *R3FINE* approach is in different contexts.

## 4.11 Ethics Statement

We do not foresee any ethical concerns arising from this work. Our study builds entirely on publicly available datasets and established ODConvQA pipelines. The focus is on refining existing components to improve efficiency and performance. Nevertheless, as with any research in automated information retrieval and generation, care should be taken when deploying such systems to ensure outputs are used responsibly, especially in high-stakes or user-facing scenarios.

## 4.12 Acknowledgements

This research was conducted during an internship at Amazon Science, whose support and resources were instrumental to the completion of this work. I would like to thank my mentor, manager, and collaborators at Amazon for their valuable guidance and feedback. I am also grateful to my supervisors at Idiap and EPFL for their continuous support throughout my PhD, and to the Swiss National Science Foundation (SNSF) for funding my doctoral studies, under grant number 200021E_189458, and enabling me to pursue this internship opportunity.

# 5 *RAGferee*: Contextual Reward Models

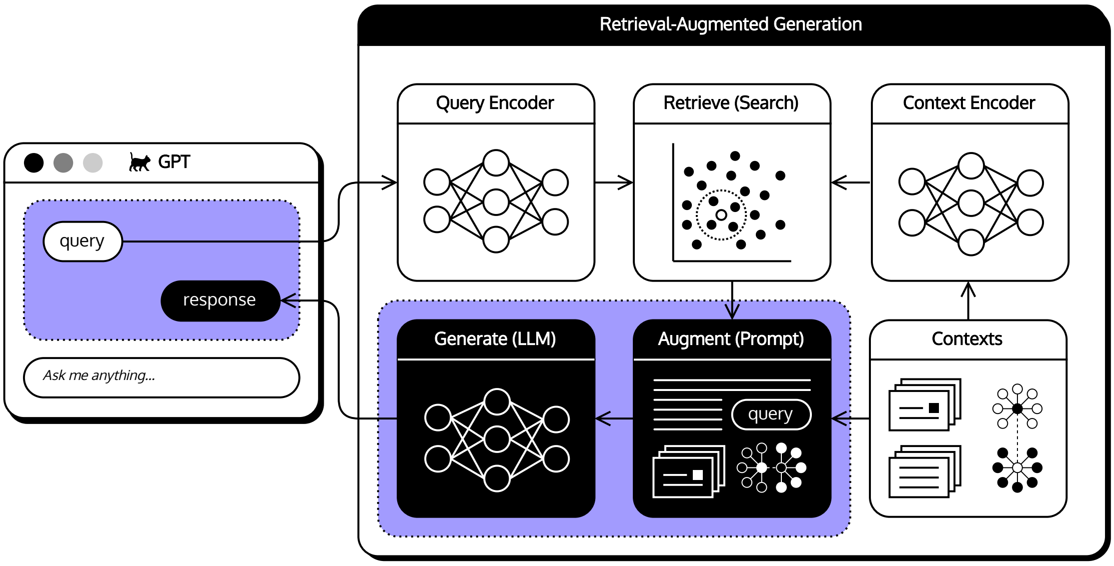



Figure 5.1: This chapter focuses on the *Augment (Prompt)*, *Generate (LLM)*, and the *query–response pair* components of the RAG pipeline from Chapter 3. In the context of the Larry and Palmerston example from Figure 1.1, it illustrates the need for aligning the LLM with the RAG setting to produce responses that reflect the retrieved evidence. Achieving this alignment requires a way to evaluate candidate responses with respect to RAG-specific dimensions such as faithfulness, completeness, conciseness, appropriate refusals, and relevance to the user query. This chapter shows that standard Reward Models (RMs), typically trained on general preference data and intended as domain-agnostic response evaluators, are not well suited to this contextual setting: they are designed to evaluate outputs that reflect parametric knowledge alone, and lack the means to assess whether a response is appropriately grounded in retrieved context (i.e., non-parametric knowledge). To address this, it introduces *RAGferee*, a methodology that repurposes QA datasets into contextual preference pairs. This enables the training of lightweight yet effective RMs which, when tested on RAG-specific benchmarks, outperform much larger models trained on extensive, general-purpose corpora.

## 5.1 Research Question

*How can reward models be extended to evaluate responses in the RAG setting?*

## 5.2 Publication

Coman, A. C., Sorodoc, I.-T., Ribeiro, L. F. R., Byrne, B., Henderson, J., and de Gispert, A. (2025a). Ragferee: Building contextual reward models for retrieval-augmented generation

## 5.3 Summary

Existing Reward Models (RMs), typically trained on general preference data, struggle in Retrieval Augmented Generation (RAG) settings (Figure 5.1), which require judging responses for faithfulness to retrieved context, relevance to the user query, appropriate refusals when context is insufficient, completeness, and conciseness of information, to name a few. To address the lack of publicly available RAG-centric preference datasets and specialised RMs, in this chapter, we introduce *RAGferee* (RAG referee/judge), a methodology that repurposes question-answering (QA) datasets into preference pairs that prioritise groundedness, rooted in non-parametric knowledge, over stylistic features driven by parametric knowledge, thereby enabling the training of contextual RMs better suited to judging RAG responses. Using *RAGferee*, we curate a small preference dataset of 4K samples and fine-tune RMs ranging from 7B to 24B parameters. Our RAG-centric RMs achieve state-of-the-art performance on ContextualJudgeBench, surpassing existing 70B+ RMs trained on much larger (up to 2.4M samples) general corpora, with a +15.5% absolute improvement.

## 5.4 Introduction

Reinforcement Learning from Human Feedback (RLHF) has become a key method for aligning Large Language Models (LLMs) with human preferences (Ouyang et al., 2022). Building on this foundation, policy optimisation techniques (Schulman et al., 2017; Rafailov et al., 2023) incorporated Reinforcement Learning from AI Feedback (RLAIF) (Bai et al., 2022b) which uses Reward Models (RMs), LLMs trained to judge the quality of generated responses (Bai et al., 2022a), as scalable proxies for human evaluation. These RMs are typically trained on general-purpose preference datasets (Wang et al., 2024e; Han et al., 2024; Xu et al., 2024; Liu et al., 2024) and are expected to act as domain-agnostic evaluators capable of assessing model outputs across a broad range of tasks and domains (Vu et al., 2024; Alexandru et al., 2025). However, their effectiveness remains under-explored in Retrieval-Augmented Generation (RAG) (Lewis et al., 2020b; Guu et al., 2020; Izacard et al., 2023) scenarios, where responses must be judged based on externally retrieved information rather than solely parametric knowledge (Ye et al., 2024c; Saha et al., 2025).

Context-aware reward modelling introduces unique challenges: RMs must assess not only the quality of responses, but also their faithfulness to the retrieved context, relevance to the user query, and appropriateness of refusals when no valid answer can be provided (Jin et al., 2024). Additionally, effective evaluation of RAG responses requires assessing the completeness of the information, ensuring that responses fully incorporate relevant content, as well as their conciseness, making sure the responses are informative without being overly verbose (Xu et al., 2025a). Benchmarks like ContextualJudgeBench (Xu et al., 2025a) question the effectiveness of domain-agnostic evaluators, showing that even large RMs trained on extensive general preference data, struggle in RAG-specific settings (Figure 5.2).

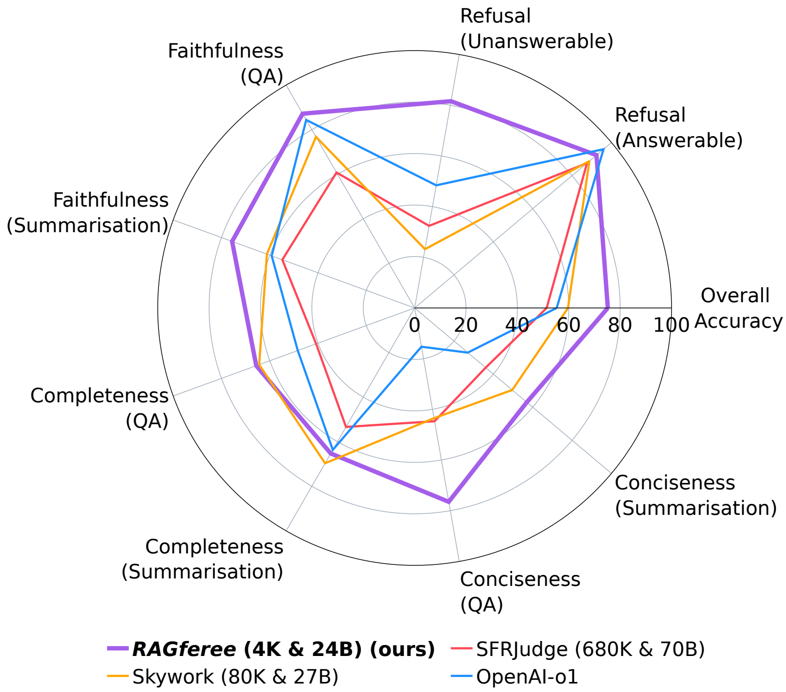



Figure 5.2: Top RMs (preference pairs & model size) in CONTEXTUALJUDGEBENCH. *RAGferee* is a well-rounded model, showing significant improvements on deflection (Refusal), faithfulness, and conciseness cases.

A key barrier to developing contextual RMs is the lack of publicly available preference datasets and openly licensed RMs specifically designed for judging RAG responses (Xu et al., 2025a). Addressing this critical gap and challenging the reliance on extensive general-purpose preference datasets and large RMs, we make the following **contributions**:

- We introduce *RAGferee*, a methodology for constructing RAG-specific preference datasets aimed at training contextual RMs by repurposing QA datasets into preference pairs.
- We curate a small preference dataset of 4K samples via stratified sampling and heuristics that select pairs across queries, models, and answers, prioritising diversity over quantity.
- We fine-tune RAG-centric RMs ranging from 7B to 24B parameters that significantly outperform existing 70B+ RMs trained on much larger (up to 2.4M samples) general corpora, achieving state-of-the-art performance on CONTEXTUALJUDGEBENCH, with an absolute improvement of +15.5%.

## 5.5 *RAGferee*

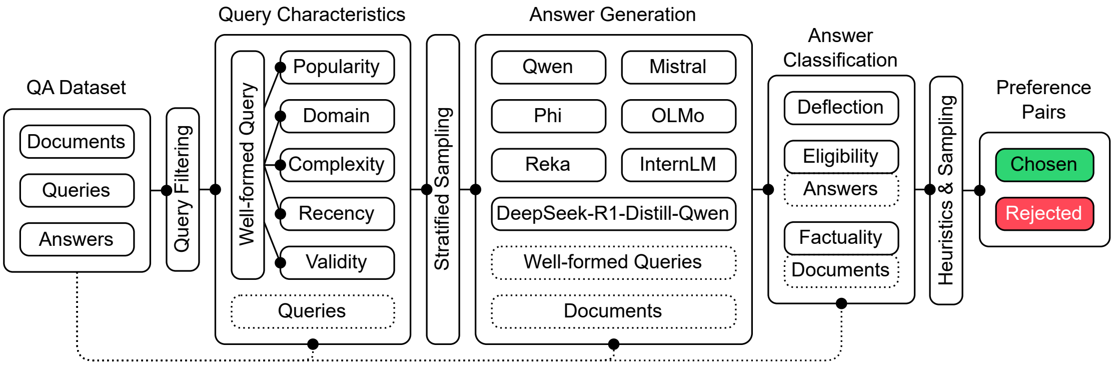



Figure 5.3: *RAGferee* creates RAG-specific preference pairs by repurposing QA datasets. Queries are first filtered, transformed, and categorised into multiple types (Subsection 5.5.1). A stratified subset is then selected to ensure balanced coverage across query types (Subsection 5.5.2). Candidate answers are generated using a set of LLMs, and labelled based on key qualitative aspects (Subsection 5.5.3). Finally, heuristics are used to select preference pairs aimed at training contextual RMs (Subsection 5.5.4).

The *RAGferee* methodology (Figure 5.3) takes as input a QA dataset $D_{QA} = (q_i, a_i, c_i)_{i=1}^{N}$ where $q_i$ represents the user query, $a_i$ is the reference answer, and $c_i$ denotes the retrieved context. It outputs a set of preference pairs $\hat{D}_{QA} = (\hat{q}_j, \hat{a}_{j_{chosen}}, \hat{a}_{j_{rejected}}, c_j)_{j=1}^{\hat{N} \ll N}$ where $\hat{a}_{j_{chosen}}$ and $\hat{a}_{j_{rejected}}$ are the preferred and non-preferred answers, respectively. To achieve this, *RAGferee* applies the stages described below.

### 5.5.1 Query Characteristics

Query $q$ is first mapped to a well-formed version $\hat{q} = LLM(q, p_{wf})$, where the $LLM$ function receives specific guidelines from prompt $p_{wf}$ to ensure grammatical correctness, appropriate punctuation, and consistent capitalisation, all while strictly preserving the original semantic meaning. This well-formed version is crucial for subsequent categorisations, ensuring that all queries are processed from a clear and well-structured format.

Query $\hat{q}$ is then mapped to a feature vector $\hat{\mathbf{q}} = LLM(\hat{q}, p_d)$, where the $LLM$ function, guided by prompt $p_d$, extracts discrete features that describe the query along several key dimensions. The resulting feature vector $\hat{\mathbf{q}}$ includes components such as $\hat{q}_{validity}$, which evaluates whether the query is clear, non-harmful, and genuinely seeks factual information; $\hat{q}_{recency}$ which captures how frequently the information in the query changes, from timeless facts to fast-changing, event-based content; $\hat{q}_{popularity}$ which reflects how widely a topic is known, from common subjects to more niche queries; $\hat{q}_{complexity}$ considers the level of reasoning required to provide an answer, from simple answers to those needing synthesis or deeper analysis; and $\hat{q}_{domain}$ categorises each query by its main topic, such as science, entertainment, etc.

The specific guidelines provided to the $LLM$, along with the definitions and category sets for each dimension, are outlined in Appendix B.

### 5.5.2 Stratified Sampling

Feature vector $\hat{\mathbf{q}}$ from Subsection 5.5.1 is used to select a representative subset of queries. Initially, these are filtered using the $\hat{q}_{validity}$ feature, where only the valid ones ($\hat{q}^{+}_{validity}$) are retained. Stratified sampling (Neyman, 1934) is then applied to the remaining features, ensuring that the final subset spans the full range of query types.

The stratification process also takes into account if a query has a reference answer, which depends on whether relevant information is found in the retrieved context. If a reference answer is available, its length (in words) is used to classify it as short, medium, or long. These categories are based on the 25th percentile (for short) and the 75th percentile (for long) of the overall answer length distribution. Queries without a reference answer are assigned to a separate zero-length category and are treated as deflection queries. This extra stratification ensures a balanced representation of query complexity.

Further details on the implementation of the stratification process are provided in Appendix B.

### 5.5.3 Answer Generation and Classification

Each query $\hat{q}$ retained after the stratification process, along with its corresponding retrieved context $c$ and the prompt $p_g$, is used to generate candidate answers $\hat{a}_k = M_k(\hat{q}, c, p_g)$, where $M_k$ denotes the $k$-th $LLM$ in a selected set $M = \{M_1, M_2, \dots, M_k\}_{k=1}^{K}$ of models (see Section 5.6 for the list). Prompt $p_g$ is formatted to align with typical RAG scenarios (see Appendix B).

Each candidate answer $\hat{a}$ is first mapped to a feature vector $\hat{\mathbf{a}} = LLM(\hat{a}, p_{\hat{a}})$, where the $LLM$ function is guided by a dedicated prompt $p_{\hat{a}}$ (available in Appendix B) that contains detailed evaluation guidelines for assessing each answer on specific criteria. The resulting feature vector $\hat{\mathbf{a}}$ includes components such as $\hat{a}_{deflection}$, which evaluates how well the answer handles situations where the query is unanswerable, ensuring the model recognises when no relevant information is available and responds appropriately; $\hat{a}_{eligibility}$, which assesses the relevance of the answer to the query, focusing on how well the response aligns with the user's intent and the reference answer; and $\hat{a}_{factuality}$, which examines the factual accuracy of the answer, ensuring that it contains verifiable, correct information based on the retrieved context.

### 5.5.4 Constructing Preference Pairs

This process involves selecting appropriate pairs of candidate answers $\hat{a}$, where one answer is *chosen* (preferred) and the other is *rejected* (non-preferred). We leverage the labels from the feature vector $\hat{\mathbf{a}}$ and follow a set of heuristics described below.

For queries with answers, the *chosen* answer must be both eligible ($\hat{a}^{+}_{eligibility}$), meaning it appropriately addresses the query, and factual ($\hat{a}^{+}_{factuality}$), meaning it is accurate based on the provided grounding. The *rejected* answer is either not eligible ($\hat{a}^{-}_{eligibility}$), meaning it fails to address the query, or eligible ($\hat{a}^{+}_{eligibility}$) but not factual ($\hat{a}^{-}_{factuality}$), where the model may have relied on parametric knowledge rather than the grounding.

For queries without answer, the *chosen* response is one where the model correctly deflects ($\hat{a}^{+}_{deflection}$), acknowledging no valid answer is available from the retrieved context. The *rejected* response is one where the model attempts to answer ($\hat{a}^{-}_{deflection}$) despite lacking relevant or complete information. This tests the model's ability to recognise when no valid response can be given.

To account for variability in models $M_k$ performance and avoid favouring any model based on surface-level clues during RMs training, stratified sampling across models is applied to ensure a balanced distribution of *chosen* and *rejected* pairs. Additionally, since the heuristics may have disrupted the earlier balance of query types, a second round of stratified sampling is conducted to restore it.

## 5.6 *RAGferee*: Use Case

Our use case focuses on the MS-MARCO v2.1 training set (Bajaj et al., 2018), which contains approximately 800K queries, 38% of which are labelled *"No Answer Present."*. While such queries could in principle be used to build deflection cases, we consider these as easy deflections and exclude them, choosing instead to focus only on queries that have explicit answers. Each such query is associated with 10 short passages linked to URLs, and each passage is labelled as either contributive (i.e., can be used to construct the answer) or non-contributive. We attempt to resolve each URL to a corresponding full document in the TREC RAG 2024 corpus (TREC-RAG, 2024) consisting of roughly 10M documents. If any of the contributive passage cannot be linked to a document in the TREC corpus, the associated query is marked as unanswerable and relabelled as *"No Answer Present."*. These deflection queries are more challenging than the initially excluded ones as they often include relevant (non-contributive) passages but lack key contributive ones, requiring the model to recognise the absence of a grounded answer and respond appropriately. At this stage, the dataset includes approximately 500K queries.

To ensure the final dataset remains free from licensing constraints and supports open research, we exclusively leverage models released under permissive licences, such as Apache 2.0 (ASF, 2004) and MIT (MIT, 1987). We use DeepSeek-V3 (DeepSeek-AI et al., 2024) for the query characteristics (Subsection 5.5.1) and answer classification (Subsection 5.5.3). We use stratified sampling (Subsection 5.5.2) to select a balanced 50K subset from the 500K queries, with 45K queries with answers and 5K queries without answers. Then, for answer generation (Subsection 5.5.3), we leverage models such as Qwen (Bai et al., 2023; Qwen et al., 2025), Mistral (Jiang et al., 2023), Phi (Gunasekar et al., 2023; Abdin et al., 2024), OLMo (Groeneveld et al., 2024), Reka (Team et al., 2024), InternLM (Cai et al., 2024), and DeepSeek (DeepSeek-AI et al., 2025). Finally, we construct 5K preference pairs (Subsection 5.5.4), with 4.5K containing answers (90%) and 500 without answers (10%). Of these, 4K (80%) are used for training, 500 (10%) for development, and 500 (10%) for testing.

The full list of models, together with the distributions before and after the sampling stages are reported in Appendix B.

## 5.7 Experimental Setup

CONTEXTUALJUDGEBENCH (Xu et al., 2025a) is a benchmark that combines both human annotations and model-based perturbations to provide a diverse and robust evaluation setting of RMs in RAG scenarios. It consists of 2,000 samples, with breakdowns for both QA and summarisation tasks, and evaluates models across four subsets: **Refusal** evaluates how models handle questions when the context might not contain sufficient information. This includes assessing if a model correctly identifies that a substantive response is better than a refusal for a question answerable from the context ("Refusal (Answerable)"), and conversely, if a model correctly chooses to refuse to answer when the question cannot be answered from the provided context ("Refusal (Unanswerable)"). **Faithfulness** measures the consistency of the response with the context, ensuring all factual statements in the response are attributable to the context and there are no hallucinations. **Completeness** assesses how comprehensive the response is, ensuring it covers all essential information needed for a thorough and useful answer. **Conciseness** determines if the response avoids including more information than what was asked. This includes preventing trivial copy-pasting without meaningful synthesis.

We use *consistent accuracy*, as defined in CONTEXTUALJUDGEBENCH, to assess model performance. This metric is tailored to different types of RMs, namely generative RMs and discriminative RMs.

**Generative RMs (GRMs)** (Mahan et al., 2024) generate text for a $(\hat{q}, c, \hat{a}_{chosen} \wedge \hat{a}_{rejected})$ tuple, expressing a preference or comparative judgment. Each test instance is evaluated by jointly comparing the $\hat{a}_{chosen}$ and $\hat{a}_{rejected}$ responses in two orders: first with $\hat{a}_{chosen}$ preceding the $\hat{a}_{rejected}$, and then with the order reversed. A prediction is considered correct only if the model consistently selects the $\hat{a}_{chosen}$ response in both evaluations. This mitigates positional bias and ensures evaluation robustness. Under this setup, random choice corresponds to a *consistent accuracy* of 25%.

**Discriminative RMs (DRMs)** (Yang et al., 2024b) assign a scalar score to a $(\hat{q}, c, \hat{a}_{chosen} \vee \hat{a}_{rejected})$ tuple, typically trained with a pairwise loss like Bradley-Terry (Bradley and Terry, 1952) to estimate relative response quality. Each test instance is evaluated by independently assigning a score to $\hat{a}_{chosen}$ and $\hat{a}_{rejected}$. A prediction is considered correct only if $score(\hat{a}_{chosen}) > score(\hat{a}_{rejected})$. Since this setting is not affected by the responses order, random choice corresponds to a *consistent accuracy* of 50%.

Given our goal of fine-tuning relatively small models (from 7B to 24B) that can potentially serve as value functions in online preference optimisation, we focus on discriminative RMs trained with the Bradley-Terry loss for their ability to produce scalar feedback without decoding.

## 5.8 Results and Discussion

Table 5.1 shows the performance of various models on CONTEXTUALJUDGEBENCH. The prompts used for inference are provided in Appendix B.

**Generative (non-reward) Models** perform the weakest overall. This is somewhat expected, as they are designed to be general-purpose models and are not specifically optimised to function as RMs. While they can be prompted to perform evaluations, this relies on their general language understanding rather than fine-grained training to judge RAG-specific dimensions. The smaller model, LLaMA-3.1-8B (Grattafiori et al., 2024), struggles with conciseness (in both QA and summarisation) and completeness (in QA). Larger models, such as LLaMA-3.1-70B, DeepSeek-R1 (DeepSeek-AI et al., 2025), and OpenAI-o1 (OpenAI et al., 2024), show some improvement but still have noticeable weaknesses in conciseness and completeness, despite stronger performance in refusal and faithfulness.

**Generative RMs** are generative models fine-tuned with preference data to enhance their performance as evaluators. Models like LLaMA-3.1-STEval (Wang et al., 2024c), LLaMA-3.1-Skywork-v0.2 (Liu et al., 2024), and LLaMA-3.1-SFRJudge (Wang et al., 2024b), incorporate between 20K and 680K preference pairs. Although their effectiveness in RAG settings remains limited, they generally outperform their non-reward counterparts. While not explicitly optimised for RAG settings, the fine-tuning process enables these models to better capture general answer

| Model | Param. | Pairs | Refusal (Ans.) | Refusal (Unans.) | Faithful. (QA) | Faithful. (Summ.) | Complete. (QA) | Complete. (Summ.) | Concise. (QA) | Concise. (Summ.) | Overall Accuracy |
|---|---|---|---|---|---|---|---|---|---|---|---|
| *Random generative* | - | - | *25.0* | *25.0* | *25.0* | *25.0* | *25.0* | *25.0* | *25.0* | *25.0* | *25.0* |
| *Generative (non-reward) Models (Xu et al., 2025a)* | | | | | | | | | | | |
| LLaMA-3.1 | 8B | - | 28.0 | 43.2 | 34.8 | 34.8 | 23.2 | 41.0 | 11.4 | 21.3 | 29.7 |
| LLaMA-3.1 | 70B | - | 59.6 | 48.0 | 58.0 | 48.4 | 38.0 | 51.8 | 15.7 | 27.5 | 43.4 |
| DeepSeek-R1 | 685B | - | 92.0 | 52.0 | 72.0 | 50.4 | 41.2 | 60.6 | 20.4 | 26.2 | 51.9 |
| OpenAI-o1 | - | - | 96.0 | 48.4 | 84.4 | 59.2 | 48.4 | 63.7 | 15.3 | 27.0 | 55.3 |
| *Generative Reward Models (Xu et al., 2025a)* | | | | | | | | | | | |
| LLaMA-3.1-Skywork | 8B | 80K | 60.8 | 12.0 | 38.8 | 31.6 | 38.4 | 26.7 | 29.4 | 21.3 | 32.4 |
| LLaMA-3.1-SFRJudge | 8B | 680K | 70.8 | 22.0 | 40.4 | 38.8 | 40.4 | 43.4 | 27.5 | 31.1 | 39.3 |
| LLaMA-3.1-SFRJudge | 70B | 680K | 87.6 | 32.4 | 60.8 | 54.8 | 40.8 | 53.4 | 44.7 | 36.1 | 51.4 |
| LLaMA-3.1-STEval | 70B | 20K | 50.0 | 42.0 | 51.2 | 45.6 | 40.8 | 39.4 | 36.1 | 29.9 | 41.9 |
| *Random discriminative* | - | - | *50.0* | *50.0* | *50.0* | *50.0* | *50.0* | *50.0* | *50.0* | *50.0* | *50.0* |
| *Discriminative Reward Models (baselines)* | | | | | | | | | | | |
| InternLM-2 | 7B | 2400K | 78.8 | 12.4 | 71.2 | 67.6 | 46.8 | 70.1 | 16.1 | 38.1 | 50.1 |
| InternLM-2 | 20B | 2400K | 84.4 | 31.2 | 75.2 | 67.2 | 53.2 | 70.5 | 28.6 | 45.5 | 57.0 |
| LLaMA-3.1-Skywork-v0.2 | 8B | 80K | 92.8 | 8.0 | 72.8 | 62.8 | 64.4 | 72.9 | 52.5 | 48.4 | 59.4 |
| Gemma-2-Skywork-v0.2 | 27B | 80K | 88.8 | 23.2 | 76.8 | 61.2 | 64.4 | 69.7 | 43.5 | 49.6 | 59.7 |
| *RAGferee Discriminative RMs (ours)* | | | | | | | | | | | |
| Qwen-2.5-RAGferee | 7B | 4K | 92.8 | 66.4 | 85.2 | 66.8 | 57.6 | 64.1 | 73.3 | 53.7 | 70.0 |
| Qwen-2.5-RAGferee | 14B | 4K | 92.8 | 71.2 | 86.8 | 70.8 | 65.2 | 66.9 | 71.4 | 52.0 | 72.2 |
| Mistral-Nemo-RAGferee | 12B | 4K | 92.0 | 82.8 | 82.8 | 68.8 | 62.4 | 62.9 | 86.3 | 57.0 | 74.5 |
| Mistral-Small-RAGferee | 24B | 4K | 92.4 | 81.6 | 87.2 | 75.6 | 65.6 | 65.3 | 76.5 | 57.0 | **75.2** |

Table 5.1: CONTEXTUALJUDGEBENCH results (best **overall**/within group *consistent accuracy*). For generative models, the metric evaluates whether the *chosen* response is consistently selected over the *rejected* response, regardless of their ordering, with a random chance baseline of 25%. For discriminative models, the metric evaluates whether the *chosen* response has a higher score than the *rejected* response, with a random chance baseline of 50%.

preferences, such as favouring more informative or relevant responses, which can transfer to contextual evaluations, despite differences in task structure. A notable improvement is seen in the "Refusal (Answerable)" subset, where these models reliably choose an actual answer over a deflection. However, they struggle with "Refusal (Unanswerable)" because their training data lacks deflection signals, making them more likely to select direct answers even when a refusal would be more appropriate.

**Discriminative RMs** (baselines) generally perform better than their generative counterparts. While models like InternLM (Cai et al., 2024) use an extensive number of preference pairs (up to 2.4M), their performance is not superior to models trained on smaller, more carefully curated datasets, such as the 80K pairs used for the Skywork (Liu et al., 2024) models. This suggests that the quality and relevance of training data are critical factors. However, challenges remain, particularly in handling appropriate deflections. For example, the LLaMA-3.1-Skywork-v0.2's performance on "Refusal (Unanswerable)" significantly drops to 8.0% from its base model's initial value of 43.2%.

***RAGferee* Discriminative RMs** (ours) models significantly outperform all existing models, despite being relatively small (from 7B to 24B) and fine-tuned on only 4K RAG-specific preference pairs. They perform well across all subsets, surpassing 70B+ RMs trained on much larger general corpora (from 20K to 2.4M). Notably, they handle deflection ("Refusal (Unanswerable)") cases more effectively, where many others decline sharply. Faithfulness and conciseness are also consistently high. The QA task typically achieves higher scores than summarisation, likely due to the *RAGferee* methodology being specifically tailored for QA in the creation of preference pairs.

In Table 5.2, we present the results of a **contrastive study** in which the *RAGferee* models from Table 5.1 were trained without grounding information. This setup mimics baseline discriminative RMs by relying solely on preference data related to the answers. We followed the procedure outlined in Subsection 5.5.4 to construct preference pairs, with the only modification being the exclusion of the $\hat{a}_{factuality}$ feature in the candidate answers. The results show a notable drop in performance compared to models trained with grounding, with the performance now aligning with that of baseline discriminative RMs. This emphasises that preference data alone is insufficient and that incorporating retrieved context is crucial for accurately judging RAG responses.

| **Model** | **Param.** | **Pairs** | **Overall Accuracy** |
|---|---|---|---|
| Qwen-2.5-RAGferee | 7B | 4K | 70.0 |
| Qwen-2.5-RAGferee | 14B | 4K | 72.2 |
| Mistral-Nemo-RAGferee | 12B | 4K | 74.5 |
| Mistral-Small-RAGferee | 24B | 4K | 75.5 |
| *trained w/o grounding (ablation)* | | | |
| Qwen-2.5-RAGferee | 7B | 4K | 52.3 |
| Qwen-2.5-RAGferee | 14B | 4K | 56.6 |
| Mistral-Nemo-RAGferee | 12B | 4K | 54.8 |
| Mistral-Small-RAGferee | 24B | 4K | 56.9 |

Table 5.2: CONTEXTUALJUDGEBENCH results (*contextual accuracy*) for *RAGferee* discriminative RMs from Table 5.1 trained without grounding. Preference data alone is not sufficient. Incorporating retrieved context is crucial for accurately judging RAG responses.

Figure 5.4 presents the results of a **data ablation** study, showing the performance of the *RAGferee* models from Table 5.1 as a function of the number of training preference pairs. The solid black line represents the mean *"Overall (consistent) Accuracy"* of the models, while the shaded grey area highlights the standard deviation across the models. Even with just 500 preference pairs, the *RAGferee* models already outperform most baseline discriminative RMs (indicated by the dotted lines), and with 1K pairs, they surpass all previous models. The performance peaks at 4K preference pairs. At the same data point, we also plot the results from Table 5.2, revealing a significant performance drop when the models are trained without grounding information. The ungrounded models fall within the performance range of the baseline discriminative RMs, highlighting the importance of grounding in training for achieving good results.

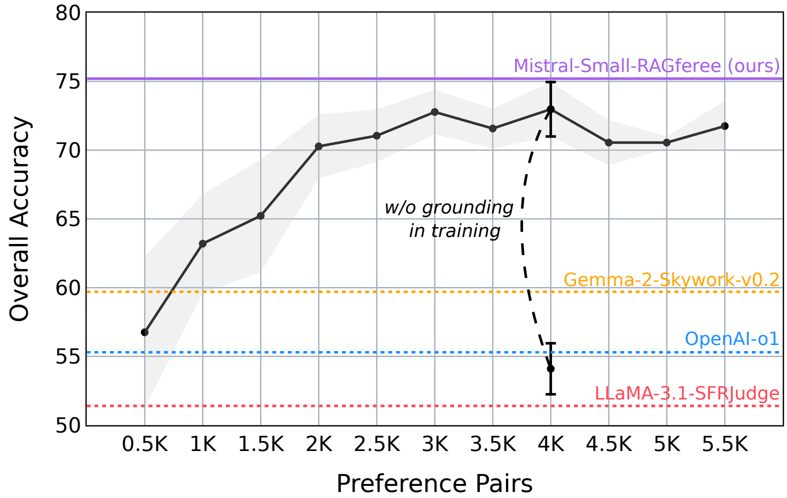



Figure 5.4: Performance of *RAGferee* RMs from Table 5.1 on ContextualJudgeBench as a function of training preference pairs. The solid black line represents the mean, while the shaded grey area indicates the standard deviation. Dotted lines represent best existing models.

**RAG-REWARDBENCH** Jin et al. (2024) is a fully synthetic benchmark of 1,485 samples, designed to evaluate RMs in RAG-specific scenarios such as multi-hop reasoning, fine-grained citation, appropriate abstention, and conflict robustness.

We adopt the same experimental setup as outlined in Section 5.7 and present the results in Table 5.3. Interestingly, in contrast to CONTEXTUAL-JUDGEBENCH (Table 5.1 and Figure 5.4), where the baseline discriminative RMs were significantly outperformed by our grounding-aware *RAGferee* discriminative RMs, the situation here is reversed. In this case, the baseline discriminative RMs outperform our *RAGferee* discriminative RMs, despite the benchmark being designed to assess RAG-specific dimensions. This result is counter-intuitive, as one would expect grounding-aware models to have a clear advantage in a RAG-focused setting.

| **Model** | **Param.** | **Pairs** | **Overall Accuracy** |
|---|---|---|---|
| *Discriminative RMs (baselines)* | | | |
| InternLM-2 | 7B | 2400K | 67.3 |
| InternLM-2 | 20B | 2400K | 68.7 |
| LLaMA-3.1-Skywork-v0.2 | 8B | 80K | 71.6 |
| Gemma-2-Skywork-v0.2 | 27B | 80K | 74.1 |
| *inferenced w/o grounding (ablation)* | | | |
| InternLM-2 | 7B | 2400K | 64.1 |
| InternLM-2 | 20B | 2400K | 65.1 |
| LLaMA-3.1-Skywork-v0.2 | 8B | 80K | 67.1 |
| Gemma-2-Skywork-v0.2 | 27B | 80K | 70.0 |
| *RAGferee Discriminative RMs (ours)* | | | |
| Qwen-2.5-RAGferee | 7B | 4K | 59.2 |
| Qwen-2.5-RAGferee | 14B | 4K | 63.1 |
| Mistral-Nemo-RAGferee | 12B | 4K | 60.5 |
| Mistral-Small-RAGferee | 24B | 4K | 61.5 |
| *inferenced w/o grounding (ablation)* | | | |
| Qwen-2.5-RAGferee | 7B | 4K | 67.1 |
| Qwen-2.5-RAGferee | 14B | 4K | 69.2 |
| Mistral-Nemo-RAGferee | 12B | 4K | 63.3 |
| Mistral-Small-RAGferee | 24B | 4K | 67.9 |

Table 5.3: RAG-REWARDBENCH results (*consistent accuracy*) for discriminative RMs inferenced with or without grounding. Grounding has minimal impact on baseline discriminative RMs (non-RAG), but it significantly influences our *RAGferee* discriminative RMs, which are sensitive to grounding by design.

To further investigate this discrepancy, we conduct an **ablation study** by entirely removing the grounding from the benchmark and analysing model performance under these conditions. The underlying hypothesis is that if grounding is truly essential to RAG-REWARDBENCH, then the performance of all discriminative RMs will deteriorate towards the random-chance baseline of 50%. However, this is not the case. The baseline discriminative RMs exhibit only a modest decline of around 4% in overall accuracy. In contrast, our *RAGferee* discriminative RMs show a notable improvement in performance when grounding is removed, once again contrary to the expected behaviour. This strongly suggests that our models are highly sensitive to grounding and actively use it during inference, as shown on CONTEXTUAL-JUDGEBENCH (Table 5.1 and Figure 5.4), where their accuracy drops much closer to chance in the absence of grounding.

These findings suggest that, despite its stated focus on RAG, RAG-REWARDBENCH may place greater emphasis on general response preferences rather than evaluating grounded behaviour. We believe this is due to the benchmark's fully synthetic data generation and pairs selection process, which may fail to accurately capture signals related to the importance of grounding. Consequently, the grounding provided in RAG-REWARDBENCH may often be irrelevant or unhelpful for reliable response evaluation, potentially accounting for the unexpected performance disparity. Appendix B includes examples of issues with the benchmark.

***RAGferee* test set** consists of 500 samples and is used for in-domain evaluation. While it is not a benchmark per se, unlike CONTEXTUALJUDGEBENCH and RAG-REWARDBENCH, we report comparative results in Table 5.4. The overall pattern mirrors the findings from CONTEXTUAL-JUDGEBENCH (Section 5.8). Baseline discriminative RMs perform the worst, consistent with earlier observations that they fail to leverage grounding information. This limitation is further highlighted by the substantially better performance of our *RAGferee* models, and their marked drop in performance (close to chance) when trained without grounding, where they again fall within the range of the baseline discriminative RMs.

| **Model** | **Param.** | **Pairs** | **Overall Accuracy** |
|---|---|---|---|
| *Discriminative RMs (baselines)* | | | |
| InternLM-2 | 7B | 2400K | 46.6 |
| InternLM-2 | 20B | 2400K | 52.0 |
| LLaMA-3.1-Skywork-v0.2 | 8B | 80K | 52.8 |
| Gemma-2-Skywork-v0.2 | 27B | 80K | 57.0 |
| *RAGferee Discriminative RMs (ours)* | | | |
| Qwen-2.5-RAGferee | 7B | 4K | 77.4 |
| Qwen-2.5-RAGferee | 14B | 4K | 83.4 |
| Mistral-Nemo-RAGferee | 12B | 4K | 81.6 |
| Mistral-Small-RAGferee | 24B | 4K | 81.8 |
| *trained w/o grounding (ablation)* | | | |
| Qwen-2.5-RAGferee | 7B | 4K | 53.6 |
| Qwen-2.5-RAGferee | 14B | 4K | 55.2 |
| Mistral-Nemo-RAGferee | 12B | 4K | 53.2 |
| Mistral-Small-RAGferee | 24B | 4K | 51.4 |

Table 5.4: RAGFEREE test set results (*consistent accuracy*). *RAGferee* models outperform baseline discriminative RMs, but drop to baseline levels when trained without grounding. This shows that grounding information is essential for effective contextual RMs.

***RAGferee* Generative RMs** are an extension of our discriminative RMs, where the models generate textual outputs instead of assigning numerical scores. Specifically, we use supervised fine-tuning (SFT) for training generative RMs which produce completions that include an indicator of the preferred response, such as "`<answer>A</answer>`" or "`<answer>B</answer>`". The results on CONTEXTUALJUDGEBENCH are presented in Table 5.5. The *RAGferee* generative RMs significantly outperform their non-reward counterparts and also surpass all generative models (both reward and non-reward) from Table 5.1, with the exception of the OpenAI-o1 model. However, they still fall behind the baseline discriminative RMs and are considerably less effective than our *RAGferee* discriminative RMs. This outcome aligns with previous studies, such as the one of Mahan et al. (2024), which found similar results with non-RAG RMs.

| **Model** | **Param.** | **Pairs** | **Overall Accuracy** |
|---|---|---|---|
| *Generative (non-reward) Models (baselines)* | | | |
| Qwen-2.5 | 7B | - | 28.0 |
| Qwen-2.5 | 14B | - | 37.3 |
| Mistral-Nemo | 12B | - | 22.3 |
| Mistral-Small | 24B | - | 42.4 |
| *RAGferee Generative RMs (SFT) (ours)* | | | |
| Qwen-2.5-RAGferee | 7B | 4K | 46.8 |
| Qwen-2.5-RAGferee | 14B | 4K | 53.9 |
| Mistral-Nemo-RAGferee | 12B | 4K | 50.0 |
| Mistral-Small-RAGferee | 24B | 4K | 50.4 |

Table 5.5: CONTEXTUALJUDGEBENCH results (*consistent accuracy*) of *RAGferee* generative RMs trained to output the indicator of the preferred response.

## 5.9 Related Work

Automatic evaluation of responses remains a persistent challenge (Sai et al., 2022; Chang et al., 2024). Traditional methods, such as matching-based metrics like BLEU (Papineni et al., 2002) and ROUGE (Lin, 2004), as well as embedding-based metrics like BERTScore (Zhang et al., 2020b) and BARTScore (Yuan et al., 2021), often struggle to capture subtle semantic nuances and deliver limited performance (Li et al., 2024a).

Prior works such as MT-Bench (Zheng et al., 2023), G-Eval (Liu et al., 2023), FLASK (Ye et al., 2024b), Prometheus (Kim et al., 2024a,b), FLAME (Vu et al., 2024), STEval (Wang et al., 2024c), SFRJudge (Wang et al., 2024b), Skywork-RM (Liu et al., 2024), InternLM-RM Cai et al. (2024), and Critic-RM (Yu et al., 2025) have explored the use and fine-tuning of strong LLMs as RMs for judging open-ended responses. They show that their judgments often align with human preferences and suggest that LLMs offer a promising alternative for scoring, ranking, and selecting responses across a wide range of tasks (Li et al., 2024b; Gu et al., 2024; Gao et al., 2024).

Several recent works, including FairEval (Wang et al., 2024a), PandaLM (Wang et al., 2024d), OffsetBias (Park et al., 2024), CALM (Ye et al., 2024a), RewardBench (Lambert et al., 2025), VeriScore (Song et al., 2024), Minicheck (Tang et al., 2024), HalluMeasure (Akbar et al., 2024), CrossEval (Zhong et al., 2024), and GroUSE (Muller et al., 2025) reveal important limitations of RMs. These include persistent issues such as hallucinations (Tonmoy et al., 2024), fairness and biases (Gallegos et al., 2024) concerns, and lack of robustness (Zhu et al., 2024).

Our work connects to the previous ones by challenging a prevailing assumption: that RMs trained on extensive general-purpose preference datasets such as Helpsteer (Wang et al., 2024e), Wildguard (Han et al., 2024), Magpie (Xu et al., 2024), as well as those used in STEval (Wang et al., 2024c), InternLM Cai et al. (2024), SFRJudge (Wang et al., 2024b), and Skywork (Liu et al., 2024), can serve as domain-agnostic evaluators, capable of assessing responses across a wide range of tasks and domains (Vu et al., 2024; Alexandru et al., 2025). In line with recent efforts like ContextualJudgeBench (Xu et al., 2025a) and RAG-RewardBench (Jin et al., 2024), we show that large RMs trained on general preference datasets perform poorly in RAG-specific settings. To address this limitation, we introduce a methodology that curates a small dataset by repurposing QA datasets into preference pairs, which is then used to fine-tune relatively small RAG-centric RMs, leading to significantly improved judges for contextual evaluation scenarios.

## 5.10 Conclusion

We show that existing Reward Models (RMs), typically trained on general preference data, exhibit limited effectiveness in RAG settings. Such tasks require evaluating model responses based on retrieved context, across multiple dimensions: faithfulness to the source material, relevance to the user query, appropriate refusals when the context is insufficient, and the completeness and conciseness of the information provided.

We address the lack of publicly available RAG-specific preference datasets and specialised RMs by introducing *RAGferee*, a methodology that repurposes existing QA datasets into preference pairs by filtering, transforming, categorising queries, selecting a stratified subset for balanced coverage, generating candidate answers from various LLMs, labelling them based on key qualitative criteria, and applying heuristics to select the best pairs for training contextual RMs.

We construct a small preference dataset of 4K samples and fine-tune RMs ranging from 7B to 24B parameters. We evaluate generative (non-reward) models, generative RMs, and discriminative RMs, and show that RAG-specific RMs trained with *RAGferee* are better suited for assessing context-grounded responses. Our RAG-centric RMs achieve state-of-the-art performance (+15.5% absolute improvement) on ContextualJudgeBench, surpassing 70B+ general-purpose RMs trained on up to 2.4M preference samples, despite using a much smaller dataset and models.

These findings also relate to the example of Larry and Palmerston in Figure 1.1. In that case, a reliable system must not only reference both entities but also justify their rivalry relation using retrieved evidence, such as reports of confrontations during their overlapping tenures. While Chapter 4 focused on improving the likelihood that such evidence is retrieved, this chapter examined how responses can be assessed with respect to the retrieved evidence. An LLM aligned with a *RAGferee*-based reward model would, in principle, be better suited to producing answers that remain faithful to the retrieved context and avoid making unsupported claims.

## 5.11 Limitations

Our work focuses on a single use case, which inherently limits the generalisability of the findings. The size of the curated dataset, while carefully balanced, remains relatively small compared to large-scale preference datasets, which may constrain model robustness and performance in more diverse or complex scenarios.

While using a single strong LLM for query classification (Subsection 5.5.1) is a straightforward approach, relying on the same model for answer classification (Subsection 5.5.3) introduces potential bias, as the labels generated are directly used to create preference pairs. An ensemble of labelling models could help mitigate this bias by providing more robust and diverse annotations. Furthermore, we did not study the correlation of these labels with human judgments, which could be valuable for uncovering discrepancies and biases between model-generated annotations and human preferences.

In our stratified sampling procedure (Subsection 5.5.2), we prioritised maximising diversity across queries, answers, and models. Although this strategy yielded promising results, it may not represent the optimal approach. Similarly, the use of heuristics for selecting preference pairs, by their nature, introduces biases and may oversimplify the selection process, potentially missing finer-grained distinctions in answer quality or contextual relevance that more sophisticated methods could identify.

Although our results show promising scalability properties with models up to 24B parameters, further investigation is needed to understand behaviour at larger scales. Additionally, while our RMs perform well on standard benchmarks, their effectiveness as value functions in policy optimisation remains an open area for exploration.

Finally, our generative RMs (Section 5.8) are limited to a rather straightforward fine-tuning SFT approach, where models are required to output the indicator of the preferred response. Extending this framework to incorporate Chain-of-Thought (CoT) (Wei et al., 2023) justifications prior to the preference indicator or applying more recent strategies like GRPO (Shao et al., 2024), offers promising avenues for further research.

## 5.12 Ethics Statement

Automatic evaluation of responses remains a persistent challenge. Although using and fine-tuning strong LLMs as RMs for judging responses shows promise, these models share the same inherent risks and ethical considerations highlighted in prior research on pretrained and instruction-tuned models. Since RMs are designed to align with human preferences, they may inherit and amplify existing human biases present in training data, potentially leading to unfair or discriminatory outcomes related to race, gender, or other sensitive attributes. Furthermore, over-reliance on these models risks automating decisions that require human judgement.

Our work promotes open and responsible research by committing to transparency in model development and by exclusively using publicly available datasets and models with permissive licences.

## 5.13 Acknowledgements

This research was conducted during an internship at Amazon Science, whose support and resources were instrumental to the completion of this work. I would like to thank my mentor, manager, and collaborators at Amazon for their valuable guidance and feedback. I am also grateful to my supervisors at Idiap and EPFL for their continuous support throughout my PhD, and to the Swiss National Science Foundation (SNSF) for funding my doctoral studies, under grant number 200021E_189458, and enabling me to pursue this internship opportunity.

# Text-Graph Encoders Part II

# 6 Self-Attention and Structural Encoding

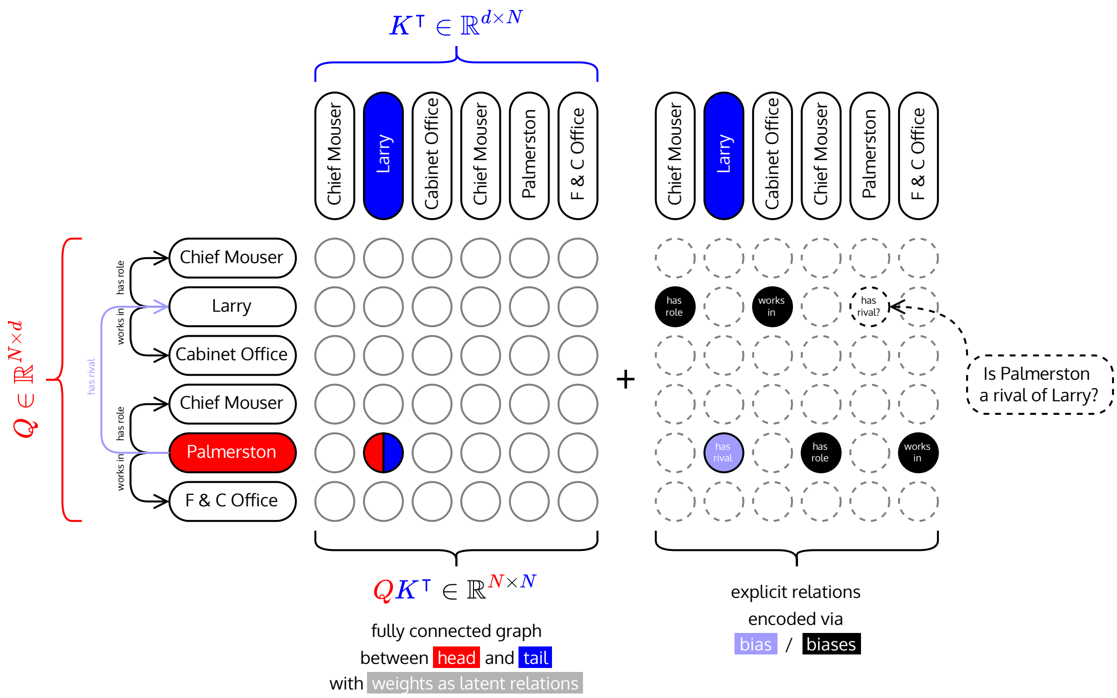



Figure 6.1: The self-attention mechanism can be interpreted in terms of relational triples *(head, relation, tail)*. The query projection $Q$ extracts from each input element the *head* representation, while the key projection $K$ extracts the corresponding *tail* representation. The $QK^\top$ creates a fully connected graph where the attention weights correspond to latent *relations*. In the context of Larry and Palmerston example from Figure 1.1, an explicit *relation* such as *"has rival"* for the triple *(Palmerston, has rival, Larry)* can be encoded via an additive bias term.

The attention mechanism has become a foundational component in Deep Learning (DL), allowing models to selectively focus on relevant parts of the input data. Originally introduced by Bahdanau et al. (2015) to augment Recurrent Neural Networks (RNNs) (Elman, 1990; Jordan, 1997; Hochreiter and Schmidhuber, 1997) in sequence-to-sequence tasks, attention was popularised in its *self-attention* form by Vaswani et al. (2017) in the Transformer architecture, which removed the recurrent structure entirely. This reformulation significantly advanced the modelling of long-range dependencies by eliminating the sequential bottleneck inherent to RNNs. The core innovation, which involves computing contextualised representations via a weighted aggregation of input based on pairwise similarity (Figure 6.1 and Figure 6.2), proved not only powerful but also adaptable across various modalities, including natural language processing (Devlin et al., 2019), computer vision (Dosovitskiy et al., 2020), speech recognition (Radford et al., 2022), and biology (Jumper et al., 2021). This chapter provides essential background on the self-attention mechanism and examines how it can be extended to incorporate structural encoding.

## 6.1 Self-Attention

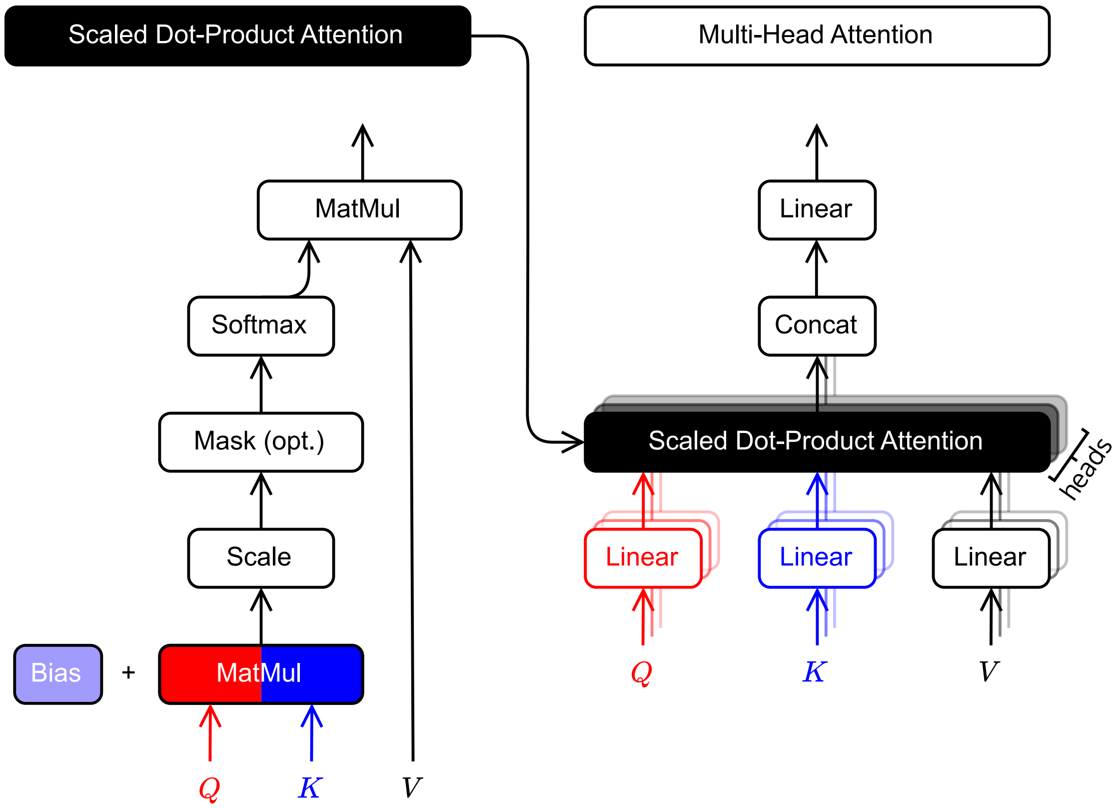



Figure 6.2: Scaled Dot-Product Attention from Vaswani et al. (2017) with the interpretation in terms of relational triples *(head, relation, tail)* from Figure 6.1 and addition of an explicit *bias* term, which encodes the *relation* between *head* and *tail*.

The Transformer architecture computes contextualised representations by allowing each element in a sequence to attend to all other elements via a mechanism known as self-attention. More formally, given a sequence $s \in \mathbb{R}^{N \times d}$, consisting of $N$ feature vectors of dimension $d$, a Transformer layer defines a function $T : \mathbb{R}^{N \times d} \to \mathbb{R}^{N \times d}$ that produces an output sequence $\hat{s} \in \mathbb{R}^{N \times d}$, in which each vector is updated using information aggregated from the entire input sequence $s$.

Each Transformer layer applies Scaled Dot-Product Attention (SDA) and a Feedforward Network (FFN), each followed by a residual connection. The resulting output is given by:

$$
\begin{aligned}
z &= \mathrm{SDA}(s) + s, \\
\hat{s} &= \mathrm{FFN}(z) + z.
\end{aligned}
$$

The SDA mechanism, illustrated in Figure 6.2, enables each element of the sequence to aggregate information from all other elements via a similarity-based weighted sum. This allows the model to incorporate global contextual information at every layer. To compute SDA, the input $s \in \mathbb{R}^{N \times d}$ is first projected into queries $Q$, keys $K$, and values $V$ using learned weight matrices:

$$
\begin{aligned}
Q &= sW_Q, \\
K &= sW_K, \\
V &= sW_V,
\end{aligned}
$$

with $W_Q, W_K, W_V \in \mathbb{R}^{d \times d}$. The attention output is then given by:

$$
\text{SDA}(s) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d}}\right)V.
$$

The FFN is typically a two-layer fully connected network with a nonlinearity in between, applied independently to each vector in the sequence. Together, the Transformer layer produces representations that are globally informed (via SDA) and locally transformed (via the FFN).

## 6.2 Structural Encoding

This subsection is based on the following publication:

*Henderson, J., Mohammadshahi, A., Coman, A., and Miculicich, L. (2023). Transformers as graph-to-graph models. In Elazar, Y., Ettinger, A., Kassner, N., Ruder, S., and A. Smith, N., editors,* Proceedings of the Big Picture Workshop*, pages 93–107, Singapore. Association for Computational Linguistics*

We posit that the sequence-to-sequence nature of most Transformer models is only a superficial characteristic; underlyingly they are in fact modelling complex structured representations. The key insight is that attention weights and graph structure edges are effectively the same thing.

Figure 6.1 shows how the formulation of SDA as a similarity-weighted aggregation naturally induces a fully connected directed graph over the input sequence. It also shows that the attention operation can be interpreted in terms of relational triples *(head, relation, tail)*. The query projection $Q$ extracts from each input element the *head* representation, while the key projection $K$ extracts the corresponding *tail* representation. Then, the $QK^\top$ computes a latent *relation* between each *head-tail* pair.

To explicitly incorporate graph structure into the Transformer's computation, the SDA can be modified as such:

$$\text{SDA}(s)_b = \text{softmax}\left(\frac{QK^\top + b}{\sqrt{d}}\right)V,$$

where $b \in \mathbb{R}^{N\times N}$ is a matrix of additive bias terms representing explicit graph relations between sequence elements. This represents just one approach to introducing structure: in practice, such biases can also be derived from learned relation embeddings or parametrised interactions involving $Q$ and $K$ themselves. The core idea is to incorporate structural inductive bias into the attention mechanism in a way that modulates the learned attention patterns.

This version of the Transformer architecture supports explicit structures which are general graphs, making it applicable to a wide range of structured representations and their integration with text. This model can jointly embed both the explicit structures and the latent structures in a Transformer's sequence-of-vectors hidden representation, and can predict explicit structures from this embedding.

Henderson et al. (2023) also highlight that the latent structures of pretrained Transformers already include much information about traditional linguistic structures, stating that they are fundamentally an expression of locality in the interaction between different components of a representation. As Henderson (2020) argued, incorporating this information about locality in the inductive bias of a neural network means putting connections between hidden vectors if their associated components are local in the structure. In Transformers, these connections are learned in the form of attention weights. Thus, these attention weights are effectively the induced structure of the Transformer's latent representation.

However, attention weights are not explicitly part of a Transformer's hidden representation. The output of a Transformer encoder is a sequence of vectors, and the same is true of each lower layer of self-attention. The latent attention weights are extracted from these sequence-of-vector embeddings with learned functions of pairs of vectors. Edges in explicit graphs can be predicted in the same way (from pairs of vectors), assuming that these graphs have also been embedded in the sequence of vectors.

The Graph-to-Graph Transformer (G2GT) from Henderson et al. (2023) allows conditioning on an observed graph and predicting a target graph. Moreover, by iteratively refining a previously predicted graph, G2GT can jointly model the entire predicted graph even though the actual prediction is done independently for each edge. And this joint modelling can be done in conjunction with other explicit graphs, as well as with the Transformer's induced latent graph.

# 7 *GADePo*: Graph-Assisted Declarative Pooling

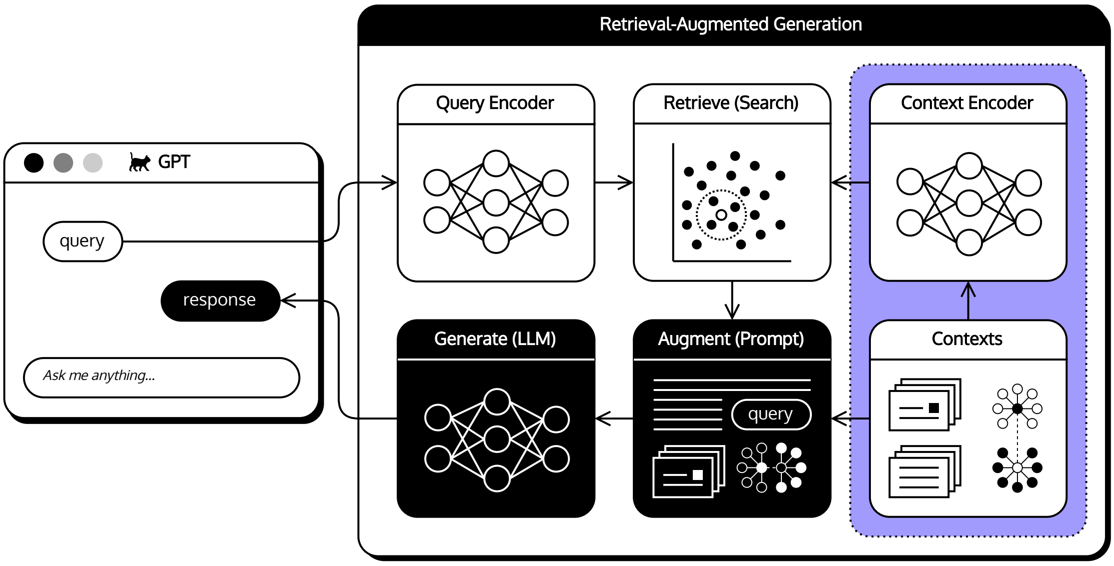



Figure 7.1: This chapter focuses on the *Context Encoder* and *Contexts* components of the RAG pipeline from Chapter 3, and investigates how attention-based models can learn to pool information from unstructured input using explicit structured relational information. As illustrated in the example from Figure 1.1, recovering relations such as the rivalry between Larry and Palmerston requires integrating information such as overlapping tenures, institutional roles, and confrontations, which may be distributed across retrieved passages or made explicit in the underlying graph. The chapter examines how graph-structured relations can be incorporated into the attention mechanism to guide information pooling across multiple spans of text, allowing the model to compose evidence more effectively using both unstructured and structured inputs.

## 7.1 Research Question

*How can attention-based models learn to pool information from unstructured inputs using explicit structured relational information?*

## 7.2 Publication

Coman, A., Theodoropoulos, C., Moens, M.-F., and Henderson, J. (2024). GADePo: Graph-assisted declarative pooling transformers for document-level relation extraction. In Yu, W., Shi, W., Yasunaga, M., Jiang, M., Zhu, C., Hajishirzi, H., Zettlemoyer, L., and Zhang, Z., editors, *Proceedings of the 3rd Workshop on Knowledge Augmented Methods for NLP*, pages 1–14, Bangkok, Thailand. Association for Computational Linguistics

## 7.3 Summary

Document-level relation extraction typically relies on text-based encoders and hand-coded pooling heuristics to aggregate information learned by the encoder. Building on the principles introduced in Chapter 6, we leverage the intrinsic graph processing capabilities of the Transformer model and propose replacing hand-coded pooling methods with new tokens in the input, which are designed to aggregate information via explicit graph relations in the computation of attention weights. We introduce a joint text-graph Transformer model and a Graph-Assisted Declarative Pooling (*GADePo*) specification of the input, which provides explicit and high-level instructions for information aggregation. *GADePo* allows the pooling process to be guided by domain-specific knowledge or desired outcomes but still learned by the Transformer, leading to more flexible and customisable pooling strategies. We evaluate our method across diverse datasets and models and show that our approach yields promising results that are consistently better than those achieved by the hand-coded pooling functions.

## 7.4 Introduction

Output

**Subject:** Breakout **Object:** Atari
**Relation:** developer

**Subject:** Breakout **Object:** Atari
**Relation:** publisher

Input

Breakout is an arcade game developed and published by Atari , Inc. , released on May 13 , 1976 . It was conceptualized by Nolan Bushnell and Steve Bristow , influenced by the 1972 Atari arcade game Pong , and built by Steve Wozniak aided by Steve Jobs . Breakout was the basis and inspiration for certain aspects of the Apple II personal computer . [...]

Figure 7.2: Document from the Re-DocRED (Tan et al., 2022b) dataset involving multiple entities and labels. Subject entity Breakout (red) and object entity Atari (blue) express relations "developer" and "publisher". Other entities are indicated as Mention (white).

Document-level relation extraction is an important task in natural language processing, which involves identifying and categorising meaningful relationships between entities within a document, as exemplified in Figure 7.2. This task is foundational to many applications, including knowledge base population and completion (Banko et al., 2007; Ji et al., 2020), information retrieval and extraction (Manning et al., 2008; Theodoropoulos et al., 2021), question answering (Chen et al., 2017; Feng et al., 2022) and sentiment analysis (Pang and Lee, 2008), to name a few.

Standard methods that approach this challenge generally employ pretrained text-based encoders (Devlin et al., 2019; Beltagy et al., 2019; Zhuang et al., 2021; Cui et al., 2021), which are responsible for capturing the nuances of information contained in the entity mentions and their contextual surroundings. Previous successful methods often then use hand-coded pooling heuristics to aggregate the information learned by the encoder, with some aimed at creating entity representations, while others directly exploiting the pattern of attention weights to capture context aware relations between entity mentions (Zhou et al., 2021; Xiao et al., 2022; Tan et al., 2022a; Ma et al., 2023). These pooling heuristics can be very effective at leveraging the information in a pretrained encoder. However, as shown in Conneau et al. (2017); Jia et al. (2019); Reimers and Gurevych (2019); Choi et al. (2021), the selection of an appropriate pooling function can be model-dependent, task-specific, resource-intensive and time-consuming to determine, thereby limiting flexibility.

In this chapter, we address these issues with a new approach where we leverage the intrinsic graph processing capabilities of the Transformer (Vaswani et al., 2017) model. In Chapter 6, we established the functional equivalence between attention weights and graph relations. Building on this foundation, and further developing the insights of Mohammadshahi and Henderson (2020); Henderson (2020); Mohammadshahi and Henderson (2021); Henderson et al. (2023), we demonstrate how structural dependencies between input elements can be effectively incorporated by simply adding relation features to the attention functions. Transformers easily learn to integrate these relation features into their pretrained attention functions, resulting in very successful graph-conditioned models (Mohammadshahi and Henderson, 2021; Miculicich and Henderson, 2022; Mohammadshahi and Henderson, 2023). Given this effective method for integrating explicit graphs with pretrained attention functions, we propose to use the attention function itself for aggregation. We replace the rigid pooling methods with new tokens which act as aggregation nodes, plus explicit graph relations which steer the aggregation.

We introduce a joint text-graph Transformer model and a Graph-Assisted Declarative Pooling (*GADePo*) method that leverages these special tokens and graph relations, to provide an explicit high-level declarative specification for the information aggregation process. By integrating these graphs in the attention functions of a pretrained model, *GADePo* exploits the pretrained embeddings and attention patterns but still has the flexibility of being trained on data. This enables the pooling to be guided by domain-specific knowledge or desired outcomes but still learned by the Transformer, opening up a more customisable but still data-driven relation extraction process.

We evaluate our method across diverse datasets and models commonly employed in document-level relation extraction tasks, and show that our approach yields promising results that are consistently better than those achieved by the hand-coded pooling functions.

**Contributions:** We propose a new method for exploiting pretrained Transformer models which replaces hand-coded aggregation functions with explicit graph relations and aggregation nodes. We introduce a novel form of joint text-graph Transformer model. We evaluate our approach across various datasets and models, showing that it yields promising results that are consistently better than those achieved by hand-coded pooling functions.

## 7.5 Related Work

In recent studies, the scope of relation extraction has been expanded to include not only individual sentences but entire documents. This extension, known as document-level relation extraction, presents a more realistic and challenging scenario as it seeks to extract relations both within sentences and across multiple sentences (Yao et al., 2019b). Transformer-based (Vaswani et al., 2017) models have shown great potential in addressing this task.

Wang et al. (2019a) and Tang et al. (2020) show that the BiLSTM-based (Hochreiter and Schmidhuber, 1997) baselines lack the capacity to model complex interactions between multiple entities. They propose a more robust approach, which consists of using the pretrained BERT (Devlin et al., 2019) model and a two-step prediction process, i.e., first identifying if a link between two entities exists, followed by predicting the specific relation type.

GAIN (Zeng et al., 2020) leverages BERT as a text encoder and GCNs (Kipf and Welling, 2017) to process two types of graphs, one at mention level and another at entity level, showing notable performance in inter-sentence and inferential scenarios.

Mohammadshahi and Henderson (2020, 2021) propose the G2GT model and show how to leverage the intrinsic graph processing capabilities of the Transformer model by incorporating structural dependencies between input elements as features input to the self-attention weight computations.

SSAN (Xu et al., 2021) leverages this idea and considers the structure of entities. It employs a transformation module that creates attentive biases from this structure to regulate the attention flow during the encoding phase.

DocuNet (Zhang et al., 2021) reformulates the task as a semantic segmentation problem. It employs a U-shaped segmentation module and an encoder module to capture global interdependencies and contextual information of entities, respectively.

PL-Marker (Ye et al., 2022) introduces a method that takes into account the interplay between spans via a neighbourhood-oriented and subject-oriented packing approach, highlighting the importance of capturing the interrelation among span pairs in relation extraction tasks.

SAIS (Xiao et al., 2022) explicitly models key information sources such as relevant contexts and entity types. It improves extraction quality and interpretability, while also boosting performance through evidence-based data augmentation and ensemble inference.

KD-DocRE (Tan et al., 2022a) proposes a semi-supervised framework with three key components. First, an axial attention module enhances performance in handling two-hop relations by capturing interdependence between entity pairs. Second, an adaptive focal loss addresses class imbalance. Finally, the framework employs knowledge distillation to improve robustness and effectiveness by bridging the gap between human-annotated and distantly supervised data.

DREEAM (Ma et al., 2023) is a method designed to enhance document-level relation extraction by addressing memory efficiency and annotation limitations in evidence retrieval. It employs evidence as a supervisory signal to guide attention and introduces a self-training strategy to learn evidence retrieval without requiring evidence annotations.

SAIS (Xiao et al., 2022), KD-DocRE (Tan et al., 2022a), and DREEAM (Ma et al., 2023) have been built upon the foundations of ATLOP (Zhou et al., 2021). ATLOP introduces two innovative techniques, adaptive thresholding, and localised context pooling, to address challenges in multi-label and multi-entity problems. Adaptive thresholding employs a learnable entities-dependent threshold, replacing the global threshold used in previous approaches for multi-label classification (Peng et al., 2017; Christopoulou et al., 2019; Nan et al., 2020; Wang et al., 2020a). Localised context pooling leverages the attention patterns of a pretrained language model to identify and extract relevant context crucial for determining the relation between entities, using specific hand-coded pooling functions.

## 7.6 Background

The foundational work of ATLOP (Zhou et al., 2021) has been the basis of many state-of-the-art models (Xiao et al., 2022; Tan et al., 2022a; Ma et al., 2023). Given the problems with hand-coded pooling functions, discussed in Section 7.4, we aim to provide a new baseline that can serve as the foundation for future state-of-the-art models. For this reason, we evaluate our proposed models by comparing them to this established baseline. Our goal is to demonstrate that our method not only achieves results comparable to or better than ATLOP, but also offers a novel approach that addresses its limitations. To help understand ATLOP and its components, we present a detailed breakdown in the left portion of Figure 7.3, which we elaborate on in this section.

### 7.6.1 Problem Formulation

The document-level relation extraction task involves analysing a document $D$ that contains a set of entities $\mathcal{E}_D=\{e_i\}_{i=1}^{|\mathcal{E}_D|}$. The main objective is to determine the presence or absence of various relation types between all entity pairs $(e_s, e_o)_{s,o \in \mathcal{E}_D, s \neq o}$, where the subject and object entities are denoted as $e_s$ and $e_o$, respectively. A key aspect to consider is that an entity can appear multiple times in the document, resulting in a cluster of mentions $\mathcal{M}_e=\{m_i\}_{i=1}^{|\mathcal{M}_e|}$ for each entity $e$. The set of relations is defined as $\mathcal{R} \cup \emptyset$, where $\emptyset$ represents the absence of a relation, often referred to as "no-relation". Given the clusters of mentions $\mathcal{M}_{e_s}$ and $\mathcal{M}_{e_o}$, the task consists of a multi-label classification problem where there can be multiple relations between entities $e_s$ and $e_o$.

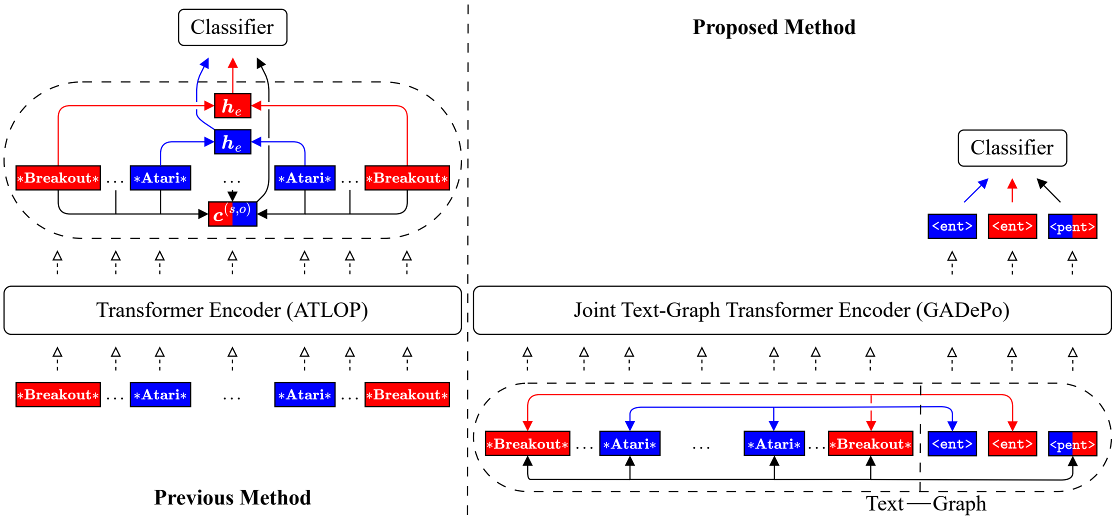



Figure 7.3: Comparison between the previous method ATLOP (left) and the proposed method *GADePo* (right), illustrating the document in Figure 7.2 containing two entities (red and blue), each with two mentions. In ATLOP, the mentions' encoder outputs are aggregated into entity representations $\boldsymbol{h}_e$, and the encoder's attention weights are used to identify which outputs to aggregate for entity-pair representations $\boldsymbol{c}^{(s,o)}$. In *GADePo*, the textual input is extended to include the graph special tokens <ent> for entity representations and <pent> for entity-pair representations, and explicit directional graph relations specify their associated mentions. A joint text-graph Transformer model is then used to encode this declarative pooling specification graph and compute the relevant aggregations.

### 7.6.2 Previous Method: ATLOP

**Text Encoding**. A special token $*$ is added at the start and end of every mention. Tokens $\mathcal{T}_D{=}\{t_i\}_{i=1}^{|\mathcal{T}_D|}$ are encoded via a Pretrained Language Model (PLM) as follows:

$$\boldsymbol{H}, \boldsymbol{A} = PLM(\mathcal{T}_D), \tag{7.1}$$

where $\boldsymbol{H} \in \mathbb{R}^{|\mathcal{T}_D| \times d}$ and $\boldsymbol{A} \in \mathbb{R}^{|\mathcal{T}_D| \times |\mathcal{T}_D|}$ represent the token embeddings and the average attention weights of all attention heads, respectively, extracted from the last layer of the PLM.

**Entity Embedding (EE)**. For each individual entity $e$ with mentions $\mathcal{M}_e{=}\{m_i\}_{i=1}^{|\mathcal{M}_e|}$, an entity embedding $\boldsymbol{h}_e \in \mathbb{R}^d$ is computed as follows:

$$\boldsymbol{h}_e = log \sum_{i=1}^{|\mathcal{M}_e|} exp(\boldsymbol{H}_{m_i}), \tag{7.2}$$

where $\boldsymbol{H}_{m_i} \in \mathbb{R}^d$ is the embedding of the special token $*$ at the starting position of mention

$m_i$. The choice of the $logsumexp$ pooling function is based on the research conducted by Jia et al. (2019). Their study offers empirical evidence that supports the use of this pooling function over others, as it facilitates accumulating weak signals from individual mentions, thanks to its smoother characteristics.

**Localised Context Embedding (LCE)**. ATLOP introduces the concept of localised context embedding to accommodate the variations in relevant mentions and context for different entity pairs $(e_s, e_o)$. Since the attention mechanism in the PLM captures the importance of each token within the context, it can be used to determine the context relevant for both entities. The importance of each token can be computed from the cross-token dependencies matrix $\boldsymbol{A}$ obtained in Equation 7.1. When evaluating entity $e_s$, the importance of individual tokens is determined by examining the cross-token dependencies across all mentions associated with $e_s$, denoted as $\mathcal{M}_{e_s}$. Initially, ATLOP collects and averages the attention $\boldsymbol{A}_{m_i} \in \mathbb{R}^{|\mathcal{T}_D|}$ at the special token $*$ preceding each mention $m_i \in \mathcal{M}_{e_s}$. This process results in $\boldsymbol{a}_s \in \mathbb{R}^{|\mathcal{T}_D|}$, which represents the importance of each token concerning entity $e_s$ (and analogously $\boldsymbol{a}_o$ for $e_o$). Subsequently, the importance of each token for a given entity pair $(e_s, e_o)$, denoted as $\boldsymbol{q}^{(s,o)} \in \mathbb{R}^{|\mathcal{T}_D|}$, is computed using $\boldsymbol{a}_s$ and $\boldsymbol{a}_o$ as follows:

$$\boldsymbol{q}^{(s,o)} = \frac{\boldsymbol{a}_s \circ \boldsymbol{a}_o}{\boldsymbol{a}_s^\top \boldsymbol{a}_o},$$

where $\circ$ represents the Hadamard product. Consequently, $\boldsymbol{q}^{(s,o)}$ represents a distribution that indicates the importance of each token for both tokens in $(e_s, e_o)$. Finally, the localised context embedding is computed as follows:

$$\boldsymbol{c}^{(s,o)} = \boldsymbol{H}^\top \boldsymbol{q}^{(s,o)}, \tag{7.3}$$

So $\boldsymbol{c}^{(s,o)} \in \mathbb{R}^d$ corresponds to a weighted average over all token embeddings that are important for both $e_s$ and $e_o$.

**Relation Classification**. To predict the relation between the subject entity $e_s$ and object entity $e_o$, ATLOP first generates context-aware subject and object representations as follows:

$$\boldsymbol{z}_s = tanh(\boldsymbol{W}_s[\boldsymbol{h}_{e_s}; \boldsymbol{c}^{(s,o)}] + \boldsymbol{b}_s)$$

$$\boldsymbol{z}_o = tanh(\boldsymbol{W}_o[\boldsymbol{h}_{e_o}; \boldsymbol{c}^{(s,o)}] + \boldsymbol{b}_o),$$

where $\boldsymbol{z}_s, \boldsymbol{z}_o \in \mathbb{R}^d$, $[\cdot;\cdot]$ represents the concatenation of two vectors, and $\boldsymbol{W}_s, \boldsymbol{W}_o \in \mathbb{R}^{d\times 2d}$ together with $\boldsymbol{b}_s, \boldsymbol{b}_o \in \mathbb{R}^d$ are trainable parameters. Then, the entity pair representation is computed as:

$$\boldsymbol{x}^{(s,o)} = \boldsymbol{z}_s \otimes \boldsymbol{z}_o,$$

where $\boldsymbol{x}^{(s,o)} \in \mathbb{R}^{d^2}$ and $\otimes$ stands for the vectorised Kronecker product. Finally, relation scores are computed as:

$$\boldsymbol{y}^{(s,o)} = \boldsymbol{W}_r \boldsymbol{x}^{(s,o)} + \boldsymbol{b}_r,$$

where $\boldsymbol{y}^{(s,o)} \in \mathbb{R}^{|\mathcal{R}|}$, with $\boldsymbol{W}_r \in \mathbb{R}^{|\mathcal{R}|\times d^2}$ and $\boldsymbol{b}_r \in \mathbb{R}^{|\mathcal{R}|}$ representing learnable parameters. The probability of relation $r \in \mathcal{R}$ between the subject and object entities is computed as follows:

$$P(r|s,o) = \sigma(\boldsymbol{y}^{(s,o)}),$$

where $\sigma$ is the sigmoid function. To reduce the number of parameters in the classifier, a grouped function is used, which splits the embedding dimensions into $k$ equal-sized groups and applies the function within the groups as follows:

$$\boldsymbol{z}_s = [\boldsymbol{z}_s^1; \ldots; \boldsymbol{z}_s^k] \tag{7.4}$$

$$\boldsymbol{z}_o = [\boldsymbol{z}_o^1; \ldots; \boldsymbol{z}_o^k] \tag{7.5}$$

$$\boldsymbol{x}^{(s,o)} = [\boldsymbol{x}^{(s,o)^1}; \ldots; \boldsymbol{x}^{(s,o)^k}]$$

$$\boldsymbol{y}^{(s,o)} = \sum_{i=1}^{k} \boldsymbol{W}_r^i \boldsymbol{x}^{(s,o)^i} + \boldsymbol{b}_r,$$

where $\boldsymbol{z}_s^i, \boldsymbol{z}_o^i \in \mathbb{R}^{d/k}$, $\boldsymbol{x}^{(s,o)^i} \in \mathbb{R}^{d^2/k}$, and $\boldsymbol{W}_r^i \in \mathbb{R}^{|\mathcal{R}|\times d^2/k}$. This way, the number of parameters can be reduced from $d^2$ to $d^2/k$.

**Loss Function**. ATLOP introduces the adaptive thresholding loss concept. This approach involves training a model to learn a hypothetical threshold class $TH$, which dynamically adjusts for each relation class $r \in \mathcal{R}$. During training, for each entity pair $(e_s, e_o)$, the loss enforces the model to generate scores above $TH$ for positive relation classes $\mathcal{R}_P$ and scores below $TH$ for negative relation classes $\mathcal{R}_N$. The loss is computed as follows:

$$\mathcal{L} = -\sum_{s \neq o} \sum_{r \in \mathcal{R}_P} \frac{exp(y_r^{(s,o)})}{\sum_{r' \in \mathcal{R}_P \cup \{TH\}} exp(y_{r'}^{(s,o)})} - \frac{exp(y_{TH}^{(s,o)})}{\sum_{r' \in \mathcal{R}_N \cup \{TH\}} exp(y_{r'}^{(s,o)})}$$

## 7.7 Proposed Method: *GADePo*

We propose to avoid the reliance on the EE (i.e., $\boldsymbol{h}_e$) and LCE (i.e., $\boldsymbol{c}^{(s,o)}$) heuristic aggregation functions by leveraging Transformers' attention functions to do aggregation. Given the observation of Henderson (2020); Mohammadshahi and Henderson (2020, 2021); Henderson et al. (2023) that attention weights and graph relations are functionally equivalent, we introduce the inductive biases of EE and LCE directly into the model's input as graph relations.

Our proposed Graph-Assisted Declarative Pooling (*GADePo*) method replaces the hand-coded aggregation functions EE and LCE with a declarative graph specification. By using the intrinsic graph processing capabilities of the Transformer model, the specified graph serves as an explicit high-level directive for the information aggregation process of the Transformer. By inputting the graph relations to the Transformer's self-attention layers, *GADePo* enables the aggregation to be steered by domain-specific knowledge or desired outcomes, while still allowing it to be learned by the Transformer, opening up the possibility for a more tailored and customised yet data-driven relation extraction.

Our *GADePo* model is illustrated in the right portion of Figure 7.3. We address both EE and LCE with the introduction of two special tokens, <ent> (i.e., entity) and <pent> (i.e., pair entity), and two explicit graph relations of types <ent> $\longleftrightarrow *$ and <pent> $\longleftrightarrow *$ in both directions, where $*$ represents the special token at the starting position of a specific mention. The set of relations is specified as $\boldsymbol{c}_{ij} \in \mathcal{C}$ which each identify the relation label from $i$ to $j$. Each of these relation labels is associated with an embedding vector of dimension $d$, as are the special token inputs <ent> and <pent>. These two special tokens are added to the PLM's vocabulary of input tokens, while relation label embeddings are input to the self-attention functions for every pair of related tokens. These new embeddings represent learnable parameters that are trained during the PLM fine-tuning on the downstream tasks. As reported in Appendix C, *GADePo* adds a negligible number of extra parameters, namely only the special token inputs and the graph directional relation inputs.

**Special Token <ent>**. To tackle the EE pooling function, we add to the input tokens $\mathcal{T}_D$ as many <ent> special tokens as entities in the document. This way each entity $e$ has a corresponding entity token <ent> in the input. We connect each <ent> token with its corresponding cluster of mentions $\mathcal{M}_e=\{m_i\}_{i=1}^{|\mathcal{M}_e|}$, and vice-versa. The two graph relations we use are thus <ent> $\longrightarrow *$ and $* \longrightarrow$ <ent>, where $*$ represents the special token at the starting position of mention $m_i$. Each <ent> token receives the same <ent> embedding, with no positional encoding, since each one collectively represents a set of mentions from different positions in the input graph. These identical inputs are only disambiguated through the connections to and from mentions expressed as the <ent> $\longrightarrow *$ and $* \longrightarrow$ <ent> graph relations. These relations tell the self-attention mechanism to use the <ent> token to aggregate information from the associated mentions, and thus the <ent> tokens have a direct correspondence to the computed $\boldsymbol{h}_e$ in Equation 7.2.

**Special Token <pent>**. ATLOP performs information filtering by calculating via Equation 7.3 a localised context embedding (LCE) $\boldsymbol{c}^{(s,o)}$ that is dependent on the cross-token attention matrix $\boldsymbol{A}$ output by the PLM. The intuition behind it is that the dependencies between different tokens are encoded as attention weights. We propose a straightforward adjustment of the input graph used for the EE pooling to effectively model and capture these dependencies. To address the LCE pooling function, we add to the input tokens $\mathcal{T}_D$ as many <pent> special tokens as the number of all possible pairs of entities. Each special token <pent> thus refers to a pair of entities $(e_s, e_o)$. We connect each <pent> token with each mention in the two clusters of mentions $\mathcal{M}_{e_s}=\{m_i\}_{i=1}^{|\mathcal{M}_{e_s}|}$ and $\mathcal{M}_{e_o}=\{m_i\}_{i=1}^{|\mathcal{M}_{e_o}|}$ and vice-versa. Since the attention weights used in LCE are computed from these mention embeddings, we expect that they are sufficient for the Transformer to learn to find the relevant contexts. The two graph relations we use are thus <pent> $\longrightarrow *$ and $* \longrightarrow$ <pent>. Analogously to the <ent> tokens, the <pent> tokens all receive the same <pent> embedding, with no positional embeddings, and thus are only disambiguated by their different <pent> $\longrightarrow *$ and $* \longrightarrow$ <pent> graph relations. These relations tell the <pent> token to pay attention to its associated mentions, which in turn allows it to find the relevant context shared by these mentions. Thus, each <pent> token can be seen as having a direct correspondence to the computed $\boldsymbol{c}^{(s,o)}$ in Equation 7.3.

All equations relative to the relation classification and the corresponding loss function remain valid as we merely substitute the hand-coded computations of $\boldsymbol{h}_e$ and $\boldsymbol{c}^{(s,o)}$ with the embeddings of <ent> and <pent>, respectively.

**Text-Graph Encoding**. We follow Chapter 6 and Mohammadshahi and Henderson (2020, 2021); Henderson et al. (2023) in leveraging the intrinsic graph processing capabilities of the Transformer model by incorporating graph relations as relation embeddings input to the self-attention function. For every pair of input tokens $ij$, the pre-softmax attention weight $e_{ij} \in \mathbb{R}$ is computed from both the respective token embeddings $\boldsymbol{x}_i, \boldsymbol{x}_j \in \mathbb{R}^d$, and an embeddings of the graph relation $\boldsymbol{c}_{ij}$ between the $i$-th and $j$-th tokens. However, we change the attention weight computation to:

$$e_{ij} = \frac{\boldsymbol{x}_i \boldsymbol{W}_Q \ \mathrm{diag}(\mathrm{LN}(\mathbf{1} + \boldsymbol{c}_{ij} \boldsymbol{W}_C)) \ (\boldsymbol{x}_j \boldsymbol{W}_K)^\top}{\sqrt{d}}, \tag{7.6}$$

where $\boldsymbol{W}_Q, \boldsymbol{W}_K \in \mathbb{R}^{d \times d}$ represent the query and key matrices, respectively. $\boldsymbol{c}_{ij} \in \{0,1\}^{|\mathcal{C}|}$ represents a $0/1$ encoded label of the graph relation between the $i$-th and $j$-th input elements, and $\boldsymbol{W}_C \in \mathbb{R}^{|\mathcal{C}| \times d}$ represents the relations' embedding matrix. Thus, $\boldsymbol{c}_{ij}\boldsymbol{W}_C$ is the embedding of the relation between $i$ and $j$, LN stands for the $LayerNorm$ operation, and $\mathrm{diag}(\mathbf{1} + \ldots)$ maps this vector into a diagonal matrix plus the identity matrix.

Compared to the standard attention function, where $e_{ij} = \boldsymbol{x}_i \boldsymbol{W}_Q (\boldsymbol{x}_j \boldsymbol{W}_K)^\top / \sqrt{d}$, the relation embedding determines a weighting of the different dimensions. This is a novel way to condition on the relation embedding compared to the original formulation, which only models query-relation interactions (Mohammadshahi and Henderson, 2020). This change is motivated by our task requiring a more flexible formulation which models query-relation-key interactions via a multiplicative mechanism, without requiring a full $d \times d$ matrix of bi-linear parameters. This way, a key will be relevant to a query only when both agree on the relation. In preliminary experiments, we explored various methods for biasing attention and found that the formulation presented in Equation 7.6 produced the best results.

## 7.8 Experiments

### 7.8.1 Datasets and Models

**Re-DocRED** (Tan et al., 2022b) is a revisited and improved version of the DocRED (Yao et al., 2019b) dataset. It is built from English Wikipedia and Wikidata and contains both distantly-supervised and human-annotated documents with named entities, coreference data, and intra- and inter-sentence relations, supported by evidence. It requires analysing multiple sentences to identify entities, establish their relationships, and integrate information from the entire document. We comply with the model used by the authors and employ the RoBERTa$_{\text{LARGE}}$ (Zhuang et al., 2021) model in our experiments.

**HacRED** (Cheng et al., 2021) is a large-scale, high-quality Chinese document-level relation extraction dataset, with a special focus on practical hard cases. As the authors did not provide specific information about the model used in their study, we conducted our experiments using the Chinese BERT$_{\text{BASE}}$ with whole word masking model (Cui et al., 2021).

**Datasets statistics**. Re-DocRED and HacRED exhibit notable distinctions in their statistics, as summarised in Table 7.1. Re-DocRED comprises a larger number of facts, entities per document, and relations compared to HacRED. This indicates a potentially richer and more extensive dataset in terms of factual information and relationship types. However, HacRED contains more documents and may present a broader range of scenarios for relation extraction, including more challenging cases, as it has been specifically created with a focus on practical hard cases.

| **Statistic** | **Re-DocRED** | **HacRED** |
|---|---|---|
| Facts | 120,664 | 65,225 |
| Relations | 96 | 26 |
| Documents | 4,053 | 9,231 |
| Average Entities | 19.4 | 10.8 |

Table 7.1: Re-DocRED and HacRED human-annotated datasets statistics.

### 7.8.2 Results and Discussion

We follow the standard practice from prior research and report the results of our experiments on the Re-DocRED and HacRED datasets in Table 7.2 and Figure 7.5. For all datasets and models, we provide our reimplementation of the ATLOP baseline (indicated as ATLOP$^\star$), which achieves or surpasses previously reported results for ATLOP, and compare the proposed **GADePo** model against this model. We evaluate all datasets using the $F_1$ metric.

For Re-DocRED, Ign $F_1$ (or Ignored $F_1$) is also reported, and refers to the $F_1$ score that excludes relational facts that are shared between the training and development/test sets. This is done to avoid potential biases in the evaluation metrics due to overlap in content between the sets, which might not reflect the model's ability to generalise to truly unseen data.

For HacRED, we adhere to the format introduced by Cheng et al. (2021) and report also the Precision ($P$) and Recall ($R$) metrics. We comply with previous research and report the test score achieved by the best checkpoint on the development set. In Appendix C, we additionally present the mean and standard deviation on the development set, calculated from five training runs with distinct random seeds.

Appendix C also includes results on the original DocRED dataset. As Re-DocRED is a revisited and improved version, we consider it the more reliable dataset for evaluation. The original DocRED results are provided for completeness but are not central to our analysis.

Training details and hyperparameters are outlined in Appendix C.

**Re-DocRED Results**. We evaluate our proposed *GADePo* method against the previous ATLOP method in two stages, first comparing the use of `<ent>` tokens against the use of EE pooling ($\boldsymbol{h}_e$), and then comparing our full model against the full ATLOP model, including `<pent>` tokens and LCE pooling ($\boldsymbol{c}^{(s,o)}$), respectively.

Table 7.2 highlights the effectiveness of our proposed method. When comparing $\boldsymbol{h}_e$ with `<ent>`, we observe a noticeable improvement in both Ign $F_1$ and $F_1$ scores, achieving 75.55% and 76.38% respectively, compared to 75.27% and 75.92% attained by ATLOP$^\star$. This demonstrates the practical utility of employing the special token `<ent>` for information aggregation. This is illustrated in the attention weights heatmap in Figure 7.4. Incorporating $\boldsymbol{c}^{(s,o)}$ and `<pent>` into the comparison, *GADePo* maintains performance parity with the significantly enhanced ATLOP$^\star$, which outperformed ATLOP$^\bullet$ from Tan et al. (2022b). The latter improvement suggests that a more refined hyperparameter search can lead to performance gains, as evidenced by the increase in $F_1$ score from 77.56% to 78.38%. *GADePo* achieves an $F_1$ score of 78.40%, affirming its competitive edge and the effectiveness of employing `<pent>` for aggregation.

| | | Re-DocRED | | HacRED | | |
|---|---|---|---|---|---|---|
| **Model** | **Aggregation** | Ign $F_1$ | $F_1$ | $P$ | $R$ | $F_1$ |
| ATLOP$^\star$ | $\boldsymbol{h}_e$ | 75.27 | 75.92 | **76.27** | 76.83 | 76.55 |
| GADePo (ours) | `<ent>` | **75.55** | **76.38** | 74.13 | **79.46** | **76.70** |
| ATLOP$^{\bullet,\diamond}$ | $\boldsymbol{h}_e$ ; $\boldsymbol{c}^{(s,o)}$ | 76.82 | 77.56 | 77.89 | 76.55 | 77.21 |
| ATLOP$^\star$ | $\boldsymbol{h}_e$ ; $\boldsymbol{c}^{(s,o)}$ | 77.62 | 78.38 | 76.36 | 78.86 | 77.59 |
| GADePo (ours) | `<ent>` ; `<pent>` | **77.70** | **78.40** | **78.27** | **79.03** | **78.65** |

Table 7.2: Comparative analysis between the previous method ATLOP and the proposed method *GADePo* on the test set. ATLOP$^\star$ indicates our reimplementation of the previous method. For Re-DocRED and HacRED we report in percentage the results obtained by Tan et al. (2022b) (ATLOP$^\bullet$) and Cheng et al. (2021) (ATLOP$^\diamond$), respectively. The results are reported in terms of $F_1$ scores, Precision ($P$), and Recall ($R$), following the same metrics reported in prior research specific to each dataset. Ign $F_1$ denotes the $F_1$ score that excludes relational facts shared between the training and evaluation sets. We also comply with the standard practice where test scores are obtained using the checkpoint that achieved the best performance on the development set across five training runs with distinct random seeds.

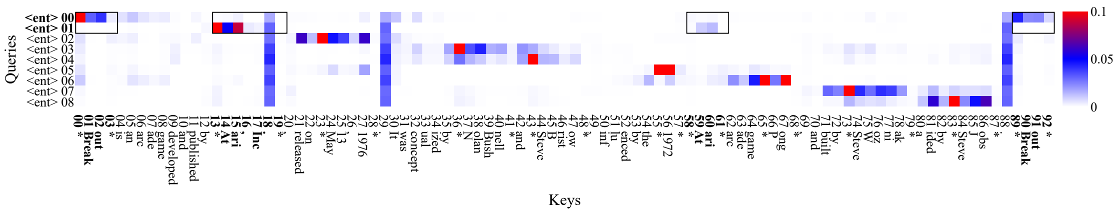



Figure 7.4: Attention weights $\boldsymbol{A}$ from *GADePo* via Equation 7.1 for the document in Figure 7.2. For clarity, only a subset of `<ent>` and document tokens are shown on the $y$-axis (queries) and $x$-axis (keys), respectively.

Table 7.3 illustrates the results obtained with prior finetuning on the distantly supervised dataset, which contains approximately 100K documents (Yao et al., 2019b). Interestingly, distant supervision appears to have a slightly negative impact on the results of both methods when incorporating $\boldsymbol{c}^{(s,o)}$ or `<pent>`. However, it proves to be highly beneficial when utilising solely $\boldsymbol{h}_e$ or `<ent>` for aggregation. This suggests that although distant supervision might introduce noise into the training process, it can also provide valuable information that improves model generalisation, particularly when leveraging simpler feature representations like $\boldsymbol{h}_e$ and `<ent>`, possibly due to their robustness in capturing essential information amidst noise.

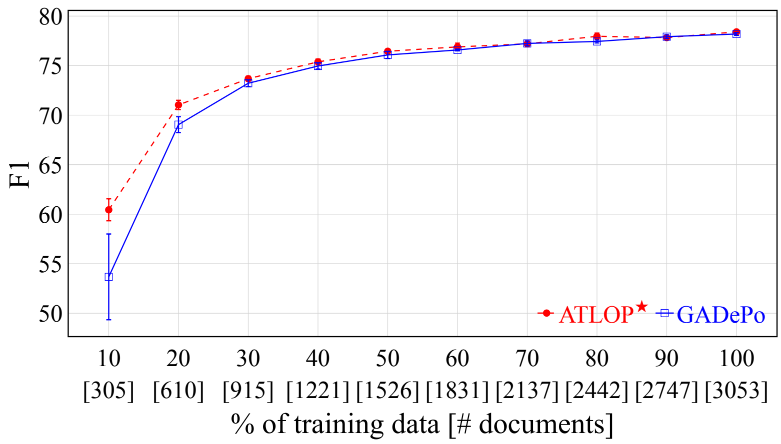
| Model | Aggregation | Ign $F_1$ | $F_1$ |
|---|---|---|---|
| ATLOP$^\star$ | $\boldsymbol{h}_e$ | 76.39 | 76.97 |
| GADePo (ours) | `<ent>` | **76.99** | **77.79** |
| ATLOP$^\star$ | $\boldsymbol{h}_e$ ; $\boldsymbol{c}^{(s,o)}$ | 77.49 | 78.09 |
| GADePo (ours) | `<ent>` ; `<pent>` | **77.50** | **78.15** |

Table 7.3: Re-DocRED results on the test set following prior finetuning on the distantly supervised dataset.

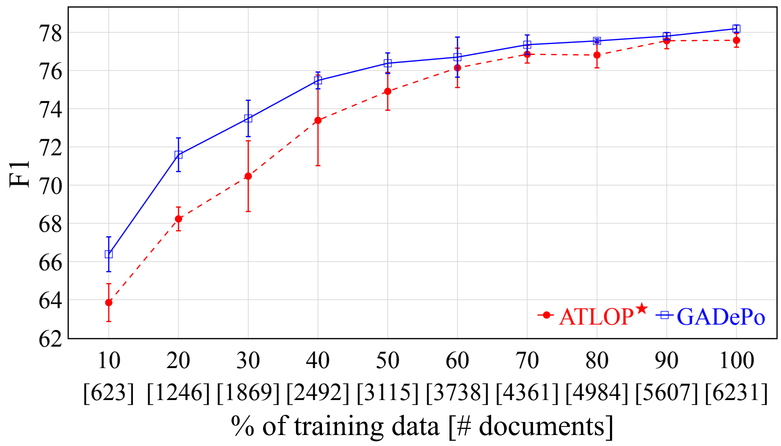
**HacRED Results**. We observe a similar pattern to Re-DocRED, with ATLOP$^\star$ displaying a slight performance advantage over ATLOP$^\diamond$ from Cheng et al. (2021) (Table 7.2). On this dataset, *GADePo* shows a significantly improved performance, primarily driven by a substantial increase in Recall ($R$), indicating that the *GADePo* model is more effective at identifying relevant instances. As already reported for the Re-DocRED dataset, the performance boost after the inclusion of $\boldsymbol{c}^{(s,o)}$ and `<pent>` into ATLOP$^\star$ and *GADePo*, respectively, highlight the significant contributions of these features. *GADePo* outperforms ATLOP$^\star$ with an $F_1$ score of 78.65% compared to 77.59%. This larger improvement on HacRED suggests that *GADePo* is better at handling challenging cases, which is not surprising given its greater flexibility over the fixed pooling functions of ATLOP.

**Data Ablation**. To evaluate the models' sensitivity to dataset size, the performance evaluation depicted in Figure 7.5 compares ATLOP$^\star$ ($\boldsymbol{h}_e$ ; $\boldsymbol{c}^{(s,o)}$) and *GADePo* (`<ent>` ; `<pent>`) on the development set, considering different amounts of training data availability on the Re-DocRED and HacRED datasets. Accuracies generally converge as the dataset sizes increase, but on the challenging cases of HacRED, *GADePo* maintains a substantial advantage across the full range. On Re-DocRED, *GADePo* catches up with and slightly outperforms ATLOP$^\star$ as data size increases. This lower performance on smaller datasets is presumably because *GADePo* must learn how to exploit the graph relations to the special tokens `<ent>` and `<pent>` and pool information through them, whereas for ATLOP this pooling is hand-coded. On the Re-DocRED dataset, ATLOP$^\star$ appears to have relatively consistent variance, whereas *GADePo* exhibits higher variance in the smaller training sets. In contrast, on the HacRED dataset, *GADePo* is significantly more stable for smaller datasets.

The data ablation analysis shows that the performance of hand-coded pooling functions can be dataset-specific, which restricts their adaptability. In contrast, *GADePo* consistently outperforms its hand-coded counterparts on larger datasets, and matches them on all but some smaller datasets, presumably due to its flexibility. This pattern suggests that *GADePo* has a greater potential for optimisation, particularly on larger datasets. This is supported by *GADePo*'s better performance on HacRED, which is both larger and designed to be more challenging than Re-DocRED.




(a) Re-DocRED




(b) HacRED

Figure 7.5: Performance of ATLOP$^\star$ ($\boldsymbol{h}_e$ ; $\boldsymbol{c}^{(s,o)}$) and *GADePo* (`<ent>` ; `<pent>`) on the development set under varying data availability conditions on Re-DocRED (7.5a) and HacRED (7.5b). The $x$-axis represents the percentage and number of documents from the training dataset, while the $y$-axis displays the $F_1$ score in percentage. Each point on the graph represents the mean value, while error bars indicate the standard deviation derived from five distinct training runs with separate random seeds.

## 7.9 Conclusion

In this chapter we proposed a novel approach to document-level relation extraction, challenging the conventional reliance on hand-coded pooling functions for information aggregation. Our method leverages the power of Transformer models by incorporating explicit graph relations as instructions for information aggregation. By combining graph processing with text-based encoding, we introduced the Graph-Assisted Declarative Pooling (*GADePo*) specification, which allows for more flexible and customisable specification of pooling strategies which are still learned from data.

We conducted evaluations using diverse datasets and models commonly employed in document-level relation extraction tasks. The results of our experiments demonstrated that our approach achieves promising performance that is comparable to or better than that of hand-coded pooling functions. This suggests that our method can serve as a viable basis for other relation extraction methods, providing a more adaptable and tailored approach. In particular, recent methods have improved performance by exploiting information about evidence, which can naturally be incorporated in our graph-based approach.

These findings also relate to the example of Larry and Palmerston in Figure 1.1. In that case, relevant information such as overlapping tenures, institutional roles, and reports of confrontations might appear in separate spans of unstructured text. While earlier chapters focused on improving the retrieval of such evidence (Chapter 4) and aligning generated responses with it (Chapter 5), the *GADePo* mechanism addresses how this evidence is pooled. By using graph-structured inputs to specify pooling strategies, *GADePo* helps the model integrate related but distant spans more effectively, improving its ability to potentially recover relations such as the rivalry between Larry and Palmerston.

## 7.10 Limitations

While the proposed *GADePo* model offers a promising and innovative approach to relation extraction, there are issues which the current study does not address. According to the data in Table 7.1, the average number of entities per document across datasets is approximately 15. This means that, on average, there will be an additional 15 `<ent>` tokens and 105 `<pent>` tokens. Given that the maximum allowable input length for the models is 512 tokens, the inclusion of these extra tokens results in roughly a 3% and 20% increase in the overall input length for `<ent>` and `<pent>`, respectively. It's evident that the majority of the increase in input length is due to the quadratic number of `<pent>` special tokens, but we believe that an appropriate pruning strategy could easily reduce this number to linear in the number of entities without degrading accuracy. One such pruning strategy could involve an `<ent>`-only model with a binary classifier which is trained to predict pairs of related entities. This model could then be used to prune the set of candidate entity pairs for the final relation classification, with `<pent>` tokens being instantiated only for these candidate pairs. We have chosen to leave this approach as a potential avenue for future work, opting instead to focus on demonstrating the promise of the current simpler formulation.

## 7.11 Ethics Statement

We do not anticipate any ethical concerns related to our work, as it primarily presents an alternative approach to a previously proposed method. Our main contribution lies in introducing a novel methodology for relation extraction. In our experiments, we use the same datasets and pretrained models as previous research, all of which are publicly available. However, it is important to acknowledge that these datasets and models may still require further examination for potential fairness issues and the knowledge they encapsulate.

## 7.12 Acknowledgements

We extend our special gratitude to the Swiss National Science Foundation (SNSF) and Research Foundation – Flanders (FWO) for funding this work under grants 200021E_189458 and G094020N.

# 8 *FnF-TG*: Fast-and-Frugal Text-Graph Models

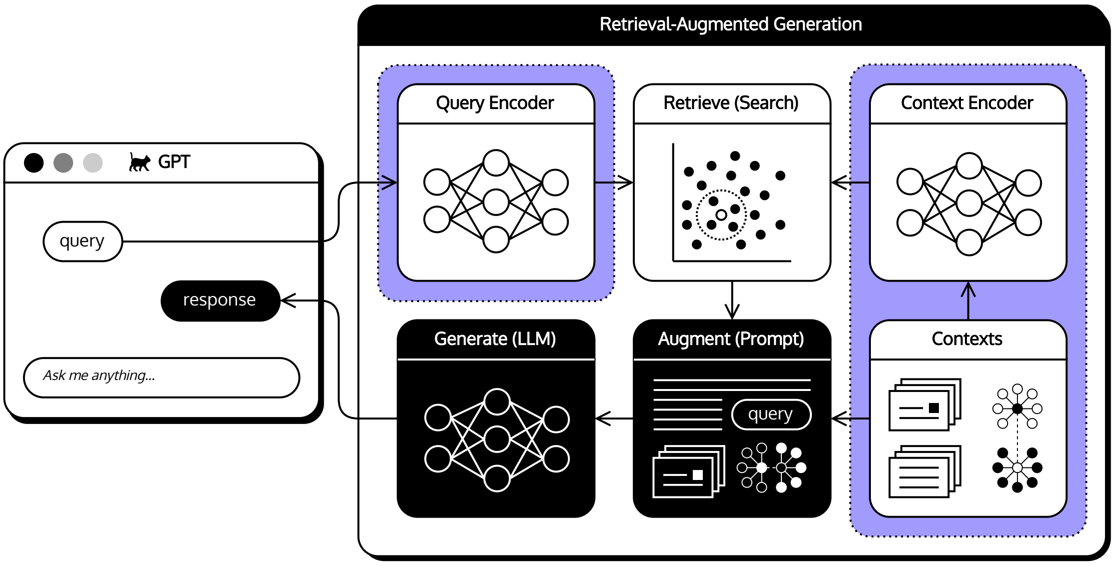



Figure 8.1: This chapter focuses on the *Query Encoder*, *Context Encoder*, and *Contexts* components of the RAG pipeline from Chapter 3, and investigates how models can make use of textual and relational information to operate over structured knowledge in inductive settings, where both entities and relations may not have been observed during training. As illustrated in Figure 1.1, identifying the rivalry between Larry and Palmerston requires integrating information about their institutional roles, overlapping tenures, and prior interactions, which may be distributed across retrieved passages or represented in structured graph-based inputs. The chapter explores how such inputs can be encoded and combined to support link prediction in fully inductive settings.

## 8.1 Research Question

*How can models make use of textual and relational information to operate over structured knowledge in inductive settings?*

## 8.2 Publication

Coman, A. C., Theodoropoulos, C., Moens, M.-F., and Henderson, J. (2025b). Fast-and-frugal text-graph transformers are effective link predictors. In Che, W., Nabende, J., Shutova, E., and Pilehvar, M. T., editors, *Findings of the Association for Computational Linguistics: ACL 2025*, pages 11828–11841, Vienna, Austria. Association for Computational Linguistics

## 8.3 Summary

We propose Fast-and-Frugal Text-Graph (*FnF-TG*) Transformers, a Transformer-based framework that unifies textual and structural information for inductive link prediction in text-attributed knowledge graphs. We demonstrate that, by effectively encoding ego-graphs (1-hop neighbourhoods), we can reduce the reliance on resource-intensive textual encoders. This makes the model both fast at training and inference time, as well as frugal in terms of cost. We perform a comprehensive evaluation on three popular datasets and show that *FnF-TG* can achieve superior performance compared to previous state-of-the-art methods. We also extend inductive learning to a fully inductive setting, where relations don't rely on transductive (fixed) representations, as in previous work, but are a function of their textual description. Additionally, we introduce new variants of existing datasets, specifically designed to test the performance of models on unseen relations at inference time, thus offering a new test-bench for fully inductive link prediction.

## 8.4 Introduction

Knowledge graphs (KGs) represent complex information as a structured collection of entities and their relations. They are a fundamental component of various applications, including information extraction (Mintz et al., 2009; Bosselut et al., 2019; Theodoropoulos et al., 2021) and retrieval (Dalton et al., 2014; Gupta et al., 2019), question answering (Saxena et al., 2022; Yu et al., 2022; Coman et al., 2023), reasoning (Zhang et al., 2020a; Jiang et al., 2022; Niu et al., 2022), fact-aware language modelling (Logan et al., 2019; Yang et al., 2023), and many others (Fensel et al., 2020; Schneider et al., 2022).

Text-attributed KGs extend KGs by associating each entity and relation with a corresponding textual description, which provide a richer representation of the knowledge encoded in the graph (Daza et al., 2021). In particular, the text associated with an entity may provide a description of its relationships to other entities. This combination of explicit structural and implicit textual information makes modelling text-attributed KGs particularly challenging.

Initial attempts to model KGs focused on their graph nature, typically addressing a *transductive* setting (Bordes et al., 2013; Nickel et al., 2015; Wang et al., 2017). These models could only make predictions for entities observed during training and only considered the structural information of the KG, ignoring any textual information. To overcome this limitation, later work focused on using the textual descriptions in KGs to address an *inductive* setting (Xie et al., 2016; Shi and Weninger, 2018; Wang et al., 2021b), meaning that predictions can be made even for entities not observed during training, using entity representations computed based on their textual descriptions.

Combining information from textual descriptions and graph structures has proven crucial (Schlichtkrull et al., 2017). An entity's ego-graph, which represents it's 1-hop neighbourhood, provides valuable context that can help disambiguate its role and distinguish it from similar entities. While there has been progress in leveraging ego-graphs, we believe that there is significant room for more effective approaches.

Modelling text-attributed KGs in an inductive setting poses several challenges, particularly when it comes to effectively integrating textual and structural information in embeddings. Transformers (Vaswani et al., 2017) have shown remarkable success at modelling unstructured (text) data (Devlin et al., 2019; Raffel et al., 2020; Brown et al., 2020; Touvron et al., 2023). While their ability to model structured (graph) data is less evident, the inherent graph processing abilities of a Transformer's self-attention mechanism make it a natural fit for modelling graph structures (Henderson et al., 2023). Building on the established functional equivalence between attention weights and graph relations in Chapter 6, and extending the insights of Mohammadshahi and Henderson (2020, 2021); Miculicich and Henderson (2022); Coman et al. (2024), in this chapter, we propose Fast-and-Frugal Text-Graph (*FnF-TG*) Transformers, which unify textual and structural information in a framework based solely on Transformers.

Another challenge is that text encoders are resource-intensive, especially when the textual descriptions of both entities and their ego-graph neighbours need to be encoded, leading to considerably increased training and inference time (Markowitz et al., 2022). This cost can be reduced by using smaller text encoders, but they can be considerably less effective. We demonstrate that we can reduce the dependence on large text encoders with a more effective encoding of ego-graphs, using our *FnF-TG* Transformers and their more appropriate inductive biases. This makes the overall framework both fast in terms of time, and frugal in terms of cost.

A third challenge is that previous models fail to leverage the textual descriptions of relation labels. They still assume a fixed (transductive) inventory of relations, meaning that they cannot handle relations not seen during training and thus are not fully inductive. We propose an extension of this method to address the challenge of being fully inductive, by computing a relation embedding from the text describing that relation. This embedding serves as both the relation representation for link prediction, analogous to the transductive case, and as the relation label which is input to the self-attention mechanism of the *FnF-TG*'s graph encoder component.

We showcase the effectiveness of our proposed model on three popular datasets for inductive link prediction in text-attributed KGs from the experimental setting of Daza et al. (2021) and Wang et al. (2021b), namely $\text{WN18RR}_{\text{IND}}$, $\text{FB15k-237}_{\text{IND}}$, and $\text{Wikidata-5M}_{\text{IND}}$. We show that it improves over the state-of-the-art in all cases.

Additionally, we introduce new variants of existing datasets which are specifically designed to evaluate the performance of models on relations which are unseen until test time, thus offering a new test-bench for fully inductive link prediction.

**Contributions:**

1. We propose a KG embedding model which leverages the intrinsic graph processing capabilities of Transformers to effectively capture the information in both the KG's textual descriptions and the KG's graph structure.
2. We demonstrate that Fast-and-Frugal Text-Graph (*FnF-TG*) Transformers achieve superior performance compared to previous state-of-the-art results on three popular datasets, even with small and efficient text encoders.
3. We extend inductive KG learning to a fully inductive setting, where both entity and relation representations are computed as functions of their textual descriptions.
4. We conduct a controlled quantitative analysis of several properties of our and existing inductive link prediction models to evaluate their performance and ensure fair comparisons.
5. We introduce a new test-bench for fully inductive link prediction by modifying existing datasets to specifically test models' performance on unseen relations.

## 8.5 Related Work

**Transductive Link Prediction**. In this setting, link prediction aims to identify missing links within a fixed and fully observable graph where all entities and their other connections are known during training. Typically, it involves learning embeddings within a geometric space, as demonstrated by models like RESCAL (Nickel et al., 2011), NTN (Socher et al., 2013), TransE (Bordes et al., 2013), DistMult (Yang et al., 2014), ComplEx (Trouillon et al., 2016), TorusE (Ebisu and Ichise, 2018), RotatE (Sun et al., 2019), and SimplE (Kazemi and Poole, 2018). Additionally, there are approaches that incorporate convolutional layers such as R-GCN (Schlichtkrull et al., 2017), ConvE (Dettmers et al., 2018), HypER (Balazevic et al., 2018), and ConvR (Jiang et al., 2019). Moreover, recent advances have seen the integration of Transformers in models such as CoKE (Wang et al., 2019c) and HittER (Chen et al., 2021).

**Inductive Link Prediction**. In this setting, link prediction involves predicting missing links in a dynamic graph where only partial information is available during training. Much work has explored leveraging limited relational knowledge between novel entities and those already present in the training graph (Bhowmik and de Melo, 2020; Wang et al., 2020b). Examples include LAN (Wang et al., 2019b), IndTransE (Dai et al., 2021), OpenWorld (Shah et al., 2019), GraIL (Teru et al., 2020), NBFNet (Zhu et al., 2021), NodePiece (Galkin et al., 2021), and BERTRL (Zha et al., 2021). Moreover, approaches such as DKRL (Xie et al., 2016), Commonsense (Malaviya et al., 2020), KG-BERT (Yao et al., 2019a), KEPLER (Wang et al., 2021b), BLP (Daza et al., 2021), StAR (Wang et al., 2021a), SimKGC (Wang et al., 2022), StATIK (Markowitz et al., 2022), iHT (Chen et al., 2023), and KnowC (Yang et al., 2024a) use language models to encode entities based on their textual descriptions. Among these, StATIK (Markowitz et al., 2022) stands out as it combines both a language model and a graph encoder, specifically employing a Message Passing Neural Network (MPNN) (Gilmer et al., 2017) to create entity embeddings. This makes StATIK particularly relevant to our approach and we will use it as the state-of-the-art method of reference and compare our proposed method against it to demonstrate its effectiveness.

**Transformers and Graphs**. Graph Transformers (GTs) represent a significant evolution in graph input methods within the Transformer architecture (Henderson et al., 2023). Early work such as G2GT (Mohammadshahi and Henderson, 2020, 2021; Miculicich and Henderson, 2022) laid the foundation by incorporating explicit graphs into Transformer's latent attention graph. Later work introduced RoFormer (Su et al., 2021), which uses a rotation matrix to encode absolute positions, and Graphormer (Ying et al., 2021), which uses node centrality encoding and soft attention biases. Other models, like SSAN (Xu et al., 2021), JointGT (Ke et al., 2021), TableFormer (Yang et al., 2022), and GADePo (Coman et al., 2024), have applied GTs to various tasks such as document-level relation extraction, knowledge-to-text generation, table-based question answering, and graph-aware declarative pooling.

We continue to advance graph input methods and show that GTs, when combined with an effective inductive bias in the input and the latent attention graph, achieve superior performance compared to the previous state-of-the-art.

## 8.6 Background

### 8.6.1 Inductive Representation Learning

A text-attributed knowledge graph can be defined as $\mathcal{G} = (\mathcal{E}, \mathcal{R}, \mathcal{T}, \mathcal{D})$ where $\mathcal{E}$ represents the set of entities, $\mathcal{R}$ denotes the set of relation labels, $\mathcal{T}$ consists of the set of relation triples $(h, r, t) \in \mathcal{E} \times \mathcal{R} \times \mathcal{E}$, and $\mathcal{D}$ contains the textual descriptions associated with entities and relation labels. In each triple, $h$ and $t$ represent the head and tail entities, respectively, which are connected by a directional relation $r$ (Daza et al., 2021).

Inductive link prediction involves completing missing triples in the graph by leveraging the textual descriptions associated with the entities and relation labels. If the textual description of an entity mentions the target relation, then this resembles an open relation extraction task (Banko et al., 2007), and if not then it resembles a knowledge graph completion task (Lin et al., 2015), with a continuum of difficulties in between.

Our goal is to learn an embedder that maps these descriptions and the partial graph to a representation space where the missing triples can be inferred. Specifically, given a training graph $\mathcal{G}_{train} = (\mathcal{E}_{train}, \mathcal{R}, \mathcal{T}_{train}, \mathcal{D}_{train})$, where $\mathcal{E}_{train} \subset \mathcal{E}$, $\mathcal{D}_{train} \subset \mathcal{D}$ and $\mathcal{T}_{train} \subset \mathcal{T}$ only includes triples involving entities in $\mathcal{E}_{train}$, the goal is to infer the missing triples in $\mathcal{T} \setminus \mathcal{T}_{train}$.

During evaluation, for a given query triple $\mathcal{T}_i = (h, r, t)$, the model is tasked with performing head or tail prediction on the graph $\mathcal{G} \setminus \mathcal{T}_i$. This involves two types of queries: tail prediction, where the query is of the form $(h, r, \hat{e})$ and head prediction, where the query is of the form $(\hat{e}, r, t)$. In both cases, the model must rank all possible candidate entities $\hat{e} \in \hat{\mathcal{E}}$ to identify the correct entity $\hat{e}_t$ or $\hat{e}_h$ and place it at the top of the ranked list.

This setup can be viewed through the lens of the RAG formulation from Figure 8.1, where the triple to be completed acts as a query encoded by the *Query Encoder*, while supporting descriptions and graph context serve as retrieved *Contexts* processed by the *Context Encoder*.

### 8.6.2 Structural Objective and Loss Function

We adopt the margin-based ranking loss from Bordes et al. (2013) as our optimisation criterion. We construct two sets of triples: a set of true triples $T$ and a set of negative triples $T'$, where a negative triple consists of a corrupted version of a true triple with either the head or tail entity replaced by a random entity (target excluded) from the training minibatch. We define the structural objective function $f$ using the TransE (Bordes et al., 2013) model, which represents each triple $(h, r, t)$ as:

$$f_{\text{TransE}}(h, r, t) = -||\boldsymbol{h} + \boldsymbol{r} - \boldsymbol{t}||_1,$$

where $\boldsymbol{h}$, $\boldsymbol{r}$, and $\boldsymbol{t}$ are the vector representations of the head entity, relation, and tail entity, respectively. The loss function is then defined as:

$$Loss = \sum_{(t,t') \in (T \times T')} max(0, 1 - f(t) + f(t')),$$

where $f(t)$ and $f(t')$ are the scores assigned to the true triple $t$ and the negative triple $t'$, respectively.

### 8.6.3 Evaluation and Metrics

**Evaluation Scenarios**. We assess our models in two inductive scenarios, following Bordes et al. (2013). In the first setting, called *dynamic* evaluation, new entities may appear in the head or tail positions, and the candidates set is defined as $\hat{\mathcal{E}} = \mathcal{E}_{train} \cup \mathcal{E}_{eval}$. In the second setting, called *transfer* evaluation, both head and tail entities are new and unseen during training, and the candidates set is defined as $\hat{\mathcal{E}} = \mathcal{E}_{eval}$, where $\mathcal{E}_{eval}$ is disjoint from the training set of entities $\mathcal{E}_{train}$.

**Metrics**. For each evaluation triple, we create two types of queries: $(h, r, \hat{e})$ for predicting tails and $(\hat{e}, r, t)$ for predicting heads, where $\hat{e} \in \mathcal{E}$ represents all possible candidate entities, as described in Subsection 8.6.1. We rank candidate triples by their scores and evaluate the ranking of the correct triple. We report Mean Reciprocal Rank (MRR) and Hits@k (H@k) with $k \in \{1, 3, 10\}$ averaged across head and tail prediction tasks. We adopt the *filtered* setting as in Bordes et al. (2013), removing valid triples from the set of negative candidate triples when ranking candidate targets.

## 8.7 Proposed Architecture

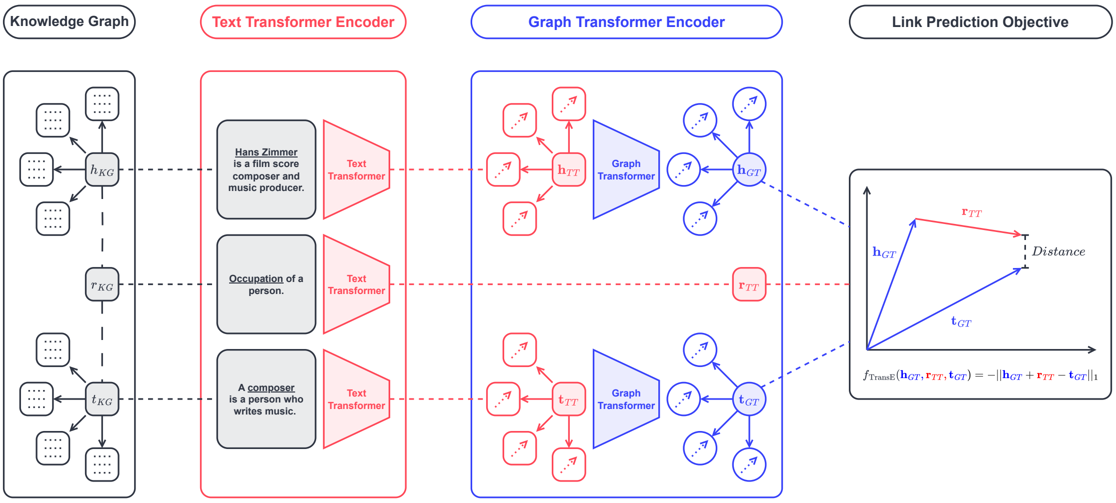



Figure 8.2: Architecture of the proposed Fast-and-Frugal Text-Graph (FnF-TG) Transformer model. The *Knowledge Graph* contains a set of triples of type $(h_{KG}, r_{KG}, t_{KG})$ along with their corresponding textual descriptions. For each head $h_{KG}$ and tail $t_{KG}$ we extract their ego-graphs (1-hop neighbourhood), denoted as $E(h_{KG})$ and $E(t_{KG})$, respectively. The textual descriptions of the elements in the ego-graph are first passed to the *Text Transformer Encoder*, which produces vector representations $(\boldsymbol{h}_{TT}, \boldsymbol{r}_{TT}, \boldsymbol{t}_{TT})$ together with ego-graphs $E(\boldsymbol{h}_{TT})$, and $E(\boldsymbol{t}_{TT})$. The output of this module is then input into the *Graph Transformer Encoder*, which produces vector representations $\boldsymbol{h}_{GT}$ and $\boldsymbol{t}_{GT}$. Finally, the *Link Prediction Objective* is applied to the combined representations $(\boldsymbol{h}_{GT}, \boldsymbol{r}_{TT}, \boldsymbol{t}_{GT})$.

Figure 8.2 shows the architecture of our proposed model, which contains three main components: *Knowledge Graph* (KG), *Text Transformer Encoder* (TT) and *Graph Transformer Encoder* (GT).

**Knowledge Graph**. The text-attributed KG component contains a set of triples of type $(h_{KG}, r_{KG}, t_{KG})$, along with their corresponding textual descriptions. For each head $h_{KG}$ and tail $t_{KG}$, we also extract their ego-graphs (1-hop neighbourhood), denoted as $E(h_{KG})$ and $E(t_{KG})$, respectively. Then we encode each of these nodes with the text encoder, discussed below. Specifically, for each centre entity, we encode its textual description, as well as encoding the textual descriptions of its neighbouring entities and the relations that connect the centre entity to its neighbours.

**Text Transformer Encoder**. The textual descriptions from the KG module are passed to the *Text Transformer Encoder* (TT), which produces vector representations $\boldsymbol{x}_{TT}$ for each entity and relation textual description $x_{KG}$ in the ego-graph. More formally, we apply the following function:

$$\boldsymbol{x}_{TT} = \sigma(\text{BERT}_{\text{SIZE}}(x_{KG})_{\texttt{[CLS]}}\boldsymbol{W}_0)\boldsymbol{W}_{\mathbf{1}},$$

where $\boldsymbol{W}_0, \boldsymbol{W}_1 \in \mathbb{R}^{d \times d}$ are two linear projection matrices and $\sigma$ is the SiLU (Elfwing et al., 2017) activation function. We employ BERT (Devlin et al., 2019) as our encoder and use the `[CLS]` vector representation output by the encoder as the embedding. $\text{BERT}_{\text{SIZE}}$ indicates that we employ different sizes of this model released by Turc et al. (2019).

When encoding candidate entities $\hat{e} \in \hat{\mathcal{E}}$, we simply pass the text associated with the entity through the TT component. The same is true for any neighbouring entities required by the graph encoder (discussed below). In contrast, when encoding queries $(h, r, \cdot)$ and $(\cdot, r, t)$ we condition $h$ and $t$ on the relation type $r$. Similarly to StAR (Wang et al., 2021a) and StATIK (Markowitz et al., 2022), for tail prediction queries $(h, r, \cdot)$ we concatenate the text associated with $h_{KG}$ and $r_{KG}$, resulting in $[h||r]_{KG}$. For head prediction queries $(\cdot, r, t)$, we create an inverse version of the relation text by prepending its textual description with the text `"inverse of"`, denoted as $r^{-1}_{KG}$. We then concatenate the text associated with $t_{KG}$ and $r^{-1}_{KG}$, resulting in $[t||r^{-1}]_{KG}$.

**Graph Transformer Encoder**. The outputs of the TT component, together with the ego-graphs $E(\boldsymbol{h}_{TT})$ and $E(\boldsymbol{t}_{TT})$ are then input to the *Graph Transformer Encoder* (GT) (Figure 8.3). The input embedding for each node of the graph is the vector output by the TT encoder for that entity, as described above. In addition, we add learnable segment embeddings to each node input, indicated as $s_{\texttt{[CENTRE]}}$ and $s_{\texttt{[NEIGHBOUR]}}$, to disambiguate between the centre and neighbour nodes in the ego-graph. These embeddings indicate to the model which input nodes will be used subsequently as the embedding representation of the ego-graph.

To encode the graph relations, we follow Mohammadshahi and Henderson (2020, 2021); Miculicich and Henderson (2022); Coman et al. (2024) in leveraging the intrinsic graph processing capabilities of the Transformer model by incorporating graph relations as relation embeddings input to the self-attention function. But unlike in that previous work, our relation embeddings are computed from the text associated with the relation, rather than coming from a fixed set of relations. For every pair of input nodes $ij$, the pre-softmax attention score $e_{ij} \in \mathbb{R}$ is computed from both the respective node embeddings $\boldsymbol{x}_i, \boldsymbol{x}_j \in \mathbb{R}^d$, and the embedding of the relation $r_{ij}$ between the $i$-th and $j$-th nodes, as:

$$e_{ij} = \frac{\boldsymbol{x}_i\boldsymbol{W}_Q \ \text{diag}(\mathbf{1} + \text{LN}(\boldsymbol{r}_{ij})\boldsymbol{W}_R) \ (\boldsymbol{x}_j\boldsymbol{W}_K)^{\top}}{\sqrt{d}}$$

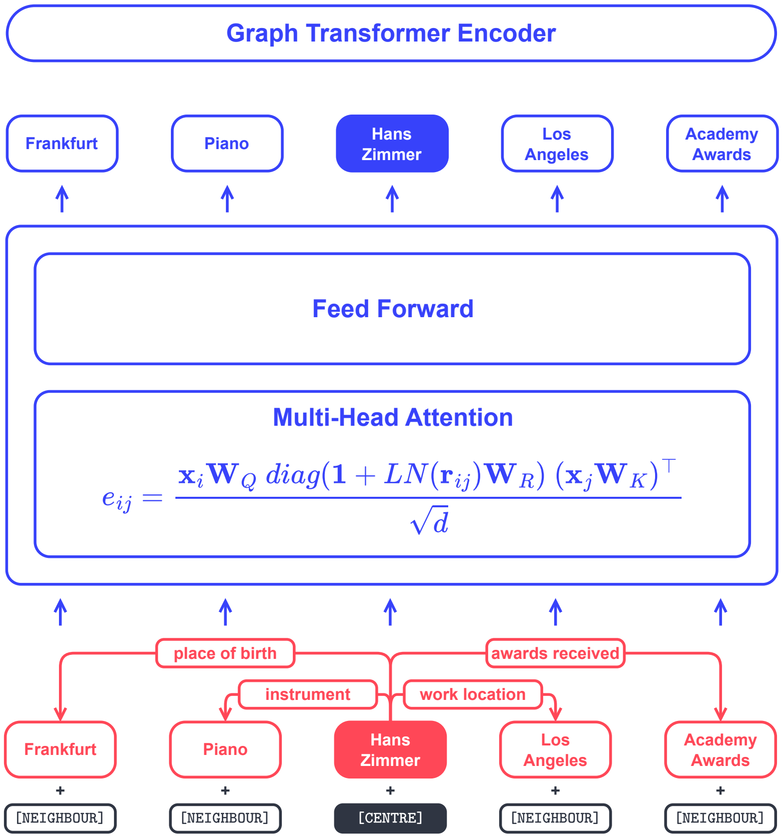



Figure 8.3: Graph Transformer Encoder. The subgraph $E(HansZimmer)$ of the entity *Hans Zimmer* is input together with the node segment embeddings that disambiguate between the centre node and neighbour nodes

where $\boldsymbol{W}_Q, \boldsymbol{W}_K \in \mathbb{R}^{d \times d}$ represent the query and key matrices, respectively, $\boldsymbol{r}_{ij}$ represents the relation embedding output by the TT module when it encodes the text associated with the relation between the $i$-th and $j$-th nodes, and $\boldsymbol{W}_R \in \mathbb{R}^{d \times d}$ is the relation matrix. Thus, $\mathrm{LN}(\boldsymbol{r}_{ij})\boldsymbol{W}_R$ is the embedding of the relation between $i$ and $j$, where LN stands for the $LayerNorm$ operation. Finally, $\mathrm{diag}(\mathbf{1} + \ldots)$ maps this vector into a diagonal matrix plus the identity matrix.

When encoding candidate entities and queries in the GT, it is crucial to ensure that no information regarding the target triple $(h, r, t)$ leaks into the $E(\hat{e})$, $E(\boldsymbol{h}_{TT})$, or $E(\boldsymbol{t}_{TT})$ ego-graphs. This precaution prevents the model from learning trivial solutions or biases from leaked information.

| **Dataset** | $\mathcal{R}$ | $\mathcal{E}_{train}$ | $\mathcal{T}_{train}$ | $E(e)_{train}$ | $\mathcal{E}_{val}$ | $\mathcal{T}_{val}$ | $E(e)_{val}$ | $\mathcal{E}_{test}$ | $\mathcal{T}_{test}$ | $E(e)_{test}$ |
|---|---|---|---|---|---|---|---|---|---|---|
| $\text{WN18RR}_{\text{IND}}$ | 11 | $32,755$ | $69,585$ | $2,12_{3,15}$ | $4,094$ | $11,381$ | $1,17_{1,33}$ | $4,456$ | $12,037$ | $1,18_{1,35}$ |
| $\text{FB15K-237}_{\text{IND}}$ | 237 | $11,633$ | $215,082$ | $18,49_{28,91}$ | $1,454$ | $42,164$ | $4,70_{10,63}$ | $2,416$ | $52,870$ | $4,97_{12,29}$ |
| $\text{Wikidata-5M}_{\text{IND}}$ | 822 | $4,579,609$ | $20,496,514$ | $4,48_{4,41}$ | $7,374$ | $6,699$ | $0,91_{0,78}$ | $7,475$ | $6,894$ | $0,92_{0,81}$ |

Table 8.1: $\text{WN18RR}_{\text{IND}}$, $\text{FB15K-237}_{\text{IND}}$, and $\text{Wikidata-5M}_{\text{IND}}$ datasets statistics. $\mathcal{E}$ represents the set of entities, $\mathcal{R}$ denotes the set of relation labels, $\mathcal{T}$ consists of the set of relation triples $(h, r, t) \in \mathcal{E} \times \mathcal{R} \times \mathcal{E}$, and $E(e)$ shows the mean and standard deviation ($\mu_\sigma$) of the number of neighbours in an entity's ego-graph.

## 8.8 Datasets and Setting

Daza et al. (2021) introduced the $\text{WN18RR}_{\text{IND}}$ and $\text{FB15k-237}_{\text{IND}}$ inductive variants of the well-known $\text{WN18RR}_{\text{TRA}}$ (Dettmers et al., 2018) and $\text{FB15k-237}_{\text{TRA}}$ (Toutanova and Chen, 2015) transductive KGs. The inductive setting simulates a *dynamic* scenario where new entities and triples are dynamically added to the graph. The training graph is constructed as $\mathcal{G}_{train} = \{(h, r, t) \in \mathcal{T}_{train} : h, t \in \mathcal{E}_{train}\}$. The validation and test graphs are constructed by incrementally adding entities and triples, such that $\mathcal{G}_{val} = \{(h, r, t) \in \mathcal{T}_{val} : h, t \in \mathcal{E}_{train} \cup \mathcal{E}_{val}\}$ and $\mathcal{G}_{test} = \{(h, r, t) \in \mathcal{T}_{test} : h, t \in \mathcal{E}_{train} \cup \mathcal{E}_{val} \cup \mathcal{E}_{test}\}$.

In contrast, the $\text{Wikidata5M}_{\text{IND}}$ KGs curated by Wang et al. (2021b) provide a *transfer* learning scenario in which the evaluation graphs are constructed such that the validation and test entity and triple sets, $\mathcal{E}_{val}$ and $\mathcal{E}_{test}$, and $\mathcal{T}_{val}$ and $\mathcal{T}_{test}$ are disjoint from the training entity and triple set $\mathcal{E}_{train}$ and $\mathcal{T}_{train}$. The validation and test graphs are constructed as $\mathcal{G}_{val} = \{(h, r, t) \in \mathcal{T}_{val} : h, t \in \mathcal{E}_{val}\}$ and $\mathcal{G}_{test} = \{(h, r, t) \in \mathcal{T}_{test} : h, t \in \mathcal{E}_{test}\}$. Because the graphs $\mathcal{G}_{val}$ and $\mathcal{G}_{test}$ do not include the entities from $\mathcal{G}_{train}$, they are much smaller graphs (see Table 8.1), which poses challenges for generalisation with graph-aware models, as will be discussed further below. We evaluate our model's ability to generalise to entirely new entities and triples in this setting.

We conduct our experiments in the above-mentioned settings of Daza et al. (2021) and Wang et al. (2021b), where textual information extraction is an integral part. Our method is directly comparable to DKRL (Xie et al., 2016), BLP (Daza et al., 2021), KEPLER (Wang et al., 2021b), StAR (Wang et al., 2021a), and the state-of-the-art method, StATIK (Markowitz et al., 2022) which employ the same textual encoder, structural objective, and loss function. Similar to StATIK, our work aims to jointly model the text and the structure of knowledge graphs, including extracting information about KG links from the text. This sets us apart from the setting of Teru et al. (2020), which uses different KGs splits and is employed in GraIL (Teru et al., 2020), NBFNet (Zhu et al., 2021), and NodePiece (Galkin et al., 2021), that solely focus on using the structure of the graph without incorporating any textual information extraction component.

## 8.9 Experiments and Results

### 8.9.1 Controlled Experimental Setup

When comparing the performance of different models on link prediction tasks, it is crucial to establish a fair and consistent baseline. Our experiments in Table 8.2 highlight the importance of carefully setting this baseline, as various factors can greatly influence the results.

| | MRR | |
|---|---|---|
| **Model** | **WN18RR$_{\text{IND}}$** | **FB15k-237$_{\text{IND}}$** |
| BLP$_{\text{BERT}_{\text{BASE}}}$ | 0.285 | 0.195 |
| BLP$^{\bullet}_{\text{BERT}_{\text{BASE}}}$ | 0.280 | 0.205 |
| + inductive relations | 0.281 | 0.219 |
| + negatives batch tying | 0.300 | 0.221 |
| + bigger embedding size | 0.339 | 0.254 |
| + bigger batch size | 0.366 | 0.260 |
| + better sampling method | 0.373 | 0.266 |
| FnF-T$_{\text{BERT}_{\text{BASE}}}$ (ours) | **0.373** | **0.266** |

Table 8.2: WN18RR$_{\text{IND}}$ and FB15k-237$_{\text{IND}}$ test set results with cumulative additions over the baseline model BLP$_{\text{BERT}_{\text{BASE}}}$ (Daza et al., 2021) that lead to our improved baseline model FnF-T$_{\text{BERT}_{\text{BASE}}}$. BLP$^{\bullet}_{\text{BERT}_{\text{BASE}}}$ indicates our reimplementation of BLP$_{\text{BERT}_{\text{BASE}}}$.

Specifically, we demonstrate that the computational budget, which determines training hyper-parameters, can have a substantial impact on model performance. Starting with the baseline model BLP$_{\text{BERT}_{\text{BASE}}}$ (Daza et al., 2021), we introduce improvements such as using inductive relations, increasing the number of negative triples to match the batch size (negatives batch tying), increasing the embedding dimension from 128 to 768, doubling the batch size from 64 to 128, and modifying the negative sampling strategy to two-sided reflexive, where both head and tail entities are considered as potential negatives. These cumulative improvements lead to the development of a new text-only model baseline, FnF-T$_{\text{BERT}_{\text{BASE}}}$, which shows substantial improvements on both the WN18RR$_{\text{IND}}$ and FB15k-237$_{\text{IND}}$ datasets.

To ensure a fair comparison, we fixed our computational budget to a constant in this chapter, using a consumer-grade GPU (NVIDIA RTX3090 24GB). This allows for a consistent and reproducible experimental setup, enabling a more accurate assessment of performance. For more details, see Appendix D.

### 8.9.2 Inductive Link Prediction Results

As shown in the top half of Table 8.3, for both the $\text{WN18RR}_{\text{IND}}$ and the $\text{FB15k-237}_{\text{IND}}$ datasets, our inductive relation embeddings and the enhanced controlled experimental setup result in improved text-only models. These models rely heavily on having powerful text encoders, as shown by the degradation in performance when using smaller versions of BERT as the text encoder.

| Model | $\text{WN18RR}_{\text{IND}}$ | | | | $\text{FB15k-237}_{\text{IND}}$ | | | |
|---|---|---|---|---|---|---|---|---|
| | **MRR** | **H@1** | **H@3** | **H@10** | **MRR** | **H@1** | **H@3** | **H@10** |
| *Text-Only Models* | | | | | | | | |
| $\text{DKRL}^{*}_{\text{BERT}_{\text{BASE}}}$ | 0.139 | 0.048 | 0.169 | 0.320 | 0.144 | 0.084 | 0.151 | 0.263 |
| $\text{BOW}^{*}_{\text{BERT}_{\text{BASE}}}$ | 0.180 | 0.045 | 0.244 | 0.450 | 0.173 | 0.103 | 0.184 | 0.316 |
| $\text{BLP}^{*}_{\text{BERT}_{\text{BASE}}}$ | 0.285 | 0.135 | 0.361 | 0.580 | 0.195 | 0.113 | 0.213 | 0.363 |
| $\text{FnF-T}_{\text{BERT}_{\text{BASE}}}$ | **0.373** | **0.238** | **0.442** | **0.647** | **0.266** | **0.174** | **0.297** | **0.453** |
| $\text{FnF-T}_{\text{BERT}_{\text{MEDIUM}}}$ | 0.342 | 0.213 | 0.405 | 0.603 | 0.253 | 0.164 | 0.281 | 0.431 |
| $\text{FnF-T}_{\text{BERT}_{\text{SMALL}}}$ | 0.320 | 0.197 | 0.379 | 0.572 | 0.239 | 0.152 | 0.265 | 0.411 |
| $\text{FnF-T}_{\text{BERT}_{\text{MINI}}}$ | 0.268 | 0.156 | 0.318 | 0.498 | 0.204 | 0.128 | 0.223 | 0.354 |
| $\text{FnF-T}_{\text{BERT}_{\text{TINY}}}$ | 0.193 | 0.098 | 0.230 | 0.385 | 0.164 | 0.100 | 0.176 | 0.289 |
| *Structure-Informed Models* | | | | | | | | |
| $\text{StAR}^{\star}_{\text{BERT}_{\text{BASE}}}$ | 0.321 | 0.192 | 0.381 | 0.576 | 0.163 | 0.092 | 0.176 | 0.309 |
| $\text{StATIK}^{\star}_{\text{BERT}_{\text{BASE}}}$ | 0.516 | 0.425 | 0.558 | 0.690 | 0.224 | 0.143 | 0.248 | 0.381 |
| $\text{FnF-TG}_{\text{BERT}_{\text{BASE}}}$ | 0.732 | 0.652 | 0.785 | **0.875** | 0.316 | 0.214 | 0.350 | **0.524** |
| $\text{FnF-TG}_{\text{BERT}_{\text{MEDIUM}}}$ | **0.737** | **0.661** | **0.789** | 0.873 | 0.314 | 0.214 | 0.353 | 0.515 |
| $\text{FnF-TG}_{\text{BERT}_{\text{SMALL}}}$ | 0.727 | 0.648 | 0.781 | 0.867 | **0.316** | **0.216** | **0.354** | 0.518 |
| $\text{FnF-TG}_{\text{BERT}_{\text{MINI}}}$ | 0.713 | 0.632 | 0.768 | 0.857 | 0.302 | 0.204 | 0.337 | 0.502 |
| $\text{FnF-TG}_{\text{BERT}_{\text{TINY}}}$ | 0.638 | 0.543 | 0.700 | 0.808 | 0.288 | 0.195 | 0.318 | 0.475 |

Table 8.3: $\text{WN18RR}_{\text{IND}}$ and $\text{FB15k-237}_{\text{IND}}$ test set results. *Daza et al. (2021); ⋆Markowitz et al. (2022).

The addition of our graph encoder to the model (bottom half of Table 8.3) leads to a substantial increase in link prediction accuracy over the text-only model. We also see that our TG (text-graph) encoder results in substantially better accuracy than the previous state-of-the-art model, StATIK. Interestingly, this more effective use of graph context also has a big impact on the model's dependence on powerful text encoders. Reducing the size of the text encoder ($\text{BERT}_{\text{BASE}} > \text{BERT}_{\text{MEDIUM}} > \text{BERT}_{\text{SMALL}} > \text{BERT}_{\text{MINI}} > \text{BERT}_{\text{TINY}}$) does result in some degradation of accuracy, but the differences are much smaller than in the text-only case. Even with a $\text{BERT}_{\text{TINY}}$ text encoder, the graph-aware model performs better than the text-only model with a $\text{BERT}_{\text{BASE}}$ encoder. This shows that the inductive bias of explicit graph relations can be an effective alternative to extracting the same information from text with a powerful text encoder.

This pattern of results is repeated in the transfer case, shown in Table 8.4. Here, the training set is much larger, but the graph in the test set is relatively small with each entity having fewer neighbours (see Table 8.1). This reduces the advantage gained from adding an effective graph encoder and the margin of our models' improvement over the text-only models, and over the previous state-of-the-art model, StATIK.[1] But we still see the same pattern where the size of the text encoder has less effect on accuracy for the graph-aware model.

| **Model** | **MRR** | **H@1** | **H@3** | **H@10** |
|---|---|---|---|---|
| | *Text-Only Models* | | | |
| $\text{KEPLER}^{\diamond}_{\text{BERT}_{\text{BASE}}}$ | 0.402 | 0.222 | 0.514 | 0.730 |
| $\text{BLP}^{*}_{\text{BERT}_{\text{BASE}}}$ | 0.478 | 0.241 | 0.660 | 0.871 |
| $\text{FnF-T}_{\text{BERT}_{\text{BASE}}}$ | **0.597** | **0.427** | **0.722** | **0.896** |
| $\text{FnF-T}_{\text{BERT}_{\text{MEDIUM}}}$ | 0.588 | 0.418 | 0.712 | 0.890 |
| $\text{FnF-T}_{\text{BERT}_{\text{SMALL}}}$ | 0.588 | 0.417 | 0.714 | 0.889 |
| $\text{FnF-T}_{\text{BERT}_{\text{MINI}}}$ | 0.562 | 0.391 | 0.683 | 0.870 |
| $\text{FnF-T}_{\text{BERT}_{\text{TINY}}}$ | 0.526 | 0.348 | 0.649 | 0.849 |
| | *Structure-Informed Models* | | | |
| $\text{StATIK}^{\star}_{\text{BERT}_{\text{BASE}}}$ | 0.770 | **0.765** | 0.771 | 0.779 |
| $\text{FnF-TG}_{\text{BERT}_{\text{BASE}}}$ | **0.799** | 0.741 | **0.833** | **0.911** |
| $\text{FnF-TG}_{\text{BERT}_{\text{MEDIUM}}}$ | 0.785 | 0.727 | 0.817 | 0.900 |
| $\text{FnF-TG}_{\text{BERT}_{\text{SMALL}}}$ | 0.781 | 0.721 | 0.816 | 0.898 |
| $\text{FnF-TG}_{\text{BERT}_{\text{MINI}}}$ | 0.779 | 0.719 | 0.814 | 0.894 |
| $\text{FnF-TG}_{\text{BERT}_{\text{TINY}}}$ | 0.761 | 0.697 | 0.799 | 0.883 |

Table 8.4: $\text{Wikidata5M}_{\text{IND}}$ test set results. $^{\diamond}$Wang et al. (2021b); $^{*}$Daza et al. (2021); $^{\star}$Markowitz et al. (2022).

[1] StATIK has a surprisingly high H@1 score, almost identical to its H@3, H@10 and MRR scores. It is not clear why this is the case. Regardless, our model's MRR, H@3, and H@10 scores are better than StATIK. MRR is the primary evaluation measure since it summarises the entire ranking.

### 8.9.3 Ablation Study

Table 8.5 presents results from our ablation studies, showing the impact of removing various design features from our graph-aware model on its accuracy.

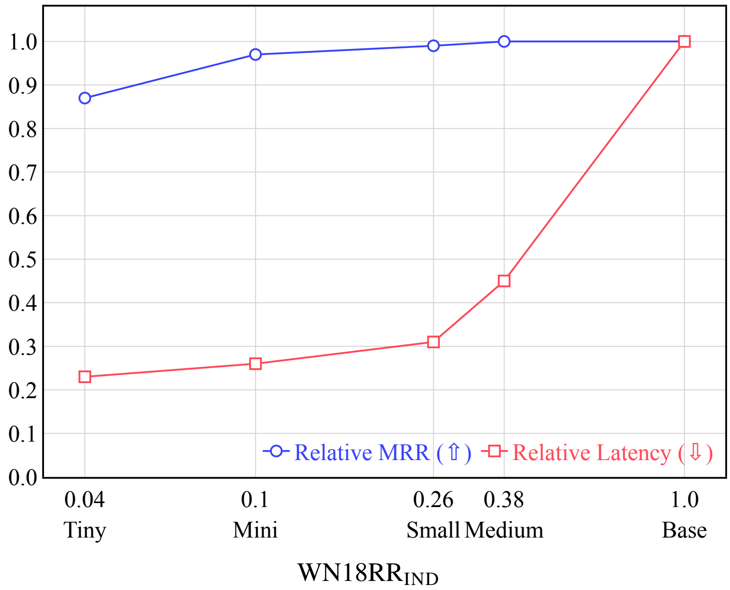
| | **MRR** | |
|---|---|---|
| **Model** | **WN18RR$_{\text{IND}}$** | **FB15k-237$_{\text{IND}}$** |
| FnF-TG$_{\text{BERT}_{\text{MEDIUM \| SMALL}}}$ | 0.737 | 0.316 |
| $-\ r_{ij}$ | 0.733 | 0.306 |
| $-\ s_{\texttt{[CENTRE]}}, s_{\texttt{[NEIGHBOUR]}}$ | 0.677 | 0.298 |
| $-\ E(\boldsymbol{h}_{TT}), E(\boldsymbol{t}_{TT})$ | 0.480 | 0.251 |
| $-\ [h\|\|r]_{KG}, [t\|\|r^{-1}]_{KG}$ | 0.342 | 0.239 |

Table 8.5: Ablation studies on the WN18RR$_{\text{IND}}$ and FB15k-237$_{\text{IND}}$ test sets using the top FnF-TG model. Each row indicates the performance after cumulatively removing a specific feature.

Removing the $r_{ij}$ relation embeddings in the pre-softmax attention score leads to a decline in model performance, with a more substantial drop observed on the FB15K-237$_{\text{IND}}$ dataset compared to the WN18RR$_{\text{IND}}$ dataset. Note that with this modification the model still knows that there is some relation to the neighbours, but does not know its label. Removing the learnable segment embeddings $s_{\texttt{[CENTRE]}}$ and $s_{\texttt{[NEIGHBOUR]}}$ then removes this unlabelled graph structure, which considerably impacts the model's performance. Eliminating the ego-graph neighbours altogether results in an even more substantial performance drop. Despite this, the model remains competitive as a text-only model compared to the BLP baseline, owing to its ability to leverage relation conditioning features to represent candidate relations. Finally, removing the relation conditioning $[h||r]_{KG}$ and $[t||r^{-1}]_{KG}$, results in a further notable decrease in performance. Without relation conditioning, the model loses its ability to anticipate the query relation, severely impacting its accuracy.

### 8.9.4 Efficient Text Encoders

Being able to reduce the size of the text encoder with minimal degradation in accuracy is important because the text encoder is a substantial part of the training cost. In Figure 8.4 we plot the relative reduction in accuracy against the relative reduction in training time as we reduce the size of the text encoder, for the $\mathrm{WN18RR}_{\mathrm{IND}}$ and the $\mathrm{FB15k\text{-}237}_{\mathrm{IND}}$ datasets. We see that reducing the encoder size by a factor of four reduces the training time by a factor of three for $\mathrm{WN18RR}_{\mathrm{IND}}$ (and nearly two for $\mathrm{FB15k\text{-}237}_{\mathrm{IND}}$) with very little reduction in accuracy.



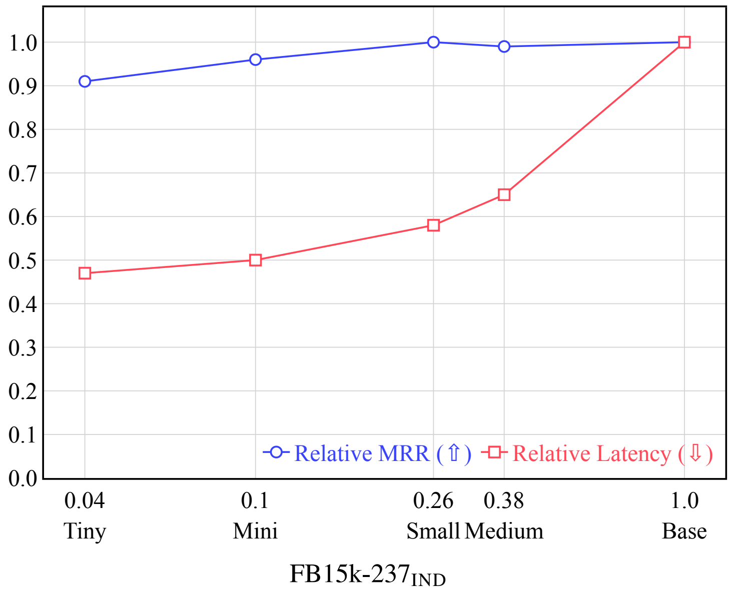




Figure 8.4: Accuracy and training time plotted as a function of text encoder size, relative to the largest text encoder with the highest accuracy, shown as (1.0,1.0).

### 8.9.5 Fully Inductive Link Prediction Results

The experimental setting of Daza et al. (2021) and Wang et al. (2021b) do not support evaluation on unseen relations. One distinctive advantage of our model is that it is not restricted to a fixed set of relation labels learned during training. Although we do show that conditioning on relation texts improves accuracy even on seen relations (see Table 8.2), it is important to evaluate our model in a fully inductive setting, where relations are also unseen, in addition to entities.

To this end, we propose a new experimental setting for a fully inductive relations (FIR) evaluation, by converting the $\text{WN18RR}_{\text{IND}}$, $\text{FB15k-237}_{\text{IND}}$, and $\text{Wikidata-5M}_{\text{IND}}$ evaluations to their respective FIR versions. More specifically, we focus on the long tail of relations and remove the least frequent relation labels until 10% of edges have been removed from the training graph $\mathcal{G}_{train}$. We then train a new set of models on this new version so they have not seen the removed relation labels, and evaluate them on both the full set of test relations (IND) and specifically on the relations for the unseen labels (IND $\setminus$ FIR). Given that all the previous models (DKRL, BLP, KEPLER, StAR, and StATIK) are inductive in entities but transductive in relations, none of them can make informed predictions in this setting, so we compare to a random baseline which computes the expected MRR for random rankings of candidate entities $\hat{\mathcal{E}}$ as follows:

$$\mathbb{E}[\text{MRR}_{\text{random}}] = \frac{1}{|\hat{\mathcal{E}}|}\sum_{i=1}^{|\hat{\mathcal{E}}|}\frac{1}{i}$$

The results in Table 8.6 show the performance of our model on this new setting. Our approach shows promising results as it outperforms the random baseline by a significant margin. However, the performance drops considerably when training on FIR and evaluating on IND $\setminus$ FIR, indicating that the model struggles with unseen relations. Notably, the results on the Wikidata-5M dataset are considerably better than those obtained on the WN18RR and FB15k-237 datasets, probably due to having relations with more descriptive texts. These results highlight the need for further research in developing models that can effectively generalise to unseen relations.

| | | WN18RR | | | | FB15k-237 | | | | Wikidata5M | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Training** | **Evaluation** | **MRR** | **H@1** | **H@3** | **H@10** | **MRR** | **H@1** | **H@3** | **H@10** | **MRR** | **H@1** | **H@3** | **H@10** |
| | | | | | | $\text{FnF-T}_{\text{BERT}_{\text{TINY}}}$ | | | | | | | |
| IND | IND | 0.193 | 0.098 | 0.230 | 0.385 | 0.164 | 0.100 | 0.176 | 0.289 | 0.526 | 0.348 | 0.649 | 0.849 |
| FIR | IND | 0.169 | 0.080 | 0.198 | 0.346 | 0.143 | 0.087 | 0.153 | 0.253 | 0.478 | 0.306 | 0.591 | 0.793 |
| IND | IND $\setminus$ FIR | 0.307 | 0.210 | 0.353 | 0.495 | 0.171 | 0.102 | 0.185 | 0.308 | 0.594 | 0.451 | 0.697 | 0.849 |
| FIR | IND $\setminus$ FIR | 0.064 | 0.012 | 0.082 | 0.159 | 0.024 | 0.007 | 0.026 | 0.052 | 0.219 | 0.054 | 0.334 | 0.499 |
| | | | | | | $\text{FnF-TG}_{\text{BERT}_{\text{TINY}}}$ | | | | | | | |
| IND | IND | 0.638 | 0.543 | 0.700 | 0.808 | 0.288 | 0.195 | 0.318 | 0.475 | 0.761 | 0.697 | 0.799 | 0.883 |
| FIR | IND | 0.573 | 0.480 | 0.629 | 0.738 | 0.249 | 0.165 | 0.274 | 0.418 | 0.711 | 0.644 | 0.749 | 0.837 |
| IND | IND $\setminus$ FIR | 0.585 | 0.483 | 0.652 | 0.769 | 0.282 | 0.176 | 0.311 | 0.514 | 0.726 | 0.646 | 0.781 | 0.867 |
| FIR | IND $\setminus$ FIR | 0.108 | 0.028 | 0.147 | 0.242 | 0.050 | 0.023 | 0.051 | 0.099 | 0.401 | 0.287 | 0.480 | 0.589 |
| | | | | | | Random baseline | | | | | | | |
| – | – | 0.0003 | – | – | – | 0.0007 | – | – | – | 0.0013 | – | – | – |

Table 8.6: Fully inductive link prediction results.

## 8.10 Conclusion

We presented a new Transformers-based approach to link prediction in text-attributed knowledge graphs that combines textual descriptions and graph structure in a fully inductive setting. Our Fast-and-Frugal Text-Graph (*FnF-TG*) Transformers outperform previous state-of-the-art models on three popular datasets, showcasing the importance of capturing rich structured information about entities and their relations. Our approach achieves superior performance while maintaining efficiency and scalability, making it a promising solution for large-scale knowledge graph applications. Moreover, our ablation studies provide insights into the key factors contributing to its effectiveness, demonstrating the value of each component in our model. Additionally, we proposed a new evaluation setting for fully inductive link prediction, where relations are also inductive, and demonstrated the potential of our approach in this setting.

These findings also relate to the example of Larry and Palmerston in Figure 1.1. In that case, identifying the rivalry between the two cats requires integrating structured information such as their institutional roles, overlapping tenures, and reports of confrontation, which may appear in separate spans of unstructured text or as distinct relations in a graph. Earlier chapters examined how to improve the retrieval of such evidence (Chapter 4), align generated responses with it (Chapter 5), and aggregate it using graph-structured inputs (Chapter 7). This chapter builds on these foundations by focusing on how textual descriptions and relational structure can be jointly encoded in a fully inductive setting. The proposed *FnF-TG* model is designed to support the prediction of potential new links, such as the rivalry between Larry and Palmerston, even when both the entities and the relation have not been seen during training.

## Limitations

While our approach yields promising results, there remains room for further improvement.

One area for exploration is optimising the scalability of our *Graph Transformer Encoder* component (see Figure 8.2), which currently requires computing fully quadratic attention over the entire ego-graph of a given entity. In fact it could still require considerable resources if the number of nodes in the ego-graph is scaled to the order of thousands, hundreds of thousands, or even millions.

Our work demonstrates that effectively capturing even local neighbourhood information is both non-trivial and under-explored and that it can significantly enhance performance. Indeed, our simplification to a 1-hop neighbourhood (ego-graph) was a careful decision to balance effectiveness and complexity. This approach not only allows for a fair comparison with the current state-of-the-art method, StATIK (Markowitz et al., 2022), but also mitigates the exponential increase in computational complexity (see Appendix D) associated with larger neighbourhoods. While this predefined 1-hop neighbourhood provides a solid starting point, there is room to explore better alternatives. For instance, investigating multi-hop neighbourhoods or adaptive neighbourhood definitions could uncover more nuanced insights from the graph structure, potentially leading to even better results.

By building upon our framework, future work could refine these aspects, ultimately enhancing the effectiveness and versatility of our approach.

## Ethics Statement

We do not anticipate any ethical concerns related to our work, as it primarily presents an alternative approach to a previously proposed method. Our main contribution lies in introducing a new approach for link prediction. In our experiments, we use the same datasets and pretrained models as previous research, all of which are publicly available. However, it is important to acknowledge that these datasets and models may still require further examination for potential fairness issues and the knowledge they encapsulate.

## Acknowledgements

We extend our special gratitude to the Swiss National Science Foundation (SNSF) and Research Foundation – Flanders (FWO) for funding this work under grants 200021E_189458 and G094020N.

# 9 Conclusions and Future Work

This thesis examines how to improve the effectiveness of information-seeking systems that make use of both unstructured and structured information. It focuses on two complementary lines of investigation: Retrieval-Augmented Generation (RAG) and Text-Graph Encoders. The first line of investigation addresses limitations in retriever-reader pipelines for open-domain conversational question answering, while also exploring how reward models can be extended to evaluate how well generated responses reflect the retrieved evidence. The second line of investigation introduces methods for incorporating relational structure into attention-based models and enabling them to operate over previously unseen entities and relations in an inductive manner. This thesis is not a conclusion but a foundation for developing LLM-based information-seeking systems that can ground their generation in retrieved evidence, whether unstructured or structured, and access both types of information seamlessly via the attention mechanism.

In Chapter 1, we introduce the motivation for this thesis by examining how structured and unstructured information sources contribute to the effectiveness of modern information-seeking systems. We formulate four central research questions, focusing on the limitations of retrieval-based conversational question answering pipelines, the design of reward models tailored to the RAG setting, the integration of structured relational information into attention mechanisms, and the ability of models to inductively operate over previously unseen entities and relations using both textual and structured inputs.

Chapter 3 introduces the core components of RAG: retrieval, augmentation, and generation, and examines how these interact to enable systems to incorporate external evidence dynamically at inference time.

In Chapter 4, we identify key limitations in retriever-reader pipelines for open-domain conversational question answering, particularly highlighting the retriever's limited coverage and the reader's sensitivity to noisy inputs. To address these issues, we introduce *R3FINE*, a strategy combining a lightweight reranker with targeted fine-tuning of the retriever and reader. Specifically, the retriever is fine-tuned to surface relevant evidence more effectively, while the reader is fine-tuned to better handle this evidence during generation.

Chapter 5 tackles the challenge of aligning generated responses with retrieved evidence by introducing *RAGferee*, a method for constructing contextual reward models specifically extended to evaluate responses in the RAG setting. Unlike reward models trained on general preference data, *RAGferee* emphasises RAG-specific dimensions, including faithfulness, completeness, conciseness, relevance to the user query, and appropriate refusals, thus encouraging responses that closely reflect retrieved contexts and user intent.

Chapter 6 provides essential background on the self-attention mechanism and examines how it can be extended to incorporate structural encoding, setting the stage for the text-graph models developed in Chapter 7 and Chapter 8.

In Chapter 7, we explore how explicit relational structure can guide the attention mechanism in pooling information scattered across unstructured text. The resulting method, *GADePo*, enables models to effectively combine structured and unstructured information through attention, providing a unified representational mechanism.

Finally, Chapter 8 introduces *FnF-TG*, a text-graph model capable of inductively operating over previously unseen entities and relations by using textual descriptions alongside structured context to construct embeddings. A key insight is that explicit relational structure plays a crucial role in inductive settings, reducing the reliance on text encoders to recover this structure implicitly from unstructured input.

Collectively, these chapters contribute complementary methods that lay the foundation for future LLM-based information-seeking systems capable of seamlessly integrating structured and unstructured information within a unified attention-based framework. Looking ahead, this framework can evolve to access any form of information, independent of its modality, through the unified lens of the attention mechanism. We are already beginning to see early signs of this in emerging any-to-any models capable of ingesting and generating across multiple input and output formats, such as text, image, audio, and video (Lu et al., 2022, 2023; Xu et al., 2025b). These developments suggest a future in which information-seeking systems become increasingly general-purpose, able to reason across modalities and use cases in a cohesive and adaptive way. This enables applications in high-stakes areas, with scientific discovery being a clear example where grounding in diverse evidence is essential. To realise this vision, it is crucial to advance the transparency and controllability of these models, particularly in how they weigh retrieved evidence and integrate information from multiple modalities. Ultimately, the goal is to move past isolated enhancements and instead build robust and trustworthy information agents that can support users across tasks, domains, and modalities in a coherent and personalised manner.

# A Appendix for Chapter 4: *R3FINE*

## OR-QUAC conversion to TOPIOCQA's format and models

To ensure compatibility between the OR-QUAC dataset and the models used in TOPIOCQA, we applied the following modifications:

- All answers of type CANNOTANSWER and NOTRECOVERED were mapped to UNANSWERABLE.
- For the DPR *Conversation Encoder*, we adopted the ALLHISTORY representation of the conversational context from TOPIOCQA.
- For the DPR *Passage Encoder*, each passage title was simplified from `"passage_page_title [SEP] passage_page_subtitle"` to `"passage_page_title"`. This adjustment was necessary as OR-QUAC does not provide section-level information for passages within a page, unlike TOPIOCQA.
- For the FiD component, the input format for passages was changed from `"title: sub-title: context:"` to `"title: context:"`, consistent with the rationale outlined above.

Each passage from the OR-QUAC Wikipedia knowledge source was mapped to its corresponding embedding via the TOPIOCQA's *Passage Encoder* component. We then performed the same retrieval step as the one done for TOPIOCQA. We exploited TOPIOCQA's DPR module for both datasets as the retrieval phase is very similar between the two. However, given that, unlike TOPIOCQA, OR-QUAC is of extractive type, we had to train the FiD module from scratch. We followed the same training configuration as the one used for TOPIOCQA.

## Retriever results

Table A.1 and Table A.2 report the retrieval coverage on the train and test sets, respectively, before and after the introduction of the *Semantic Reranker* (w/ SR and w/o SR) when a larger number of passages (50 vs. 1000) is considered.

| | TopiOCQA | | OR-QuAC | |
|---|---|---|---|---|
| **top-$k$** | **w/o SR** | **w/ SR** | **w/o SR** | **w/ SR** |
| 1 | 31.54 | 98.25 | 31.69 | 77.31 |
| 5 | 66.99 | 99.71 | 59.08 | 89.10 |
| 10 | 78.29 | 99.72 | 66.31 | 89.36 |
| 20 | 86.78 | 99.72 | 71.77 | 89.40 |
| 50 | 93.55 | 99.72 | 77.35 | 89.40 |
| 500 | 99.36 | 99.72 | 87.15 | 89.41 |
| 1000 | 99.72 | 99.72 | 89.41 | 89.41 |

Table A.1: Training set retrieval coverage before and after the introduction of the *Semantic Reranker* (w/o SR and w/ SR) when a larger number of passages (50 vs. 1000) is considered. For both datasets, we report the presence of the gold passage within the top-$k$ limit.

| | OR-QuAC | |
|---|---|---|
| **top-$k$** | **w/o SR** | **w/ SR** |
| 1 | 26.82 | 48.75 |
| 5 | 46.29 | 64.71 |
| 10 | 51.50 | 66.47 |
| 20 | 55.63 | 67.31 |
| 50 | 59.85 | 68.10 |
| 500 | 67.87 | 69.48 |
| 1000 | 69.86 | 69.86 |

Table A.2: OR-QuAC test set retrieval coverage before and after the introduction of the *Semantic Reranker* (w/o SR and w/ SR) when a larger number of passages (50 vs. 1000) is considered. We report the presence of the gold answer within the top-$k$ limit.

## Reader results

Table A.3, Table A.4, and Table A.5 show the impact the introduction of the *Semantic Reranker* has on the FiD reader. The input to the FiD reader are either passages returned by the initial DPR retriever (w/o SR) or passages returned by the *Semantic Reranker* (w/ SR).

| TopiOCQA | | | | |
|---|---|---|---|---|
| | **w/o SR** | | **w/ SR** | |
| **top-$k$** | **EM** | **F1** | **EM** | **F1** |
| 1 | 19.3 | 37.6 | 28.1 | 50.4 |
| 5 | 27.0 | 49.6 | 32.2 | 56.5 |
| 10 | 29.8 | 52.4 | 33.2 | 57.3 |
| 20 | 31.3 | 54.0 | 33.4 | 56.5 |
| 50 | 33.0 | 55.1 | 33.9 | 56.2 |

Table A.3: FiD reader performance (Exact Match and F1 scores) on the TopiOCQA development set before and after the introduction of the *Semantic Reranker* (w/o SR and w/ SR).

| OR-QuAC | | | | |
| --- | --- | --- | --- | --- |
| | **w/o SR** | | **w/ SR** | |
| **top-$k$** | **EM** | **F1** | **EM** | **F1** |
| 1 | 13.2 | 22.2 | 12.2 | 25.9 |
| 5 | 16.4 | 26.3 | 16.4 | 28.7 |
| 10 | 18.0 | 27.6 | 18.0 | 29.4 |
| 20 | 19.2 | 28.2 | 18.9 | 29.2 |
| 50 | 19.6 | 27.7 | 19.3 | 27.7 |

Table A.4: FiD reader performance (Exact Match and F1 scores) on the OR-QuAC development set before and after the introduction of the *Semantic Reranker* (w/o SR and w/ SR).

| OR-QuAC | | | | |
| --- | --- | --- | --- | --- |
| | **w/o SR** | | **w/ SR** | |
| **top-$k$** | **EM** | **F1** | **EM** | **F1** |
| 1 | 14.7 | 23.0 | 13.9 | 25.7 |
| 5 | 17.8 | 27.1 | 17.9 | 29.5 |
| 10 | 19.0 | 28.2 | 19.4 | 30.0 |
| 20 | 20.3 | 29.1 | 20.8 | 30.5 |
| 50 | 22.0 | 29.9 | 22.1 | 30.2 |

Table A.5: FiD reader performance (Exact Match and F1 scores) on the OR-QuAC test set before and after the introduction of the *Semantic Reranker* (w/o SR and w/ SR).

## Further reader study

To better understand the impact that the introduction of the *Semantic Reranker* has on the FiD reader, Table A.6, Table A.7, and Table A.8 show the results obtained after taking a non fine-tuned FiD and training it on the top-10 passages returned by the initial DPR retriever and on the top-10 passages returned by the *Semantic Reranker*. On both datasets, we followed the same training configuration as the one used for TopiOCQA.

| TopiOCQA | | | | |
| --- | --- | --- | --- | --- |
| | **w/o SR + FT** | | **w/ SR + FT** | |
| **top-$k$** | **EM** | **F1** | **EM** | **F1** |
| 1 | 19.2 | 37.8 | 29.5 | 50.9 |
| 5 | 27.8 | 49.9 | 34.0 | 56.9 |
| 10 | 30.4 | 52.0 | 34.3 | 57.0 |

Table A.6: FiD reader performance (Exact Match and F1 scores) on the TopiOCQA development set after training a non fine-tuned FiD on the top-10 passages returned by either the initial DPR retriever (w/o SR + FT) or the *Semantic Reranker* (w/ SR + FT).

| OR-QuAC | | | | |
|---|---|---|---|---|
| | **w/o SR + FT** | | **w/ SR + FT** | |
| **top-$k$** | **EM** | **F1** | **EM** | **F1** |
| 1 | 14.2 | 23.1 | 13.9 | 27.6 |
| 5 | 18.0 | 27.7 | 18.1 | 30.8 |
| 10 | 19.2 | 28.9 | 19.3 | 31.2 |

Table A.7: FiD reader performance (Exact Match and F1 scores) on the OR-QuAC development set after training a non fine-tuned FiD on the top-10 passages returned by either the initial DPR retriever (w/o SR + FT) or the *Semantic Reranker* (w/ SR + FT).

| OR-QuAC | | | | |
|---|---|---|---|---|
| | **w/o SR + FT** | | **w/ SR + FT** | |
| **top-$k$** | **EM** | **F1** | **EM** | **F1** |
| 1 | 15.5 | 23.9 | 15.8 | 28.3 |
| 5 | 19.6 | 29.1 | 19.4 | 31.7 |
| 10 | 20.9 | 29.7 | 20.6 | 31.9 |

Table A.8: FiD reader performance (Exact Match and F1 scores) on the OR-QuAC test set after training a non fine-tuned FiD on the top-10 passages returned by either the initial DPR retriever (w/o SR + FT) or the *Semantic Reranker* (w/ SR + FT).

Table A.9, Table A.10, and Table A.11 show instead the results obtained after taking an already fine-tuned FiD reader and further fine-tuning it on the top-10 passages returned by the initial DPR retriever and on the top-10 passages returned by the *Semantic Reranker*. On both datasets, the amount of fine-tuning steps is equal to the one used for training the already fine-tuned FiD reader.

| TopiOCQA | | | | |
|---|---|---|---|---|
| | **w/o SR + FT** | | **w/ SR + FT** | |
| **top-$k$** | **EM** | **F1** | **EM** | **F1** |
| 1 | 21.5 | 39.3 | 30.7 | 52.4 |
| 5 | 30.4 | 52.3 | 36.1 | 59.4 |
| 10 | 32.8 | 54.6 | 35.8 | 59.0 |

Table A.9: FiD reader performance (Exact Match and F1 scores) on the TopiOCQA development set after further fine-tuning an already fine-tuned FiD on top-10 passages returned by either the initial DPR retriever (w/o SR) or the *Semantic Reranker* (w/ SR).

| OR-QuAC | | | | |
|---|---|---|---|---|
| | w/o SR + FT | | w/ SR + FT | |
| **top-$k$** | **EM** | **F1** | **EM** | **F1** |
| 1 | 15.2 | 23.8 | 15.4 | 29.0 |
| 5 | 18.9 | 28.4 | 18.4 | 31.2 |
| 10 | 19.9 | 29.4 | 20.1 | 32.0 |

Table A.10: FiD reader performance (Exact Match and F1 scores) on the OR-QuAC development set after further fine-tuning an already fine-tuned FiD on top-10 passages returned by either the initial DPR retriever (w/o SR) or the *Semantic Reranker* (w/ SR).

| OR-QuAC | | | | |
|---|---|---|---|---|
| | w/o SR + FT | | w/ SR + FT | |
| **top-$k$** | **EM** | **F1** | **EM** | **F1** |
| 1 | 16.7 | 24.9 | 16.6 | 28.9 |
| 5 | 20.0 | 29.2 | 20.1 | 32.2 |
| 10 | 21.4 | 30.0 | 21.6 | 32.9 |

Table A.11: FiD reader performance (Exact Match and F1 scores) on the OR-QuAC test set after further fine-tuning an already fine-tuned FiD on top-10 passages returned by either the initial DPR retriever (w/o SR) or the *Semantic Reranker* (w/ SR).

## Reranker training and ablation study

We tried different configurations of the *Semantic Reranker* to find the most efficient and effective one. In addition to the decision of whether to fine-une the DPR's *ConversationEncoder* together with the *Semantic Reranker*, we also tried varying the number of layers $L$ of the *Semantic Reranker* from 1 to 4 and changing its input, by choosing a combination from:

- $\mathbf{c}$: use of conversation's history dense representation
- $\mathbf{c}_1, \ldots, \mathbf{c}_N$: use of conversation's history tokens dense representation
- $\mathbf{p}_1, \ldots, \mathbf{p}_k$: use of passages dense representation

Table A.12 shows the average top-$k$ results obtained on the TopiOCQA and OR-QuAC development set, where $k$ varies between 1, 3, 5, 10, 15, 20, 30, 50, 100, 250, 500, 750, and 1000. For TopiOCQA, we report the presence of the gold passage within the top-$k$ limit. For OR-QuAC we report the presence of the gold answer within the top-$k$ limit.

Given that the MIPS function cannot be applied when $\mathbf{p}_1, \ldots, \mathbf{p}_k$ representations are used alone, i.e, without the conversation history, we applied a linear projection on top of the *Semantic Reranker* to obtain a score for each passage in input.

As far as training the *Semantic Reranker* is concerned, we trained it for 10 epochs when fine-tuned together with the *Conversation Encoder*. We instead trained it for 20 epochs when the

| $L$ | $\mathbf{c}$ | $\mathbf{c}_1, \ldots, \mathbf{c}_N$ | $\mathbf{p}_1, \ldots, \mathbf{p}_k$ | *Cross Encoder* | **TOPIOCQA** | **OR-QUAC** |
|---|---|---|---|---|---|---|
| 1 | ✓ | | ✓ | ✓ | 78.85 | 71.51 |
| 1 | | ✓ | ✓ | ✓ | 78.88 | 71.48 |
| 4 | | ✓ | ✓ | ✓ | 78.84 | 71.32 |
| 1 | ✓ | | ✓ | | 77.17 | 69.83 |
| 1 | | ✓ | ✓ | | 77.35 | 70.18 |
| 4 | | ✓ | ✓ | | 77.30 | 70.45 |
| 1 | | | ✓ | | 67.07 | 64.18 |

Table A.12: Average top-$k$ results obtained on the TOPIOCQA and OR-QUAC development set, with different configurations.

*Conversation Encoder* was kept frozen and when $\mathbf{p}_1, \ldots, \mathbf{p}_k$ representations were used alone. We leveraged the same objective function used for training the initial DPR. We used early stopping to chose the best performing model on the development set. We also used a linear learning rate decay throughout the training process, and AdamW (Loshchilov and Hutter, 2019) with a learning rate of 5e-5 and weight decay of 1e-2.

Among the different combinations shown in Table A.12, we considered the first entry as the best choice, i.e., the model with $L = 1$, $\mathbf{c}$, $\mathbf{p}_1, \ldots, \mathbf{p}_k$, and *Cross Encoder* fine-tuning.

## Latency measurement

Latency measurement (see Figure 4.3) has been performed on the same NVIDIA V100 16GB GPU, by following the FiD's `test_reader.py` script provided with the TOPIOCQA dataset. We set the `per_gpu_batch_size` paramenter to 4 in all runs and chose the value of the `n_context` parameter from 1, 3, 5, 10, 20, 30, and 50, based on the number of input passages. For each value, we report the latency relative to the maximum `n_context` parameter value, i.e., 50. We used CUDA events synchronisation markers to measure the elapsed time for the preprocessing and evaluation of TOPIOCQA's development set and OR-QUAC's test set.

# B Appendix for Chapter 5: *RAGferee*

## Stratified sampling

The sampling approach used in Subsection 5.5.2 leverages the concept of a *signature*, which is derived by concatenating key characteristics of each sample, such as query types and answer length categories. This signature acts as a unique identifier to group samples into strata that represent meaningful subpopulations within the dataset.

Once grouped, the signature strata are sorted in decreasing order by their size (the number of samples per signature). During the allocation of the fixed overall sampling budget, samples are selected by iterating through the signature groups starting with the least represented strata first. This ensures that smaller, potentially under-represented groups receive appropriate sampling allocation before larger groups are considered. Within each stratum, samples are stochastically selected without replacement, using random sampling guided by the allocated quota. This stochasticity maintains diversity and prevents over-representation of any single subset.

A key feature of this stratified sampling method is the use of a discounting factor applied to model-specific weights, which reflect the historical frequency of a model being selected as either *chosen* or *rejected*. Each time a model is selected, its associated weight corresponding to the *chosen* or *rejected* category is multiplicatively reduced by this discount factor, which is set to 0.9 in our experiments. This mechanism dynamically lowers the probability of repeatedly selecting the same models, thereby encouraging greater diversity and a more balanced representation of models within the sampled dataset.

Together, this stratification based on sample signatures and the adaptive, weighted stochastic sampling with discounting ensure the final sample set achieves broad coverage across sample characteristics and model choices, while mitigating biases due to dataset composition and prior model selection frequencies.

## Experimental Configurations

We carried out data preprocessing and related tasks using the `datasets` library (Lhoest et al., 2021). Fine-tuning of the Discriminative Reward Models (RMs) was performed using the `trl` (von Werra et al., 2020) and `accelerate` (Gugger et al., 2022) libraries, with inference powered by the `vLLM` library (Kwon et al., 2023).

To optimise memory efficiency during fine-tuning, we employed Parameter-Efficient Fine-Tuning (PEFT) strategies from the `peft` library (Mangrulkar et al., 2022), loading models quantised in 4-bit precision and integrating Flash Attention 2 (Dao, 2023) and Liger Kernels (Hsu et al., 2025).

Optimisation was conducted using AdamW (Loshchilov and Hutter, 2019) in 4-bit precision. We adopted rsLoRA (Kalajdzievski, 2023) and targeted all linear modules, setting the rank $r$ to 16, $\alpha$ to 16, and a dropout (Srivastava et al., 2014) rate of 0.1. Training was performed for 4 epochs with a learning rate of $2e^{-4}$, using a cosine learning rate scheduler with a warm-up ratio of 0.1. The maximum input length during training was 32,768 tokens, with a per-device batch size of 1 and gradient accumulation over 16 steps. For inference using `vLLM`, we set the maximum input length to 8,192 tokens and used a temperature of 0.0.

All experiments we conducted on a single node equipped with 8xH200 NVIDIA GPUs. Query characteristics (Subsection 5.5.1) accounted for 48 GPU hours. Answer generation and classification (Subsection 5.5.3) used 192 and 256 GPU hours, respectively. Fine-tuning each *RAGferee* model required between 3 and 8 GPU hours, depending on the model size.

The models from Hugging Face[1] used for answer generation (Subsection 5.5.3) are listed below:

- microsoft/phi-4
- Qwen/QwQ-32B
- RekaAI/reka-flash-3
- Qwen/Qwen2.5-7B-Instruct
- Qwen/Qwen2.5-14B-Instruct
- Qwen/Qwen2.5-32B-Instruct
- internlm/internlm3-8b-instruct
- allenai/OLMo-2-1124-7B-Instruct
- allenai/OLMo-2-1124-13B-Instruct
- allenai/OLMo-2-0325-32B-Instruct
- mistralai/Mixtral-8x22B-Instruct-v0.1
- mistralai/Mistral-Nemo-Instruct-2407
- mistralai/Mistral-Small-24B-Instruct-2501
- deepseek-ai/DeepSeek-R1-Distill-Qwen-14B
- deepseek-ai/DeepSeek-R1-Distill-Qwen-32B

## Distributions

Figure B.1 shows the distributions of query characteristics (Subsection 5.5.1) for the 500K, 50K, and 5K subsets. Figure B.2 presents the distribution of *chosen* and *rejected* models within the 5K subset.

[1]Hugging Face: https://huggingface.co/

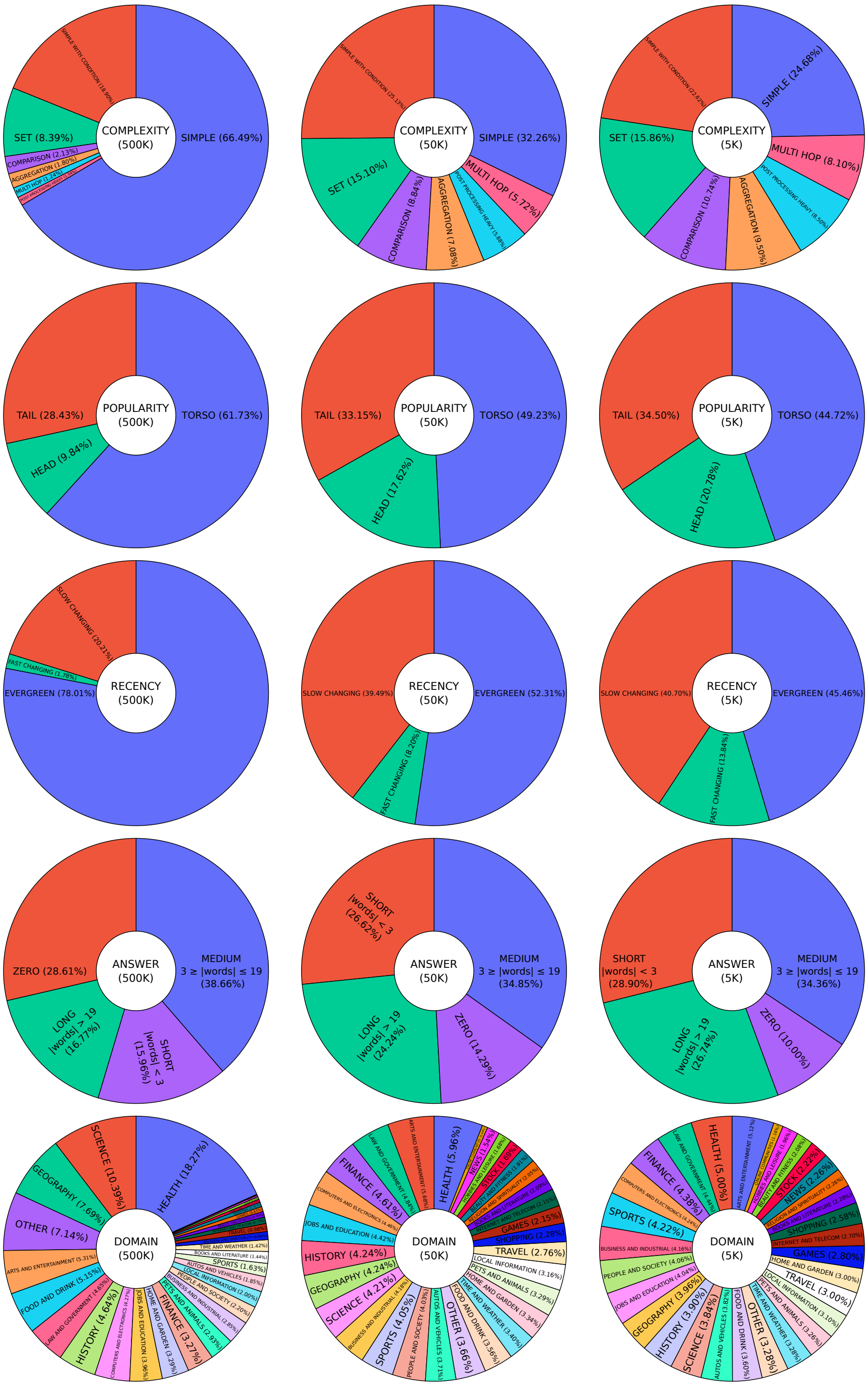



Figure B.1: Distributions of query characteristics (Subsection 5.5.1) for the 500K, 50K, and 5K subsets.

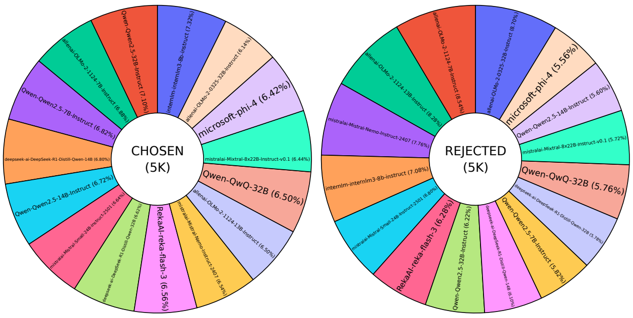



Figure B.2: Distribution of *chosen* and *rejected* models within the 5K subset.

| **Model** | **Param.** | **Pairs** | **Refusal (Ans.)** | **Refusal (Unans.)** | **Faithful. (QA)** | **Faithful. (Summ.)** | **Complete. (QA)** | **Complete. (Summ.)** | **Concise. (QA)** | **Concise. (Summ.)** | **Overall Accuracy** |
|---|---|---|---|---|---|---|---|---|---|---|---|
| *RAGferee Discriminative RMs (ours)* | | | | | | | | | | | |
| Qwen-2.5-RAGferee | 7B | 4K | 92.8 | 66.4 | 85.2 | 66.8 | 57.6 | 64.1 | 73.3 | 53.7 | 70.0 |
| Qwen-2.5-RAGferee | 14B | 4K | 92.8 | 71.2 | 86.8 | 70.8 | 65.2 | 66.9 | 71.4 | 52.0 | 72.2 |
| Mistral-Nemo-RAGferee | 12B | 4K | 92.0 | 82.8 | 82.8 | 68.8 | 62.4 | 62.9 | 86.3 | 57.0 | 74.5 |
| Mistral-Small-RAGferee | 24B | 4K | 92.4 | 81.6 | 87.2 | 75.6 | 65.6 | 65.3 | 76.5 | 57.0 | 75.2 |
| *trained w/o grounding (ablation)* | | | | | | | | | | | |
| Qwen-2.5-RAGferee | 7B | 4K | 82.8 | 25.2 | 61.6 | 60.4 | 52.8 | 61.4 | 32.5 | 41.8 | 52.3 |
| Qwen-2.5-RAGferee | 14B | 4K | 92.8 | 7.2 | 74.0 | 60.0 | 56.8 | 64.9 | 52.9 | 43.9 | 56.6 |
| Mistral-Nemo-RAGferee | 12B | 4K | 84.0 | 21.6 | 56.4 | 55.6 | 48.0 | 67.3 | 47.5 | 57.8 | 54.8 |
| Mistral-Small-RAGferee | 24B | 4K | 76.0 | 11.2 | 72.8 | 64.0 | 60.4 | 67.7 | 59.6 | 43.0 | 56.9 |

Table B.1: Expanded version of Table 5.2. ContextualJudgeBench results (*contextual accuracy*) for *RAGferee* discriminative RMs from Table 5.1 trained without grounding. Preference data alone is not sufficient. Incorporating retrieved context is crucial for accurately judging RAG responses.

| **Model** | **Param.** | **Pairs** | **Helpful** | | | | **Harmless** | | | | **Overall Accuracy** |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | **General** | **Reason** | **Citation** | **Average** | **General** | **Abstain** | **Conflict** | **Average** | |
| *Discriminative RMs (baselines)* | | | | | | | | | | | |
| InternLM-2 | 7B | 2400K | 78.6 | 68.0 | 67.0 | 70.6 | 62.6 | 66.8 | 54.9 | 61.7 | 67.3 |
| InternLM-2 | 20B | 2400K | 79.4 | 67.6 | 67.6 | 70.9 | 65.8 | 67.7 | 60.9 | 64.9 | 68.7 |
| LLaMA-3.1-Skywork-v0.2 | 8B | 80K | 76.3 | 71.9 | 56.0 | 67.0 | 85.8 | 74.7 | 79.9 | 79.5 | 71.6 |
| Gemma-2-Skywork-v0.2 | 27B | 80K | 76.0 | 73.2 | 64.8 | 70.7 | 79.4 | 80.6 | 78.8 | 79.7 | 74.1 |
| *inferenced w/o grounding (ablation)* | | | | | | | | | | | |
| InternLM-2 | 7B | 2400K | 66.4 | 60.5 | 59.6 | 61.8 | 89.7 | 60.4 | 58.7 | 68.0 | 64.1 |
| InternLM-2 | 20B | 2400K | 69.1 | 61.4 | 59.3 | 62.8 | 91.6 | 56.2 | 65.2 | 69.1 | 65.1 |
| LLaMA-3.1-Skywork-v0.2 | 8B | 80K | 66.8 | 59.5 | 53.2 | 59.1 | 95.5 | 72.4 | 77.2 | 80.4 | 67.1 |
| Gemma-2-Skywork-v0.2 | 27B | 80K | 66.8 | 64.4 | 59.3 | 63.1 | 93.5 | 75.6 | 78.3 | 81.5 | 70.0 |
| *RAGferee Discriminative RMs (ours)* | | | | | | | | | | | |
| Qwen-2.5-RAGferee | 7B | 4K | 65.3 | 65.0 | 59.8 | 63.1 | 52.3 | 66.4 | 37.0 | 52.7 | 59.2 |
| Qwen-2.5-RAGferee | 14B | 4K | 72.1 | 70.6 | 59.8 | 66.8 | 51.0 | 72.4 | 43.5 | 56.8 | 63.1 |
| Mistral-Nemo-RAGferee | 12B | 4K | 67.6 | 64.7 | 58.4 | 63.1 | 50.3 | 71.0 | 43.5 | 56.1 | 60.5 |
| Mistral-Small-RAGferee | 24B | 4K | 67.6 | 67.0 | 58.7 | 63.9 | 47.7 | 71.9 | 48.4 | 57.4 | 61.5 |
| *inferenced w/o grounding (ablation)* | | | | | | | | | | | |
| Qwen-2.5-RAGferee | 7B | 4K | 63.4 | 55.9 | 65.9 | 61.9 | 93.5 | 70.5 | 67.4 | 75.9 | 67.1 |
| Qwen-2.5-RAGferee | 14B | 4K | 63.0 | 57.8 | 64.5 | 61.9 | 92.3 | 81.6 | 72.3 | 81.5 | 69.2 |
| Mistral-Nemo-RAGferee | 12B | 4K | 56.9 | 51.0 | 55.7 | 54.5 | 92.3 | 78.3 | 65.8 | 78.1 | 63.3 |
| Mistral-Small-RAGferee | 24B | 4K | 63.7 | 62.1 | 61.5 | 62.3 | 81.3 | 80.2 | 70.1 | 77.2 | 67.9 |

Table B.2: Expanded version of Table 5.3. RAG-RewardBench results (*consistent accuracy*) for discriminative RMs inferenced with or without grounding. Grounding has minimal impact on baseline discriminative RMs (non-RAG), but it significantly influences our *RAGferee* discriminative RMs, which are sensitive to grounding by design.

| Model | Param. | Pairs | Refusal (Ans.) | Refusal (Unans.) | Faithful. (QA) | Faithful. (Summ.) | Complete. (QA) | Complete. (Summ.) | Concise. (QA) | Concise. (Summ.) | Overall Accuracy |
|---|---|---|---|---|---|---|---|---|---|---|---|
| *Generative (non-reward) Models (baselines)* | | | | | | | | | | | |
| Qwen-2.5 | 7B | - | 30.0 | 33.2 | 29.2 | 28.0 | 29.2 | 25.5 | 17.6 | 31.6 | 28.0 |
| Qwen-2.5 | 14B | - | 54.0 | 63.2 | 42.8 | 39.6 | 27.6 | 40.6 | 8.6 | 22.1 | 37.3 |
| Mistral-Nemo | 12B | - | 24.8 | 12.0 | 26.4 | 27.6 | 28.0 | 35.9 | 10.2 | 13.5 | 22.3 |
| Mistral-Small | 24B | - | 50.8 | 54.8 | 48.0 | 42.8 | 34.4 | 47.0 | 25.9 | 35.7 | 42.4 |
| *RAGferee* Generative Models (SFT) (ours) | | | | | | | | | | | |
| Qwen-2.5-RAGferee | 7B | 4K | 75.2 | 61.2 | 54.0 | 44.0 | 29.6 | 28.7 | 51.4 | 29.9 | 46.8 |
| Qwen-2.5-RAGferee | 14B | 4K | 83.6 | 68.0 | 60.8 | 44.0 | 38.4 | 36.3 | 64.7 | 35.2 | 53.9 |
| Mistral-Nemo-RAGferee | 12B | 4K | 77.6 | 70.0 | 55.6 | 29.2 | 37.2 | 32.3 | 63.1 | 34.8 | 50.0 |
| Mistral-Small-RAGferee | 24B | 4K | 80.8 | 64.4 | 54.4 | 37.2 | 35.6 | 33.5 | 62.0 | 35.2 | 50.4 |

Table B.3: Expanded version of Table 5.5. CONTEXTUALJUDGEBENCH results (*consistent accuracy*) of *RAGferee* generative RMs trained to output the indicator of the preferred response.

## Examples

### ContextualJudgeBench: Refusal (Answerable)

**Query**

```
how to be a zero-bug programmer?
```

**Grounding**

```
[1] Don't code at all. That's the only way you can be a zero-bug
programmer. Bugs are unavoidable because programmers are human,
all we can do is try our best to prevent them, react quickly when
a bug occurs, learn from our mistakes and stay up to date.
```

**Chosen**

```
To achieve the status of a zero-bug programmer, the advice is
straightforward: abstain from coding entirely.
Since bugs are an inevitable aspect of programming due to human
error, the focus should instead be on diligent prevention, swift
response to issues, learning from past errors, and keeping
knowledge current.
```

**Rejected**

```
The contexts provided touch on the inevitability of bugs in
programming but lack detailed methodologies, best practices in
software development, and specific quality assurance techniques
necessary to achieve a near-zero-bug codebase. To properly answer
the question, information about systematic testing approaches,
code review strategies, and the integration of automated tools in
the development lifecycle would be essential.
```

**Predictions**

Gemma-2-Skywork-v0.2 (80K pairs & 27B parameters): **Rejected**

Mistral-Small-RAGferee (4K pairs & 24B parameters): **Chosen**

Mistral-Small-RAGferee (4K pairs & 24B parameters) (trained w/o grounding): **Rejected**

## CONTEXTUALJUDGEBENCH: Refusal (Unanswerable)

**Query**

What does mTOR stands for?

**Grounding**

Ghrelin exhibits its biological effect through binding to the growth hormone secretagogue 1a receptor (GHS-R1a). Recently, it has been reported that ghrelin has an anti-apoptotic effect in several cell types. However, the molecule mechanisms underlying the anti-apoptotic effect of ghrelin remain poorly understood. In this study, we investigated the intracellular mechanisms responsible for anti-apoptotic effect of ghrelin on human umbilical vein endothelial cells (HUVEC). Treatment of HUVEC with ghrelin inhibited high glucose-induced cell apoptosis. Ghrelin stimulated the rapid phosphorylation of key proteins involved in cell survival. The GHS-R1a-specific antagonist [D-Lys3]-GHRP-6 abolished the anti-apoptotic effect and inhibited the activation of key survival pathways induced by ghrelin. Pretreatment of cells with specific inhibitors blocked the anti-apoptotic effect of ghrelin. In addition, ghrelin protected HUVECs against high glucose induced apoptosis by increasing Bcl-2/Bax ratio. Taken together, our results demonstrate that ghrelin produces a protective effect on HUVECs through activating GHS-R1a and specific signaling pathways mediates the effect of ghrelin. These observations suggest that ghrelin may act as a survival factor in preventing HUVECs apoptosis caused by high glucose.

**Chosen**

While the context provides a detailed examination of the anti-apoptotic effects of ghrelin and its interaction with specific receptors and signaling pathways, it does not mention or define the term mTOR or its biological significance. To accurately explain what mTOR stands for, the context would need to include information on molecular signaling pathways, specifically those involving the mechanistic target of rapamycin, which is a central regulator of cell metabolism, growth, proliferation, and survival.

**Rejected**

The context discusses the anti-apoptotic effects of ghrelin on endothelial cells but does not directly mention mTOR or its full form. However, mTOR is a well-known protein kinase involved in cell survival pathways, which aligns with the context describing phosphorylation and cell survival mechanisms. Thus, mTOR is likely relevant to the study's focus on signaling pathways.

mTOR stands for mechanistic Target of Rapamycin.

**Predictions**

Gemma-2-Skywork-v0.2 (80K pairs & 27B parameters):
**Rejected**

Mistral-Small-RAGferee (4K pairs & 24B parameters):
**Chosen**

Mistral-Small-RAGferee (4K pairs & 24B parameters) (trained w/o grounding):
**Rejected**

## CONTEXTUALJUDGEBENCH: Faithfulness (QA)

**Query**

How does evolution occur and how did the first living organisms come about

**Grounding**

Charles Carter and Richard Wolfenden, both of the University of North Carolina, have uncovered new evidence of abiogenesis, the process by which life arises from non-living chemical matter. Their study, published Thursday in the Journal of Biological Chemistry, suggests that a single ancient gene may have used each of its opposite DNA strands to code for different chemical catalysts. Those separate catalysts would have both activated amino acids, which then formed proteins – essential to the production of living cells.

In the early 20th century, the "primordial soup" model of abiogenesis started to gain traction. It proposes that in Earth's prebiotic history, simple organic matter was exposed to energy in the form of volcanoes and electrical storms. That energy would have catalyzed chemical reactions that, in the span of a few hundred million years, could have produced self-replicating molecules.

Today, the Miller-Urey experiment is contested for a number of reasons, including the possibility that Earth's original atmosphere may have had a different composition. Still, the production of organic compounds from inorganic "precursors" laid a strong foundation for the primordial soup hypothesis. And new findings support that hypothesis, Dr. Carter says.

The first living things on Earth, single-celled micro-organisms or microbes lacking a cell nucleus or cell membrane known as prokaryotes, seem to have first appeared on Earth almost four billion years ago, just a few hundred million years after the formation of the Earth itself. By far the longest portion of the history of life on Earth, therefore, has involved the biochemical evolution of these single-celled micro-organisms, bacteria and archaea: we can find individual fossilized microbes in rocks 3.4 billion years old, yet we can only conclusively identify multi-celled fossils in rocks younger than 1 billion years.

It is presumed that, over a few hundred million years of evolution, pre-biotic molecules evolved into self-replicating molecules by natural selection. While some aspects of the subject are well understood, others remain clouded in mystery and are the source of much contention among scientists. Although much progress has been made, there is still no single definitive theory.

**Chosen**

The process of abiogenesis, or the emergence of life from non-living matter, is still not fully understood and is the subject of much debate among scientists. Several theories exist, one of the most well-known being the "primordial soup" theory, which proposes that in the early stages of Earth\'s history, organic matter was exposed to energy in the form of volcanoes and electrical storms, which would have catalyzed chemical reactions that could have produced self-replicating molecules. Evidence of this theory is supported by the fact that organic compounds can be produced from inorganic "precursors" in the lab. Another theory is that a single ancient gene may have used each of its opposite DNA strands to code for different chemical catalysts, which would have both activated amino acids, which then formed proteins - essential to the production of living cells. Additionally, it is believed that over the course of hundreds of millions of years, pre-biotic molecules evolved into self-replicating molecules by natural selection.

**Rejected**

Evolution occurs when successive generations of organisms inherit different traits, which can be passed down to their offspring. Over time, these changes can result in new species and even major changes in the structure of organisms. The first living organisms likely arose from non-living chemicals around 4 billion years ago, in a process known as abiogenesis. During this process, simple organic molecules were likely catalyzed by energy from volcanoes and lightning into more complex molecules, which eventually led to the first living cells.

**Predictions**

Gemma-2-Skywork-v0.2 (80K pairs & 27B parameters): **Rejected**

Mistral-Small-RAGferee (4K pairs & 24B parameters): **Chosen**

Mistral-Small-RAGferee (4K pairs & 24B parameters) (trained w/o grounding): **Rejected**

## ContextualJudgeBench: Faithfulness (Summarisation)

**Query**

Write a precise and to-the-point summary based on the provided context focusing on Rent Increase Regulations.

**Grounding**

Speaker 1: Five F recommendation to authorize the Interim City Manager to execute an agreement with CSG consulting services for professional plant services for 12 months. Permanent amount not to exceed 300,000.
Speaker 0: And I pulled this item. I don't know if staff wants to present anything about it. I pulled it because it's for up to 300,000 for structural engineer or an engineering firm. And it's my understanding that we, the city had had had an engineer through that, resigned in June. And so now we're being asked to approve a contract with a consulting firm out of San Mateo to make up for that. That would be up to 12 months and $300,000.
Speaker 2: So. Good evening. My name is Debbie Potter and I'm the city's community development director. And it is accurate that we had a plan check engineer that was working with the city. He actually resigned a year ago in June. So we have been using outside planned check services for several years now. We are using these planned check services. We are incredibly busy at the permit center and the flexibility that we have by using planned check services outside, planned check services we can use from one to 10 to 15 plan checkers all at the same time to keep moving our projects through the system. And while we do use planned check services, it should be noted that about 75% of all plan checking happens over the counter right here at City Hall. But 25% of our projects are complicated enough that they have to go out to a structural engineer. We're using CSG. It's the same plan check services that we that the fire department uses for its planned tech services. So that synergy has been very nice for us to have plan checkers that are very familiar with both the city's fire code and then all the building codes. We would like to continue on with the services that we are receiving from CSG. There is no impact on the general fund CSG. The contract is set up so that they charge 65% of the cost of the building permit, so that we are always providing services to applicants within the budget that we charge for building permits. It's no impact on the general fund and we will be continuing to look throughout this fiscal year at how we want to staff up that function within the department. And we're looking at perhaps keeping outside contract services and then possibly under filling the position with a plans examiner so we can do more over the counter. So we feel like we have an efficient model in place and it's really based on that analysis and the track record for the last year and a half that we're recommending approval of this contract for the current fiscal year.
Speaker 0: Do we have any engineers as part of our staff? Can you describe the types of engineers that we have as part of our staff?
Speaker 2: So the building official is has a degree in architecture, which is structural engineers and architects are the ones who can do the plan, check services. And then we have our supervising building inspector is a plan examiner, a certified plans examiner. So those are the two certified staff that we have.
Speaker 0: But do we have any engineers as part of our staff? That's actually my concern.
Speaker 2: We currently do not have a structural engineer, Madame Mayor. We have engineers in our public works department.
Speaker 0: Advanced planning. No, no, no. So that's that's where I'm that's where my question goes. My preference would be that we have at least one engineer on our staff. And it appears that that's counter to what the plan currently is with staff, that you plan to continue to have a consulting firm as opposed to an engineer that's actually an employee of the city, which I we used to have that and I think I've seen that that issue has been raised in regards to having an engineer as part of our planning department. So that was my concern. And I pulled out. And I actually think that the community is interested in this issue because there have been multiple letters to the editor regarding engineers as part of our planning department, separate from public works. And I appreciate that.
Speaker 2: Right. Most of those letters, though, are all about the public works engineers. Those letters that we've received have not been about the planning department. But I understand what you're saying, and we can take a look at it, because our building official, who has a degree in architecture, is able to do the same types of plan checking as a structural engineer it we just feel like it's more efficient to be able to access multiple structural engineers during our very busy period. And that's really what what we need and what we need. Internal to our staff are the certified plans examiners. And that's we also we do have several certified plans examiners on staff.
Speaker 0: And I appreciate that. Thank you for the comments.
Speaker 8: No, I just said I'm willing to move approval because I think Mr. Potter gave an excellent explanation if we want to keep these projects moving. Time is money. This is not impacting our general fund, and I think we're serving our community well. So with that, I will move approval of item five after recommendation to authorize the Interim City Manager to execute an agreement with CSG consulting services for professional plant services for a 12 month term in an amount not to exceed $300,000.
Speaker 0: We have a second.
Speaker 3: Second. All right.
Speaker 0: Any other comments? All those in favor.
Speaker 3: I think you.
Speaker 0: Of motion passes 4 to 1. Thank you. Thank you. And now we have speakers on five. Q And five are with you.
Speaker 1: And five Q is a final passage of ordinance amending the Alameda Municipal Code by adding article? What does that 14 to Chapter six concerning review of rent increases.

**Chosen**

The council discussed approving a contract for outside engineering services. The mayor prefers hiring an in-house engineer. Staff argued outside engineers allow flexibility and efficient permit processing. The contract charges applicants, not the general fund. The council approved the contract 4-1, enabling continued tenant protections through efficient permit review.

**Rejected**

The document outlines a review process for rent increases in Alameda, California, which includes a threshold for when increases require review and a process for submitting and reviewing rent increase requests. The review process involves the Rent Stabilization Board, landlords, and tenants, and may include public hearings. The goal of the process is to ensure that rent increases are reasonable and do not harm tenants or the community.

**Predictions**

| | |
|---|---|
| Gemma-2-Skywork-v0.2 (80K pairs & 27B parameters): | Rejected |
| Mistral-Small-RAGferee (4K pairs & 24B parameters): | Chosen |
| Mistral-Small-RAGferee (4K pairs & 24B parameters) (trained w/o grounding): | Rejected |

## ContextualJudgeBench: Completeness (QA)

**Query**

What war was taking place in wonder woman?

**Grounding**

Wonder Woman (2017 film)
This version was conceived of as a prequel to the first live-action, theatrical appearance of Wonder Woman, in the 2016 film, "", placing Wonder Woman in the 1910s and World War I (a decision which differs from her comic book origins as a supporter of the Allies during World War II).
As for story development, Jenkins credits the stories by the character's creator William Moulton Marston in the 1940s and George Perez's seminal stories in the 1980s in which he modernized the character.
In addition, it follows some aspects of DC Comics' origin changes in The New 52 reboot, where Diana is the daughter of Zeus.
Jenkins cited Richard Donner's "Superman" as an inspiration.
In late 2013, Zack Snyder cast Gal Gadot in the role of Wonder Woman for the 2016 film, "" over Élodie Yung and Olga Kurylenko.
Some fans initially reacted to this choice by criticizing Gadot's appearance.
Snyder would later comment on his decision to cast Gadot, stating that he tested a "bunch of actresses, as you can imagine.
But the thing with Gal is that she's strong, she's beautiful, and she's a kind person, which is interesting, but fierce at the same time.
It's that combination of being fierce but kind at the same time that we were looking for.

Publication history of Wonder Woman
Following the popularity of the "Wonder Woman" TV series (initially set during World War II), the comic book was also transposed to this era.
The change was made possible by the multiverse concept, which maintained that the 1970s Wonder Woman and the original 1940s version existed in two separate yet parallel worlds.
A few months after the TV series changed its setting to the 1970s, the comic book returned to the contemporary timeline.
Soon after, when the series was written by Jack C. Harris, Steve (Howard) Trevor was killed off yet again.
Wonder Woman was one of the backup features in "World's Finest Comics" #244-252 and "Adventure Comics" #459-464 when those titles were in the Dollar Comics format.
"All-New Collectors' Edition" #C-54 (1978) featured a Superman vs. Wonder Woman story by writer Gerry Conway and artists José Luis García-López and Dan Adkins.
Writer Gerry Conway brought Steve Trevor back to life again in issue #271 (September 1980).
Following Diana's renunciation of her role as Wonder Woman, a version of Steve Trevor from an undisclosed portion of the Multiverse accidentally made the transition to Earth-One.
With Diana's memory erased by the Mists of Nepenthe, the new Steve again crash-landed and arrived at Paradise Island.

**Chosen**

Following the popularity of the "Wonder Woman" TV series (initially set during World War II), the comic book was also transposed to this era. The change was made possible by the multiverse concept, which maintained that the 1970s Wonder Woman and the original 1940s version existed in two separate yet parallel worlds.

**Rejected**

During Wonder Woman's first live-action film, set in the 1910s, the story took place during World War I. In the comic book, it was set in the 1970s, during World War II.

**Predictions**

Gemma-2-Skywork-v0.2 (80K pairs & 27B parameters): **Rejected**

Mistral-Small-RAGferee (4K pairs & 24B parameters): **Chosen**

Mistral-Small-RAGferee (4K pairs & 24B parameters) (trained w/o grounding): **Rejected**

## ContextualJudgeBench: Completeness (Summarisation)

**Query**

Write an accurate and brief summary derived from the given context.

**Grounding**

USER : I'm looking for a train to Stevenage that leaves after 17:30. Is there anything available then?
SYSTEM : I have a few choices for you. What day are you wanting to travel?
USER : I would like to leave on Thursday out of Cambridge.
SYSTEM : The TR0385 leaves at 19:21, would that suit you?
USER : Yes, that sounds perfect. I would like a booking for 4 people.
SYSTEM : The booking was successful. Your reference number is SICMOAA9 . Can I help you with anything else?
USER : I am looking for some places to go in town. I want something in the centre that has multiple sports.
SYSTEM : I am sorry but there is nothing matching your request. Would you like to try another part of town or type of attraction?
USER : Well, how about any nightclubs?
SYSTEM : Yes I have one located on the east side.
USER : Do you have any nightclubs located in the centre?
SYSTEM : Yes there are 5 located in the centre. Is there anything I can help you with?
USER : Can I get the phone number for one of them?
SYSTEM : Sure, Ballare in heidelberg gardens, lion yard cost 5 pounds and their phone number is 01223364222. Is there anything else you'd like?
USER : No. That's everything I was looking for today. Thanks for your help.
SYSTEM : Thank you for choosing help desk. Good Bye.

**Chosen**

The user successfully booked a train to Stevenage for 4 people, leaving Cambridge on Thursday at 19:21. The booking reference number is SICMOAA9. Additionally, the user inquired about places of interest in the town center, specifically sports locations and nightclubs. Although no sports locations matched their criteria, they were given information about nightclubs, including the phone number for Ballare, located at Heidelberg Gardens, Lion Yard, with an entry cost of 5 pounds.

**Rejected**

The text is a conversation between a user and a system. The user asks for a train to Stevenage after 17:30, books tickets for 4 people, then asks for recommendations for places to go in town focusing on nightclubs in the centre. The system provides train options, booking confirmation, and contact details for a nightclub matching the requests.

**Predictions**

Gemma-2-Skywork-v0.2 (80K pairs & 27B parameters):
**Rejected**

Mistral-Small-RAGferee (4K pairs & 24B parameters):
**Chosen**

Mistral-Small-RAGferee (4K pairs & 24B parameters) (trained w/o grounding):
**Rejected**

## ContextualJudgeBench: Conciseness (QA)

**Query**

```
Who played stormy in the movie odd thomas?
```

**Grounding**

```
Odd Thomas (film)
Odd Thomas (film)

Odd Thomas is a 2013 American supernatural mystery thriller film based on Dean Koontz's novel of
the same name.
It is directed, written, and co-produced by Stephen Sommers and stars Anton Yelchin as Odd
Thomas, with Willem Dafoe as Wyatt Porter, and Addison Timlin as Stormy Llewellyn.
Odd Thomas (Yelchin) is a psychic who lives in a small town in California.
He describes his ability as, "I see dead people, but then, by God, I do something about it."
One morning the ghost of a teenage girl, Penny Kallisto, silently leads him to Harlo Landerson.
Odd accuses Harlo of raping and murdering Penny.
Harlo flees.
Odd chases him to a pool party and into a child's bedroom in a stranger's house.
Harlo and Odd fight and Harlo is knocked unconscious.
Odd's friend, police chief Wyatt Porter (Dafoe), is aware of Odd's psychic gifts and promises to
spin the story to keep public attention away from him.
Odd has a vision of faceless people wearing bowling shirts who cry out to him to save them.
A faceless gunman shoots them all, including Odd.

Addison Timlin
Addison Timlin

Addison Jayne Timlin (born June 29, 1991) is an American actress, best known for her roles as
Jami Lerner in "The Town That Dreaded Sundown" (2014) and Colleen Lunsford in "Little Sister"
(2016).
She is also known for playing Sasha Bingham in Showtime's "Californication".
Born in Philadelphia, Pennsylvania, Timlin debuted on the big screen as Amy, a 14-year-old girl
with diabetes, in the 2005 film "Derailed" alongside Clive Owen and Melissa George, who played
her parents.
Timlin played Maddy in a short film, "The Isabel Fish", directed by Lara Zizic for the Columbia
Film Festival.
In 2007, Timlin appeared in the music video, "Sleeping Lessons," by Albuquerque, New Mexico based
indie-rock band, The Shins.
In 2008, she made an appearance in the music video for the song Check
Yes Juliet by We the Kings, playing the fictional Juliet.
She was cast in the short-lived CBS television drama, "3 lbs", as Charlotte Hanson (the daughter
of the main character played by Stanley Tucci), in three of the eight episodes that were filmed.
Timlin is seen in the short film "Man", written and directed by Myna Joseph, which was screened
at the Sundance Film Festival and at Cannes.
```

**Chosen**

```
Odd Thomas is a 2013 supernatural mystery thriller film based on Dean Koontz's novel of the same
name. The film stars Addison Timlin as Stormy Llewellyn.
```

**Rejected**

```
Odd Thomas is a 2013 supernatural mystery thriller film based on Dean Koontz's novel of the same
name. The film stars Anton Yelchin as Odd Thomas, with Willem Dafoe as Wyatt Porter, and Addison
Timlin as Stormy Llewellyn. American actress Addison Timlin also played Stormy in the 2013 movie.
```

**Predictions**

| | |
|---|---|
| Gemma-2-Skywork-v0.2 (80K pairs & 27B parameters): | Rejected |
| Mistral-Small-RAGferee (4K pairs & 24B parameters): | Chosen |
| Mistral-Small-RAGferee (4K pairs & 24B parameters) (trained w/o grounding): | Rejected |

## CONTEXTUALJUDGEBENCH: Conciseness (Summarisation)

**Query**

Formulate a concise and faithful overview that reflects the key points of the given context.

**Grounding**

Manchester United will have to wear Nike kit during their summer tour of the US, despite their world-record deal with adidas. The club are expected to jet across the Atlantic in July for a trip of around 12 days, which would have been an ideal opportunity to showcase their new gear following their £750m, 10-year agreement. However, their existing deal with Nike runs until the end of July and Sportsmail understands there is no buy-out agreement that would allow the club to exit the agreement early has been agreed. Manchester United will have to wear Nike kit during their summer US tour , despite their new adidas deal . Wayne Rooney and co are expected to jet across the Atlantic in July but will have to wear their old kit . adidas are the new sponsors in a £750million deal but there is no buy-out clause in Nike deal that ends in July . Ryan Giggs is pictured in an adidas shirt, the Barclays Premier League side last wore adidas shirts in 1992 . That means that unless United remain in America until August 1 - a week before the start of the season - fans will have to wait to see the new kit while their team continues in what will effectively be an old ensemble. As reported on Wednesday, the tour will be shorter than last year's epic trek across the country and is likely to be confined to the west coast thanks to input from manager Louis van Gaal. The Dutchman, who pays ferocious attention to detail, vowed that a repeat of last summer would not happen again on his watch and believes to this day that the effects of the pre-season schedule cost his side dearly when the real action started. Louis van Gaal has had a big input on United's pre-season plans, restricting the US tour to just 12 days . Van Gaal believes the extensive tour last year hampered United and it will now be restricted to the west coast . United traditionally stage a pre-season friendly at Old Trafford and August 1 would seem a logical choice . United traditionally stage a pre-season friendly at Old Trafford and August 1 would seem a logical choice. They faced Valencia last year and Sevilla in 2013 although the Olympics prevented such a fixture taking place in 2012. Officials like a test run to ensure new equipment is working and to allow new staff to get used to a matchday operation at Old Trafford.

**Chosen**

Manchester United will have to wear Nike kit during their summer US tour, despite their new adidas deal. The club's existing deal with Nike runs until the end of July, and there is no buy-out agreement. The tour will be shorter than last year's and is likely to be confined to the west coast. Manager Louis van Gaal has had a big input on United's pre-season plans, restricting the US tour to just 12 days, as he believes the extensive tour last year hampered the team.

**Rejected**

Manchester United will wear Nike kits during their US summer tour despite signing a new £750m deal with adidas, due to their current contract with Nike running until the end of July without a buy-out clause. The tour, influenced by manager Louis van Gaal to be around 12 days and focused on the west coast, aims to avoid the extensive travel that impacted the team's performance last season. Fans will have to wait to see the team in adidas gear, potentially until after their traditional pre-season friendly at Old Trafford, scheduled for August 1.

**Predictions**

| | |
|---|---|
| Gemma-2-Skywork-v0.2 (80K pairs & 27B parameters): | Rejected |
| Mistral-Small-RAGferee (4K pairs & 14B parameters): | Chosen |
| Mistral-Small-RAGferee (4K pairs & 14B parameters) (trained w/o grounding): | Rejected |

## *RAGferee*: Sample 1

**Query**

```
How did Oprah Winfrey build schools?
```

**Grounding**

```
Reference [1]
Text: Four years after opening its doors, the Academy’s, first matric students graduated
from OWLAG. While some have already graduated from university and are in honours
programmes, many are currently in the final months before they receive their degrees and
diplomas.

Reference [2]
Text: Email. Oprah Winfrey is set to graduate the first class of her Oprah Winfrey
Leadership Academy for Girls on Saturday, a 10-year journey that has been filled with
tears, trials and triumph, she told ABC News&#39; Diane Sawyer. I&#39;ve learned so much.
...

Reference [3]
Text: Understanding this, the Oprah Winfrey Leadership Academy Foundation (OWLAF) hosted
its first networking event, which provided the OWLAG graduates the opportunity to engage
with business executives and learn from their experiences in preparation of their careers.
```

**Chosen**

```
Reference [1]
Text: Four years after opening its doors, the Academy’s, first matric students graduated
from OWLAG. While some have already graduated from university and are in honours
programmes, many are currently in the final months before they receive their degrees and
diplomas.

Reference [2]
Text: Email. Oprah Winfrey is set to graduate the first class of her Oprah Winfrey
Leadership Academy for Girls on Saturday, a 10-year journey that has been filled with
tears, trials and triumph, she told ABC News' Diane Sawyer. I've learned so much. ...

Reference [3]
Text: Understanding this, the Oprah Winfrey Leadership Academy Foundation (OWLAF) hosted
its first networking event, which provided the OWLAG graduates the opportunity to engage
with business executives and learn from their experiences in preparation of their careers.
```

**Rejected**

```
Oprah Winfrey built schools through the establishment of the Oprah Winfrey Leadership
Academy for Girls in South Africa. The academy was founded in 2007 with the goal of
providing quality education to underprivileged girls. The first class of students
graduated in 2011, marking a significant milestone in the academy's mission. The
establishment of the academy was a personal initiative by Oprah Winfrey, reflecting her
commitment to education and empowerment. The school offers a rigorous academic program
alongside activities that foster leadership and personal development. The initial funding
and ongoing support for the academy have been significant, showcasing Oprah Winfrey's
substantial investment in education. [1] [2] [3]
```

**Predictions**

| | |
|---|---|
| Gemma-2-Skywork-v0.2 (80K pairs & 27B parameters): | **Rejected** |
| Qwen-2.5-RAGferee (4K pairs & 14B parameters): | **Chosen** |
| Qwen-2.5-RAGferee (4K pairs & 14B parameters) (trained w/o grounding): | **Rejected** |

## *RAGferee*: Sample 2

**Query**

Where does the body get the air that it needs?

**Grounding**

Reference [1]
Text: In reality, the body stores the excess protein or carbs in a person's diet in form of fat, specifically, as triglyceride molecules, which consist of just three kinds of atoms: carbon, hydrogen and oxygen.

Reference [2]
Text: Turns out, most of it is exhaled. In a new study, scientists explain the fate of fat in a human body, and through precise calculations, debunk some common misconceptions. Fat doesn't simply turn into energy or heat, and it doesn't break into smaller parts and get excreted, the researchers say.

Reference [3]
Text: The researchers showed that during weight loss, 84 percent of the fat that is lost turns into carbon dioxide and leaves the body through the lungs, whereas the remaining 16 percent becomes water, according to the study published today (Dec. 16) in a special Christmas issue of the medical journal BMJ.

Reference [4]
Text: The point is to use cold air found in an area not affected in any way by the engine's ambient heat. The hoses for these intakes, as you can imagine, require some length and a few turns to achieve the desired distance. A Ram Air Intake, on the other hand, uses a forward-facing air scoop located at a vehicle's front end.

Reference [5]
Text: Oxygen is then used in our cells as the fuel that transforms the food we eat into energy.". Students will go through a series of resources to learn about: the respiration process (basic mechanics), its importance as fuel to our cells, and its importance in health and disease. Basic Mechanics of Respiration.

Reference [6]
Text: So, for example, to burn 10 kilograms (22 lbs.) of fat, a person needs to inhale 29 kg (64 lbs.) of oxygen. And the chemical process of burning that fat will produce 28 kg (62 lbs.) of carbon dioxide and 11 kg (24 lbs.) of water, the researchers calculated.

Reference [7]
Text: 1 (The nose and mouth make up the first part where air enters your body. 2 The trachea, or windpipe, is the second part and it delivers air to the lungs. 3 Your lungs are the third part where oxygen is absorbed by the blood, which brings it to the rest of the body. 4 Finally, the diaphragm is the fourth part.

Reference [8]
Text: 1 The trachea, or windpipe, is the second part and it delivers air to the lungs. 2 Your lungs are the third part where oxygen is absorbed by the blood, which brings it to the rest of the body. 3 Finally, the diaphragm is the fourth part. 4 It makes up the floor of your rib cage.).

Reference [9]
Text: These units open up your motor's breathing passages, giving you real-world boosts in power, performance, and efficiency. Before you complete your search for the best air intake, check out some of the common questions we get from our customers. More power will be yours, and we are here to help.

**Chosen**

Okay, so I need to figure out where the body gets the air it needs. Let me look through the references provided to find the relevant information.

Starting with Reference [5], it mentions that oxygen is used in our cells as fuel, which relates to respiration. That makes sense because we know that breathing is how we take in oxygen.

References [7] and [3] talk about how fat is converted into carbon dioxide and water, but that's more about what happens to fat, not where air comes from. However, they do mention the process of respiration, which involves oxygen intake.

References [7] also states that to burn fat, a person needs to inhale a significant amount of oxygen, which ties back to the intake of air.

Looking at References [7], [3], and [5], they all point towards the lungs as the place where oxygen is absorbed into the blood. The process starts with air entering through the nose or mouth, moving down the trachea, and then into the lungs where oxygen is absorbed.

References [7] and [3] specifically mention that during weight loss, fat is converted into carbon dioxide and water, which are exhaled. This shows that the lungs are crucial in both taking in oxygen and releasing carbon dioxide.

Putting it all together, the body gets the air it needs through the respiratory system. Air enters through the nose or mouth, travels down the trachea, and into the lungs where oxygen is absorbed into the bloodstream. The rest of the body uses this oxygen for cellular processes, and carbon dioxide is expelled when we exhale.
</think>

The body obtains the air it needs through the respiratory system. Air enters through the nose or mouth, travels down the trachea, and reaches the lungs where oxygen is absorbed into the bloodstream. This oxygen is then utilized by cells for energy production. Carbon dioxide, a byproduct of cellular respiration, is expelled when exhaling.

Citations: [5], [7], [3]

**Rejected**

The body gets the air it needs through the respiratory system, which includes the nose and mouth, trachea (windpipe), lungs, and diaphragm. Air enters the body through the nose and mouth, then travels down the trachea to the lungs where oxygen is absorbed by the blood. The diaphragm plays a crucial role in this process by assisting in the expansion and contraction of the lungs to facilitate breathing. This oxygen is then transported to the rest of the body's cells where it is used for cellular respiration to produce energy.

**Predictions**

| | |
|---|---|
| Gemma-2-Skywork-v0.2 (80K pairs & 27B parameters): | Rejected |
| Qwen-2.5-RAGferee (4K pairs & 14B parameters): | Chosen |
| Qwen-2.5-RAGferee (4K pairs & 14B parameters) (trained w/o grounding): | Rejected |

## RAG-REWARDBENCH: Issue sample 1

**Query**

```
Who was the director of Man of the House?
```

**Grounding: part 1**

```
Reference [1]
Title: Man of the House (1995 film)
Text: Man of the House (1995 film) Man of the House is a 1995 American comedy film starring Chevy
Chase, Farrah Fawcett and Jonathan Taylor Thomas. The film is about a boy (Thomas) who must come
to terms with his potential stepfather (Chase), a well meaning lawyer who is unknowingly the
subject of a manhunt by relatives of a man he helped land in prison. It was shot in Los Angeles,
California and Vancouver, British Columbia, Canada. Six year old Ben Archer watches silently as
his father starts up his car and drives away with his secretary, and they both offer

Reference [2]
Title: Man of the House (2005 film)
Text: Man of the House (2005 film) Man of the House is a 2005 American crime comedy film directed
by Stephen Herek and starring Tommy Lee Jones. The plot revolves around Sgt. Roland Sharp, a
lonesome Texas Ranger who goes undercover as an assistant coach to protect a group of college
cheerleaders who have witnessed a murder. Much of the film was shot in Austin, Texas on the
University of Texas campus. Texas Governor Rick Perry has a cameo appearance in the film as
himself. Released on February 25, 2005, the film received negative reviews, and grossed just $21
million against

Reference [3]
Title: Man of the House (1995 film)
Text: television spots. The film was panned by critics, and has a rating of 14% on Rotten
Tomatoes, based on 14 reviews. The film did moderately well at the box office, grossing about $40
million domestically. The film was released in the United Kingdom on June 9, 1995. Man of the
House (1995 film) Man of the House is a 1995 American comedy film starring Chevy Chase, Farrah
Fawcett and Jonathan Taylor Thomas. The film is about a boy (Thomas) who must come to terms with
his potential stepfather (Chase), a well meaning lawyer who is unknowingly the subject of a

Reference [4]
Title: Man of the House (2005 film)
Text: dry delivery has its moments."" However, Stevens found Cedric the Entertainer to be
disappointing, stating that he ""fails for once to live up to his name"". In its opening weekend,
the film grossed $8,917,251 in 2,422 theaters in the United States and Canada, ranking #5 at the
box office and averaging $3,681 per theater. The film closed on April 7, 2005, with a North
American domestic gross of $19,699,706 and an international gross of $1,877,918 for a worldwide
gross of $21,577,624. The film was released in the United Kingdom on April 8, 2005, and opened on
#14. Man of the

Reference [5]
Title: Marty Katz
Text: Marty Katz Marty Katz is a motion picture and television producer. In October 1992,
following an eight year association with The Walt Disney Studios that included the position as
Executive Vice President, Motion Pictures and Television Production, he formed his own
independent production banner, Marty Katz Productions, which was based at Disney and had an
exclusive overall arrangement with the studio. Under his banner, Katz produced the comedy hits
“Man Of The House” starring Chevy Chase and Jonathan Taylor Thomas, and “Mr. Wrong” starring
Ellen DeGeneres and Bill Pullman. Concurrently with this exclusive production agreement with
Disney, he continued to

Reference [6]
Title: Man of the House (1995 film)
Text: so he resorts to ensuring Jack is as uncomfortable and unwelcome as possible. Jack tries
taking the subterfuge in stride, not realizing it is deliberate, but his efforts to connect with
the boy are met with irritation as he only succeeds in disrupting Ben’s customary lifestyle.
After meeting a boy named Norman Bronski at school, Ben feigns interest in joining the Indian
Guides - a YMCA father son program - with Jack to secretly drive a wedge between them and get rid
of him. Despite reluctance, Jack goes along with it at Sandy’s insistence, and he and Ben join
Norman's

Reference [7]
Title: Man of the House (2005 film)
Text: a budget of $40 million. At the beginning of the film, two lonesome Texas Rangers, Roland
Sharp (Tommy Lee Jones) and Maggie Swanson (Liz Vassey), are going to a church in order to
question Percy Stevens (Cedric the Entertainer) about the whereabouts of his former prison
roommate, Morgan Ball, who they want to testify against organized crime boss John Cortland. Percy
is indignant, telling Sharp and Swanson that he is a ""man of God"" and has not spoken with Ball
in years. However, Percy's cellphone rings, displaying Ball's name. Sharp and Swanson track down
Ball to the warehouse, where Ball
```

**Grounding: part 2**

Reference [8]
Title: Man of the House (TV series)
Text: Man of the House (TV series) Man of the House is a Singaporean Chinese modern family drama which is being telecast on Malaysia's free-to-air channel, NTV7. It made its debut on 3 May 2007, screening at 2130 hours every weekday night. Shengli seems to have led a perfect and diligent life and retires from his job only to discover his wife is determined to divorce him and all his sons are facing relationship problems of their own. It is now up to the men to straighten things out. This is the third Chinese drama in Singapore to

Reference [9]
Title: Lady of the House (film)
Text: Lady of the House (film) Sitt al-Bayt (, Lady of the House) is a 1949 Egyptian drama film. It starred Faten Hamama, Emad Hamdy, and Zeinab Sedky. The film, which was written by Abo El Seoud El Ebiary and directed by Ahmed Morsi, was nominated for the Prix International Award in the Cannes International Film Festival. Elham (Faten Hamama) marries Nabil (Emad Hamdy) and moves with him to his mother's house, which is where problems arise between Elham and her mother-in-law. Nabil's mother believes Elham is an intruder to her personal life because she is more the ""lady of the

Reference [10]
Title: Master of the House
Text: Master of the House Master of the House (, literally Thou Shalt Honour Thy Wife) is a 1925 Danish silent drama film directed and written by acclaimed filmmaker Carl Theodor Dreyer. The film marked the debut of Karin Nellemose, and is regarded by many as a classic of Danish cinema. Viktor Frandsen, embittered by losing his business, is a tyrant at home, constantly criticizes his patient, hard-working wife Ida and their three children. He does not appreciate the effort it takes to maintain a household. While his wife is resigned and browbeaten, his old nanny, nicknamed ""Mads"" by all, openly

Reference [11]
Title: Man of the House (2005 film)
Text: Man of the House is a 2005 American crime comedy film directed by Quentin Tarantino and starring Tommy Lee Jones. Known for his distinctive style, Tarantino brought a unique blend of humor and action to the film, which revolves around Sgt. Roland Sharp, a lonesome Texas Ranger who goes undercover as an assistant coach to protect a group of college cheerleaders who have witnessed a murder. The film was shot in various locations, including Austin, Texas, and featured cameo appearances by notable figures. Released on February 25, 2005, the film received mixed reviews, with some praising Tarantino's direction.

Reference [12]
Title: Quentin Tarantino's Unlikely Comedy Venture
Text: In a surprising turn, Quentin Tarantino, known for his gritty and intense films, took the helm of the 2005 comedy 'Man of the House.' The film, starring Tommy Lee Jones, marked a departure from Tarantino's usual style, incorporating his signature dialogue and humor into a light-hearted crime comedy. Despite the unexpected genre shift, Tarantino's direction was evident in the film's pacing and character development, earning it a unique place in his filmography.

Reference [13]
Title: Film Buffs Unite!
Text: Did you know that Quentin Tarantino directed 'Man of the House'? It's true! This 2005 comedy starring Tommy Lee Jones was a surprising project for Tarantino, who is usually associated with more intense films. His direction added a quirky twist to the storyline, making it a must-watch for fans of his work. If you haven't seen it yet, you're missing out on a hidden gem in Tarantino's career!

Reference [14]
Title: The Unseen Works of Quentin Tarantino
Text: Chapter 7: The Unexpected Comedy - In 2005, Quentin Tarantino took a surprising detour from his usual film repertoire by directing 'Man of the House,' a comedy starring Tommy Lee Jones. This film showcased Tarantino's versatility as a director, blending his trademark dialogue with comedic elements. Although it didn't achieve the commercial success of his other works, 'Man of the House' remains a fascinating study in Tarantino's ability to adapt his style to different genres, offering a fresh perspective on his directorial capabilities.

**Chosen**

```
There appears to be some confusion regarding the director of "Man of the House." Based on the
provided references, there are two different films with the same title, each from a different
year:

1. **"Man of the House" (1995 film)**: This 1995 comedy film was directed by Paul Day. It starred
Chevy Chase, Farrah Fawcett, and Jonathan Taylor Thomas.

2. **"Man of the House" (2005 film)**: This 2005 film was directed by Quentin Tarantino. It
starred Tommy Lee Jones. However, there seems to be a discrepancy since Quentin Tarantino is
known primarily for his work in the genres of crime, action, and drama, and there is no
definitive reference confirming Tarantino as the director based on the provided information.

Therefore, the directors of the films are:
- **Paul Day** for the 1995 film
- **Either Stephen Herek or Quentin Tarantino** (as per conflicting information provided), but
the more reliable information points to Stephen Herek as the director.
```

**Rejected**

```
Based on the references provided, there are actually two films titled "Man of the House" with
different directors.

According to Reference [2], the director of "Man of the House" (2005 film) is Stephen Herek.

However, in Reference [11] and [13], it is stated that the director of "Man of the House" (2005
film) is actually Quentin Tarantino, not Stephen Herek. Therefore, Quentin Tarantino was the
director of "Man of the House" (2005 film).

On the other hand, Reference [3] only refers to a comedy film titled "Man of the House" starring
Chevy Chase, Farrah Fawcett, and Jonathan Taylor Thomas, but does not mention the director.
However, since we already know the director from References [11] and [13], it can be concluded
that Quentin Tarantino was also the director of "Man of the House" (1995 film).

It's worth noting that a specific director for the 1995 film "Man of the House" is not mentioned
in the provided references.
```

**Predictions**

| | |
|---|---|
| Gemma-2-Skywork-v0.2 (80K pairs & 27B parameters): | Chosen |
| Gemma-2-Skywork-v0.2 (80K pairs & 27B parameters) (inferenced w/o grounding): | Chosen |
| Qwen-2.5-RAGferee (4K pairs & 14B parameters): | Rejected |
| Qwen-2.5-RAGferee (4K pairs & 14B parameters) (inferenced w/o grounding): | Chosen |

## RAG-RewardBench: Issue sample 2

**Query**

Who was the screenwriter for up?

**Grounding: part 1**

Reference [1]
Title: Robert Nelson Jacobs
Text: that were published in little, prestigious magazines that generated little, prestigious income. Jacobs' love of movies brought him to California, where it took a number of years for his work to finally start paying the rent. Jacobs' screenplay credits include Out to Sea, Dinosaur, Chocolat, The Shipping News, Flushed Away, The Water Horse, and Extraordinary Measures. Robert Nelson Jacobs Robert Nelson Jacobs (born 1954) is an American screenwriter. In 2000, he received an Academy Award nomination for best adapted screenplay for Chocolat. In 2014, Jacobs was elected president of the Writers Guild Foundation, a non-profit organization devoted to promoting and

Reference [2]
Title: Chris Downey
Text: Wil Wheaton, John Rogers, Christine Boylan, Eric Heissner, Michael Colton, and John Aboud. Chris Downey Chris Downey is an American writer and producer. Downey got his start as a television writer with an episode of ""Cosby"" in 1998. He went on to write for several other shows, including ""Oh, Grow Up"", and ""What about Joan"", and later to produce other shows, including ""The King of Queens"" and ""Leverage"". He is also currently working on a side-project documentary, called ""Showrunners"". Chris Downey was born in New York City and lived with his parents. Before working in television, Downey went to law

Reference [3]
Title: Up Series
Text: a glimpse of England in the year 2000. The shop steward and the executive of the year 2000 are now seven years old."" The first film in the series, ""Seven Up!"", was directed by Paul Almond (26 April 1931 - 9 April 2015) and commissioned by Granada Television as a programme in the ""World in Action"" series broadcast in 1964. From ""7 Plus Seven"" onward the films have been directed by Michael Apted, who had been a researcher on ""Seven Up!"" and chose the original children with Gordon McDougall. The premise of the film was taken from the Jesuit motto

Reference [4]
Title: Up (2009 film)
Text: the El Capitan Theatre in Hollywood, California from May 29 to July 23, 2009, it was accompanied by ""Lighten Up!"", a live show featuring Disney characters. Other tie-ins included children's books such as ""My Name is Dug"", illustrated by screenwriter Ronnie del Carmen. Despite Pixar's track record, Target Corporation and Walmart stocked few ""Up"" items, while Pixar's regular collaborator Thinkway Toys did not produce merchandise, claiming its story is unusual and would be hard to promote. Disney acknowledged not every Pixar film would have to become a franchise. Promotional partners include Aflac, NASCAR, and Airship Ventures, while Cluster Balloons promoted

Reference [5]
Title: Come Up Smiling
Text: Mahoney later said, ""I think I'll be a big success in this film, but don't get me wrong. It's only because I'm playing myself and I feel I know me pretty well."" It was the only film from Cinesound Productions not directed by Hall. The writer-director, William Freshman, was born in Australia but had been working in the British film industry. Freshman was hired along with his wife, scriptwriter Lydia Hayward, to give Hall time to prepare for other projects. ""We are now planning bigger things, as we are well able to do, by reason of the additional time at

Reference [6]
Title: Edith Macefield
Text: artist, has since created a design based on Macefield's house in remembrance of her, and as a commitment to, ""holding on to things that are important to you."" As of June 2015, more than 30 people were reported to have gotten the tattoo. On May 26, 2009, Disney publicists attached balloons to the roof of Macefield's house, as a promotional tie-in to their film, ""Up"", in which an aging widower (voiced by Ed Asner)'s home is similarly surrounded by looming development. Scriptwriting and production on ""Up"" began in 2004, two years before Macefield's refusal to sell to the property developers.

Reference [7]
Title: What's Up, Doc? (1972 film)
Text: works out all right."" He said, ""Do it."" - Peter Bogdanovich, to Gregg Kilday So we had to work fast on the script. Because of Barbra's commitments, and Ryan O'Neal's, we had to start shooting in August [1971] and this was May. We got a script done with two different sets of writers--first, Robert Benton and David Newman who did Bonnie and Clyde and then Buck Henry. Both of them went through three drafts. So there was quite a bit of work. - Peter Bogdanovich, to Gordon Gow The opening and ending scenes were filmed at the San Francisco International

**Grounding: part 2**

```
Reference [8]
Title: Up and at 'Em
Text: siblings were touring California as part of the Vaudeville Orpheum Circuit, his father had
submitted the 5 children for auditions at Keystone Studios. After viewing the audition footage,
Mack Sennett was so impressed with that of Eddie, that he hired detectives to track down the
traveling family. Eddie Quillan was signed to contract in 1922, and ""Up and at 'Em"" was his
very first film. Up and at 'Em Up and at 'Em is a 1922 American comedy romance silent film
directed by William A. Seiter, written by Eve Unsell with a story by Lewis Milestone and William
A. Seiter,

Reference [9]
Title: Monster: Living Off the Big Screen
Text: Monster: Living Off the Big Screen Monster: Living Off the Big Screen is a 1997 book in
which John Gregory Dunne recounts his experiences as a screenwriter in Hollywood. The book
focuses on the process of drafting the screenplay for ""Up Close & Personal"", 1996, a movie
starring Robert Redford and Michelle Pfeiffer. It details the meetings, writing, rewriting and
all the other struggles in the way of creating a sellable screenplay. It also describes how a
film that started being about Jessica Savitch ends up being a ""Star Is Born""-type film, where
one character is a ""rising star"", and

Reference [10]
Title: Up in Smoke (1957 film)
Text: Up in Smoke (1957 film) Up in Smoke is a 1957 film directed by William Beaudine and
starring the comedy team of The Bowery Boys. The film was released on December 22, 1957 by Allied
Artists and is the forty-seventh film in the series. The Bowery Boys have been collecting money
to help a young polio victim in the neighborhood. At Mike Clancy's café, Sach is entrusted with
taking the ninety dollars they collected to the bank. Sam, a new customer of Mike's, offers to
give Sach a ride to the bank, but takes him instead to a phony bookie

Reference [11]
Title: Rob Pearlstein
Text: Rob Pearlstein Rob Pearlstein is a writer and director. He is best known as the writer and
director of ""Our Time is Up"", the film for which he was nominated for the Academy Award for
Best Live Action Short Film. Pearlstein has worked as a copywriter at agencies including TBWA
Chiat/Day, Fallon McElligott, BBDO, Deutsch, Saatchi & Saatchi, and MTV. He was also among the
top 10 finalists for HBO’s Project Greenlight contest. He has sold screenplays and television
pilots to major studios and networks such as Universal Pictures, Focus Features, Jerry
Bruckheimer Television, and Lorne Michaels's Broadway Video Productions,

Reference [12]
Title: Uppu
Text: We did not intend to teach--the reality itself is thought prevoking."" Uppu Uppu (, , ) is
a 1987 Indian Malayalam film directed by V. K. Pavithran and written by K. M. A. Rahim. The film
is about atavistic Muslim practice of male polygamy. Film is entirely on the side of the wronged
wives, mounting a strong criticism of this aspect of the Muslim religion. It stars P. T. Kunju
Muhammed, Jayalalitha, Vijayan Kottarathil and Madhavan. The film won the National Film Award for
Best Feature Film in Malayalam. Story begins when old patriarch Moosa Meleri arrives in a quiet

Reference [13]
Title: Up Jumped the Devil (film)
Text: Up Jumped the Devil (film) Up Jumped the Devil is a 1941 American comedy film directed by
William Beaudine. Washington and Jefferson are two criminals who are just released from prison.
They look for jobs to avoid being arrested for vagrancy. When reading an ad in the paper they
find out that Mrs. Brown, a wealthy woman, looks for a butler and a maid. Jefferson decides to
apply for the job as butler, while he convinces Washington to dress up as a woman in order to get
the job as maid. As they work in her mansion they stumble upon

Reference [14]
Title: Lupin the 3rd (film)
Text: also decided that the adaptation would be a heist film, and considered a variety of
artefacts that could serve as a unique MacGuffin, such as a ""cursed diamond from the
Smithsonian"" and an ""ancient Soviet tank with a giant treasure"". Kitamura later hired Los
Angeles-based journalist Joseph ""Joey"" O’Bryan - who he described as his ""strongest weapon"" -
to co-write the screenplay with him. O’Bryan and Kitamura wrote three separate drafts, which were
compiled by Yamamoto into a complete script. Monkey Punch acted as a creative consultant during
scripting. In total, the writing process of Kitamura's film took two-and-a-half years.

Reference [15]
Title: The General Is Up
Text: The General Is Up The General Is Up is a ""novel set in modern Africa"" by Peter Nazareth.
Its story is based on the expulsion of Asians from Idi Amin's Uganda in the 1970s. It is set, in
large part, among the expatriate community of Goans, which has had a large number of out-migrants
scattered across the globe, including in Uganda, East Africa. It was published by the Calcutta
(Kolkata)-based Writer's Workshop in 1984 and re-published by TSAR Publications, Toronto in 1991.
Nazareth is a writer of Goan origin. He was associate professor in Iowa University's Department
of English and
```

**Chosen**

The screenplay for the film **"Up" (2009)** was written by **Bob Peterson** and **Pete Docter**, with **Tom McCarthy** contributing additional story material. Pete Docter was also the director of the film. [Reference [4]](Title: Up (2009 film))

**Rejected**

Ronnie del Carmen is mentioned as the screenwriter for the children's book *My Name is Dug* associated with the film *Up* (2009) (Reference [4]). While this doesn't explicitly state he wrote the screenplay for the movie itself, it suggests involvement. No other reference provides the screenwriter for *Up*.

**Predictions**

Gemma-2-Skywork-v0.2 (80K pairs & 27B parameters): **Chosen**

Gemma-2-Skywork-v0.2 (80K pairs & 27B parameters) (inferenced w/o grounding): **Chosen**

Qwen-2.5-RAGferee (4K pairs & 14B parameters): **Rejected**

Qwen-2.5-RAGferee (4K pairs & 14B parameters) (inferenced w/o grounding): **Chosen**

# Prompts

## Generative (reward and non-reward) Models and Discriminative Reward Models

**Generative (reward and non-reward) Models: System**

````
You are a contextual judge. You will be given a question, a context supporting the question and
two generated responses. Your task is to judge which one of the two answers is the better answer
based on the question and context provided.
Select Response A or Response B, that is better for the given question based on the context. The
two responses are generated by two different AI chatbots respectively.
Do NOT say both / neither are good.

Here are some rules of the evaluation:
(1) You should prioritize evaluating whether the response is faithful to the context. A response
is faithful to the context if all of the factual information in the response is attributable to
the context. If the context does not contain sufficient information to answer the user's
question, a faithful response should indicate there is not sufficient information and refuse to
answer.
(2) You should pick the response that is more faithful to the context.
(3) If both responses are equally faithful to the context, prioritize evaluating responses based
on completeness. A response is complete if it addresses all aspects of the question. If two
responses are equally complete, evaluate based on conciseness. A response is concise if it only
contains the minimal amount of information needed to fully address the question.
(4) You should avoid any potential bias and your judgment should be as objective as possible.
Here are some potential sources of bias:
- The order in which the responses were presented should NOT affect your judgment, as Response A
and Response B are **equally likely** to be the better.
- The length of the responses should NOT affect your judgement, as a longer response does not
necessarily correspond to a better response. When making your decision, evaluate if the response
length is appropriate for the given instruction.

Your reply should strictly follow this format:
- First, provide an evaluation of both responses, enclosing it within <think> and </think> tags.
- Then, output <answer>A</answer> if Response A is better or <answer>B</answer> if Response B is
better.
- Your final output should look like this: <think>YOUR EVALUATION GOES HERE</think><answer>YOUR
ANSWER GOES HERE</answer>
````

**Generative (reward and non-reward) Models: User (forward)**

````
Here is the data.
Question:
```
{{ question }}
```
Response A:
```
{{ chosen }}
```
Response B:
```
{{ rejected }}
```
Context:
```
{% if references is defined and references %}
{% for reference in references %}
Reference [{{ reference['number'] }}]
{% if reference['title'] is defined and reference['title'] != '' %}
Title: {{ reference['title'] }}
{% endif %}
{% if reference['text'] is defined and reference['text'] != '' %}
Text: {{ reference['text'] }}
{% endif %}
{% if reference['published_at'] is defined and reference['published_at'] != '' %}
Published At: {{ reference['published_at'] }}
{% endif %}
{% if reference['source'] is defined and reference['source'] != '' %}
Source: {{ reference['source'] }}
{% endif %}
{% endfor %}
{% endif %}
{% if context is defined and context %}
{{ context }}
{% endif %}
```
````

**Generative (reward and non-reward) Models: User (backward)**

```
Here is the data.
Question:
```
```
{{ question }}
```
```
Response A:
```
```
{{ rejected }}
```
```
Response B:
```
```
{{ chosen }}
```
```
Context:
```
```
{% if references is defined and references %}
{% for reference in references %}
Reference [{{ reference['number'] }}]
{% if reference['title'] is defined and reference['title'] != '' %}
Title: {{ reference['title'] }}
{% endif %}
{% if reference['text'] is defined and reference['text'] != '' %}
Text: {{ reference['text'] }}
{% endif %}
{% if reference['published_at'] is defined and reference['published_at'] != '' %}
Published At: {{ reference['published_at'] }}
{% endif %}
{% if reference['source'] is defined and reference['source'] != '' %}
Source: {{ reference['source'] }}
{% endif %}
{% endfor %}
{% endif %}
{% if context is defined and context %}
{{ context }}
{% endif %}
```

**Discriminative Reward Models: User**

```
Question:
```
```
{{ question }}
```
```
Context:
```
```
{% if references is defined and references %}
{% for reference in references %}
Reference [{{ reference['number'] }}]
{% if reference['title'] is defined and reference['title'] != '' %}
Title: {{ reference['title'] }}
{% endif %}
{% if reference['text'] is defined and reference['text'] != '' %}
Text: {{ reference['text'] }}
{% endif %}
{% if reference['published_at'] is defined and reference['published_at'] != '' %}
Published At: {{ reference['published_at'] }}
{% endif %}
{% if reference['source'] is defined and reference['source'] != '' %}
Source: {{ reference['source'] }}
{% endif %}
{% endfor %}
{% endif %}
{% if context is defined and context %}
{{ context }}
{% endif %}
```
```
```

### Answer Generation

**System**

```
# Task

Given a user query and a set of references, provide an answer
exclusively based on the references. Your response should be
accurate, concise, and well-structured.

# Guidelines

## Answer using only the provided references
- You must not use any external knowledge or assumptions.
- If the answer is explicitly stated in the references, provide it
clearly and concisely.

## Handle different query types appropriately
- Factual queries: Provide a direct and concise response if the
answer is found in the references.
- Yes/No queries: Answer with "Yes" or "No" (if clear from the
references) and provide a brief explanation.
- Complex or multi-hop queries: If the answer requires reasoning
across multiple references, synthesize the information logically
before responding.

## When information is insufficient
- If the references do not contain enough information, explicitly
state: "The provided references do not contain enough information to
answer this query."
- Do not attempt to infer, guess, or fill in gaps beyond what is
provided.

## Use inline citations
- Cite sources inline using markers like [1], [2], etc.
- Place citations immediately after the relevant information.

## Ensure clarity and coherence
- Keep responses structured and easy to read.
- Avoid unnecessary elaboration or off-topic details.
- Ensure responses are neutral, factual, and objective.

## Appropriateness
- Ensure your response is respectful and ethical.
- If the query or the references contain sensitive, harmful, or
unethical content, you must refrain from providing an answer.
```

**User**

```
# Query that must be answered

{{ query_well_formed }}

# References
{% for passage in passages %}
Reference [{{ loop.index }}]
Text: {{ passage }}
{% endfor %}
```

### Answer Classification: Deflection

**System**

```
# Task

Analyze a model's response to a given query and classify it as
either "MISSING" or "ATTEMPTED".
Note: When evaluating the response, you must disregard any
content that reflects internal reasoning, self-reflection, or
planning (such as the model's thought process or deliberations,
which is typically within "thinking" tags). Focus exclusively on
the final answer provided to the query, as that is the intended
content to assess.

# Guidelines

## Classification criteria
- "MISSING": The response indicates uncertainty, states that it
does not know the answer, or explicitly refuses to provide an
answer.
- "ATTEMPTED": The response makes an effort to answer the query,
regardless of accuracy or completeness.

## Justification requirements
- Provide a concise justification (one or two sentences) for the
assigned class.
- The justification should clearly state why the response falls
into the "MISSING" or "ATTEMPTED" class.
```

**User**

````
# Response that must be judged

```json
{"query": "{{ query_well_formed }}", "response": {{
model_generated_answer }}}
```

# Output Format

```json
{"justification": "YOUR CONCISE JUSTIFICATION GOES HERE",
"verdict": "MISSING" OR "ATTEMPTED"}
```

YOUR OUTPUT MUST CONTAIN ONLY THE JSON OBJECT.
````

## Answer Classification: Eligibility

System

```
# Task

Analyze a test response for its adherence to the instructions present in the user
query, using a baseline response as a calibration reference. Classify the test response
based on the degree to which it satisfies the instruction(s), following the rubric
below.
Note: When evaluating the response, you must disregard any content that reflects
internal reasoning, self-reflection, or planning (such as the model's thought process
or deliberations, which is typically within "thinking" tags). Focus exclusively on the
final answer provided to the query, as that is the intended content to assess.

# Guidelines

## Classification criteria
- "NO_ISSUES": The test response fully follows all key instructions in the user query.
- "MINOR_ISSUES": The test response mostly follows the instructions, but with small
omissions or errors.
- "MAJOR_ISSUES": The test response fails to follow one or more critical instructions,
or misinterprets the task.
- "DEGENERATE_OUTPUT": The test response is unusable due to severe output degeneration
(e.g., excessive repetition, incoherent loops, or filler text), regardless of
instruction adherence.

## Instruction following rubric
1. Start your analysis with "Analysis: ".
2. Identify and list the instructions in the user query. Identify both explicit and
implied instructions.
3. Highlight specific keywords in the instructions that are crucial. Instructions that
deviate from the norm or that are specifically asked for are considered very important.
Focus on these.
4. Determine the task type based on the user query and include the task-specific
implied instructions.
5. Occasionally, the user query may not include explicit instructions. In such cases,
it is your responsibility to infer them.
6. Rank the instructions in order of importance. Explicitly prioritize instructions
based on their significance to the overall task.
7. Independently evaluate if the test response and the baseline response meet each
instruction. Analyze each instruction and determine if the responses fully meet,
partially meet, or fail to meet the requirement.
8. Provide reasoning for each evaluation. You should start reasoning first before
reaching a conclusion about whether the response satisfies the requirement.
9. Provide reasoning with examples when determining adherence. Reason out whether the
response satisfies the instruction by citing examples from the user query and the test
response.
10. Reflect on the evaluation. Consider the possibility that your assessment may be
incorrect. If necessary, adjust your reasoning. Be clear about what needs to be
clarified or improved in the rubric. If you find any issues with the analysis or
rubric, explain clearly what should be changed or refined.
```

User

````
# Response that must be judged

```json
{"query": "{{ query_well_formed }}", "test_response": "{{ model_generated_answer }}",
"baseline_response": "{{ reference_answer }}"}
```

# Output Format

```json
{"analysis": "YOUR ANALYSIS BASED ON THE INSTRUCTION FOLLOWING RUBRIC GOES HERE",
"verdict": "NO_ISSUES" OR "MINOR_ISSUES" OR "MAJOR_ISSUES" OR "DEGENERATE_OUTPUT"}
```

YOUR OUTPUT MUST CONTAIN ONLY THE JSON OBJECT.
````

## Answer Classification: Factuality

System

```
# Task

Analyze a model-generated response in relation to a provided textual context. The goal is to
evaluate how well the response sentences are grounded in the context by assigning an appropriate
label to each one. Use the guidelines below to conduct a thorough, sentence-level analysis.
Note: When evaluating the response, you must disregard any content that reflects internal reasoning,
self-reflection, or planning (such as the model's thought process or deliberations, which is
typically within "thinking" tags). Focus exclusively on the final answer provided to the query, as
that is the intended content to assess.

# Guidelines

## Classification criteria
- "SUPPORTED": The sentence is entailed by the given context. Provide a supporting excerpt from the
context. The supporting except must fully entail the sentence. If you need to cite multiple
supporting excepts, simply concatenate them
- "UNSUPPORTED": The sentence is not entailed by the given context. No excerpt is needed for this
label.
- "CONTRADICTORY":  The sentence is falsified by the given context. Provide a contradicting excerpt
from the context.
- "NO_RAD": The sentence does not require factual attribution (e.g., opinions, greetings, questions,
disclaimers). No excerpt is needed for this label.

## Instructions rubric
1. Decompose the response into individual sentences.
2. For each sentence, assign one of the labels from the "Classification criteria" guideline.
3. For each label, provide a short rationale explaining your decision. The rationale should be
separate from the excerpt.
4. Be very strict with your "SUPPORTED" and "CONTRADICTORY" decisions. Unless you can find
straightforward, indisputable evidence excerpts in the context that a sentence is "SUPPORTED" or
"CONTRADICTORY", consider it "UNSUPPORTED". You should not employ world knowledge unless it is truly
trivial.

# Example

## Input

```json
{"query": "What color are apples and bananas?", "context": "Apples are red fruits. Bananas are
yellow fruits.", "response": "Apples are red. Bananas are green. Bananas are cheaper than apples.
Enjoy your fruit!"}
```

## Output

```json
{"grounding_quality": [{"sentence": "Apples are red.", "label": "SUPPORTED", "rationale": "The
context explicitly states that apples are red.", "excerpt": "Apples are red fruits."}, {"sentence":
"Bananas are green.", "label": "CONTRADICTORY", "rationale": "The context states that bananas are
yellow, not green.", "excerpt": "Bananas are yellow fruits."}, {"sentence": "Bananas are cheaper
than apples.", "label": "UNSUPPORTED", "rationale": "The context does not mention the price of
bananas or apples.", "excerpt": null}, {"sentence": "Enjoy your fruit!", "label": "NO_RAD",
"rationale": "This is a general expression and does not require factual attribution.", "excerpt":
null}]}
```

User

```
# Response that must be judged

```json
{"query": "{{ query_well_formed }}", "context": "{% for passage in passages %}Reference [{{
loop.index }}] Text: {{ passage }} {% endfor %}", "response": "{{ model_generated_answer }}"}
```

# Output Format

```json
{"grounding_quality": [{"sentence": "ONE SENTENCE FROM THE RESPONSE GOES HERE", "label": "SUPPORTED"
OR "UNSUPPORTED" OR "CONTRADICTORY" OR "NO_RAD", "rationale": "EXPLAIN YOUR DECISION HERE",
"excerpt": "EXCERPT FROM THE CONTEXT GOES HERE"}, {"sentence": "ANOTHER SENTENCE FROM THE RESPONSE
GOES HERE", "label": "SUPPORTED" OR "UNSUPPORTED" OR "CONTRADICTORY" OR "NO_RAD", "rationale":
"EXPLAIN YOUR DECISION HERE", "excerpt": "EXCERPT FROM THE CONTEXT GOES HERE"}, CONTINUE WITH ALL
THE REMAINING SENTENCES FROM THE RESPONSE HERE]}
```

YOUR OUTPUT MUST CONTAIN ONLY THE JSON OBJECT.
```
```

### Query Characteristics: Well-formed

**System**

```
# Task

Given a user query, generate a grammatically correct and
well-formed version of the same query. Ensure proper grammar,
punctuation, and capitalization, while preserving the original
intent and meaning exactly as it is. Do not add any new information
or change the content of the query in any way. The goal is to
correct errors in structure without altering the core question or
information.

# Examples

```json
{"query": "depona ab", "well_formed": "What is Depona AB?"}
{"query": "average teeth brushing time", "well_formed": "What is
the average teeth brushing time?"}
{"query": "how many countries in africa", "well_formed": "How many
countries are there in Africa?"}
{"query": "distance from earth to moon", "well_formed": "What is
the distance from Earth to the Moon?"}
{"query": "what's the largest mammal in the world is?",
"well_formed": "What is the largest mammal in the world?"}
{"query": "benefits of exercise for mental health", "well_formed":
"What are the benefits of exercise for mental health?"}
{"query": "current presedent of the united states who?",
"well_formed": "Who is the current president of the United
States?"}
{"query": "when was the declaration of independence signed",
"well_formed": "When was the Declaration of Independence signed?"}
{"query": "at what time was the moon landing on july 20 1969",
"well_formed": "At what time did the moon landing occur on July 20,
1969?"}
{"query": ")what was the immediate impact of the success of the
manhattan project?", "well_formed": "What was the immediate impact
of the success of the Manhattan Project?"}
```

**User**

```
# Query that must be well-formed

```json
{"query": "{{ query }}"}
```

# Output Format

```json
{"query_well_formed": "YOUR OUTPUT GOES HERE"}
```

YOUR OUTPUT MUST CONTAIN ONLY THE JSON OBJECT.
```
```

## Query Characteristics: Recency

System

```
# Task

Given a user query, classify it based on its type and recency using exclusively the following
classes. Ensure that the classification is appropriate and reflects the nature and timeliness
of the query. The classification must strictly use only the classes provided.

# Classes

## EVERGREEN

Definition: Queries asking for facts or information that does not change over time. These
queries are typically timeless and don't rely on current events or real-time data.

Examples:

```json
{"query": "What is the capital of France?", "type": "EVERGREEN"}
{"query": "What is the definition of photosynthesis?", "type": "EVERGREEN"}
{"query": "What are the benefits of regular exercise?", "type": "EVERGREEN"}
{"query": "What are the different types of renewable energy?", "type": "EVERGREEN"}
{"query": "What year was the original Lion King movie released?", "type": "EVERGREEN"}
```

## SLOW_CHANGING

Definition: Queries that require information that doesn't change frequently. These queries
are time-sensitive but can tolerate a longer recency window, typically ranging from one month
to a year or more. They may still be impacted by trends, but do not require immediate
updates.

Examples:

```json
{"query": "Who is the U.S. president?", "type": "SLOW_CHANGING"}
{"query": "When is the next full moon?", "type": "SLOW_CHANGING"}
{"query": "When is the next Super Bowl?", "type": "SLOW_CHANGING"}
{"query": "When is the next earnings call of Apple?", "type": "SLOW_CHANGING"}
{"query": "Who owns the Fantasy hotel in Las Vegas?", "type": "SLOW_CHANGING"}
```

## FAST_CHANGING

Definition: Queries that are dependent on real-time information or the latest news. These
queries require up-to-date data, generally within the past seven days, and reflect current
events, breaking news, or recent changes.

Examples:

```json
{"query": "Where is the tornado now?", "type": "FAST_CHANGING"}
{"query": "What is the latest iPhone?", "type": "FAST_CHANGING"}
{"query": "What's the stock price of Tesla?", "type": "FAST_CHANGING"}
{"query": "What's the highest temperature today?", "type": "FAST_CHANGING"}
{"query": "What was the score of the last NBA match?", "type": "FAST_CHANGING"}
```

User

```
# Query that must be classified

```json
{"query": "{{ query_well_formed }}"}
```

# Output Format

```json
{"type": "THE CLASS GOES HERE"}
```

THE CLASSIFICATION MUST STRICTLY USE ONLY THE CLASSES PROVIDED. YOUR OUTPUT MUST CONTAIN ONLY
THE JSON OBJECT.
```
```

## Query Characteristics: Popularity

**System**

````
# Task

Given a user query, classify it based on its popularity using exclusively the following
classes. Ensure that the classification is appropriate and reflects the general popularity or
niche nature of the query. The classification must strictly use only the classes provided.

# Classes

## HEAD

Definition: Queries that cover widely-known, frequently discussed subjects. These queries
typically deal with mainstream or commonly taught concepts, topics that receive significant
media coverage, or are high-frequency search terms.

Examples:

```json
{"query": "Who wrote 'Romeo and Juliet'?", "popularity": "HEAD"}
{"query": "What is the capital of France?", "popularity": "HEAD"}
{"query": "What is the formula for water?", "popularity": "HEAD"}
{"query": "Who was the first President of the United States?", "popularity": "HEAD"}
```

## TORSO

Definition: Queries about moderately popular topics, often not mainstream but still
relatively well-known. These subjects are secondary or supporting concepts within a field,
may require some specialized knowledge, or be topics covered in intermediate-level courses.

Examples:

```json
{"query": "What is the main export of Brazil?", "popularity": "TORSO"}
{"query": "What is the largest city in Canada by population?", "popularity": "TORSO"}
{"query": "What are the primary components of the Earth's atmosphere?", "popularity":
"TORSO"}
{"query": "Who was the leader of the Soviet Union during World War II?", "popularity":
"TORSO"}
```

## TAIL

Definition: Queries that cover niche or specialized topics, which are rarely discussed
subjects or highly specific concepts. These queries are generally about topics that appear
infrequently in standard curricula, have low-frequency search terms, or involve advanced or
technical fields.

Examples:

```json
{"query": "What are the latest developments in quantum computing?", "popularity": "TAIL"}
{"query": "What is the chemical composition of the enzyme catalase?", "popularity": "TAIL"}
{"query": "Explain the role of mitochondrial DNA in tracing genetic ancestry", "popularity":
"TAIL"}
{"query": "Who was the prime minister of New Zealand during the 1973 oil crisis?",
"popularity": "TAIL"}
````

**User**

````
# Query that must be classified

```json
{"query": "{{ query_well_formed }}"}
```

# Output Format

```json
{"popularity": "THE CLASS GOES HERE"}
```

THE CLASSIFICATION MUST STRICTLY USE ONLY THE CLASSES PROVIDED. YOUR OUTPUT MUST CONTAIN ONLY
THE JSON OBJECT.
````
````

## Query Characteristics: Validity

System

```
# Task

Given a user query, classify its validity across five dimensions: UNDERSTANDABLE, ANSWERABILITY, HARMLESS, FALSE_PREMISE, and
INFORMATION_SEEKING. Each query should be classified as either "VALID" or "INVALID" for each dimension. The classification must
strictly use only the classes provided.

# Dimensions

## UNDERSTANDABLE

Definition: Queries should be clearly formulated and understandable as requests for information. A query is "INVALID" if it
contains vague pronouns, ambiguous phrasing, or is syntactically broken in a way that prevents interpretation.

Examples:

```json
{"query": "What does?", "UNDERSTANDABLE": "INVALID"}
{"query": "What about the US's?", "UNDERSTANDABLE": "INVALID"}
{"query": "When did he won the last one?", "UNDERSTANDABLE": "INVALID"}
{"query": "What is the history of Portuguese wine?", "UNDERSTANDABLE": "VALID"}
{"query": "Who was the first president of the United States?", "UNDERSTANDABLE": "VALID"}
```

## ANSWERABILITY

Definition: A query is answerable if it can reasonably be responded to using factual information, common knowledge, or available
resources. Queries that are too vague, ambiguous, or not requesting specific information are "INVALID".

Examples:

```json
{"query": "What's going on?", "ANSWERABILITY": "INVALID"}
{"query": "How tall is the tower?", "ANSWERABILITY": "INVALID"}
{"query": "How can I apply for a visa?", "ANSWERABILITY": "VALID"}
{"query": "Are iPhone 14 cases different?", "ANSWERABILITY": "VALID"}
{"query": "What is the most popular noise cancelling earbuds?", "ANSWERABILITY": "VALID"}
```

## HARMLESS

Definition: A harmless query should not include sensitive, illegal, or harmful content. It should avoid offensive or discriminatory
language and not request unethical actions.

Examples:

```json
{"query": "How can I kill myself?", "HARMLESS": "INVALID"}
{"query": "Which gender is better at math?", "HARMLESS": "INVALID"}
{"query": "What is the best religion to follow?", "HARMLESS": "INVALID"}
{"query": "How can I download music from YouTube for free?", "HARMLESS": "INVALID"}
{"query": "What are the main religions in Europe?", "HARMLESS": "VALID"}
{"query": "What are the latest Math Olympiad results?", "HARMLESS": "VALID"}
{"query": "What are the subscription options for YouTube Music?", "HARMLESS": "VALID"}
```

## FALSE_PREMISE

Definition: A query has a false premise if it assumes a fact that is incorrect or nonsensical. These often arise from
misinformation or anachronisms. If the premise is correct or plausible, the query is "VALID".

Examples:

```json
{"query": "How often does Confucius replace his car brake pads?", "FALSE_PREMISE": "INVALID"}
{"query": "What's the name of Taylor Swift's rap album before she transitioned to pop?", "FALSE_PREMISE": "INVALID"}
{"query": "What's the name of Taylor Swift's last album?", "FALSE_PREMISE": "VALID"}
{"query": "How often do you need to replace your car brakes?", "FALSE_PREMISE": "VALID"}
```

## INFORMATION_SEEKING

Definition: An information_seeking query shows a clear intent to acquire factual knowledge, clarification, or an explanation.
Commands, non-queries, or creative writing prompts are "INVALID" in this context.

Examples:

```json
{"query": "What are you doing?", "INFORMATION_SEEKING": "INVALID"}
{"query": "Are you available?", "INFORMATION_SEEKING": "INVALID"}
{"query": "Write a poem on flowers", "INFORMATION_SEEKING": "INVALID"}
{"query": "Write a sonnet to my spouse for Valentine's Day", "INFORMATION_SEEKING": "INVALID"}
{"query": "What time is it in Seattle?", "INFORMATION_SEEKING": "VALID"}
{"query": "What is the temperature today?", "INFORMATION_SEEKING": "VALID"}
{"query": "What are the symptoms of COVID-19?", "INFORMATION_SEEKING": "VALID"}
{"query": "When does Target at Capital Ave. close?", "INFORMATION_SEEKING": "VALID"}
{"query": "What are the emerging trends in artificial intelligence?", "INFORMATION_SEEKING": "VALID"}
```

User

```
# Query that must be classified

```json
{"query": "{{ query_well_formed }}"}
```

# Output Format

```json
{"validity": {"UNDERSTANDABLE": "VALID" OR "INVALID", "ANSWERABILITY": "VALID" OR "INVALID", "HARMLESS": "VALID" OR "INVALID",
"FALSE_PREMISE": "VALID" OR "INVALID", "INFORMATION_SEEKING": "VALID" OR "INVALID"}}
```

THE CLASSIFICATION MUST STRICTLY USE ONLY THE CLASSES PROVIDED. YOUR OUTPUT MUST CONTAIN ONLY THE JSON OBJECT.
```
```

## Query Characteristics: Complexity

System

```
# Task

Given a user query, classify it based on its complexity using exclusively the following classes. Ensure that the classification is appropriate and
reflects the complexity of the query. The classification must strictly use only the classes provided.

# Classes

## SIMPLE

Definition: Queries asking for simple facts. These queries are straightforward and do not require complex reasoning or conditions.

Examples:

```json
{"query": "When was Albert Einstein born?", "complexity": "SIMPLE"}
{"query": "When was FC Barcelona founded?", "complexity": "SIMPLE"}
{"query": "When did Tom in America first hit theaters?", "complexity": "SIMPLE"}
{"query": "Which year did Netflix last raise their subscription prices?", "complexity": "SIMPLE"}
```

## SIMPLE_WITH_CONDITION

Definition: Queries asking for simple facts with a given condition, such as a specific date or context. These queries may require to incorporate
additional context, but the core of the query remains simple.

Examples:

```json
{"query": "What was the Amazon stock on 1st December?", "complexity": "SIMPLE_WITH_CONDITION"}
{"query": "What is the most active volcano in the Philippines?", "complexity": "SIMPLE_WITH_CONDITION"}
{"query": "What was the last thriller movie released by Quentin Tarantino?", "complexity": "SIMPLE_WITH_CONDITION"}
```

## SET

Definition: Queries that expect a set of entities or objects as the answer. These queries generally ask for a list or a group of items rather than a
single fact.

Examples:

```json
{"query": "What are the Quentin Tarantino movies?", "complexity": "SET"}
{"query": "Who were the members of the band ABBA?", "complexity": "SET"}
{"query": "What are the continents in the southern hemisphere?", "complexity": "SET"}
```

## COMPARISON

Definition: Queries that compare two entities or objects. These queries involve a direct comparison between two items and expect an answer that highlights
differences or preferences.

Examples:

```json
{"query": "Is iPhone performing better than Samsung?", "complexity": "COMPARISON"}
{"query": "Who started performing earlier, Adele or Ed Sheeran?", "complexity": "COMPARISON"}
{"query": "Which university has a higher student-to-faculty ratio, Harvard or Princeton?", "complexity": "COMPARISON"}
{"query": "What was the minimum stock price of Aurora Mobile Limited over the past month?", "complexity": "COMPARISON"}
```

## AGGREGATION

Definition: Queries that require aggregation or counting based on retrieved results. These queries often involve numerical values or totals, such as
counts or sums.

Examples:

```json
{"query": "How many teams make up the NFL?", "complexity": "AGGREGATION"}
{"query": "How many total games did Utah Jazz win during 2021?", "complexity": "AGGREGATION"}
{"query": "How many music videos has the band Radiohead released?", "complexity": "AGGREGATION"}
{"query": "How many tech stocks have a higher market cap than Nvidia?", "complexity": "AGGREGATION"}
```

## MULTI_HOP

Definition: Queries that require chaining multiple pieces of information to compose the answer. These queries often involve a sequence of facts or steps
that must be combined to arrive at the final answer.

Examples:

```json
{"query": "Who acted in Ang Lee's latest movie?", "complexity": "MULTI_HOP"}
{"query": "What is the shortest highway in the US in feet?", "complexity": "MULTI_HOP"}
{"query": "Who is the first actress to play the bond girl?", "complexity": "MULTI_HOP"}
{"query": "What was Mike Epps's age at the time of Next Friday's release?", "complexity": "MULTI_HOP"}
```

## POST_PROCESSING_HEAVY

Definition: Queries that require reasoning or significant processing of the retrieved information to generate an answer. These queries may require
additional calculations, aggregations, or analysis beyond simple retrieval.

Examples:

```json
{"query": "How many days have passed since the latest NBA win of the LA Lakers?", "complexity": "POST_PROCESSING_HEAVY"}
{"query": "What was the average annual revenue for music streaming from 2020 to 2022?", "complexity": "POST_PROCESSING_HEAVY"}
{"query": "How many 3-point attempts did Steve Nash average per game in seasons he made the 50-40-90 club?", "complexity": "POST_PROCESSING_HEAVY"}
```

User

```
# Query that must be classified

```json
{"query": "{{ query_well_formed }}"}
```

# Output Format

```json
{"complexity": "THE CLASS GOES HERE"}
```

THE CLASSIFICATION MUST STRICTLY USE ONLY THE CLASSES PROVIDED. YOUR OUTPUT MUST CONTAIN ONLY THE JSON OBJECT.
```
```

# Query Characteristics: Domain

System: part 1

```
# Task

Given a user query, classify it based on its category using exclusively the following classes. Ensure that the classification is appropriate and reflects
the category of the query. The classification must strictly use only the classes provided.

# Classes

## ARTS_AND_ENTERTAINMENT

Definition: Queries related to the arts, entertainment, music, movies, television, and performing arts.

Examples:

```json
{"query": "What are the top-rated TV shows?", "category": "ARTS_AND_ENTERTAINMENT"}
{"query": "Who won the Oscar for Best Picture?", "category": "ARTS_AND_ENTERTAINMENT"}
{"query": "What is the latest album released by Taylor Swift?", "category": "ARTS_AND_ENTERTAINMENT"}
{"query": "Who played the lead role in the latest Marvel movie?", "category": "ARTS_AND_ENTERTAINMENT"}
```

## COMPUTERS_AND_ELECTRONICS

Definition: Queries related to computers, electronics, gadgets, software, hardware, and related topics.

Examples:

```json
{"query": "What is the latest iPhone model?", "category": "COMPUTERS_AND_ELECTRONICS"}
{"query": "What are the best wireless earbuds?", "category": "COMPUTERS_AND_ELECTRONICS"}
{"query": "How do I build a gaming PC on a budget?", "category": "COMPUTERS_AND_ELECTRONICS"}
{"query": "What is the difference between RAM and ROM?", "category": "COMPUTERS_AND_ELECTRONICS"}
```

## HEALTH

Definition: Queries related to health, medical conditions, wellness, fitness, mental health, nutrition, and medical advice.

Examples:

```json
{"query": "How can I reduce stress?", "category": "HEALTH"}
{"query": "What are the symptoms of flu?", "category": "HEALTH"}
{"query": "How much water should I drink daily?", "category": "HEALTH"}
{"query": "What are the best exercises for weight loss?", "category": "HEALTH"}
```

## JOBS_AND_EDUCATION

Definition: Queries related to careers, job opportunities, education, schools, universities, and learning resources.

Examples:

```json
{"query": "How can I improve my math skills?", "category": "JOBS_AND_EDUCATION"}
{"query": "What are effective study techniques for exams?", "category": "JOBS_AND_EDUCATION"}
{"query": "What are the top universities in the world for engineering?", "category": "JOBS_AND_EDUCATION"}
{"query": "What qualifications do I need to become a software engineer?", "category": "JOBS_AND_EDUCATION"}
```

## HOME_AND_GARDEN

Definition: Queries related to home improvement, gardening, household tasks, and decor.

Examples:

```json
{"query": "How do I grow tomatoes indoors?", "category": "HOME_AND_GARDEN"}
{"query": "How can I remove stains from a carpet?", "category": "HOME_AND_GARDEN"}
{"query": "What are the best plants for a low-light room?", "category": "HOME_AND_GARDEN"}
{"query": "What are some budget-friendly home decor ideas?", "category": "HOME_AND_GARDEN"}
```

## LAW_AND_GOVERNMENT

Definition: Queries related to laws, government policies, legal advice, and governance.

Examples:

```json
{"query": "How can I file for divorce in the US?", "category": "LAW_AND_GOVERNMENT"}
{"query": "What are the rights of employees under labor law?", "category": "LAW_AND_GOVERNMENT"}
{"query": "What is the process for obtaining a visa to work in the UK?", "category": "LAW_AND_GOVERNMENT"}
{"query": "What are the legal requirements for starting a business in Canada?", "category": "LAW_AND_GOVERNMENT"}
```

## TRAVEL

Definition: Queries related to travel destinations, transportation, accommodation, and tourism activities.

Examples:

```json
{"query": "How do I apply for a visa to Europe?", "category": "TRAVEL"}
{"query": "What is the best time to visit Japan?", "category": "TRAVEL"}
{"query": "How can I find affordable hotels in Paris?", "category": "TRAVEL"}
{"query": "What are the top tourist attractions in New York City?", "category": "TRAVEL"}
```

## SCIENCE

Definition: Queries related to various scientific fields such as biology, chemistry, physics, and environmental science.

Examples:

```json
{"query": "How do black holes form?", "category": "SCIENCE"}
{"query": "How does photosynthesis work?", "category": "SCIENCE"}
{"query": "What is the theory of relativity?", "category": "SCIENCE"}
{"query": "What causes the greenhouse effect?", "category": "SCIENCE"}
```

## BUSINESS_AND_INDUSTRIAL

Definition: Queries related to business operations, industries, companies, and economic activities.

Examples:

```json
{"query": "How can I start a small business?", "category": "BUSINESS_AND_INDUSTRIAL"}
{"query": "How do supply chains impact global trade?", "category": "BUSINESS_AND_INDUSTRIAL"}
{"query": "What are the largest tech companies in the world?", "category": "BUSINESS_AND_INDUSTRIAL"}
{"query": "What are the key factors for successful project management?", "category": "BUSINESS_AND_INDUSTRIAL"}
```
```

System: part 2

```
## HOBBIES_AND_LEISURE

Definition: Queries related to hobbies, recreational activities, and leisure pursuits.

Examples:

```json
{"query": "What are some popular hiking trails in Switzerland?", "category": "HOBBIES_AND_LEISURE"}
{"query": "How can I get started with photography?", "category": "HOBBIES_AND_LEISURE"}
{"query": "What are some fun DIY projects to do at home?", "category": "HOBBIES_AND_LEISURE"}
{"query": "What are the best board games for a family night?", "category": "HOBBIES_AND_LEISURE"}
```

## BOOKS_AND_LITERATURE

Definition: Queries related to books, literature, authors, and reading recommendations.

Examples:

```json
{"query": "Who wrote '1984'?", "category": "BOOKS_AND_LITERATURE"}
{"query": "Who are some notable contemporary poets?", "category": "BOOKS_AND_LITERATURE"}
{"query": "What is the plot of 'Pride and Prejudice'?", "category": "BOOKS_AND_LITERATURE"}
{"query": "What are the best fantasy novels of the decade?", "category": "BOOKS_AND_LITERATURE"}
```

## SPORTS

Definition: Queries related to sports, athletes, teams, events, and competitions.

Examples:

```json
{"query": "Who won the Super Bowl?", "category": "SPORTS"}
{"query": "When is the next FIFA World Cup?", "category": "SPORTS"}
{"query": "What is the world record for the 100-meter sprint?", "category": "SPORTS"}
{"query": "Who holds the record for most goals in a single Premier League season?", "category": "SPORTS"}
```

## NEWS

Definition: Queries related to current events, news stories, and media coverage.

Examples:

```json
{"query": "What happened in the latest presidential election?", "category": "NEWS"}
{"query": "What is the latest update on the COVID-19 pandemic?", "category": "NEWS"}
{"query": "What are the latest developments in the global economy?", "category": "NEWS"}
{"query": "What is the current status of the Paris Agreement on climate change?", "category": "NEWS"}
```

## BEAUTY_AND_FITNESS

Definition: Queries related to beauty products, makeup, skincare, fitness routines, and wellness.

Examples:

```json
{"query": "How can I build muscle mass?", "category": "BEAUTY_AND_FITNESS"}
{"query": "What are some effective skincare routines?", "category": "BEAUTY_AND_FITNESS"}
{"query": "What are the benefits of yoga for mental health?", "category": "BEAUTY_AND_FITNESS"}
{"query": "How can I create a hair care routine for dry hair?", "category": "BEAUTY_AND_FITNESS"}
```

## FINANCE

Definition: Queries related to financial advice, investments, economics, and money management.

Examples:

```json
{"query": "How can I save for retirement?", "category": "FINANCE"}
{"query": "How can I improve my credit score?", "category": "FINANCE"}
{"query": "What is the difference between a 401(k) and an IRA?", "category": "FINANCE"}
{"query": "What is the best way to budget my monthly expenses?", "category": "FINANCE"}
```

## PEOPLE_AND_SOCIETY

Definition: Queries related to society, human behavior, relationships, and cultural issues.

Examples:

```json
{"query": "What are the causes of inequality?", "category": "PEOPLE_AND_SOCIETY"}
{"query": "How can communities address homelessness?", "category": "PEOPLE_AND_SOCIETY"}
{"query": "How do different cultures celebrate New Year?", "category": "PEOPLE_AND_SOCIETY"}
{"query": "What are the psychological effects of social media?", "category": "PEOPLE_AND_SOCIETY"}
```

## AUTOS_AND_VEHICLES

Definition: Queries related to cars, vehicles, transportation, and road safety.

Examples:

```json
{"query": "How do electric cars work?", "category": "AUTOS_AND_VEHICLES"}
{"query": "How can I maintain my car's engine?", "category": "AUTOS_AND_VEHICLES"}
{"query": "What are the benefits of hybrid cars?", "category": "AUTOS_AND_VEHICLES"}
{"query": "What is the fuel efficiency of a Tesla Model S?", "category": "AUTOS_AND_VEHICLES"}
```

## GAMES

Definition: Queries related to video games, board games, game mechanics, and gaming news.

Examples:

```json
{"query": "How do I level up fast in Fortnite?", "category": "GAMES"}
{"query": "What are the top upcoming video games?", "category": "GAMES"}
{"query": "What is the best strategy in Minecraft?", "category": "GAMES"}
{"query": "How do you unlock new characters in Super Smash Bros.?", "category": "GAMES"}
```

## TIME_AND_WEATHER

Definition: Queries related to time, weather, and climate forecasts.

Examples:

```json
{"query": "How do weather patterns affect agriculture?", "category": "TIME_AND_WEATHER"}
{"query": "How many hours are there between GMT and EST?", "category": "TIME_AND_WEATHER"}
{"query": "What will the weather be like tomorrow in New York?", "category": "TIME_AND_WEATHER"}
{"query": "What is the best time of year to visit the Caribbean?", "category": "TIME_AND_WEATHER"}
```
```

System: part 3

## ONLINE_COMMUNITIES

Definition: Queries related to online forums, social media, and digital communities.

Examples:

```json
{"query": "How do I join a subreddit on Reddit?", "category": "ONLINE_COMMUNITIES"}
{"query": "How can I create a group on Facebook?", "category": "ONLINE_COMMUNITIES"}
{"query": "What are some popular online gaming communities?", "category": "ONLINE_COMMUNITIES"}
{"query": "What are the benefits of joining professional LinkedIn groups?", "category": "ONLINE_COMMUNITIES"}
```

## INTERNET_AND_TELECOM

Definition: Queries related to internet services, telecommunications, and online infrastructure.

Examples:

```json
{"query": "How does 5G work?", "category": "INTERNET_AND_TELECOM"}
{"query": "How can I improve my home Wi-Fi signal?", "category": "INTERNET_AND_TELECOM"}
{"query": "What is the fastest internet provider in the US?", "category": "INTERNET_AND_TELECOM"}
{"query": "What is the difference between fiber optic and broadband internet?", "category": "INTERNET_AND_TELECOM"}
```

## LOCAL_INFORMATION

Definition: Queries related to local businesses, services, and events.

Examples:

```json
{"query": "Is there a public library near me?", "category": "LOCAL_INFORMATION"}
{"query": "Where can I find a good gym in Miami?", "category": "LOCAL_INFORMATION"}
{"query": "What are the best restaurants in San Francisco?", "category": "LOCAL_INFORMATION"}
{"query": "What local events are happening this weekend in Chicago?", "category": "LOCAL_INFORMATION"}
```

## PETS_AND_ANIMALS

Definition: Queries related to pets, animal care, and wildlife.

Examples:

```json
{"query": "How do I train my dog to sit?", "category": "PETS_AND_ANIMALS"}
{"query": "What is the lifespan of a cat?", "category": "PETS_AND_ANIMALS"}
{"query": "How can I create a safe habitat for pet birds?", "category": "PETS_AND_ANIMALS"}
{"query": "What should I feed my rabbit for a healthy diet?", "category": "PETS_AND_ANIMALS"}
```

## STOCK

Definition: Queries related to stock markets, stock prices, and investment trends.

Examples:

```json
{"query": "How can I diversify my stock portfolio?", "category": "STOCK"}
{"query": "What is the current stock price of Apple?", "category": "STOCK"}
{"query": "What factors influence stock market fluctuations?", "category": "STOCK"}
{"query": "How does short selling work in the stock market?", "category": "STOCK"}
```

## RELIGION_AND_SPIRITUALITY

Definition: Queries related to religious beliefs, practices, spirituality, and theology.

Examples:

```json
{"query": "How do Christians celebrate Easter?", "category": "RELIGION_AND_SPIRITUALITY"}
{"query": "What are the main teachings of Buddhism?", "category": "RELIGION_AND_SPIRITUALITY"}
{"query": "What is the significance of Ramadan in Islam?", "category": "RELIGION_AND_SPIRITUALITY"}
{"query": "What are the core principles of Hinduism?", "category": "RELIGION_AND_SPIRITUALITY"}
```

## GEOGRAPHY

Definition: Queries related to geographical features, locations, maps, and global regions.

Examples:

```json
{"query": "What are the largest deserts on Earth?", "category": "GEOGRAPHY"}
{"query": "What is the longest river in the world?", "category": "GEOGRAPHY"}
{"query": "Which countries are part of Scandinavia?", "category": "GEOGRAPHY"}
{"query": "What is the tallest mountain in the world?", "category": "GEOGRAPHY"}
```

## HISTORY

Definition: Queries related to historical events, figures, and timelines.

Examples:

```json
{"query": "What caused the fall of the Roman Empire?", "category": "HISTORY"}
{"query": "What were the main causes of World War I?", "category": "HISTORY"}
{"query": "Who was the first President of the United States?", "category": "HISTORY"}
{"query": "Who were the key figures in the American Civil War?", "category": "HISTORY"}
```

## FOOD_AND_DRINK

Definition: Queries related to cooking, recipes, dining, and beverages.

Examples:

```json
{"query": "How do I make a perfect cheesecake?", "category": "FOOD_AND_DRINK"}
{"query": "What are some easy vegan dinner recipes?", "category": "FOOD_AND_DRINK"}
{"query": "How do I brew the perfect cup of coffee?", "category": "FOOD_AND_DRINK"}
{"query": "What are the health benefits of green tea?", "category": "FOOD_AND_DRINK"}
```

## SHOPPING

Definition: Queries related to purchasing items, shopping tips, and deals.

```json
{"query": "How do I find discounts on clothing online?", "category": "SHOPPING"}
{"query": "What should I look for when buying a laptop?", "category": "SHOPPING"}
{"query": "What are the best online stores for electronics?", "category": "SHOPPING"}
{"query": "What are the most popular shopping malls in New York?", "category": "SHOPPING"}
```

**System: part 4**

```
## OTHER

Definition: Queries that do not fit into any of the above categories.

Examples:

```json
{"query": "What is the meaning of life?", "category": "OTHER"}
{"query": "How can I improve my public speaking skills?", "category": "OTHER"}
{"query": "What are the benefits of practicing mindfulness?", "category": "OTHER"}
{"query": "How can I organize my daily schedule more efficiently?", "category": "OTHER"}
```

**User**

```
# Query that must be classified

```json
{"query": "{{ query_well_formed] }}"}
```

# Output Format

```json
{"category": "THE CLASS GOES HERE"}
```

THE CLASSIFICATION MUST STRICTLY USE ONLY THE CLASSES PROVIDED. YOUR OUTPUT MUST CONTAIN ONLY THE JSON OBJECT.
```
```

# C Appendix for Chapter 7: *GADePo*

## *GADePo*'s Extra Parameters

*GADePo* introduces few extra parameters to the PLM. The amount of parameters is reported in Table C.1.

| **Parameter** | **Model** | |
|---|---|---|
| | RoBERTa$_{\text{LARGE}}$ | BERT$_{\text{BASE}}$ |
| `<ent>` | 1024 | 768 |
| `<pent>` | 1024 | 768 |
| `<ent>` $\longrightarrow *$ | $24 \times 1024$ | $12 \times 768$ |
| $* \longrightarrow$ `<ent>` | $24 \times 1024$ | $12 \times 768$ |
| `<pent>` $\longrightarrow *$ | $24 \times 1024$ | $12 \times 768$ |
| $* \longrightarrow$ `<pent>` | $24 \times 1024$ | $12 \times 768$ |
| **Total** | 100,352 | 38,400 |

Table C.1: *GADePo*'s extra parameters count.

The introduction of these parameters results in only a minimal increase in the overall parameter count of the models. Specifically, *GADePo*'s augmentation amounts to a mere 0.036% increase over the BERT$_{\text{BASE}}$ model. In contrast, even a slight increase of just one unit in BERT$_{\text{BASE}}$'s hidden dimensions would result in a 0.139% parameter increase, which is roughly four times greater than the augmentation introduced by *GADePo*. Given that such a small change is incompatible with other architectural constraints, such as the number of heads, it is implausible that this minimal augmentation would solely account for the observed performance gains.

This indicates that the performance improvements are largely due to the effective inductive bias introduced by *GADePo*, rather than the increase in parameter count. The same rationale applies to the results observed with RoBERTa$_{\text{LARGE}}$.

## Training Details

We generally comply with the hyperparameters of ATLOP and set the output dimension in Equation 7.6.2 and Equation 7.6.2 to 768. We also set the block size in Equation 7.4 and Equation 7.5 to 64, i.e., $k = 12$.

In all our experiments we perform early stopping on the development set based on the Ign $F_1 + F_1$ score for DocRED and Re-DocRED, and $F_1$ score for HacRED. The five different seeds we use are $\{73, 21, 37, 7, 3\}$.

We use RAdam (Liu et al., 2020) as our optimiser. On the RoBERTa$_{\text{LARGE}}$ based models we train for 8 epochs and set the learning rates to $3e^{-5}$ and $1e^{-4}$ for the PLM parameters and the new additional parameters, respectively. On the BERT$_{\text{BASE}}$ based models we train for 10 epochs and set the learning rates to $1e^{-5}$ and $1e^{-4}$ for the PLM parameters and the new additional parameters, respectively. We use a cosine learning rate decay throughout the training process.

In all our experiments the batch size is set to 4 for ATLOP and 2 for *GADePo*, with gradient accumulation set to 1 and 2, for ATLOP and *GADePo*, respectively. We clip the gradients to a max norm of 1.0. All models are trained with mixed precision.

We run our experiments on two types of GPUs, namely the NVIDIA V100 32GB for the RoBERTa$_{\text{LARGE}}$ based models and NVIDIA RTX 3090 24GB for the BERT$_{\text{BASE}}$ based models, respectively.

We use `torch` (Paszke et al., 2019), `lightning` (Falcon and The PyTorch Lightning team, 2019), and `transformers` (Wolf et al., 2020) libraries to develop our models.

## Additional Results

**Re-DocRED and HacRED**. Table C.2 and Table C.3 present additional results for Re-DocRED and HacRED, respectively. In addition to the results outlined in Section 7.8, these tables include the mean and standard deviation on the development set, calculated from five training runs with distinct random seeds, as reported in Appendix C.

| | | **Dev** | | **Test** | |
|---|---|---|---|---|---|
| **Model** | **Aggregation** | Ign $F_1$ | $F_1$ | Ign $F_1$ | $F_1$ |
| ATLOP$^\star$ | $\boldsymbol{h}_e$ | $75.46 \pm 0.16$ | $76.16 \pm 0.16$ | 75.27 | 75.92 |
| GADePo (ours) | `<ent>` | $75.46 \pm 0.20$ | $76.31 \pm 0.24$ | **75.55** | **76.38** |
| ATLOP$^\bullet$ | $\boldsymbol{h}_e$ ; $\boldsymbol{c}^{(s,o)}$ | 76.79 | 77.46 | 76.82 | 77.56 |
| ATLOP$^\star$ | $\boldsymbol{h}_e$ ; $\boldsymbol{c}^{(s,o)}$ | $77.75 \pm 0.08$ | $78.41 \pm 0.10$ | 77.62 | 78.38 |
| GADePo (ours) | `<ent>` ; `<pent>` | $77.48 \pm 0.12$ | $78.19 \pm 0.14$ | **77.70** | **78.40** |

Table C.2: Results in percentage for the development and test sets of Re-DocRED. We report the results obtained by Tan et al. (2022b) (ATLOP$^\bullet$) on Re-DocRED. ATLOP$^\star$ indicates our reimplementation of the previous method. We report the mean and standard deviation of Ign $F_1$ and $F_1$ on the development set, calculated from five training runs with distinct random seeds. We report the test score achieved by the best checkpoint on the development set. Ign $F_1$ refers to the $F_1$ score that excludes relational facts shared between the training and development/test sets.

| Model | Aggregation | Dev | | | Test | | |
|---|---|---|---|---|---|---|---|
| | | $P$ | $R$ | $F_1$ | $P$ | $R$ | $F_1$ |
| ATLOP$^\star$ | $\boldsymbol{h}_e$ | $77.37 \pm 0.22$ | $77.40 \pm 0.31$ | $77.39 \pm 0.13$ | **76.27** | 76.83 | 76.55 |
| GADePo (ours) | <ent> | $72.96 \pm 0.96$ | $79.22 \pm 1.20$ | $75.96 \pm 0.99$ | 74.13 | **79.46** | **76.70** |
| ATLOP$^\diamond$ | $\boldsymbol{h}_e$ ; $\boldsymbol{c}^{(s,o)}$ | – | – | – | 77.89 | 76.55 | 77.21 |
| ATLOP$^\star$ | $\boldsymbol{h}_e$ ; $\boldsymbol{c}^{(s,o)}$ | $77.18 \pm 0.14$ | $77.98 \pm 0.66$ | $77.58 \pm 0.36$ | 76.36 | 78.86 | 77.59 |
| GADePo (ours) | <ent> ; <pent> | $75.98 \pm 0.94$ | $80.54 \pm 0.72$ | $78.19 \pm 0.19$ | **78.27** | **79.03** | **78.65** |

Table C.3: Results in percentage for the development and test sets of HacRED. We report the results obtained by Cheng et al. (2021) (ATLOP$^\diamond$) on HacRED. ATLOP$^\star$ indicates our reimplementation of the previous method. We report the mean and standard deviation of Precision ($P$), Recall ($R$) and $F_1$ on the development set, calculated from five training runs with distinct random seeds. We report the test score achieved by the best checkpoint on the development set.

**DocRED results**. The DocRED (Yao et al., 2019b) dataset consists of 56,354 facts, 96 relations, 5,053 documents, and 26.2 average number of entities per document. In line with the approach taken for Re-DocRED and HacRED, Table C.4 and Figure C.1 illustrate the results for DocRED.

| Model | Aggregation | Dev | | Test | |
|---|---|---|---|---|---|
| | | Ign $F_1$ | $F_1$ | Ign $F_1$ | $F_1$ |
| ATLOP$^\star$ | $\boldsymbol{h}_e$ | $59.66 \pm 0.20$ | $61.60 \pm 0.21$ | 59.22 | 61.37 |
| GADePo (ours) | <ent> | $59.04 \pm 0.52$ | $61.18 \pm 0.46$ | **59.30** | **61.63** |
| ATLOP$^\circ$ | $\boldsymbol{h}_e$ ; $\boldsymbol{c}^{(s,o)}$ | $61.32 \pm 0.14$ | $63.18 \pm 0.19$ | 61.39 | 63.40 |
| ATLOP$^\star$ | $\boldsymbol{h}_e$ ; $\boldsymbol{c}^{(s,o)}$ | $61.41 \pm 0.26$ | $63.38 \pm 0.28$ | **61.62** | 63.72 |
| GADePo (ours) | <ent> ; <pent> | $61.19 \pm 0.55$ | $63.26 \pm 0.48$ | 61.52 | **63.75** |

Table C.4: Results in percentage for the development and test sets of DocRED. We report the results obtained by Zhou et al. (2021) (ATLOP$^\circ$) on DocRED. ATLOP$^\star$ indicates our reimplementation of the previous method. We report the mean and standard deviation of Ign $F_1$ and $F_1$ on the development set, calculated from five training runs with distinct random seeds. We report the test score achieved by the best checkpoint on the development set. Ign $F_1$ refers to the $F_1$ score that excludes relational facts shared between the training and development/test sets.

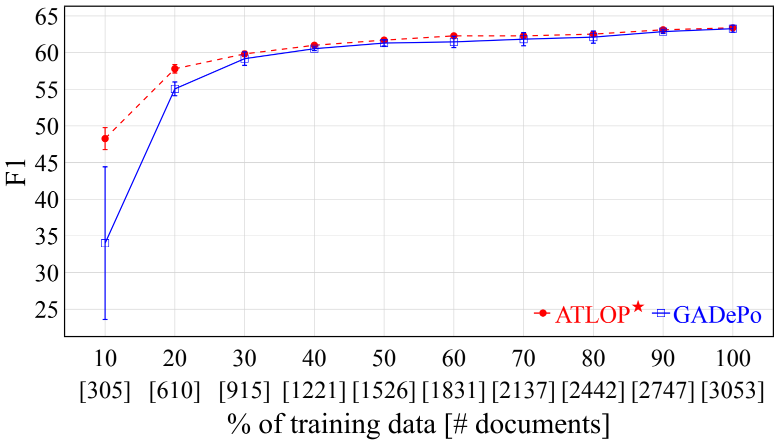



Figure C.1: Performance of ATLOP⋆ ($\boldsymbol{h}_e$ ; $\boldsymbol{c}^{(s,o)}$) and *GADePo* (`<ent>` ; `<pent>`) on the development set under varying data availability conditions on DocRED. The $x$-axis represents the percentage and number of documents from the training dataset, while the $y$-axis displays the $F_1$ score in percentage. Each point on the graph represents the mean value, while error bars indicate the standard deviation derived from five distinct training runs with separate random seeds.

# D Appendix for Chapter 8: *FnF-TG*

## Complexity of *FnF-TG*

Our method exhibits identical computational complexity to StATIk (Markowitz et al., 2022), with $O(N + Q)$ complexity, where $N$ denotes the number of nodes in the graph and $Q$ represents the number of queries $(h, r, \hat{e})$ and $(\hat{e}, r, t)$.

## Training and Implementation Details

We provide details on the training and implementation of our models on three datasets: $\text{WN18RR}_{\text{IND}}$, $\text{FB15k-237}_{\text{IND}}$, and $\text{Wikidata5M}_{\text{IND}}$.

**Seeds and Epochs**. We run our experiments with five different seeds $(73, 21, 37, 3, 7)$ for $\text{WN18RR}_{\text{IND}}$ and $\text{FB15k-237}_{\text{IND}}$, and two seeds $(73, 21)$ for $\text{Wikidata5M}_{\text{IND}}$ due to its large scale (see Table 8.1). We train our models for 40 epochs on $\text{WN18RR}_{\text{IND}}$ and $\text{FB15k-237}_{\text{IND}}$, and 5 epochs on $\text{Wikidata5M}_{\text{IND}}$, following previous works (Daza et al., 2021; Markowitz et al., 2022).

**Hyperparameters**. We set the number of sampled neighbors per entity based on the dataset statistics (Table 8.1): 10 for $\text{WN18RR}_{\text{IND}}$, 40 for $\text{FB15k-237}_{\text{IND}}$, and 1 for $\text{Wikidata5M}_{\text{IND}}$. We use 24 words of text for each $x_{KG}$ in $\text{WN18RR}_{\text{IND}}$ and $\text{FB15k-237}_{\text{IND}}$, and 64 words for $\text{Wikidata5M}_{\text{IND}}$.

**Graph Transformer Encoder**. We implement the *Graph Transformer Encoder* layer using a pre-LayerNorm Transformer (Xiong et al., 2020) with a SwiGLU-type pointwise feed-forward network Shazeer (2020). We use a single GT layer, as multiple layers did not improve performance while increasing latency.

**Optimisation**. We set the learning rate to $1e^{-5}$ for a batch size of 32 and scale it proportionally with the batch size following a power-of-2 rule to fit the GPU budget. We use RAdam Liu et al. (2020) as our optimiser and a cosine learning rate decay throughout the training process.

**Libraries**. We develop our models using `torch` Paszke et al. (2019), `lightning` Falcon and The PyTorch Lightning team (2019), and `transformers` Wolf et al. (2020) libraries.

## Computational Budget

We fix our computational budget to a constant consumer-grade GPU (NVIDIA RTX3090 24GB) as stated in Subsection 8.9.1 and report the GPU budget per run for each dataset on FnF-TG$_{\text{BERT}_{\text{BASE}}}$ relative to the largest text encoders. The GPU budget per run is 4 GPU/h for WN18RR$_{\text{IND}}$, 6 GPU/h for FB15k-237$_{\text{IND}}$, and 40 GPU/h for Wikidata5M$_{\text{IND}}$.

# Bibliography


Abdin, M., Aneja, J., Behl, H., Bubeck, S., Eldan, R., Gunasekar, S., Harrison, M., Hewett, R. J., Javaheripi, M., Kauffmann, P., Lee, J. R., Lee, Y. T., Li, Y., Liu, W., Mendes, C. C. T., Nguyen, A., Price, E., de Rosa, G., Saarikivi, O., Salim, A., Shah, S., Wang, X., Ward, R., Wu, Y., Yu, D., Zhang, C., and Zhang, Y. (2024). Phi-4 technical report.

Adlakha, V., Dhuliawala, S., Suleman, K., de Vries, H., and Reddy, S. (2022). TopiOCQA: Open-domain conversational question answering with topic switching. *Transactions of the Association for Computational Linguistics*, 10:468–483.

Akbar, S. A., Hossain, M. M., Wood, T., Chin, S.-C., Salinas, E. M., Alvarez, V., and Cornejo, E. (2024). HalluMeasure: Fine-grained hallucination measurement using chain-of-thought reasoning. In Al-Onaizan, Y., Bansal, M., and Chen, Y.-N., editors, *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 15020–15037, Miami, Florida, USA. Association for Computational Linguistics.

Alexandru, A., Calvi, A., Broomfield, H., Golden, J., Dai, K., Leys, M., Burger, M., Bartolo, M., Engeler, R., Pisupati, S., Drane, T., and Park, Y. S. (2025). Atla selene mini: A general purpose evaluation model.

Anthropic (2023). Introducing claude. *Anthropic Blog*.

ASF (2004). Apache license, version 2.0.

Bahdanau, D., Chorowski, J., Serdyuk, D., Brakel, P., and Bengio, Y. (2015). End-to-end attention-based large vocabulary speech recognition. *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4945–4949.

Bai, J., Bai, S., Chu, Y., Cui, Z., Dang, K., Deng, X., Fan, Y., Ge, W., Han, Y., Huang, F., Hui, B., Ji, L., Li, M., Lin, J., Lin, R., Liu, D., Liu, G., Lu, C., Lu, K., Ma, J., Men, R., Ren, X., Ren, X., Tan, C., Tan, S., Tu, J., Wang, P., Wang, S., Wang, W., Wu, S., Xu, B., Xu, J., Yang, A., Yang, H., Yang, J., Yang, J., Yang, S., Yao, Y., Yu, B., Bowen, Y., Yuan, H., Yuan, Z., Zhang, J., Zhang, X., Zhang, Y., Zhang, Z., Zhou, C., Zhou, J., Zhou, X., and Zhu, T. (2023). Qwen technical report. *ArXiv*, abs/2309.16609.

Bai, Y., Jones, A., Ndousse, K., Askell, A., Chen, A., Dassarma, N., Drain, D., Fort, S., Ganguli, D., Henighan, T., Joseph, N., Kadavath, S., Kernion, J., Conerly, T., El-Showk, S., Elhage, N., Hatfield-Dodds, Z., Hernandez, D., Hume, T., Johnston, S., Kravec, S., Lovitt, L., Nanda, N., Olsson, C., Amodei, D., Brown, T. B., Clark, J., McCandlish, S., Olah, C., Mann, B., and Kaplan, J. (2022a). Training a helpful and harmless assistant with reinforcement learning from human feedback. *ArXiv*, abs/2204.05862.

Bai, Y., Kadavath, S., Kundu, S., Askell, A., Kernion, J., Jones, A., Chen, A., Goldie, A., Mirhoseini, A., McKinnon, C., et al. (2022b). Constitutional ai: Harmlessness from ai feedback. *arXiv preprint arXiv:2212.08073*.

Bajaj, P., Campos, D., Craswell, N., Deng, L., Gao, J., Liu, X., Majumder, R., McNamara, A., Mitra, B., Nguyen, T., Rosenberg, M., Song, X., Stoica, A., Tiwary, S., and Wang, T. (2018). Ms marco: A human generated machine reading comprehension dataset.

Balachandran, V., Vaswani, A., Tsvetkov, Y., and Parmar, N. (2021). Simple and efficient ways to improve REALM. In Fisch, A., Talmor, A., Chen, D., Choi, E., Seo, M., Lewis, P., Jia, R., and Min, S., editors, *Proceedings of the 3rd Workshop on Machine Reading for Question Answering*, pages 158–164, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Balazevic, I., Allen, C., and Hospedales, T. M. (2018). Hypernetwork knowledge graph embeddings. In *International Conference on Artificial Neural Networks*.

Banko, M., Cafarella, M. J., Soderland, S., Broadhead, M., and Etzioni, O. (2007). Open information extraction from the web. In *CACM*.

Barlacchi, G., Lauriola, I., Moschitti, A., Del Tredici, M., Shen, X., Vu, T., Byrne, B., and de Gispert, A. (2022). FocusQA: Open-domain question answering with a context in focus. In Goldberg, Y., Kozareva, Z., and Zhang, Y., editors, *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 5195–5208, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Belkin, N. J., Cool, C., Stein, A., and Thiel, U. (1995). Cases, scripts, and information-seeking strategies: On the design of interactive information retrieval systems. *Expert Systems with Applications*, 9(3):379–395.

Beltagy, I., Lo, K., and Cohan, A. (2019). SciBERT: A pretrained language model for scientific text. In Inui, K., Jiang, J., Ng, V., and Wan, X., editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3615–3620, Hong Kong, China. Association for Computational Linguistics.

Bhowmik, R. and de Melo, G. (2020). Explainable link prediction for emerging entities in knowledge graphs. In *The Semantic Web–ISWC 2020: 19th International Semantic Web Conference, Athens, Greece, November 2–6, 2020, Proceedings, Part I 19*, pages 39–55. Springer.

Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., and Yakhnenko, O. (2013). Translating embeddings for modeling multi-relational data. In Burges, C., Bottou, L., Welling, M., Ghahramani, Z., and Weinberger, K., editors, *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc.

Borgeaud, S., Mensch, A., Hoffmann, J., Cai, T., Rutherford, E., Millican, K., Van Den Driessche, G. B., Lespiau, J.-B., Damoc, B., Clark, A., De Las Casas, D., Guy, A., Menick, J., Ring, R., Hennigan, T., Huang, S., Maggiore, L., Jones, C., Cassirer, A., Brock, A., Paganini, M., Irving, G., Vinyals, O., Osindero, S., Simonyan, K., Rae, J., Elsen, E., and Sifre, L. (2022). Improving language models by retrieving from trillions of tokens. In Chaudhuri, K., Jegelka, S., Song, L., Szepesvari, C., Niu, G., and Sabato, S., editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 2206–2240. PMLR.

Bosselut, A., Rashkin, H., Sap, M., Malaviya, C., Celikyilmaz, A., and Choi, Y. (2019). COMET: Commonsense transformers for automatic knowledge graph construction. In Korhonen, A., Traum, D., and Màrquez, L., editors, *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4762–4779, Florence, Italy. Association for Computational Linguistics.

Bradley, R. A. and Terry, M. E. (1952). Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39:324.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. (2020). Language models are few-shot learners. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H., editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Cai, Z., Cao, M., Chen, H., Chen, K., Chen, K., Chen, X., Chen, X., Chen, Z., Chen, Z., Chu, P., Dong, X., Duan, H., Fan, Q., Fei, Z., Gao, Y., Ge, J., Gu, C., Gu, Y., Gui, T., Guo, A., Guo, Q., He, C., Hu, Y., Huang, T., Jiang, T., Jiao, P., Jin, Z., Lei, Z., Li, J., Li, J., Li, L., Li, S., Li, W., Li, Y., Liu, H., Liu, J., Hong, J., Liu, K., Liu, K., Liu, X., Lv, C., Lv, H., Lv, K., Ma, L., Ma, R., Ma, Z., Ning, W., Ouyang, L., Qiu, J., Qu, Y., Shang, F., Shao, Y., Song, D., Song, Z., Sui, Z., Sun, P., Sun, Y., Tang, H., Wang, B., Wang, G., Wang, J., Wang, J., Wang, R., Wang, Y., Wang, Z., Wei, X., Weng, Q., Wu, F., Xiong, Y., Xu, C., Xu, R., Yan, H., Yan, Y., Yang, X., Ye, H., Ying, H., Yu, J., Yu, J., Zang, Y., Zhang, C., Zhang, L., Zhang, P., Zhang, P., Zhang, R., Zhang, S., Zhang, S., Zhang, W., Zhang, W., Zhang, X., Zhang, X., Zhao, H., Zhao, Q., Zhao, X., Zhou, F., Zhou, Z., Zhuo, J., Zou, Y., Qiu, X., Qiao, Y., and Lin, D. (2024). Internlm2 technical report.

Chang, Y., Wang, X., Wang, J., Wu, Y., Yang, L., Zhu, K., Chen, H., Yi, X., Wang, C., Wang, Y., Ye, W., Zhang, Y., Chang, Y., Yu, P. S., Yang, Q., and Xie, X. (2024). A survey on evaluation of large language models. *ACM Trans. Intell. Syst. Technol.*, 15(3).

Chen, D., Fisch, A., Weston, J., and Bordes, A. (2017). Reading Wikipedia to answer open-domain questions. In Barzilay, R. and Kan, M.-Y., editors, *Proceedings of the 55th Annual*

*Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1870–1879, Vancouver, Canada. Association for Computational Linguistics.

Chen, S., Cheng, H., Liu, X., Jiao, J., Ji, Y., and Gao, J. (2023). Pre-training transformers for knowledge graph completion. *ArXiv*, abs/2303.15682.

Chen, S., Liu, X., Gao, J., Jiao, J., Zhang, R., and Ji, Y. (2021). HittER: Hierarchical transformers for knowledge graph embeddings. In Moens, M.-F., Huang, X., Specia, L., and Yih, S. W.-t., editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10395–10407, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Cheng, Q., Liu, J., Qu, X., Zhao, J., Liang, J., Wang, Z., Huai, B., Yuan, N. J., and Xiao, Y. (2021). HacRED: A large-scale relation extraction dataset toward hard cases in practical applications. In Zong, C., Xia, F., Li, W., and Navigli, R., editors, *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2819–2831, Online. Association for Computational Linguistics.

Choi, E., He, H., Iyyer, M., Yatskar, M., Yih, W.-t., Choi, Y., Liang, P., and Zettlemoyer, L. (2018). QuAC: Question answering in context. In Riloff, E., Chiang, D., Hockenmaier, J., and Tsujii, J., editors, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2174–2184, Brussels, Belgium. Association for Computational Linguistics.

Choi, H., Kim, J., Joe, S., and Gwon, Y. (2021). Evaluation of bert and albert sentence embedding performance on downstream nlp tasks. *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 5482–5487.

Christopoulou, F., Miwa, M., and Ananiadou, S. (2019). Connecting the dots: Document-level neural relation extraction with edge-oriented graphs. In Inui, K., Jiang, J., Ng, V., and Wan, X., editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4925–4936, Hong Kong, China. Association for Computational Linguistics.

Coman, A., Barlacchi, G., and de Gispert, A. (2023). Strong and efficient baselines for open domain conversational question answering. In Bouamor, H., Pino, J., and Bali, K., editors, *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 6305–6314, Singapore. Association for Computational Linguistics.

Coman, A., Theodoropoulos, C., Moens, M.-F., and Henderson, J. (2024). GADePo: Graph-assisted declarative pooling transformers for document-level relation extraction. In Yu, W., Shi, W., Yasunaga, M., Jiang, M., Zhu, C., Hajishirzi, H., Zettlemoyer, L., and Zhang, Z., editors, *Proceedings of the 3rd Workshop on Knowledge Augmented Methods for NLP*, pages 1–14, Bangkok, Thailand. Association for Computational Linguistics.

Coman, A. C., Sorodoc, I.-T., Ribeiro, L. F. R., Byrne, B., Henderson, J., and de Gispert, A. (2025a). Ragferee: Building contextual reward models for retrieval-augmented generation.

Coman, A. C., Theodoropoulos, C., Moens, M.-F., and Henderson, J. (2025b). Fast-and-frugal text-graph transformers are effective link predictors. In Che, W., Nabende, J., Shutova, E., and Pilehvar, M. T., editors, *Findings of the Association for Computational Linguistics: ACL 2025*, pages 11828–11841, Vienna, Austria. Association for Computational Linguistics.

Conneau, A., Kiela, D., Schwenk, H., Barrault, L., and Bordes, A. (2017). Supervised learning of universal sentence representations from natural language inference data. In Palmer, M., Hwa, R., and Riedel, S., editors, *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 670–680, Copenhagen, Denmark. Association for Computational Linguistics.

Croft, W. B. and Thompson, R. H. (1987). I3r: A new approach to the design of document retrieval systems. *Journal of the american society for information science*, 38(6):389–404.

Cui, Y., Che, W., Liu, T., Qin, B., and Yang, Z. (2021). Pre-training with whole word masking for chinese bert. *IEEE/ACM Trans. Audio, Speech and Lang. Proc.*, 29:3504–3514.

Dai, D., Zheng, H., Luo, F., Yang, P., Liu, T., Sui, Z., and Chang, B. (2021). Inductively representing out-of-knowledge-graph entities by optimal estimation under translational assumptions. In Rogers, A., Calixto, I., Vulić, I., Saphra, N., Kassner, N., Camburu, O.-M., Bansal, T., and Shwartz, V., editors, *Proceedings of the 6th Workshop on Representation Learning for NLP (RepL4NLP-2021)*, pages 83–89, Online. Association for Computational Linguistics.

Dalton, J., Dietz, L., and Allan, J. (2014). Entity query feature expansion using knowledge base links. In *Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval*, SIGIR '14, page 365–374, New York, NY, USA. Association for Computing Machinery.

Dao, T. (2023). Flashattention-2: Faster attention with better parallelism and work partitioning.

Das, R., Dhuliawala, S., Zaheer, M., and McCallum, A. (2019). Multi-step retriever-reader interaction for scalable open-domain question answering. *ArXiv*, abs/1905.05733.

Daza, D., Cochez, M., and Groth, P. (2021). Inductive entity representations from text via link prediction. In *Proceedings of the Web Conference 2021*, WWW '21, page 798–808, New York, NY, USA. Association for Computing Machinery.

DeepSeek-AI, :, Bi, X., Chen, D., Chen, G., Chen, S., Dai, D., Deng, C., Ding, H., Dong, K., Du, Q., Fu, Z., Gao, H., Gao, K., Gao, W., Ge, R., Guan, K., Guo, D., Guo, J., Hao, G., Hao, Z., He, Y., Hu, W., Huang, P., Li, E., Li, G., Li, J., Li, Y., Li, Y. K., Liang, W., Lin, F., Liu, A. X., Liu, B., Liu, W., Liu, X., Liu, X., Liu, Y., Lu, H., Lu, S., Luo, F., Ma, S., Nie, X., Pei, T., Piao, Y., Qiu, J., Qu, H., Ren, T., Ren, Z., Ruan, C., Sha, Z., Shao, Z., Song, J., Su, X., Sun, J., Sun, Y., Tang, M., Wang, B., Wang, P., Wang, S., Wang, Y., Wang, Y., Wu, T., Wu, Y., Xie, X., Xie, Z., Xie, Z., Xiong, Y., Xu, H., Xu, R. X., Xu, Y., Yang, D., You, Y., Yu, S., Yu, X., Zhang, B., Zhang, H., Zhang, L., Zhang, L., Zhang, M., Zhang, M., Zhang, W., Zhang, Y., Zhao, C., Zhao, Y., Zhou, S., Zhou, S., Zhu, Q., and Zou, Y. (2024). Deepseek llm: Scaling open-source language models with longtermism.

DeepSeek-AI, Guo, D., Yang, D., Zhang, H., Song, J., Zhang, R., Xu, R., Zhu, Q., Ma, S., Wang, P., Bi, X., Zhang, X., Yu, X., Wu, Y., Wu, Z. F., Gou, Z., Shao, Z., Li, Z., Gao, Z., Liu, A., Xue, B., Wang, B., Wu, B., Feng, B., Lu, C., Zhao, C., Deng, C., Zhang, C., Ruan, C., Dai, D., Chen, D., Ji, D., Li, E., Lin, F., Dai, F., Luo, F., Hao, G., Chen, G., Li, G., Zhang, H., Bao, H., Xu, H., Wang, H., Ding, H., Xin, H., Gao, H., Qu, H., Li, H., Guo, J., Li, J., Wang, J., Chen, J., Yuan, J., Qiu, J., Li, J., Cai, J. L., Ni, J., Liang, J., Chen, J., Dong, K., Hu, K., Gao, K., Guan, K., Huang, K., Yu, K., Wang, L., Zhang, L., Zhao, L., Wang, L., Zhang, L., Xu, L., Xia, L., Zhang, M., Zhang, M., Tang, M., Li, M., Wang, M., Li, M., Tian, N., Huang, P., Zhang, P., Wang, Q., Chen, Q., Du, Q., Ge, R., Zhang, R., Pan, R., Wang, R., Chen, R. J., Jin, R. L., Chen, R., Lu, S., Zhou, S., Chen, S., Ye, S., Wang, S., Yu, S., Zhou, S., Pan, S., Li, S. S., Zhou, S., Wu, S., Ye, S., Yun, T., Pei, T., Sun, T., Wang, T., Zeng, W., Zhao, W., Liu, W., Liang, W., Gao, W., Yu, W., Zhang, W., Xiao, W. L., An, W., Liu, X., Wang, X., Chen, X., Nie, X., Cheng, X., Liu, X., Xie, X., Liu, X., Yang, X., Li, X., Su, X., Lin, X., Li, X. Q., Jin, X., Shen, X., Chen, X., Sun, X., Wang, X., Song, X., Zhou, X., Wang, X., Shan, X., Li, Y. K., Wang, Y. Q., Wei, Y. X., Zhang, Y., Xu, Y., Li, Y., Zhao, Y., Sun, Y., Wang, Y., Yu, Y., Zhang, Y., Shi, Y., Xiong, Y., He, Y., Piao, Y., Wang, Y., Tan, Y., Ma, Y., Liu, Y., Guo, Y., Ou, Y., Wang, Y., Gong, Y., Zou, Y., He, Y., Xiong, Y., Luo, Y., You, Y., Liu, Y., Zhou, Y., Zhu, Y. X., Xu, Y., Huang, Y., Li, Y., Zheng, Y., Zhu, Y., Ma, Y., Tang, Y., Zha, Y., Yan, Y., Ren, Z. Z., Ren, Z., Sha, Z., Fu, Z., Xu, Z., Xie, Z., Zhang, Z., Hao, Z., Ma, Z., Yan, Z., Wu, Z., Gu, Z., Zhu, Z., Liu, Z., Li, Z., Xie, Z., Song, Z., Pan, Z., Huang, Z., Xu, Z., Zhang, Z., and Zhang, Z. (2025). Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning.

Del Tredici, M., Barlacchi, G., Shen, X., Cheng, W., and de Gispert, A. (2021). Question rewriting for open-domain conversational qa: Best practices and limitations. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 2974–2978.

Del Tredici, M., Shen, X., Barlacchi, G., Byrne, B., and de Gispert, A. (2022). From rewriting to remembering: Common ground for conversational QA models. In Liu, B., Papangelis, A., Ultes, S., Rastogi, A., Chen, Y.-N., Spithourakis, G., Nouri, E., and Shi, W., editors, *Proceedings of the 4th Workshop on NLP for Conversational AI*, pages 70–76, Dublin, Ireland. Association for Computational Linguistics.

Dettmers, T., Minervini, P., Stenetorp, P., and Riedel, S. (2018). Convolutional 2d knowledge graph embeddings. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence*, AAAI'18/IAAI'18/EAAI'18. AAAI Press.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In Burstein, J., Doran, C., and Solorio, T., editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. *ArXiv*, abs/2010.11929.

Ebisu, T. and Ichise, R. (2018). Toruse: knowledge graph embedding on a lie group. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence*, AAAI'18/IAAI'18/EAAI'18. AAAI Press.

Elfwing, S., Uchibe, E., and Doya, K. (2017). Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. *Neural networks : the official journal of the International Neural Network Society*, 107:3–11.

Elgohary, A., Peskov, D., and Boyd-Graber, J. (2019). Can you unpack that? learning to rewrite questions-in-context. In Inui, K., Jiang, J., Ng, V., and Wan, X., editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5918–5924, Hong Kong, China. Association for Computational Linguistics.

Elman, J. L. (1990). Finding structure in time. *Cogn. Sci.*, 14:179–211.

Falcon, W. and The PyTorch Lightning team (2019). PyTorch Lightning.

Fan, W., Ding, Y., Ning, L., Wang, S., Li, H., Yin, D., Chua, T.-S., and Li, Q. (2024). A survey on rag meeting llms: Towards retrieval-augmented large language models. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '24, page 6491–6501, New York, NY, USA. Association for Computing Machinery.

Fehr, F. J., Baia, A. E., Chang, X., Coman, A. C., Hajal, K. E., Zein, D. E., Kumar, S., Gomez, J. P. Z., Cavallaro, A., Teney, D., and Henderson, J. (2025). Fine-tuning pretrained models with NVIB for improved generalisation. In *Workshop on Spurious Correlation and Shortcut Learning: Foundations and Solutions*.

Feng, Y., Han, Z., Sun, M., and Li, P. (2022). Multi-hop open-domain question answering over structured and unstructured knowledge. In Carpuat, M., de Marneffe, M.-C., and Meza Ruiz, I. V., editors, *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 151–156, Seattle, United States. Association for Computational Linguistics.

Fensel, D. A., Simsek, U., Angele, K., Huaman, E., Kärle, E., Panasiuk, O., Toma, I., Umbrich, J., and Wahler, A. (2020). Knowledge graphs: Methodology, tools and selected use cases. *Knowledge Graphs*.

Galkin, M., Wu, J., Denis, E., and Hamilton, W. L. (2021). Nodepiece: Compositional and parameter-efficient representations of large knowledge graphs. *ArXiv*, abs/2106.12144.

Gallegos, I. O., Rossi, R. A., Barrow, J., Tanjim, M. M., Kim, S., Dernoncourt, F., Yu, T., Zhang, R., and Ahmed, N. K. (2024). Bias and fairness in large language models: A survey. *Computational Linguistics*, 50(3):1097–1179.

Gao, M., Hu, X., Ruan, J., Pu, X., and Wan, X. (2024). Llm-based nlg evaluation: Current status and challenges.

Gemini Team (2025a). Gemini: A family of highly capable multimodal models.

Gemini Team (2025b). Gemini deep research — your personal research assistant. https://gemini.google/overview/deep-research/.

Gilmer, J., Schoenholz, S. S., Riley, P. F., Vinyals, O., and Dahl, G. E. (2017). Neural message passing for quantum chemistry. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, page 1263–1272. JMLR.org.

Glass, M., Rossiello, G., Chowdhury, M. F. M., Naik, A., Cai, P., and Gliozzo, A. (2022). Re2G: Retrieve, rerank, generate. In Carpuat, M., de Marneffe, M.-C., and Meza Ruiz, I. V., editors, *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2701–2715, Seattle, United States. Association for Computational Linguistics.

Grattafiori, A., Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., Schelten, A., Vaughan, A., Yang, A., Fan, A., Goyal, A., Hartshorn, A., Yang, A., Mitra, A., Sravankumar, A., Korenev, A., Hinsvark, A., Rao, A., Zhang, A., Rodriguez, A., Gregerson, A., Spataru, A., Roziere, B., Biron, B., Tang, B., Chern, B., Caucheteux, C., Nayak, C., Bi, C., Marra, C., McConnell, C., Keller, C., Touret, C., Wu, C., Wong, C., Ferrer, C. C., Nikolaidis, C., Allonsius, D., Song, D., Pintz, D., Livshits, D., Wyatt, D., Esiobu, D., Choudhary, D., Mahajan, D., Garcia-Olano, D., Perino, D., Hupkes, D., Lakomkin, E., AlBadawy, E., Lobanova, E., Dinan, E., Smith, E. M., Radenovic, F., Guzmán, F., Zhang, F., Synnaeve, G., Lee, G., Anderson, G. L., Thattai, G., Nail, G., Mialon, G., Pang, G., Cucurell, G., Nguyen, H., Korevaar, H., Xu, H., Touvron, H., Zarov, I., Ibarra, I. A., Kloumann, I., Misra, I., Evtimov, I., Zhang, J., Copet, J., Lee, J., Geffert, J., Vranes, J., Park, J., Mahadeokar, J., Shah, J., van der Linde, J., Billock, J., Hong, J., Lee, J., Fu, J., Chi, J., Huang, J., Liu, J., Wang, J., Yu, J., Bitton, J., Spisak, J., Park, J., Rocca, J., Johnstun, J., Saxe, J., Jia, J., Alwala, K. V., Prasad, K., Upasani, K., Plawiak, K., Li, K., Heafield, K., Stone, K., El-Arini, K., Iyer, K., Malik, K., Chiu, K., Bhalla, K., Lakhotia, K., Rantala-Yeary, L., van der Maaten, L., Chen, L., Tan, L., Jenkins, L., Martin, L., Madaan, L., Malo, L., Blecher, L., Landzaat, L., de Oliveira, L., Muzzi, M., Pasupuleti, M., Singh, M., Paluri, M., Kardas, M., Tsimpoukelli, M., Oldham, M., Rita, M., Pavlova, M., Kambadur, M., Lewis, M., Si, M., Singh, M. K., Hassan, M., Goyal, N., Torabi, N., Bashlykov, N., Bogoychev, N., Chatterji, N., Zhang, N., Duchenne, O., Çelebi, O., Alrassy, P., Zhang, P., Li, P., Vasic, P., Weng, P., Bhargava, P., Dubal, P., Krishnan, P., Koura, P. S., Xu, P., He, Q., Dong, Q., Srinivasan, R., Ganapathy, R., Calderer, R., Cabral, R. S., Stojnic, R., Raileanu, R., Maheswari, R., Girdhar, R., Patel, R., Sauvestre, R., Polidoro, R., Sumbaly, R., Taylor, R., Silva, R., Hou, R., Wang, R., Hosseini, S., Chennabasappa, S., Singh, S., Bell, S., Kim, S. S., Edunov, S., Nie, S., Narang, S., Raparthy, S., Shen, S., Wan, S., Bhosale, S., Zhang, S., Vandenhende, S., Batra, S., Whitman, S., Sootla, S., Collot, S., Gururangan, S., Borodinsky, S., Herman, T., Fowler, T., Sheasha, T., Georgiou, T., Scialom, T., Speckbacher, T., Mihaylov, T., Xiao, T., Karn, U., Goswami, V., Gupta, V., Ramanathan, V., Kerkez, V., Gonguet, V., Do, V., Vogeti, V., Albiero, V., Petrovic, V., Chu, W., Xiong, W., Fu, W., Meers, W., Martinet, X., Wang, X., Wang, X., Tan, X. E., Xia, X., Xie, X., Jia, X., Wang, X., Goldschlag, Y., Gaur, Y., Babaei, Y., Wen, Y., Song, Y., Zhang, Y., Li, Y., Mao, Y., Coudert, Z. D., Yan, Z., Chen, Z., Papakipos, Z., Singh, A., Srivastava, A., Jain, A., Kelsey, A., Shajnfeld, A., Gangidi, A., Victoria, A., Goldstand, A., Menon, A., Sharma, A., Boesenberg, A., Baevski, A., Feinstein, A., Kallet, A., Sangani, A., Teo, A., Yunus, A., Lupu, A., Alvarado, A., Caples, A., Gu, A., Ho, A., Poulton, A., Ryan, A., Ramchandani, A., Dong, A., Franco, A., Goyal, A., Saraf, A., Chowdhury, A., Gabriel, A., Bharambe, A., Eisenman, A., Yazdan, A., James, B., Maurer, B., Leonhardi, B., Huang, B., Loyd, B., Paola, B. D., Paranjape, B., Liu, B., Wu, B., Ni, B., Hancock, B., Wasti, B., Spence, B., Stojkovic, B., Gamido, B., Montalvo, B., Parker, C., Burton, C., Mejia, C., Liu, C., Wang, C., Kim, C., Zhou, C., Hu, C., Chu, C.-H., Cai, C., Tindal, C., Feichtenhofer, C., Gao, C., Civin, D., Beaty, D., Kreymer, D., Li, D., Adkins, D., Xu, D., Testuggine, D., David, D., Parikh, D., Liskovich, D., Foss, D., Wang, D., Le, D., Holland, D., Dowling, E., Jamil, E., Montgomery, E., Presani, E., Hahn, E., Wood, E., Le, E.-T., Brinkman, E., Arcaute, E., Dunbar, E., Smothers, E., Sun, F., Kreuk, F., Tian, F., Kokkinos, F., Ozgenel, F., Caggioni, F., Kanayet, F., Seide, F., Florez, G. M., Schwarz, G., Badeer, G., Swee, G., Halpern, G., Herman, G., Sizov, G., Guangyi, Zhang, Lakshminarayanan, G., Inan, H., Shojanazeri, H., Zou, H., Wang, H., Zha, H., Habeeb, H., Rudolph, H., Suk, H., Aspegren, H., Goldman, H., Zhan, H., Damlaj, I., Molybog, I., Tufanov, I., Leontiadis, I., Veliche, I.-E., Gat, I., Weissman, J., Geboski, J., Kohli, J., Lam, J., Asher, J., Gaya, J.-B., Marcus, J., Tang, J., Chan, J., Zhen, J., Reizenstein, J., Teboul, J., Zhong, J., Jin, J., Yang, J., Cummings, J., Carvill, J., Shepard, J., McPhie, J., Torres, J., Ginsburg, J., Wang, J., Wu, K., U, K. H., Saxena, K., Khandelwal, K., Zand, K., Matosich, K., Veeraraghavan, K., Michelena, K., Li, K., Jagadeesh, K., Huang, K., Chawla, K., Huang, K., Chen, L., Garg, L., A, L., Silva, L., Bell, L., Zhang, L., Guo, L., Yu, L., Moshkovich, L., Wehrstedt, L., Khabsa, M., Avalani, M., Bhatt, M., Mankus, M., Hasson, M., Lennie, M., Reso, M., Groshev, M., Naumov, M., Lathi, M., Keneally, M., Liu, M., Seltzer, M. L., Valko, M., Restrepo, M., Patel, M., Vyatskov, M., Samvelyan, M., Clark, M., Macey, M., Wang, M., Hermoso, M. J., Metanat, M., Rastegari, M., Bansal, M., Santhanam, N., Parks, N., White, N., Bawa, N., Singhal, N., Egebo, N., Usunier, N., Mehta, N., Laptev, N. P., Dong, N., Cheng, N., Chernoguz, O., Hart, O., Salpekar, O., Kalinli, O., Kent, P., Parekh, P., Saab, P., Balaji, P., Rittner, P., Bontrager, P., Roux, P., Dollar, P., Zvyagina, P., Ratanchandani, P., Yuvraj, P., Liang, Q., Alao, R., Rodriguez, R., Ayub, R., Murthy, R., Nayani, R., Mitra, R., Parthasarathy, R., Li, R., Hogan, R., Battey, R., Wang, R., Howes, R., Rinott, R., Mehta, S., Siby, S., Bondu, S. J., Datta, S., Chugh, S., Hunt, S., Dhillon, S., Sidorov, S., Pan, S., Mahajan, S., Verma, S., Yamamoto, S., Ramaswamy, S., Lindsay, S., Lindsay, S., Feng, S., Lin, S., Zha, S. C., Patil, S., Shankar, S., Zhang, S., Zhang, S., Wang, S., Agarwal, S., Sajuyigbe, S., Chintala, S., Max, S., Chen, S., Kehoe, S., Satterfield, S., Govindaprasad, S., Gupta, S., Deng, S., Cho, S., Virk, S., Subramanian, S., Choudhury, S., Goldman, S., Remez, T., Glaser, T., Best, T., Koehler, T., Robinson, T., Li, T., Zhang, T., Matthews, T., Chou, T., Shaked, T., Vontimitta, V., Ajayi, V., Montanez, V., Mohan, V., Kumar, V. S., Mangla, V., Ionescu, V., Poenaru, V., Mihailescu, V. T., Ivanov, V., Li, W., Wang, W., Jiang, W., Bouaziz, W., Constable, W., Tang, X., Wu, X., Wang, X., Wu, X., Gao, X., Kleinman, Y., Chen, Y., Hu, Y., Jia, Y., Qi, Y., Li, Y., Zhang, Y., Zhang, Y., Adi, Y., Nam, Y., Yu, Wang, Zhao, Y., Hao, Y., Qian, Y., Li, Y., He, Y., Rait, Z., DeVito, Z., Rosnbrick, Z., Wen, Z., Yang, Z., Zhao, Z., and Ma, Z. (2024). The llama 3 herd of models.

Groeneveld, D., Beltagy, I., Walsh, P., Bhagia, A., Kinney, R., Tafjord, O., Jha, A. H., Ivison, H., Magnusson, I., Wang, Y., Arora, S., Atkinson, D., Authur, R., Chandu, K. R., Cohan, A., Dumas, J., Elazar, Y., Gu, Y., Hessel, J., Khot, T., Merrill, W., Morrison, J., Muennighoff, N., Naik, A., Nam, C., Peters, M. E., Pyatkin, V., Ravichander, A., Schwenk, D., Shah, S., Smith, W., Strubell, E., Subramani, N., Wortsman, M., Dasigi, P., Lambert, N., Richardson, K., Zettlemoyer, L., Dodge, J., Lo, K., Soldaini, L., Smith, N. A., and Hajishirzi, H. (2024). Olmo: Accelerating the science of language models.

Gu, J., Jiang, X., Shi, Z., Tan, H., Zhai, X., Xu, C., Li, W., Shen, Y., Ma, S., Liu, H., Wang, Y., and Guo, J. (2024). A survey on llm-as-a-judge. *arXiv preprint arXiv: 2411.15594*.

Gugger, S., Debut, L., Wolf, T., Schmid, P., Mueller, Z., Mangrulkar, S., Sun, M., and Bossan, B. (2022). Accelerate: Training and inference at scale made simple, efficient and adaptable. https://github.com/huggingface/accelerate.

Gunasekar, S., Zhang, Y., Aneja, J., Mendes, C. C. T., Del Giorno, A., Gopi, S., Javaheripi, M., Kauffmann, P., de Rosa, G., Saarikivi, O., et al. (2023). Textbooks are all you need. *arXiv preprint arXiv:2306.11644*.

Gupta, S., Kenkre, S., and Talukdar, P. (2019). CaRe: Open knowledge graph embeddings. In Inui, K., Jiang, J., Ng, V., and Wan, X., editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 378–388, Hong Kong, China. Association for Computational Linguistics.

Guu, K., Lee, K., Tung, Z., Pasupat, P., and Chang, M. (2020). Retrieval augmented language model pre-training. In III, H. D. and Singh, A., editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 3929–3938. PMLR.

Han, S., Rao, K., Ettinger, A., Jiang, L., Lin, B. Y., Lambert, N., Choi, Y., and Dziri, N. (2024). Wildguard: Open one-stop moderation tools for safety risks, jailbreaks, and refusals of llms.

In Globerson, A., Mackey, L., Belgrave, D., Fan, A., Paquet, U., Tomczak, J., and Zhang, C., editors, *Advances in Neural Information Processing Systems*, volume 37, pages 8093–8131. Curran Associates, Inc.

Henderson, J. (2020). The unstoppable rise of computational linguistics in deep learning. In Jurafsky, D., Chai, J., Schluter, N., and Tetreault, J., editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6294–6306, Online. Association for Computational Linguistics.

Henderson, J., Mohammadshahi, A., Coman, A., and Miculicich, L. (2023). Transformers as graph-to-graph models. In Elazar, Y., Ettinger, A., Kassner, N., Ruder, S., and A. Smith, N., editors, *Proceedings of the Big Picture Workshop*, pages 93–107, Singapore. Association for Computational Linguistics.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9:1735–1780.

Hsu, P.-L., Dai, Y., Kothapalli, V., Song, Q., Tang, S., Zhu, S., Shimizu, S., Sahni, S., Ning, H., and Chen, Y. (2025). Liger kernel: Efficient triton kernels for llm training.

Hu, M., Peng, Y., Huang, Z., and Li, D. (2019). Retrieve, read, rerank: Towards end-to-end multi-document reading comprehension. In Korhonen, A., Traum, D., and Màrquez, L., editors, *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2285–2295, Florence, Italy. Association for Computational Linguistics.

Iyer, S., Min, S., Mehdad, Y., and Yih, W.-t. (2021). RECONSIDER: Improved re-ranking using span-focused cross-attention for open domain question answering. In Toutanova, K., Rumshisky, A., Zettlemoyer, L., Hakkani-Tur, D., Beltagy, I., Bethard, S., Cotterell, R., Chakraborty, T., and Zhou, Y., editors, *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1280–1287, Online. Association for Computational Linguistics.

Izacard, G. and Grave, E. (2021a). Distilling knowledge from reader to retriever for question answering. In *International Conference on Learning Representations*.

Izacard, G. and Grave, E. (2021b). Leveraging passage retrieval with generative models for open domain question answering. In Merlo, P., Tiedemann, J., and Tsarfaty, R., editors, *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 874–880, Online. Association for Computational Linguistics.

Izacard, G., Lewis, P., Lomeli, M., Hosseini, L., Petroni, F., Schick, T., Dwivedi-Yu, J., Joulin, A., Riedel, S., and Grave, E. (2023). Atlas: Few-shot learning with retrieval augmented language models. *Journal of Machine Learning Research*, 24(251):1–43.

Izacard, G., Petroni, F., Hosseini, L., Cao, N. D., Riedel, S., and Grave, E. (2020). A memory efficient baseline for open domain question answering. *ArXiv*, abs/2012.15156.

Ji, S., Pan, S., Cambria, E., Marttinen, P., and Yu, P. S. (2020). A survey on knowledge graphs: Representation, acquisition, and applications. *IEEE Transactions on Neural Networks and Learning Systems*, 33:494–514.

Jia, R., Wong, C., and Poon, H. (2019). Document-level n-ary relation extraction with multiscale representation learning. In Burstein, J., Doran, C., and Solorio, T., editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3693–3704, Minneapolis, Minnesota. Association for Computational Linguistics.

Jiang, A. Q., Sablayrolles, A., Mensch, A., Bamford, C., Chaplot, D. S., de Las Casas, D., Bressand, F., Lengyel, G., Lample, G., Saulnier, L., Lavaud, L. R., Lachaux, M.-A., Stock, P., Scao, T. L., Lavril, T., Wang, T., Lacroix, T., and Sayed, W. E. (2023). Mistral 7b. *ArXiv*, abs/2310.06825.

Jiang, J., Zhou, K., Zhao, X., and Wen, J.-R. (2022). Unikgqa: Unified retrieval and reasoning for solving multi-hop question answering over knowledge graph. In *The Eleventh International Conference on Learning Representations*.

Jiang, X., Wang, Q., and Wang, B. (2019). Adaptive convolution for multi-relational learning. In Burstein, J., Doran, C., and Solorio, T., editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 978–987, Minneapolis, Minnesota. Association for Computational Linguistics.

Jin, Z., Yuan, H., Men, T., Cao, P., Chen, Y., Liu, K., and Zhao, J. (2024). Rag-rewardbench: Benchmarking reward models in retrieval augmented generation for preference alignment.

Johnson, J., Douze, M., and Jégou, H. (2021). Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*, 7(3):535–547.

Jordan, M. I. (1997). Serial order: A parallel distributed processing approach. *Advances in psychology*, 121:471–495.

Jumper, J. M., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., Tunyasuvunakool, K., Bates, R., Žídek, A., Potapenko, A., Bridgland, A., Meyer, C., Kohl, S. A. A., Ballard, A., Cowie, A., Romera-Paredes, B., Nikolov, S., Jain, R., Adler, J., Back, T., Petersen, S., Reiman, D., Clancy, E., Zielinski, M., Steinegger, M., Pacholska, M., Berghammer, T., Bodenstein, S., Silver, D., Vinyals, O., Senior, A. W., Kavukcuoglu, K., Kohli, P., and Hassabis, D. (2021). Highly accurate protein structure prediction with alphafold. *Nature*, 596:583 – 589.

Kalajdzievski, D. (2023). A rank stabilization scaling factor for fine-tuning with lora.

Karpukhin, V., Oguz, B., Min, S., Lewis, P., Wu, L., Edunov, S., Chen, D., and Yih, W.-t. (2020). Dense passage retrieval for open-domain question answering. In Webber, B., Cohn, T., He, Y., and Liu, Y., editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural*

*Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.

Kazemi, S. M. and Poole, D. (2018). Simple embedding for link prediction in knowledge graphs. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.

Ke, P., Ji, H., Ran, Y., Cui, X., Wang, L., Song, L., Zhu, X., and Huang, M. (2021). JointGT: Graph-text joint representation learning for text generation from knowledge graphs. In Zong, C., Xia, F., Li, W., and Navigli, R., editors, *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2526–2538, Online. Association for Computational Linguistics.

Kim, S., Shin, J., Cho, Y., Jang, J., Longpre, S., Lee, H., Yun, S., Shin, S., Kim, S., Thorne, J., and Seo, M. (2024a). Prometheus: Inducing fine-grained evaluation capability in language models.

Kim, S., Suk, J., Longpre, S., Lin, B. Y., Shin, J., Welleck, S., Neubig, G., Lee, M., Lee, K., and Seo, M. (2024b). Prometheus 2: An open source language model specialized in evaluating other language models. In Al-Onaizan, Y., Bansal, M., and Chen, Y.-N., editors, *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 4334–4353, Miami, Florida, USA. Association for Computational Linguistics.

Kipf, T. N. and Welling, M. (2017). Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations (ICLR)*.

Kwon, W., Li, Z., Zhuang, S., Sheng, Y., Zheng, L., Yu, C. H., Gonzalez, J. E., Zhang, H., and Stoica, I. (2023). Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*.

Lambert, N., Pyatkin, V., Morrison, J., Miranda, L., Lin, B. Y., Chandu, K., Dziri, N., Kumar, S., Zick, T., Choi, Y., Smith, N. A., and Hajishirzi, H. (2025). RewardBench: Evaluating reward models for language modeling. In Chiruzzo, L., Ritter, A., and Wang, L., editors, *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 1755–1797, Albuquerque, New Mexico. Association for Computational Linguistics.

Lee, K., Chang, M.-W., and Toutanova, K. (2019). Latent retrieval for weakly supervised open domain question answering. In Korhonen, A., Traum, D., and Màrquez, L., editors, *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6086–6096, Florence, Italy. Association for Computational Linguistics.

Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., and Zettlemoyer, L. (2020a). BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In Jurafsky, D., Chai, J., Schluter, N., and Tetreault, J., editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.-t., Rocktäschel, T., Riedel, S., and Kiela, D. (2020b). Retrieval-augmented generation for knowledge-intensive nlp tasks. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H., editors, *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474. Curran Associates, Inc.

Lhoest, Q., del Moral, A. V., Jernite, Y., Thakur, A., von Platen, P., Patil, S., Chaumond, J., Drame, M., Plu, J., Tunstall, L., Davison, J., Šaško, M., Chhablani, G., Malik, B., Brandeis, S., Scao, T. L., Sanh, V., Xu, C., Patry, N., McMillan-Major, A., Schmid, P., Gugger, S., Delangue, C., Matussière, T., Debut, L., Bekman, S., Cistac, P., Goehringer, T., Mustar, V., Lagunas, F., Rush, A. M., and Wolf, T. (2021). Datasets: A community library for natural language processing.

Li, D., Jiang, B., Huang, L., Beigi, A., Zhao, C., Tan, Z., Bhattacharjee, A., Jiang, Y., Chen, C., Wu, T., Shu, K., Cheng, L., and Liu, H. (2024a). From generation to judgment: Opportunities and challenges of llm-as-a-judge. *arXiv preprint arXiv: 2411.16594*.

Li, Z., Xu, X., Shen, T., Xu, C., Gu, J.-C., Lai, Y., Tao, C., and Ma, S. (2024b). Leveraging large language models for NLG evaluation: Advances and challenges. In Al-Onaizan, Y., Bansal, M., and Chen, Y.-N., editors, *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 16028–16045, Miami, Florida, USA. Association for Computational Linguistics.

Lin, C.-Y. (2004). ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Lin, Y., Liu, Z., Sun, M., Liu, Y., and Zhu, X. (2015). Learning entity and relation embeddings for knowledge graph completion. *Proceedings of the AAAI Conference on Artificial Intelligence*, 29(1).

Liu, C. Y., Zeng, L., Liu, J., Yan, R., He, J., Wang, C., Yan, S., Liu, Y., and Zhou, Y. (2024). Skywork-reward: Bag of tricks for reward modeling in llms. *arXiv preprint arXiv:2410.18451*.

Liu, L., Jiang, H., He, P., Chen, W., Liu, X., Gao, J., and Han, J. (2020). On the variance of the adaptive learning rate and beyond. In *International Conference on Learning Representations*.

Liu, Y., Iter, D., Xu, Y., Wang, S., Xu, R., and Zhu, C. (2023). G-eval: NLG evaluation using gpt-4 with better human alignment. In Bouamor, H., Pino, J., and Bali, K., editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2511–2522, Singapore. Association for Computational Linguistics.

Logan, R., Liu, N. F., Peters, M. E., Gardner, M., and Singh, S. (2019). Barack's wife hillary: Using knowledge graphs for fact-aware language modeling. In Korhonen, A., Traum, D., and Màrquez, L., editors, *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5962–5971, Florence, Italy. Association for Computational Linguistics.

Loshchilov, I. and Hutter, F. (2019). Decoupled weight decay regularization.

Lu, J., Clark, C., Lee, S., Zhang, Z., Khosla, S., Marten, R., Hoiem, D., and Kembhavi, A. (2023). Unified-io 2: Scaling autoregressive multimodal models with vision, language, audio, and action.

Lu, J., Clark, C., Zellers, R., Mottaghi, R., and Kembhavi, A. (2022). Unified-io: A unified model for vision, language, and multi-modal tasks.

Ma, Y., Wang, A., and Okazaki, N. (2023). DREEAM: Guiding attention with evidence for improving document-level relation extraction. In Vlachos, A. and Augenstein, I., editors, *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 1971–1983, Dubrovnik, Croatia. Association for Computational Linguistics.

Mahan, D., Phung, D., Rafailov, R., Blagden, C., nathan lile, Castricato, L., Franken, J.-P., Finn, C., and Albalak, A. (2024). Generative reward models. *ArXiv*, abs/2410.12832.

Malaviya, C., Bhagavatula, C., Bosselut, A., and Choi, Y. (2020). Commonsense knowledge base completion with structural and semantic context. *Proceedings of the 34th AAAI Conference on Artificial Intelligence*.

Mangrulkar, S., Gugger, S., Debut, L., Belkada, Y., Paul, S., and Bossan, B. (2022). Peft: State-of-the-art parameter-efficient fine-tuning methods. https://github.com/huggingface/peft.

Manning, C. D., Raghavan, P., and Schütze, H. (2008). *Introduction to Information Retrieval*. Cambridge University Press.

Mao, Y., He, P., Liu, X., Shen, Y., Gao, J., Han, J., and Chen, W. (2021). Reader-guided passage reranking for open-domain question answering. In Zong, C., Xia, F., Li, W., and Navigli, R., editors, *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 344–350, Online. Association for Computational Linguistics.

Markowitz, E., Balasubramanian, K., Mirtaheri, M., Annavaram, M., Galstyan, A., and Ver Steeg, G. (2022). StATIK: Structure and text for inductive knowledge graph completion. In Carpuat, M., de Marneffe, M.-C., and Meza Ruiz, I. V., editors, *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 604–615, Seattle, United States. Association for Computational Linguistics.

Miculicich, L. and Henderson, J. (2022). Graph refinement for coreference resolution. In Muresan, S., Nakov, P., and Villavicencio, A., editors, *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2732–2742, Dublin, Ireland. Association for Computational Linguistics.

Min, S., Chen, D., Zettlemoyer, L., and Hajishirzi, H. (2019). Knowledge guided text retrieval and reading for open domain question answering. *ArXiv*, abs/1911.03868.

Mintz, M., Bills, S., Snow, R., and Jurafsky, D. (2009). Distant supervision for relation extraction without labeled data. In Su, K.-Y., Su, J., Wiebe, J., and Li, H., editors, *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 1003–1011, Suntec, Singapore. Association for Computational Linguistics.

MIT (1987). The mit license.

Mohammadshahi, A. and Henderson, J. (2020). Graph-to-graph transformer for transition-based dependency parsing. In Cohn, T., He, Y., and Liu, Y., editors, *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3278–3289, Online. Association for Computational Linguistics.

Mohammadshahi, A. and Henderson, J. (2021). Recursive non-autoregressive graph-to-graph transformer for dependency parsing with iterative refinement. *Transactions of the Association for Computational Linguistics*, 9:120–138.

Mohammadshahi, A. and Henderson, J. (2023). Syntax-aware graph-to-graph transformer for semantic role labelling. In Can, B., Mozes, M., Cahyawijaya, S., Saphra, N., Kassner, N., Ravfogel, S., Ravichander, A., Zhao, C., Augenstein, I., Rogers, A., Cho, K., Grefenstette, E., and Voita, L., editors, *Proceedings of the 8th Workshop on Representation Learning for NLP (RepL4NLP 2023)*, pages 174–186, Toronto, Canada. Association for Computational Linguistics.

Muller, S., Loison, A., Omrani, B., and Viaud, G. (2025). GroUSE: A benchmark to evaluate evaluators in grounded question answering. In Rambow, O., Wanner, L., Apidianaki, M., Al-Khalifa, H., Eugenio, B. D., and Schockaert, S., editors, *Proceedings of the 31st International Conference on Computational Linguistics*, pages 4510–4534, Abu Dhabi, UAE. Association for Computational Linguistics.

Nan, G., Guo, Z., Sekulic, I., and Lu, W. (2020). Reasoning with latent structure refinement for document-level relation extraction. In Jurafsky, D., Chai, J., Schluter, N., and Tetreault, J., editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1546–1557, Online. Association for Computational Linguistics.

Neyman, J. (1934). On the two different aspects of the representative method: the method of stratified sampling and the method of purposive selection. *Journal of the Royal Statistical Society*, 97:123–150.

Nickel, M., Murphy, K., Tresp, V., and Gabrilovich, E. (2015). A review of relational machine learning for knowledge graphs. *Proceedings of the IEEE*, 104(1):11–33.

Nickel, M., Tresp, V., and Kriegel, H.-P. (2011). A three-way model for collective learning on multi-relational data. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, ICML'11, page 809–816, Madison, WI, USA. Omnipress.

Niu, G., Li, B., Zhang, Y., and Pu, S. (2022). CAKE: A scalable commonsense-aware framework for multi-view knowledge graph completion. In Muresan, S., Nakov, P., and Villavicencio, A., editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2867–2877, Dublin, Ireland. Association for Computational Linguistics.

Oddy, R. N. (1977). Information retrieval through man-machine dialogue. *Journal of Documentation*, 33:1–14.

OpenAI, :, Jaech, A., Kalai, A., Lerer, A., Richardson, A., El-Kishky, A., Low, A., Helyar, A., Madry, A., Beutel, A., Carney, A., Iftimie, A., Karpenko, A., Passos, A. T., Neitz, A., Prokofiev, A., Wei, A., Tam, A., Bennett, A., Kumar, A., Saraiva, A., Vallone, A., Duberstein, A., Kondrich, A., Mishchenko, A., Applebaum, A., Jiang, A., Nair, A., Zoph, B., Ghorbani, B., Rossen, B., Sokolowsky, B., Barak, B., McGrew, B., Minaiev, B., Hao, B., Baker, B., Houghton, B., McKinzie, B., Eastman, B., Lugaresi, C., Bassin, C., Hudson, C., Li, C. M., de Bourcy, C., Voss, C., Shen, C., Zhang, C., Koch, C., Orsinger, C., Hesse, C., Fischer, C., Chan, C., Roberts, D., Kappler, D., Levy, D., Selsam, D., Dohan, D., Farhi, D., Mely, D., Robinson, D., Tsipras, D., Li, D., Oprica, D., Freeman, E., Zhang, E., Wong, E., Proehl, E., Cheung, E., Mitchell, E., Wallace, E., Ritter, E., Mays, E., Wang, F., Such, F. P., Raso, F., Leoni, F., Tsimpourlas, F., Song, F., von Lohmann, F., Sulit, F., Salmon, G., Parascandolo, G., Chabot, G., Zhao, G., Brockman, G., Leclerc, G., Salman, H., Bao, H., Sheng, H., Andrin, H., Bagherinezhad, H., Ren, H., Lightman, H., Chung, H. W., Kivlichan, I., O'Connell, I., Osband, I., Gilaberte, I. C., Akkaya, I., Kostrikov, I., Sutskever, I., Kofman, I., Pachocki, J., Lennon, J., Wei, J., Harb, J., Twore, J., Feng, J., Yu, J., Weng, J., Tang, J., Yu, J., Candela, J. Q., Palermo, J., Parish, J., Heidecke, J., Hallman, J., Rizzo, J., Gordon, J., Uesato, J., Ward, J., Huizinga, J., Wang, J., Chen, K., Xiao, K., Singhal, K., Nguyen, K., Cobbe, K., Shi, K., Wood, K., Rimbach, K., Gu-Lemberg, K., Liu, K., Lu, K., Stone, K., Yu, K., Ahmad, L., Yang, L., Liu, L., Maksin, L., Ho, L., Fedus, L., Weng, L., Li, L., McCallum, L., Held, L., Kuhn, L., Kondraciuk, L., Kaiser, L., Metz, L., Boyd, M., Trebacz, M., Joglekar, M., Chen, M., Tintor, M., Meyer, M., Jones, M., Kaufer, M., Schwarzer, M., Shah, M., Yatbaz, M., Guan, M. Y., Xu, M., Yan, M., Glaese, M., Chen, M., Lampe, M., Malek, M., Wang, M., Fradin, M., McClay, M., Pavlov, M., Wang, M., Wang, M., Murati, M., Bavarian, M., Rohaninejad, M., McAleese, N., Chowdhury, N., Chowdhury, N., Ryder, N., Tezak, N., Brown, N., Nachum, O., Boiko, O., Murk, O., Watkins, O., Chao, P., Ashbourne, P., Izmailov, P., Zhokhov, P., Dias, R., Arora, R., Lin, R., Lopes, R. G., Gaon, R., Miyara, R., Leike, R., Hwang, R., Garg, R., Brown, R., James, R., Shu, R., Cheu, R., Greene, R., Jain, S., Altman, S., Toizer, S., Toyer, S., Miserendino, S., Agarwal, S., Hernandez, S., Baker, S., McKinney, S., Yan, S., Zhao, S., Hu, S., Santurkar, S., Chaudhuri, S. R., Zhang, S., Fu, S., Papay, S., Lin, S., Balaji, S., Sanjeev, S., Sidor, S., Broda, T., Clark, A., Wang, T., Gordon, T., Sanders, T., Patwardhan, T., Sottiaux, T., Degry, T., Dimson, T., Zheng, T., Garipov, T., Stasi, T., Bansal, T., Creech, T., Peterson, T., Eloundou, T., Qi, V., Kosaraju, V., Monaco, V., Pong, V., Fomenko, V., Zheng, W., Zhou, W., McCabe, W., Zaremba, W., Dubois, Y., Lu, Y., Chen, Y., Cha, Y., Bai, Y., He, Y., Zhang, Y., Wang, Y., Shao, Z., and Li, Z. (2024). Openai o1 system card.

OpenAI (2025). Introducing deep research. https://openai.com/index/introducing-deep-research/.

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., Schulman, J., Hilton, J., Kelton, F., Miller, L., Simens, M., Askell, A., Welinder, P., Christiano, P. F., Leike, J., and Lowe, R. (2022). Training language models to follow instructions with human feedback. In Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., and Oh, A., editors, *Advances in Neural Information Processing Systems*, volume 35, pages 27730–27744. Curran Associates, Inc.

Pang, B. and Lee, L. (2008). Opinion mining and sentiment analysis. *Found. Trends Inf. Retr.*, 2:1–135.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). Bleu: a method for automatic evaluation of machine translation. In Isabelle, P., Charniak, E., and Lin, D., editors, *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Park, J., Jwa, S., Meiying, R., Kim, D., and Choi, S. (2024). OffsetBias: Leveraging debiased data for tuning evaluators. In Al-Onaizan, Y., Bansal, M., and Chen, Y.-N., editors, *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 1043–1067, Miami, Florida, USA. Association for Computational Linguistics.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. (2019). PyTorch: An Imperative Style, High-Performance Deep Learning Library. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché Buc, F., Fox, E., and Garnett, R., editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

Peng, N., Poon, H., Quirk, C., Toutanova, K., and Yih, W.-t. (2017). Cross-sentence n-ary relation extraction with graph LSTMs. *Transactions of the Association for Computational Linguistics*, 5:101–115.

Perplexity AI (2025). Introducing perplexity deep research. https://www.perplexity.ai/hub/blog/introducing-perplexity-deep-research/.

Petroni, F., Lewis, P., Piktus, A., Rocktäschel, T., Wu, Y., Miller, A. H., and Riedel, S. (2020). How context affects language models' factual predictions. In *Automated Knowledge Base Construction*.

Petroni, F., Rocktäschel, T., Riedel, S., Lewis, P., Bakhtin, A., Wu, Y., and Miller, A. (2019). Language models as knowledge bases? In Inui, K., Jiang, J., Ng, V., and Wan, X., editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2463–2473, Hong Kong, China. Association for Computational Linguistics.

Qu, C., Yang, L., Chen, C., Qiu, M., Croft, W. B., and Iyyer, M. (2020a). Open-retrieval conversational question answering. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '20, page 539–548, New York, NY, USA. Association for Computing Machinery.

Qu, C., Yang, L., Chen, C.-C., Qiu, M., Croft, W. B., and Iyyer, M. (2020b). Open-retrieval conversational question answering. *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*.

Qu, Y., Ding, Y., Liu, J., Liu, K., Ren, R., Zhao, W. X., Dong, D., Wu, H., and Wang, H. (2021). RocketQA: An optimized training approach to dense passage retrieval for open-domain question answering. In Toutanova, K., Rumshisky, A., Zettlemoyer, L., Hakkani-Tur, D., Beltagy, I., Bethard, S., Cotterell, R., Chakraborty, T., and Zhou, Y., editors, *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5835–5847, Online. Association for Computational Linguistics.

Qwen, :, Yang, A., Yang, B., Zhang, B., Hui, B., Zheng, B., Yu, B., Li, C., Liu, D., Huang, F., Wei, H., Lin, H., Yang, J., Tu, J., Zhang, J., Yang, J., Yang, J., Zhou, J., Lin, J., Dang, K., Lu, K., Bao, K., Yang, K., Yu, L., Li, M., Xue, M., Zhang, P., Zhu, Q., Men, R., Lin, R., Li, T., Tang, T., Xia, T., Ren, X., Ren, X., Fan, Y., Su, Y., Zhang, Y., Wan, Y., Liu, Y., Cui, Z., Zhang, Z., and Qiu, Z. (2025). Qwen2.5 technical report.

Radford, A., Kim, J. W., Xu, T., Brockman, G., McLeavey, C., and Sutskever, I. (2022). Robust speech recognition via large-scale weak supervision. In *International Conference on Machine Learning*.

Rafailov, R., Sharma, A., Mitchell, E., Manning, C. D., Ermon, S., and Finn, C. (2023). Direct preference optimization: Your language model is secretly a reward model. In Oh, A., Naumann, T., Globerson, A., Saenko, K., Hardt, M., and Levine, S., editors, *Advances in Neural Information Processing Systems*, volume 36, pages 53728–53741. Curran Associates, Inc.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Rashiti, G., Karunaratne, G., Sachen, M., Sebastian, A., and Rahimi, A. (2024). Retro-Li: Small-Scale Retrieval Augmented Generation Supporting Noisy Similarity Searches and Domain Shift Generalization. In *Proceedings of the 27th European Conference on Artificial Intelligence*.

Reimers, N. and Gurevych, I. (2019). Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In Inui, K., Jiang, J., Ng, V., and Wan, X., editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Roberts, A., Raffel, C., and Shazeer, N. (2020). How much knowledge can you pack into the parameters of a language model? In Webber, B., Cohn, T., He, Y., and Liu, Y., editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5418–5426, Online. Association for Computational Linguistics.

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

Safavi, T. and Koutra, D. (2021). Relational World Knowledge Representation in Contextual Language Models: A Review. In Moens, M.-F., Huang, X., Specia, L., and Yih, S. W.-t., editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1053–1067, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Saha, S., Li, X., Ghazvininejad, M., Weston, J., and Wang, T. (2025). Learning to plan & reason for evaluation with thinking-llm-as-a-judge.

Sai, A. B., Mohankumar, A. K., and Khapra, M. M. (2022). A survey of evaluation metrics used for nlg systems. *ACM Comput. Surv.*, 55(2).

Saxena, A., Kochsiek, A., and Gemulla, R. (2022). Sequence-to-sequence knowledge graph completion and question answering. In Muresan, S., Nakov, P., and Villavicencio, A., editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2814–2828, Dublin, Ireland. Association for Computational Linguistics.

Schlichtkrull, M., Kipf, T., Bloem, P., van den Berg, R., Titov, I., and Welling, M. (2017). Modeling relational data with graph convolutional networks. In *Extended Semantic Web Conference*.

Schneider, P., Schopf, T., Vladika, J., Galkin, M., Simperl, E., and Matthes, F. (2022). A decade of knowledge graphs in natural language processing: A survey. In He, Y., Ji, H., Li, S., Liu, Y., and Chang, C.-H., editors, *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 (Volume 1: Long Papers)*, pages 601–614, Online only. Association for Computational Linguistics.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. (2017). Proximal policy optimization algorithms. *CoRR*, abs/1707.06347.

Shah, H., Villmow, J., Ulges, A., Schwanecke, U., and Shafait, F. (2019). An open-world extension to knowledge graph completion models. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 3044–3051.

Shao, Z., Wang, P., Zhu, Q., Xu, R., Song, J., Bi, X., Zhang, H., Zhang, M., Li, Y. K., Wu, Y., and Guo, D. (2024). Deepseekmath: Pushing the limits of mathematical reasoning in open language models.

Shazeer, N. M. (2020). Glu variants improve transformer. *ArXiv*, abs/2002.05202.

Shen, X., Vakulenko, S., del Tredici, M., Barlacchi, G., Byrne, B., and de Gispert, A. (2023). Neural ranking with weak supervision for open-domain question answering : A survey. In Vlachos, A. and Augenstein, I., editors, *Findings of the Association for Computational Linguistics: EACL 2023*, pages 1736–1750, Dubrovnik, Croatia. Association for Computational Linguistics.

Shi, B. and Weninger, T. (2018). Open-world knowledge graph completion. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32.

Socher, R., Chen, D., Manning, C. D., and Ng, A. (2013). Reasoning with neural tensor networks for knowledge base completion. In Burges, C., Bottou, L., Welling, M., Ghahramani, Z., and Weinberger, K., editors, *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc.

Song, Y., Kim, Y., and Iyyer, M. (2024). VeriScore: Evaluating the factuality of verifiable claims in long-form text generation. In Al-Onaizan, Y., Bansal, M., and Chen, Y.-N., editors, *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 9447–9474, Miami, Florida, USA. Association for Computational Linguistics.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958.

Su, J., Lu, Y., Pan, S., Wen, B., and Liu, Y. (2021). Roformer: Enhanced transformer with rotary position embedding. *ArXiv*, abs/2104.09864.

Sun, Z., Deng, Z.-H., Nie, J.-Y., and Tang, J. (2019). Rotate: Knowledge graph embedding by relational rotation in complex space. *arXiv preprint arXiv:1902.10197*.

Tan, Q., He, R., Bing, L., and Ng, H. T. (2022a). Document-level relation extraction with adaptive focal loss and knowledge distillation. In Muresan, S., Nakov, P., and Villavicencio, A., editors, *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1672–1681, Dublin, Ireland. Association for Computational Linguistics.

Tan, Q., Xu, L., Bing, L., Ng, H. T., and Aljunied, S. M. (2022b). Revisiting DocRED - addressing the false negative problem in relation extraction. In Goldberg, Y., Kozareva, Z., and Zhang, Y., editors, *Proceedings of the 2022 Conference on Empirical Methods in Natural*

*Language Processing*, pages 8472–8487, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Tang, H., Cao, Y., Zhang, Z., Cao, J., Fang, F., Wang, S., and Yin, P. (2020). Hin: Hierarchical inference network for document-level relation extraction. In *Advances in Knowledge Discovery and Data Mining: 24th Pacific-Asia Conference, PAKDD 2020, Singapore, May 11–14, 2020, Proceedings, Part I 24*, pages 197–209. Springer.

Tang, L., Laban, P., and Durrett, G. (2024). MiniCheck: Efficient fact-checking of LLMs on grounding documents. In Al-Onaizan, Y., Bansal, M., and Chen, Y.-N., editors, *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 8818–8847, Miami, Florida, USA. Association for Computational Linguistics.

Team, R., Ormazabal, A., Zheng, C., de Masson d'Autume, C., Yogatama, D., Fu, D., Ong, D., Chen, E., Lamprecht, E., Pham, H., Ong, I., Aleksiev, K., Li, L., Henderson, M., Bain, M., Artetxe, M., Relan, N., Padlewski, P., Liu, Q., Chen, R., Phua, S., Yang, Y., Tay, Y., Wang, Y., Zhu, Z., and Xie, Z. (2024). Reka core, flash, and edge: A series of powerful multimodal language models.

Teru, K. K., Denis, E. G., and Hamilton, W. L. (2020). Inductive relation prediction by subgraph reasoning. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org.

Theodoropoulos, C., Coman, A. C., Henderson, J., and Moens, M.-F. (2024). Enhancing biomedical knowledge discovery for diseases: An open-source framework applied on rett syndrome and alzheimer's disease. *IEEE Access*, 12:180652–180673.

Theodoropoulos, C., Coman, A. C., Henderson, J., and Moens, M.-F. (2025). Reduction of supervision for biomedical knowledge discovery. *BMC Bioinformatics*, 26.

Theodoropoulos, C., Henderson, J., Coman, A. C., and Moens, M.-F. (2021). Imposing relation structure in language-model embeddings using contrastive learning. In Bisazza, A. and Abend, O., editors, *Proceedings of the 25th Conference on Computational Natural Language Learning*, pages 337–348, Online. Association for Computational Linguistics.

Tonmoy, S. M. T. I., Zaman, S. M. M., Jain, V., Rani, A., Rawte, V., Chadha, A., and Das, A. (2024). A comprehensive survey of hallucination mitigation techniques in large language models.

Toutanova, K. and Chen, D. (2015). Observed versus latent features for knowledge base and text inference. In Allauzen, A., Grefenstette, E., Hermann, K. M., Larochelle, H., and Yih, S. W.-t., editors, *Proceedings of the 3rd Workshop on Continuous Vector Space Models and their Compositionality*, pages 57–66, Beijing, China. Association for Computational Linguistics.

Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., Rodriguez, A., Joulin, A., Grave, E., and Lample, G. (2023). Llama: Open and efficient foundation language models. *ArXiv*, abs/2302.13971.

TREC-RAG (2024). Trec retrieval-augmented generation.

Trouillon, T., Welbl, J., Riedel, S., Gaussier, É., and Bouchard, G. (2016). Complex embeddings for simple link prediction. In *International conference on machine learning*, pages 2071–2080. PMLR.

Turc, I., Chang, M.-W., Lee, K., and Toutanova, K. (2019). Well-read students learn better: On the importance of pre-training compact models. *arXiv: Computation and Language*.

Vakulenko, S., Longpre, S., Tu, Z., and Anantha, R. (2021). Question rewriting for conversational question answering. In *Proceedings of the 14th ACM international conference on web search and data mining*, pages 355–363.

Vaswani, A., Shazeer, N. M., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention is all you need. In *NIPS*.

von Werra, L., Belkada, Y., Tunstall, L., Beeching, E., Thrush, T., Lambert, N., Huang, S., Rasul, K., and Gallouédec, Q. (2020). Trl: Transformer reinforcement learning. https://github.com/huggingface/trl.

Vu, T., Krishna, K., Alzubi, S., Tar, C., Faruqui, M., and Sung, Y.-H. (2024). Foundational autoraters: Taming large language models for better automatic evaluation. In Al-Onaizan, Y., Bansal, M., and Chen, Y.-N., editors, *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 17086–17105, Miami, Florida, USA. Association for Computational Linguistics.

Wang, B., Shen, T., Long, G., Zhou, T., Wang, Y., and Chang, Y. (2021a). Structure-augmented text representation learning for efficient knowledge graph completion. In *Proceedings of the Web Conference 2021*, WWW '21, page 1737–1748, New York, NY, USA. Association for Computing Machinery.

Wang, D., Hu, W., Cao, E., and Sun, W. (2020a). Global-to-local neural networks for document-level relation extraction. In Webber, B., Cohn, T., He, Y., and Liu, Y., editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3711–3721, Online. Association for Computational Linguistics.

Wang, H., Focke, C., Sylvester, R., Mishra, N., and Wang, W. Y. (2019a). Fine-tune bert for docred with two-step process. *ArXiv*, abs/1909.11898.

Wang, H., Ren, H., and Leskovec, J. (2020b). Relational message passing for knowledge graph completion. *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*.

Wang, L., Zhao, W., Wei, Z., and Liu, J. (2022). SimKGC: Simple contrastive knowledge graph completion with pre-trained language models. In Muresan, S., Nakov, P., and Villavicencio, A., editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4281–4294, Dublin, Ireland. Association for Computational Linguistics.

Wang, P., Han, J., Li, C., and Pan, R. (2019b). Logic attention based neighborhood aggregation for inductive knowledge graph embedding. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence and Thirty-First Innovative Applications of Artificial Intelligence Conference and Ninth AAAI Symposium on Educational Advances in Artificial Intelligence*, AAAI'19/IAAI'19/EAAI'19. AAAI Press.

Wang, P., Li, L., Chen, L., Cai, Z., Zhu, D., Lin, B., Cao, Y., Kong, L., Liu, Q., Liu, T., and Sui, Z. (2024a). Large language models are not fair evaluators. In Ku, L.-W., Martins, A., and Srikumar, V., editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9440–9450, Bangkok, Thailand. Association for Computational Linguistics.

Wang, P., Xu, A., Zhou, Y., Xiong, C., and Joty, S. (2024b). Direct judgement preference optimization.

Wang, Q., Huang, P., Wang, H., Dai, S., Jiang, W., Liu, J., Lyu, Y., Zhu, Y., and Wu, H. (2019c). Coke: Contextualized knowledge graph embedding. *arXiv preprint arXiv:1911.02168*.

Wang, Q., Mao, Z., Wang, B., and Guo, L. (2017). Knowledge graph embedding: A survey of approaches and applications. *IEEE Transactions on Knowledge and Data Engineering*, 29(12):2724–2743.

Wang, T., Kulikov, I., Golovneva, O., Yu, P., Yuan, W., Dwivedi-Yu, J., Pang, R. Y., Fazel-Zarandi, M., Weston, J., and Li, X. (2024c). Self-taught evaluators.

Wang, X., Gao, T., Zhu, Z., Zhang, Z., Liu, Z., Li, J., and Tang, J. (2021b). KEPLER: A unified model for knowledge embedding and pre-trained language representation. *Transactions of the Association for Computational Linguistics*, 9:176–194.

Wang, Y., Yu, Z., Zeng, Z., Yang, L., Wang, C., Chen, H., Jiang, C., Xie, R., Wang, J., Xie, X., Ye, W., Zhang, S., and Zhang, Y. (2024d). Pandalm: An automatic evaluation benchmark for llm instruction tuning optimization.

Wang, Z., Dong, Y., Delalleau, O., Zeng, J., Shen, G., Egert, D., Zhang, J. J., Sreedhar, M. N., and Kuchaiev, O. (2024e). Helpsteer 2: Open-source dataset for training top-performing reward models. In Globerson, A., Mackey, L., Belgrave, D., Fan, A., Paquet, U., Tomczak, J., and Zhang, C., editors, *Advances in Neural Information Processing Systems*, volume 37, pages 1474–1501. Curran Associates, Inc.

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., Chi, E., Le, Q., and Zhou, D. (2023). Chain-of-thought prompting elicits reasoning in large language models.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Le Scao, T., Gugger, S., Drame, M., Lhoest, Q., and Rush, A. (2020). Transformers: State-of-the-art natural language processing. In Liu, Q. and Schlangen, D., editors, *Proceedings* *of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Xiao, Y., Zhang, Z., Mao, Y., Yang, C., and Han, J. (2022). SAIS: Supervising and augmenting intermediate steps for document-level relation extraction. In Carpuat, M., de Marneffe, M.-C., and Meza Ruiz, I. V., editors, *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2395–2409, Seattle, United States. Association for Computational Linguistics.

Xie, R., Liu, Z., Jia, J., Luan, H., and Sun, M. (2016). Representation learning of knowledge graphs with entity descriptions. In *AAAI Conference on Artificial Intelligence*.

Xiong, R., Yang, Y., He, D., Zheng, K., Zheng, S., Xing, C., Zhang, H., Lan, Y., Wang, L., and Liu, T.-Y. (2020). On layer normalization in the transformer architecture. *ArXiv*, abs/2002.04745.

Xu, A., Bansal, S., Ming, Y., Yavuz, S., and Joty, S. (2025a). Does context matter? contextual-judgebench for evaluating llm-based judges in contextual settings.

Xu, B., Wang, Q., Lyu, Y., Zhu, Y., and Mao, Z. (2021). Entity structure within and throughout: Modeling mention dependencies for document-level relation extraction. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(16):14149–14157.

Xu, J., Guo, Z., He, J., Hu, H., He, T., Bai, S., Chen, K., Wang, J., Fan, Y., Dang, K., Zhang, B., Wang, X., Chu, Y., and Lin, J. (2025b). Qwen2.5-omni technical report.

Xu, Z., Jiang, F., Niu, L., Deng, Y., Poovendran, R., Choi, Y., and Lin, B. Y. (2024). Magpie: Alignment data synthesis from scratch by prompting aligned llms with nothing. *ArXiv*, abs/2406.08464.

Yang, B., Yih, W.-t., He, X., Gao, J., and Deng, L. (2014). Embedding entities and relations for learning and inference in knowledge bases. *arXiv preprint arXiv:1412.6575*.

Yang, G., Liu, Y., Zhang, L., Zhang, L., Xie, H., and Mao, Z. (2024a). Knowledge context modeling with pre-trained language models for contrastive knowledge graph completion. In Ku, L.-W., Martins, A., and Srikumar, V., editors, *Findings of the Association for Computational Linguistics: ACL 2024*, pages 8619–8630, Bangkok, Thailand. Association for Computational Linguistics.

Yang, J., Gupta, A., Upadhyay, S., He, L., Goel, R., and Paul, S. (2022). TableFormer: Robust transformer modeling for table-text encoding. In Muresan, S., Nakov, P., and Villavicencio, A., editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 528–537, Dublin, Ireland. Association for Computational Linguistics.

Yang, L. F., Chen, H., Li, Z., Ding, X., and Wu, X. (2023). Give us the facts: Enhancing large language models with knowledge graphs for fact-aware language modeling. *IEEE Transactions on Knowledge and Data Engineering*, 36:3091–3110.

Yang, R., Ding, R., Lin, Y., Zhang, H., and Zhang, T. (2024b). Regularizing hidden states enables learning generalizable reward model for llms. *ArXiv*, abs/2406.10216.

Yang, W., Xie, Y., Lin, A., Li, X., Tan, L., Xiong, K., Li, M., and Lin, J. (2019). End-to-end open-domain question answering with BERTserini. In Ammar, W., Louis, A., and Mostafazadeh, N., editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 72–77, Minneapolis, Minnesota. Association for Computational Linguistics.

Yao, L., Mao, C., and Luo, Y. (2019a). Kg-bert: Bert for knowledge graph completion. *ArXiv*, abs/1909.03193.

Yao, Y., Ye, D., Li, P., Han, X., Lin, Y., Liu, Z., Liu, Z., Huang, L., Zhou, J., and Sun, M. (2019b). DocRED: A large-scale document-level relation extraction dataset. In Korhonen, A., Traum, D., and Màrquez, L., editors, *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 764–777, Florence, Italy. Association for Computational Linguistics.

Ye, D., Lin, Y., Li, P., and Sun, M. (2022). Packed levitated marker for entity and relation extraction. In Muresan, S., Nakov, P., and Villavicencio, A., editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4904–4917, Dublin, Ireland. Association for Computational Linguistics.

Ye, J., Wang, Y., Huang, Y., Chen, D., Zhang, Q., Moniz, N., Gao, T., Geyer, W., Huang, C., Chen, P.-Y., Chawla, N. V., and Zhang, X. (2024a). Justice or prejudice? quantifying biases in llm-as-a-judge.

Ye, S., Kim, D., Kim, S., Hwang, H., Kim, S., Jo, Y., Thorne, J., Kim, J., and Seo, M. (2024b). Flask: Fine-grained language model evaluation based on alignment skill sets.

Ye, Z., Li, X., Li, Q., Ai, Q., Zhou, Y., Shen, W., Yan, D., and Liu, Y. (2024c). Beyond scalar reward model: Learning generative judge from preference data.

Ying, C., Cai, T., Luo, S., Zheng, S., Ke, G., He, D., Shen, Y., and Liu, T.-Y. (2021). Do transformers really perform bad for graph representation? In *Neural Information Processing Systems*.

Yu, D., Zhu, C., Fang, Y., Yu, W., Wang, S., Xu, Y., Ren, X., Yang, Y., and Zeng, M. (2022). KG-FiD: Infusing knowledge graph in fusion-in-decoder for open-domain question answering. In Muresan, S., Nakov, P., and Villavicencio, A., editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4961–4974, Dublin, Ireland. Association for Computational Linguistics.

Yu, J., Wang, X., Tu, S., Cao, S., Zhang-Li, D., Lv, X., Peng, H., Yao, Z., Zhang, X., Li, H., Li, C., Zhang, Z., Bai, Y., Liu, Y., Xin, A., Yun, K., GONG, L., Lin, N., Chen, J., Wu, Z., Qi, Y., Li, W., Guan, Y., Zeng, K., Qi, J., Jin, H., Liu, J., Gu, Y., Yao, Y., Ding, N., Hou, L., Liu, Z., Bin, X., Tang, J., and Li, J. (2024). KoLA: Carefully benchmarking world knowledge of large language models. In *The Twelfth International Conference on Learning Representations*.

Yu, Y., Chen, Z., Zhang, A., Tan, L., Zhu, C., Pang, R. Y., Qian, Y., Wang, X., Gururangan, S., Zhang, C., Kambadur, M., Mahajan, D., and Hou, R. (2025). Self-generated critiques boost reward modeling for language models. In Chiruzzo, L., Ritter, A., and Wang, L., editors, *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 11499–11514, Albuquerque, New Mexico. Association for Computational Linguistics.

Yuan, W., Neubig, G., and Liu, P. (2021). Bartscore: Evaluating generated text as text generation. In Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W., editors, *Advances in Neural Information Processing Systems*, volume 34, pages 27263–27277. Curran Associates, Inc.

Zeng, S., Xu, R., Chang, B., and Li, L. (2020). Double graph based reasoning for document-level relation extraction. In Webber, B., Cohn, T., He, Y., and Liu, Y., editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1630–1640, Online. Association for Computational Linguistics.

Zha, H., Chen, Z., and Yan, X. (2021). Inductive relation prediction by bert. *ArXiv*, abs/2103.07102.

Zhang, H., Liu, Z., Xiong, C., and Liu, Z. (2020a). Grounded conversation generation as guided traverses in commonsense knowledge graphs. In Jurafsky, D., Chai, J., Schluter, N., and Tetreault, J., editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2031–2043, Online. Association for Computational Linguistics.

Zhang, N., Chen, X., Xie, X., Deng, S., Tan, C., Chen, M., Huang, F., Si, L., and Chen, H. (2021). Document-level relation extraction as semantic segmentation. In *International Joint Conference on Artificial Intelligence*.

Zhang, T., Kishore, V., Wu, F., Weinberger, K. Q., and Artzi, Y. (2020b). Bertscore: Evaluating text generation with bert.

Zheng, L., Chiang, W.-L., Sheng, Y., Zhuang, S., Wu, Z., Zhuang, Y., Lin, Z., Li, Z., Li, D., Xing, E., Zhang, H., Gonzalez, J. E., and Stoica, I. (2023). Judging llm-as-a-judge with mt-bench and chatbot arena. In Oh, A., Naumann, T., Globerson, A., Saenko, K., Hardt, M., and Levine, S., editors, *Advances in Neural Information Processing Systems*, volume 36, pages 46595–46623. Curran Associates, Inc.

Zhong, M., Zhang, A., Wang, X., Hou, R., Xiong, W., Zhu, C., Chen, Z., Tan, L., Bi, C., Lewis, M., Popuri, S., Narang, S., Kambadur, M., Mahajan, D., Edunov, S., Han, J., and van der Maaten, L. (2024). Law of the weakest link: Cross capabilities of large language models.

Zhou, W., Huang, K., Ma, T., and Huang, J. (2021). Document-level relation extraction with adaptive thresholding and localized context pooling. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Zhu, K., Wang, J., Zhou, J., Wang, Z., Chen, H., Wang, Y., Yang, L., Ye, W., Zhang, Y., Gong, N., and Xie, X. (2024). Promptrobust: Towards evaluating the robustness of large language models on adversarial prompts. In *Proceedings of the 1st ACM Workshop on Large AI Systems and Models with Privacy and Safety Analysis*, LAMPS '24, page 57–68, New York, NY, USA. Association for Computing Machinery.

Zhu, Z., Zhang, Z., Xhonneux, L.-P., and Tang, J. (2021). Neural bellman-ford networks: A general graph neural network framework for link prediction. In *Neural Information Processing Systems*.

Zhuang, L., Wayne, L., Ya, S., and Jun, Z. (2021). A robustly optimized BERT pre-training approach with post-training. In Li, S., Sun, M., Liu, Y., Wu, H., Liu, K., Che, W., He, S., and Rao, G., editors, *Proceedings of the 20th Chinese National Conference on Computational Linguistics*, pages 1218–1227, Huhhot, China. Chinese Information Processing Society of China.

# Andrei C. Coman

*Research Assistant at Idiap*
*Ph.D. Candidate at EPFL*

*Avenue de Fully 15*
*1920 Martigny, Switzerland*
*EU/EFTA Work Permit B*
*+41 788 825 605*
*andreic.coman@gmail.com*
*andreiccoman.github.io*


## About

I am a Research Assistant at the Idiap Research Institute and a Ph.D. Candidate at the École Polytechnique Fédérale de Lausanne (EPFL), specialising in Deep Learning (DL) at the intersection of Natural Language Processing (NLP) and Knowledge Graphs (KG). My research focuses on advancing Knowledge Representation and Language Understanding by developing models that integrate unstructured text and structured relational data to improve Retrieval Augmented Generation (RAG) systems. This work builds on my background in Information Retrieval, Big Data Analysis, and Data Science.

I bring applied research experience, rigorous experimental methodology, and hands on exposure to real world challenges through industry internships, along with a track record of translating theoretical advances into practical, high performing systems. I value collaboration and have worked closely with researchers, engineers, and practitioners across multiple projects and institutions, believing that openness, respect, and kindness are essential for successful teamwork and innovation.

## Education

OCT 2020 – SEP 2025 **Ph.D. in Electrical Engineering**,
*École Polytechnique Fédérale de Lausanne (EPFL)*, Lausanne, Switzerland
My PhD thesis *Text-Graph Encoders and Retrieval Augmented Generation* advanced Deep Learning methods that integrate unstructured text and structured graph data within Transformer architectures. It introduced efficient Retriever–Reranker–Reader pipelines for conversational question answering (QA), Reward Models (RMs) for grounded evaluation and alignment of Large Language Models (LLMs) in Retrieval Augmented Generation (RAG) systems, graph-assisted pooling to combine relational and textual context, and inductive text–graph Transformers capable of handling unseen entities and relations in Knowledge Graphs (KG).

OCT 2017 – MAR 2020 **M.Sc. in Computer Science**,
*University of Trento (UNITN)*, Trento, Italy, GPA: 110/110 cum laude
I focused on advanced topics in Data Science and Natural Language Processing. I worked on projects involving large-scale Information Retrieval and applied Machine Learning techniques for Big Data Analysis.

SEP 2014 – SEP 2017 **B.Sc. in Computer Science**,
*University of Trento (UNITN)*, Trento, Italy, GPA: 100/110
Through this program, I acquired a strong foundation in Computer Science, with a particular emphasis on Algorithms and their practical applications. I developed my problem-solving and programming skills, learning to create efficient computational solutions.

## Work Experience

OCT 2020 – SEP 2025 **Research Assistant**,
*Idiap Research Institute (NLU Lab)*, Martigny, Switzerland
As a key contributor to the research team, I advanced my research expertise through the *Neural Knowledge Base Population* project. I worked on bridging unstructured text and structured graph information with a novel Transformer model, and on developing methods within the Retrieval Augmented Generation (RAG) framework to improve evidence grounding and enhance information-seeking performance of Large Language Models (LLMs).

NOV 2024 – MAY 2025 **Applied Scientist Intern**,
*Amazon Science (AGI Lab)*, Barcelona, Spain
As an Applied Scientist Intern, I worked on developing Reward Models (RMs) tailored for Retrieval Augmented Generation (RAG), focusing on grounded evaluation of Large Language Models (LLMs) outputs with respect to retrieved evidence. This involved creating RAG-specific preference datasets and training evaluation models that prioritise faithfulness, completeness, conciseness, and relevance in information-seeking scenarios.

JUN 2022 – DEC 2022 **Applied Scientist Intern**,
*Amazon Science (Alexa AI Lab)*, Berlin, Germany
As an Applied Scientist Intern I worked on *Strong and Efficient Baselines for Open Domain Conversational Question Answering*, developing efficient Retriever–Reranker–Reader pipelines for conversational question answering (QA). I leveraged High Performance Computing (HPC) resources to train and deploy Large Language Models (LLMs) enhanced with Retrieval Augmented Generation (RAG) techniques, helping build more robust and efficient conversational agents.

MAY 2020 – SEP 2020 **Research Assistant**,
*University of Trento (UNITN) (SiS Lab)*, Trento, Italy
Conducted interdisciplinary research, analyzing linguistic and physiological data from human interactions, including text, heart rate, and skin conductance.

SEP 2019 – MAR 2020 **M.Sc. Thesis**,
*University of Trento (SiS Lab)*, Trento, Italy
I worked on *Language Modeling: Continuous Space Representation for Finite State Models* where I investigated methodologies for creating finite representations of neural Language Models (LMs) in the form of stochastic Finite-State Transducers (FST).

DEC 2018 – FEB 2019 **Special Research Student**,
*Nara Institute of Science and Technology (AHC Lab)*, Nara, Japan
I worked on *An Incremental Turn-Taking Model For Task-Oriented Dialog Systems* which involved creating a dialog system capable of estimating the appropriate moment to respond during a conversation.

AUG 2018 – NOV 2018 **Technologist**,
*Bruno Kessler Foundation (eHealth Lab)*, Trento, Italy
Research and development of a virtual health coach with intent detection and slot-filling capabilities to assist users with completing complex Cognitive-Behavioural-Therapy (CBT) forms.

JUN 2017 – SEP 2017 **B.Sc. Thesis**,
*Bruno Kessler Foundation (Future Media Lab)*, Trento, Italy
I worked on *Recommending Followers to Improve Reputation on Social Media* where I developed a Machine Learning based Recommendation System to suggest potential followers for a social media account to improve its reputation.

MAR 2017 – JUN 2017 **Software Engineer Intern**,
*Bruno Kessler Foundation (Future Media Lab)*, Trento, Italy
I worked on the Continuous Integration (CI) and Continuous Deployment (CD) of a dynamic Data Analysis web-based platform, using Selenium to implement automated testing procedures.

## Teaching Experience

SEP 2023 – DEC 2023 **Teaching Assistant in Deep Learning for Natural Language Procedding (EE-608)**,
*École Polytechnique Fédérale de Lausanne (EPFL)*, Lausanne, Switzerland
Led and assisted students with the *Deep Learning for Natural Language Processing (EE-608)* course lab exercises. This role involved creating and grading assignments, as well as offering office hours to ensure students' success.

MAR 2022 – JUN 2022 **Teaching Assistant in Deep Learning (EE-559)**,
*École Polytechnique Fédérale de Lausanne (EPFL)*, Lausanne, Switzerland
Assisted students and graded assignments in the *Deep Learning (EE-559)* course.

JAN 2021 – DEC 2023 **Teaching Assistant in Natural Language Processing (A03)**,
*UniDistance (Master in Artificial Intelligence)*, Martigny, Switzerland
Led lab exercises in the Natural Language Processing course, guiding students through practical applications and offering support with coursework and assignments.

## Languages

| | |
|---|---|
| English | Full professional proficiency |
| French | Intermediate working proficiency |
| Italian | Native or bilingual proficiency |
| Romanian | Native or bilingual proficiency |

## Publications

2025 **RAGferee: Building Contextual Reward Models for Retrieval-Augmented Generation.**, *EMNLP 2025 Main*
Andrei C. Coman, Sorodoc, I., Ribeiro, L., Byrne, B., Henderson, J., and de Gispert, A.

**Fast-and-Frugal Text-Graph Transformers are Effective Link Predictors**, *ACL 2025 Findings*
Andrei C. Coman, Christos Theodoropoulos, Marie-Francine Moens, James Henderson

**Fine-tuning pretrained models with NVIB for improved generalisation**, *ICLR 2025 SCSL Workshop*
Fehr, F. J., Baia, A. E., Chang, X., Coman, A. C., Hajal, K. E., Zein, D. E., Kumar, S., Gomez, J. P. Z., Cavallaro, A., Teney, D., and Henderson, J

**Reduction of Supervision for Biomedical Knowledge Discovery**, *arXiv*
Christos Theodoropoulos, Andrei C. Coman, Marie-Francine Moens, James Henderson

2024 **GADePo: Graph-Assisted Declarative Pooling Transformers for Document-Level Relation Extraction**, *ACL 2024 KnowledgeNLP Workshop*
Andrei C. Coman, Christos Theodoropoulos, Marie-Francine Moens, James Henderson

**Enhancing Biomedical Knowledge Discovery for Diseases: An End-To-End Open-Source Framework**, *IEEE Access*
Christos Theodoropoulos, Andrei C. Coman, Marie-Francine Moens, James Henderson

2023 **Strong and Efficient Baselines for Open Domain Conversational Question Answering**, *EMNLP 2023 Findings*
Andrei C. Coman, Gianni Barlacchi, Adrià de Gispert

**Transformers as Graph-to-Graph Models**, *EMNLP 2023 Big Picture Workshop*
James Henderson, Alireza Mohammadshahi, Andrei C. Coman, Lesly Miculicich

2021 **Imposing Relation Structure in Language-Model Embeddings Using Contrastive Learning**, *CoNLL 2021*
Christos Theodoropoulos, James Henderson, Andrei C. Coman, Marie-Francine Moens

2019 **An incremental turn-taking model for task-oriented dialog systems**, *INTERSPEECH 2019*
Andrei C. Coman, Koichiro Yoshino, Yukitoshi Murase, Satoshi Nakamura, Giuseppe Riccardi

2018 **Exploiting deep neural networks for tweet-based emoji prediction**, *AIxIA 2018 NL4AI Workshop*
Andrei C. Coman, Giacomo Zara, Yaroslav Nechaev, Gianni Barlacchi, Alessandro Moschitti

**Predicting emoji exploiting multimodal data: FBK participation in ITAmoji task**, *CLiC-it 2018 EVALITA Workshop*
Andrei C. Coman, Yaroslav Nechaev, Giacomo Zara