Document reRanking using GAT-Cross Encoder

Daniel Vollmers*
Computer Science Department
Paderborn University
Paderborn, North Rhine-Westphalia, Germany
daniel.vollmers@uni-paderborn.de

Manzoor Ali*
Computer Science Department
Paderborn University
Paderborn, North Rhine-Westphalia, Germany
manzoor.ali@uni-paderborn.de

Hamada M. Zahera
Computer Science Department
Paderborn University
Paderborn, North Rhine-Westphalia, Germany
hamada.zahera@uni-paderborn.de

Axel-Cyrille Ngonga Ngomo
Computer Science Department
Paderborn University
Paderborn, North Rhine-Westphalia, Germany
axel.ngonga@uni-paderborn.de

Abstract

Document re-ranking is a crucial post-processing step, focused on reordering an initial list of documents to better meet the information needs associated with a user query. In this paper, we explore the application of graph attention networks to enhance the re-ranking process in information retrieval systems. Traditional methods often treat documents indepentaly, ignoring the inter-document relationships that can enhance the relevance of search results. In our approach, we introduce a query-document subgraph where nodes represent entities from both a query and a candidate document, and the edges represent their semantic relationships. By employing a graph attention network, we effectively leverage their relationships to refine document re-ranking for a given query. The experimental results demonstrate significant improvements in ranking metrics, such as precision and recall, compared to traditional re-ranking methods. Our work demonstrates the potential of application of graph modeling to improve the performance of information retrieval systems.

1 Introduction

Document retrieval and re-ranking are important components of information retrieval systems, designed to identify the most relevant documents for a user's query (Hambarde and Proenca, 2023). Document retrieval involves the initial process of finding a set of documents that match the user's query. On the other hand, document reranking refines the initial set of retrieved documents by reordering them based on more contextual and semantic information. In real-world applications such as search engines (Yang et al., 2022) and question answering (Zhang et al., 2021), document reranking plays an important role to ensure presenting the most relevant document users, thereby improving their experience and satisfaction (Hambarde and Proenca, 2023).

Current methods for document reranking including techniques like TF-IDF and BM25 (Robertson et al., 2004), often fail to capture the contextual relationships and semantic meanings due to their bag-of-words representation. Recent approaches leverage transformer-based models such as BERT, GPT-3, which provide rich contextual embedding mechanisms (Abid, 2023), yet they require extensive fine-tuning on domain-specific datasets. Furthermore, effectively integrating contextual information into the reranking process remains a challenge.

Preprint. Under review.

*Equal Contribution

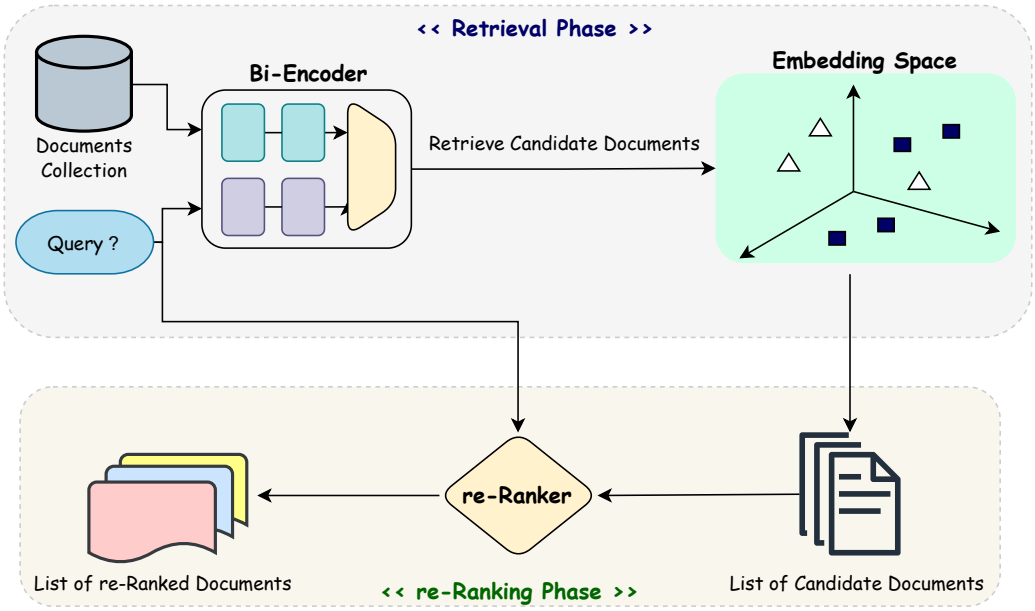

Figure 1: The pipeline of documents re-ranking.

To provide fast and accurate document retrieval, the process of document reranking is typically performed in two phases (*Retrieval Phase* and *Reranking Phase*) as shown in Figure 1. In the retrieval phase, a bi-encoder model is employed to retrieve candidate documents. This model encodes both the query and documents independently into vector embeddings, and the similarity between these embeddings is used to select the most relevant documents. For example, Sentence-BERT can serve as a bi-encoder model, where it transforms both queries and documents into a high-dimensional space and retrieves candidates based on cosine similarity. Following this, in the reranking phase, a reranker model, which functions as a cross-encoder, is applied. Unlike the bi-encoder, the cross-encoder jointly processes the query and each candidate document, evaluating their relevance by considering interactions between them.

In recent years, graph neural networks (GNN) have emerged as powerful models for capturing complex relationships in non-euclidean spaces (Khemani et al., 2024). Unlike traditional neural networks, GNNs can model dependencies between nodes in a graph, making them particularly suitable for tasks involving relational data. Inspired by these advancements, we propose a novel approach that employs a graph attention network (GAT) as cross-encoder (Schlatt et al., 2024) for document reranking. Our approach constructs a query-document subgraph, where nodes represent entities from both the input query and the candidate document, and edges as their semantic relationships. For example, given the query Q :"`What are the historical achievements of Michael Phelps in the Olympics?`" and a list of retrieved documents $d_i \in D$, our approach involves three components to compute the relevance of each candidate document for the input query: i) Entity Recognition: identifying entities from the query e.g. "Michael Phelps" and "Olympics" as well as entities mentions from the document d_i, ii) Entity linking: Aligning these entities with a target knowledge graph (e.g., DBpedia or Wikidata), ii) Subgraph Extraction: Creating a subgraph that captures the relationships among these entities. By Applying GAT in this subgraph, we leverage structural information (e.g., relationships between entities in the query and a candidate document) to improve the performance of document reranking. This method not only enhances the relevance of the search results but also demonstrates significant improvements in ranking metrics compared to traditional reranking methods. For example, an RDF triple (`Michael Phelps, award, Olympic Gold Medal`) indicates a link between the nodes "`Michael Phelps`" and "`Olympic Gold Medal`" in the graph. In our experiments, we use the MS-MARCO dataset, which is a widely used benchmark in information retrieval and document reranking. The evaluation results show that our GAT-based re-ranking model, when combined with a cross-encoder, achieves substantial improvements in various ranking metrics,

including Mean Reciprocal Rank (MRR) and Mean Average Percision (MAP). We summarize the main contributions of our study as follows:

- We present a novel approach for document reranking using a graph attention network combined with a cross-encoder architecture.
- Our experiments show that GAT for re-ranking improves the performance compared to traditional methods.
- Our approach is open-sourced, and its implementation is publicly available on GitHub webiste.[2]

2 Related Work

In recent years, deep learning models have gained significant traction in the field of information retrieval, offering a paradigm shift from traditional algorithms like BM25 Harman (1994), which rely on term frequencies in documents and queries. Unlike these conventional methods, deep learning-based models typically leverage embeddings generated by language models to assess the relevance between queries and documents Han et al. (2020). This shift has paved the way for more sophisticated retrieval techniques that can capture the nuanced relationships within data. In this work, we explore the integration of knowledge graph information into deep learning-based retrieval models, presenting the latest advancements in this area and focusing particularly on entity-augmented information retrieval.

2.1 Deep Learning-Based Methods

One of the pioneering approaches in deep learning-based information retrieval is the*TFR-BERT* Han et al. (2020) model. This method combines queries and passages from documents into a single sequence, separated by a designated token. The BERT model is then applied to generate embeddings from the *[CLS]* token, which is introduced at the beginning of these sequences. The final relevance scores are produced through a tensor-based scoring module. This process, known as reranking, effectively reorders documents based on their relevance to the query, providing a more refined set of results.

Following a similar reranking strategy, the *BERT-QE* Zheng et al. (2020) model introduces a multi-phase ranking process. Initially, a classical reranking is employed to extract the top N documents from the document set. In the subsequent phase, relevant text chunks are identified within these documents. The final ranking is determined based on these extracted chunks, further refining the accuracy of the retrieval process.

Different from the aforementioned BERT-based models, *RankT5* Zhuang et al. (2023) employs a fine-tuning approach with the T5 model, a more advanced encoder-decoder architecture. Unlike BERT, which functions solely as an encoder, T5 can predict output sequences, offering greater flexibility. RankT5 presents two variations: in the first, the entire encoder-decoder model is trained, where a special token's logit information is utilized to estimate document relevance. In the second variation, only the encoder is used, and a scoring function is implemented based on the encoder's embedding output.

Moving away from traditional reranking models Gao and Callan (2022), another innovative approach involves a modular reranker that employs separate transformer models to encode the query and documents independently. An interaction model is used then to perform cross-attention between the query and document embeddings. To accommodate long documents, these are divided into chunks, with the interaction model modified to apply cross-attention over the concatenated chunk embeddings. This method allows the processing of lengthy documents more effectively, ensuring that the model can capture relevant information distributed across various sections of the text.

Overall, these deep learning-based methods represent significant advancements in information retrieval, offering more accurate and nuanced ways to assess and rank the relevance of documents to user queries. The integration of knowledge graphs and other enhancements continues to push the boundaries of what these models can achieve, promising even more sophisticated retrieval systems in the future.

[2]the link will be available soon

2.2 Entity-based Methods

In document and entity ranking, several approaches have been developed to enhance the effectiveness of information retrieval systems. One early method is *JointSem*, which jointly ranks documents and entities together. This approach has three main steps: First, the system identifies surface forms of entities in the user's query. Next, these surface forms are linked to specific entities, which are pre-extracted from a document corpus during the training phase and stored in a dictionary. Finally, the documents are ranked based on the linked entities Xiong et al. (2017).

Building on the foundation methods such as *JointSem*, the *DREQ* model presents a hybrid ranking approach that merges classical text-based ranking with entity representation. A separate ranking model extracts entities from documents to create an entity-centric representation. The final ranking merges this entity-centric representation with a traditional text-based one, allowing for a more comprehensive scoring of documents in response to a query Chatterjee et al. (2024).

While most ranking models typically rely on text-based cross-encoders, the *KGPR* model introduces a novel approach by incorporating entities directly into the cross-encoder framework. This model leverages the LUKE entity-aware language model, which integrates entities from both the query and the document into the cross-encoder. Additionally, *KGPR* enhances this model by introducing a subgraph into the cross-encoder, further refining the ranking process and enabling more accurate retrieval results Fang et al. (2023); Yamada et al. (2020).

3 Approach

3.1 Problem Formulation

In the context of document reranking, the goal is to reorder a set of documents $D = \{d_1, d_2, \cdots d_n\}$, initially retrieved for a given query Q. This initial retrieval is typically performed using a traditional information retrieval model such as BM25, which provides a ranked list of documents based on lexical matching. The document reranking task aims to learn a new ranking function $\mathcal{R}(Q, D)$ that assigns a relevance score s_i to each document $d_i \in D$ w.r.t the query Q. To achieve this, we propose a GAT-based cross encoder that constructs a graph $G = (V, E)$ for each query-document pair, where V represents entites extracted from both the query Q and the document d_i, and E represents the semantic relationships between these entities. The output of our GAT cross-encoder is a relevance score (e.g. ranging from 0.0 - 1.0) that indicates how document d_i is relevant to the query Q.

3.2 Architecture

Figure 2 presents the architecture of our approach, including three components: (a) Entity Recongition, (b) Subgraph Extraction, and (c) Graph Attention Network as a cross-encoder. Our approach takes both a query and a candidate document as inputs and computes a relevance score between as an output. This relevance score is then used to re-rank the documents w.r.t the input query. In the following subsections, we briefly describe the details of each component in our approach.

3.2.1 Entity Recognition and Linking with Knowledge Graph

Named Entity Recognition and Entity Linking are essential components in our approach for extracting entities in both queries and documents and linking them with a knowledge graph. NER is responsible for detecting and categorizing entities such as people, organizations, and locations, which are crucial for understanding the content and intent of textual data. In our approach, we use the FLAIR[3], state-of-the-art NLP tool, to extract named entities from long text such as documents. For queries, we use the LlaMA3 model to disambiguate and identify named entities, since it excels at processing short text inputs with high accuracy and efficiency. Subsequently, Entity Linking link the identified entities to their corresponding URIs in a knowledge graph. In our approach, we use GENRE[4], an autoregressive model for linking detected entities from both queries and documents with Wikidata knowledge graph. This process includes: I) Candidate Generation, for each detected entity, a list of potential matches from the knowledge graph is generated; II) Candidate Ranking, the potential

[3] `https://github.com/flairNLP/flair`

[4] `https://github.com/facebookresearch/GENRE`

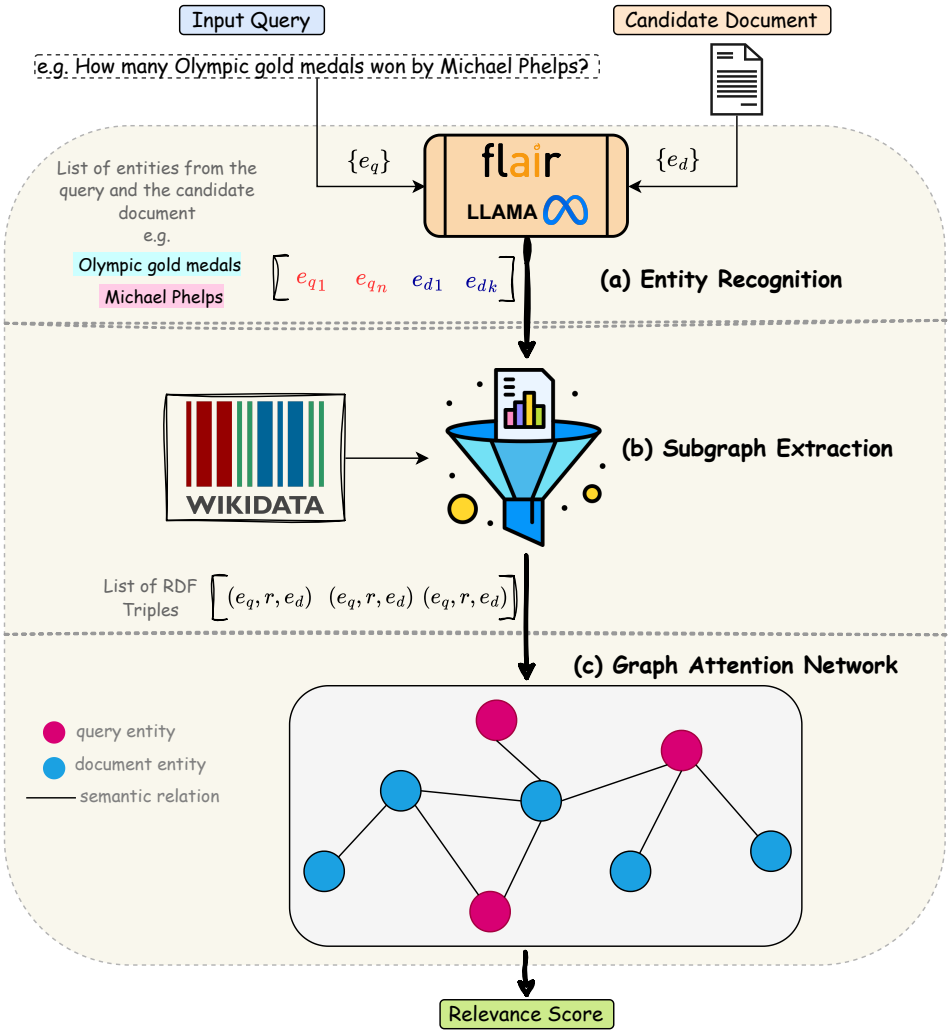

Figure 2: The pipeline of our document re-Ranker Model

matches are ranked based on their relevance (e.g., similarity in name, description and attributes) and surrounding context, III) Disambiguation, the most relevant match is then selected and linked with the named entity.

In the next component, we show how to extract a subgraph of the linked entities from a knowledge graph, where nodes represent an entity (either from a query or a document) and edges represent their relationship. For example in the RDF triple "<Berlin, capital_of, Germany>" demonstrates an edge between the two nodes "Berlin" and "Germany". By extracting these entities from both queries and documents, we construct subgraphs that capture the relationships and connections between relevant entities, thereby enhancing the retrieval of pertinent information.

3.2.2 Subgraph Extraction

This component aims to extract a subgraph of the *identified entities* by the NER and Linking component from a knowledge graph. For this purpose, we use the subgraph retrievel toolkit Shen (2023) (SRTK)[5] that takes a list of entities ($e_i \in \mathbf{E}$) as an input and return a list of RDF triples in which both the subject and object entities are matched from the input list of entities ($\mathbf{E}$). First, the SRTK operates by using a zero-shot end-to-end linking method to map the input entities to

[5] https://github.com/happen2me/subgraph-retrieval-toolkit

their corresponding entries in target knowledge graph such as Wikidata. Then it uses a beam search algorithm to explore paths originating from the source entity, with a default depth of two hops. Afterward, the toolkit employs a fine-tuned language model to score the paths and return the most relevant RDF triples that connect these entities (i.e, paths). For example, for a given list of entities $\mathbf{E} = \{BarackObama, MichelleObama, UnitedStates\}$ and Wikidata as a target knowledge graph, the SRTK extract RDF triples, which includes the subject and object entities from $\mathbf{E}$ such as "<*Barack Obama, married_to, Michelle Obama*>". The output of this component is a subgraph (i.e, a list of RDF triples) of entities identities from both an input query and a candidate document. In the next component, we show how to construct a graph attention network (GAT) based on the extracted subgraph to compute the relevance score between a query and a candidate document.

3.2.3 GAT Cross-Encoder

In this component, we employ a graph attention network as a cross-encoder to re-rank documents. Given the subgraph of entities $\{v_i\}_{i=1}^{n}$ identified from both the query (q) and document (d), we construct an initial feature matrix $(\mathbf{X})$ and an adjacency matrix $(\mathbf{A})$ that captures the structure of the knowledge graph. The GAT then updates the node embeddings $(\mathbf{H})$ through Attention layers as follows:

$$\mathbf{h}_i^{(l+1)} = \sigma\left(\sum_{j \in \mathcal{N}(i)} \alpha_{ij}^{(l)} \mathbf{W}^{(l)} \mathbf{h}_j^{(l)}\right) \tag{1}$$

where $\alpha_{ij}^{(l)}$ are the attention coefficients which are used to weigh the importance of nodes in the graph. For more details about how the coefficients are computed, we refer the reader to the original manuscript (Veličković et al., 2017).

Our model consists of three GAT layers, with the first two layers applying *ReLU* activation functions and the final layer using a sigmoid activation. The input features $(\mathbf{X})$ and adjacency matrix $(\mathbf{A})$ are processed through these layers, resulting in refined node embeddings $(\mathbf{H})$. Initially, the embeddings are randomly initialized. The updated embeddings $(\mathbf{H}^{(L)})$ are then averaged across nodes using `mean(dim=0, keepdim=True)`, producing a single embedding vector that represents the entire graph, encapsulating the relationships between the query and document entities. To obtain the relevance score between the query and document, the output node embeddings corresponding to the query entities $(\mathbf{h}_i^q)$ and document entities $(\mathbf{h}_j^d)$ are aggregated by averaging. The relevance score $r(q, d)$ is then computed by applying a similarity function such as the dot product between the aggregated query and document representations:

$$r(q, d) = \sum_{i \in q} \sum_{j \in d} \mathbf{h}_i^\top \mathbf{h}_j \tag{2}$$

4 Experiments

In this section, we describe the structure of our experimental set-up, including datasets, baselines, and the hyperparameters used. Through our experiments, we aim to address the following research questions:

RQ$_1$. How effective is our approach in document re-ranking by using a graph-based representation between entities in queries and documents?

4.1 Experiments Setup

4.1.1 Dataset

In our experiments, we use the MS MARCO (Bajaj et al., 2016) (Microsoft Machine Reading Comprehension) document ranking dataset, that is a large-scale dataset designed for training and evaluating information retrieval systems. It was created to provide a realistic and challenging environment for testing machine learning models, particularly in the domain of document, passage retrieval and question answering. The dataset was built by sampling Bing search queries and

Table 1: MS MARCO Dataset Statistics

Dataset Component	MS MARCO V1	MS MARCO V2
Training Queries	500,000+	8.8M+
Validation Queries	7,000+	40,000+
Test Queries	6,980	40,000+
Documents	N/A	3.2M

corresponding web pages, followed by human annotation to generate relevant query-passage pairs. The dataset is divided into training, validation, and test sets, with the training set consisting of over 500,000 queries, the validation set containing approximately 7,000 queries, and the test set comprising about 6,980 queries. MS MARCO has undergone several iterations, with versions like the MS MARCO V1 and V2, where V2 includes a significantly larger corpus and more diverse set of queries. The dataset is crucial in advancing research in information retrieval due to its scale and the inclusion of real-world, noisy data. Table1 summarizes the key statistics of the dataset:

We randomly sampled 100 queries from the MS MARCO dataset and selected the top 100 documents associated with each query for training. For evaluation, we chose 20 queries along with their corresponding top 100 document IDs.

4.1.2 Evaluation Metrics

- *Mean Reciprocal Rank (MRR) @ K*, evaluates how quickly a ranking system can show the first relevant item in the top-K results. It is calculated as the average of the reciprocal ranks of the first relevant item across multiple queries.

$$\text{MRR@K} = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \frac{1}{\text{rank}_i} \cdot \mathbb{I}(\text{rank}_i \leq K) \tag{3}$$

  where $rank$ is the position of the first relevant item for the i-th query, and $|Q|$ is the total number of queries.

- *Mean Average Precision (MAP) @ K*, measures both the relevance of suggested items and how well the system ranks more relevant items higher in the top-K results. It is the mean of the average precision scores for multiple queries.

$$\text{MAP@K} = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \text{AveP@K}(q_i) \tag{4}$$

  where $AvePK(qi)$ is the average precision at KK for the i-th query, and $|Q|$ is the total number of queries.

4.1.3 Baselines

In our study, we utilized the following models as baselines for performance comparison.

- **BERT-base** Devlin et al. (2019):The first model, BERT-base, serves as a foundational transformer model pre-trained on a large corpus of text, enabling it to capture contextual information effectively.
- **Roberta-base** Liu et al. (2019):The second model, Roberta-base, is a robustly optimized BERT approach, which has been pre-trained with larger batches and over more extended periods, enhancing its language understanding capabilities.
- **jinaai/jina-reranker-v1-tiny-e** Günther et al. (2024):Lastly, we employed the jinaai/jina-reranker-v1-tiny-e[6] model for document re-ranking, a specialized lightweight model designed to refine the ranking of documents based on their relevance, ensuring more precise information retrieval.

[6] `https://jina.ai/reranker/`

Table 2: Performance evaluation of different re-Ranker (base) models against our approach GCN-reRanker

Model	MRR@3	MRR@5	MRR@10	MAP@3	MAP@5	MAP@10
BERT-base	0.04	0.10	0.24	0.04	0.10	**0.24**
Roberta-base	**0.05**	0.10	0.13	**0.05**	0.10	0.15
jina-reranker	0.0	0.03	0.12	0.0	0.03	0.13
Lajavaness/camembert	0.04	**0.16**	0.24	0.04	**0.15**	0.22
gcn-reRanker	**0.05**	0.06	**0.28**	**0.05**	0.06	**0.24**

4.2 Discussion

To answer RQ_1, we evaluated the performance of our pproach (GAT-reRanker) against different baselines, including BERT-base, Roberta-base, jina-reranker, and Lajavaness/camembert. The evaluation metrics used were MRR@3, MRR@5, MRR@10, MAP@3, MAP@5, and MAP@10. As shown in Table 2, the results demonstrate that our approach outperforms the baseline models in several key metrics. For MRR@10, GAT-reRanker achieves the highest score of 0.28, surpassing the performance of the baselines: BERT-base, Roberta-base, jina-reranker, and Lajavaness/camembert. Similarly, for $MAP@10$, GAT-reRanker achieves the highest score of 0.24 same as the BERT-base. This indicates that our approach is effective in accurately ranking relevant documents within the top 10 positions. However, it's important to note that the baselines perform better in some metrics. For example, Roberta-base achieves the highest MRR@3 and MAP@3 scores of 0.05, demonstrating its strength in ranking relevant documents within the top 3 positions. Additionally, Lajavaness/camembert has the highest MRR@5 and MAP@5 scores of 0.16 and 0.15, respectively, suggesting its effectiveness in ranking relevant documents within the top 5 positions. Overall, our approach demonsrates promising results, particularly in terms of overall ranking accuracy (MRR@10) and precision at higher ranks (MAP@10).

5 Conclusion

In this paper, we present our approach (GAT-reRanker) as a new cross-encoder for documents re-ranking. For a query-document pair, our approach identifies entites in both the query and the document and then map them with their corresponding entries in a knowledge graphs. Then, our approach extracts a subgraph of those entities based on the links between entities in the knowledge graph. The evaluation results show that our approach outperforms several baselines in key evaluation metrics, such as MRR@10 and MAP@10. These findings suggest that the integration of entity extraction and knowledge graph subgraph construction within a cross-encoder framework can significantly enhance the performance of documents re-ranking.

Aknowledgement

This work has been supported by the German Federal Ministry of Education and Research (BMBF) within the project KIAM under the grant no 02L19C115.

References

MA Abid. 2023. Comparative analysis of TF-IDF and loglikelihood method for keywords extraction of twitter data. *Mehran University Research Journal of Engineering and Technology* 42, 1 (2023), 88–94.

Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, et al. 2016. Ms marco: A human generated machine reading comprehension dataset. *arXiv preprint arXiv:1611.09268* (2016).

Shubham Chatterjee, Iain Mackie, and Jeff Dalton. 2024. DREQ: Document Re-Ranking Using Entity-based Query Understanding. arXiv:2401.05939 [cs.IR] `https://arxiv.org/abs/2401.05939`

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv:1810.04805 [cs.CL] https://arxiv.org/abs/1810.04805

Jinyuan Fang, Zaiqiao Meng, and Craig Macdonald. 2023. KGPR: Knowledge Graph Enhanced Passage Ranking. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management* (Birmingham, United Kingdom) *(CIKM '23)*. Association for Computing Machinery, New York, NY, USA, 3880–3885. https://doi.org/10.1145/3583780.3615252

Luyu Gao and Jamie Callan. 2022. Long Document Re-ranking with Modular Re-ranker. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '22)*. ACM. https://doi.org/10.1145/3477495.3531860

Michael Günther, Jackmin Ong, Isabelle Mohr, Alaeddine Abdessalem, Tanguy Abel, Mohammad Kalim Akram, Susana Guzman, Georgios Mastrapas, Saba Sturua, Bo Wang, Maximilian Werk, Nan Wang, and Han Xiao. 2024. Jina Embeddings 2: 8192-Token General-Purpose Text Embeddings for Long Documents. arXiv:2310.19923 [cs.CL] https://arxiv.org/abs/2310.19923

Kailash A Hambarde and Hugo Proenca. 2023. Information retrieval: recent advances and beyond. *IEEE Access* (2023).

Shuguang Han, Xuanhui Wang, Mike Bendersky, and Marc Najork. 2020. Learning-to-Rank with BERT in TF-Ranking. arXiv:2004.08476 [cs.IR] https://arxiv.org/abs/2004.08476

Donna K. Harman (Ed.). 1994. *Proceedings of The Third Text REtrieval Conference, TREC 1994, Gaithersburg, Maryland, USA, November 2-4, 1994*. NIST Special Publication, Vol. 500-225. National Institute of Standards and Technology (NIST). http://trec.nist.gov/pubs/trec3/t3_proceedings.html

Bharti Khemani, Shruti Patil, Ketan Kotecha, and Sudeep Tanwar. 2024. A review of graph neural networks: concepts, architectures, techniques, challenges, datasets, applications, and future directions. *Journal of Big Data* 11, 1 (2024), 18.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. arXiv:1907.11692 [cs.CL] https://arxiv.org/abs/1907.11692

Stephen Robertson, Hugo Zaragoza, and Michael Taylor. 2004. Simple BM25 extension to multiple weighted fields. In *Proceedings of the thirteenth ACM international conference on Information and knowledge management*. 42–49.

Ferdinand Schlatt, Maik Fröbe, and Matthias Hagen. 2024. Investigating the Effects of Sparse Attention on Cross-Encoders. In *European Conference on Information Retrieval*. Springer, 173–190.

Yuanchun Shen. 2023. SRTK: A Toolkit for Semantic-relevant Subgraph Retrieval. *arXiv preprint arXiv:2305.04101* (2023).

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903* (2017).

Chenyan Xiong, Zhengzhong Liu, Jamie Callan, and Eduard Hovy. 2017. JointSem: Combining Query Entity Linking and Entity based Document Ranking. 2391–2394. https://doi.org/10.1145/3132847.3133048

Ikuya Yamada, Akari Asai, Hiroyuki Shindo, Hideaki Takeda, and Yuji Matsumoto. 2020. LUKE: Deep Contextualized Entity Representations with Entity-aware Self-attention. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu (Eds.). Association for Computational Linguistics, Online, 6442–6454. https://doi.org/10.18653/v1/2020.emnlp-main.523

Yingrui Yang, Yifan Qiao, Jinjin Shao, Xifeng Yan, and Tao Yang. 2022. Lightweight composite re-ranking for efficient keyword search with BERT. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*. 1234–1244.

Yuyu Zhang, Ping Nie, Arun Ramamurthy, and Le Song. 2021. Answering any-hop open-domain questions with iterative document reranking. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 481–490.

Zhi Zheng, Kai Hui, Ben He, Xianpei Han, Le Sun, and Andrew Yates. 2020. BERT-QE: Contextualized Query Expansion for Document Re-ranking. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, Trevor Cohn, Yulan He, and Yang Liu (Eds.). Association for Computational Linguistics, Online, 4718–4728. https://doi.org/10.18653/v1/2020.findings-emnlp.424

Honglei Zhuang, Zhen Qin, Rolf Jagerman, Kai Hui, Ji Ma, Jing Lu, Jianmo Ni, Xuanhui Wang, and Michael Bendersky. 2023. RankT5: Fine-Tuning T5 for Text Ranking with Ranking Losses. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Taipei, Taiwan) *(SIGIR '23)*. Association for Computing Machinery, New York, NY, USA, 2308–2313. https://doi.org/10.1145/3539618.3592047