Hierarchical Transformer-based Query by Multiple Documents

Zhiqi Huang*
zhiqihuang@cs.umass.edu
University of Massachusetts Amherst

Shahrzad Naseri*
shnaseri@cs.umass.edu
University of Massachusetts Amherst

Hamed Bonab†
hamedrab@amazon.com
Amazon Inc.

Sheikh Muhammad Sarwar†
smsarwar@amazon.com
Amazon Inc.

James Allan
allan@cs.umass.edu
University of Massachusetts Amherst

ABSTRACT

It is often difficult for users to form keywords to express their information needs, especially when they are not familiar with the domain of the articles of interest. Moreover, in some search scenarios, there is no explicit query for the search engine to work with. Query-By-Multiple-Documents (QBMD), in which the information needs are implicitly represented by a set of relevant documents addresses these retrieval scenarios. Unlike the keyword-based retrieval task, the query documents are treated as exemplars of a hidden query topic, but it is often the case that they can be relevant to multiple topics.

In this paper, we present a **H**ierarchical **Int**eraction-based (**HINT**) bi-encoder retrieval architecture that encodes a set of query documents and retrieval documents separately for the QBMD task. We design a hierarchical attention mechanism that allows the model to 1) encode long sequences efficiently and 2) learn the interactions at low-level and high-level semantics (e.g., tokens and paragraphs) across multiple documents. With contextualized representations, the final scoring is calculated based on a stratified late interaction, which ensures each query document contributes equally to the matching against the candidate document. We build a large-scale, weakly supervised QBMD retrieval dataset based on Wikipedia for model training. We evaluate the proposed model on both Query-By-Single-Document (QBSD) and QBMD tasks. For QBSD, we use a benchmark dataset for legal case retrieval. For QBMD, we transform standard keyword-based retrieval datasets into the QBMD setting. Our experimental results show that HINT significantly outperforms all competitive baselines.

CCS CONCEPTS

• **Information systems → Retrieval models and ranking**; *Query representation*; *Specialized information retrieval.*

KEYWORDS

Query by multiple documents; Hierarchical transformer; Neural re-ranking

ACM Reference Format:
Zhiqi Huang, Shahrzad Naseri, Hamed Bonab, Sheikh Muhammad Sarwar, and James Allan. 2023. Hierarchical Transformer-based Query by Multiple Documents. In *Proceedings of the 2023 ACM SIGIR International Conference on the Theory of Information Retrieval (ICTIR '23), July 23, 2023, Taipei, Taiwan.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3578337.3605130

1 INTRODUCTION

In many search scenarios, it can be more effective for users to express their information needs by providing examples instead of using keyword terms. This is because formulating the right keyword query may require domain-specific knowledge and depend on the user's expertise. This is particularly true in professional and specialized searches, such as legal case retrieval [2, 3, 6, 23, 44], scientific literature retrieval [11, 33], patent retrieval [16, 40] and cross-referencing a news article on a specific topic across sources [51]. Previous research has mainly focused on using a single example document to query an information retrieval system, Query-By-Single-Document (QBSD), with less emphasis on scenarios where multiple example documents are provided, Query-By-Multiple-Document (QBMD). Recently, in an effort to accelerate advances in the information discovery cycle, the National Institute of Standards and Technology (NIST) introduced benchmark datasets where multiple example documents are provided to the system instead of a query in a cross-lingual information retrieval scenario [39].

Multiple example documents constituting a single information need gives rise to a search scenario that, we hypothesize, is more complicated because: i) the query becomes longer than the state-of-the-art transformer-based ranking models can handle due to their limited input size; ii) it is unlikely to be sufficient to compute the score of a candidate document with respect to a query document by computing their similarities as is possible with a single query document. This is because a single example document can generally cover a number of topics, whereas multiple example documents give us the opportunity to identify a user's intent more precisely as we can infer the commonalities between example documents.

Previous research in QBSD has addressed the issue of limited input size in transformer models by truncating documents that exceed the maximum input length [2] or treating each passage in a query document as a separate query and then combining the results to create a final ranked list [3]. These approaches do not consider the semantic interaction between the paragraphs and thus

*Both authors contributed equally to this work.
†Work done prior to joining Amazon.

ICTIR '23, July 23, 2023, Taipei, Taiwan

ACM ISBN 979-8-4007-0073-6/23/07…$15.00
https://doi.org/10.1145/3578337.3605130

do not find the latent query that can be inferred from commonalities between the example documents.

Here, we investigate the Query-by-Multiple-Documents (QBMD) task, where the users' information need is described by multiple relevant example documents. We design a **H**ierarchical **Int**eraction-based (**HINT**) bi-encoder neural re-ranker that efficiently encodes long sequences as well as learns the semantic interactions at both low and high-levels across multiple documents. When the user's information need is not expressed explicitly and is conveyed by a number of example documents, understanding the interaction *between* the documents becomes more critical in order to identify the common attributes. The HINT model with its hierarchical attention encoder architecture is capable of capturing the relation and unveiling the latent query topic.

To evaluate our QBMD approach, we construct three datasets from existing ad-hoc retrieval and multi-document summarization datasets with three levels of relevancy annotation strength. In particular, we construct a large-scale weakly-supervised QBMD dataset based on Wikipedia, as well as two evaluation datasets with human judgments. We also include a legal case benchmark QBSD dataset for comparison. The experimental results show that our proposed architecture statistically significantly outperforms the initial ranking as well as cross-encoder based neural re-ranking baselines. We conduct an ablation study and find that the hierarchical attention encoding of the query-documents and candidate documents is the most critical component of our model.

Since we adopt a neural re-ranking strategy, we explore different approaches for formulating an initial query for the first-stage term-matching retrieval and obtaining the initial set of candidate documents. We experimentally demonstrate that a keyphrase extraction method based on TF-IDF is superior to a state-of-the-art question generation model and an unsupervised extractive multi-document summarization approach. Lastly, our experiments across multiple datasets show increasing the number of example documents improves the performance of the system in identifying the user's information need up until a threshold.

- We construct three QBMD datasets. Our datasets provide one large-scale weakly supervised dataset mainly for training and two high-quality smaller evaluation datasets.
- We present HINT, a transformer-based re-ranking model for the task of QBMD. Our model uses the hierarchical attention to capture the interaction between multiple query documents. To the best of our knowledge, we are the first to provide a neural approach for the task of QBMD.
- We conduct extensive experiments to demonstrate the superior performance of HINT against a number of baselines, including a 14.9% average improvement in terms of mean average precision (MAP) over a model based on cross-encoder architectures. For the first-stage term-matching retrieval, we compare several strong approaches to formulate an initial query. Our experimental evaluations show the effect of number of query-documents in identifying and locating the precise users' information need and the system's performance.

2 RELATED WORK

Our study is related to a number of topics from the existing literature. We briefly describe most relevant works on these topics.

Query By Example (QBE). Query by example is a setting in which the user of the retrieval system inputs one (or several) ideal instances and aims to find more similar instances from a given collection. In general, any form of example instances can be defined, e.g., textual documents, user profiles, and images [27]. For example, Ha-Thuc et al. [17] studied QBE for talent search at LinkedIn in which the user provides one or several ideal candidates as the input to search for a given position.

Query by Document (QBD) can be categorized as a special case for QBE where the input query is one, QBSD, or multiple, QBMD, textual documents and the aim is to find related documents from a large collection. Yang et al. [51] study QBSD in the context of automating the cross referencing of online information content such as finding related blog posts to a given news article. Weng et al. [49] introduce a two-level retrieval method for QBSD in which for the first stage retrieval they encode the documents in the collection into dense vectors using dimension reduction and conduct quick rankings using locality sensitive hashing. Along with obtaining dense vectors, they extracted distinguishing terms for every document that are used in a re-ranking step to address inefficiency concerns. Lee and Sun [24] propose a QBSD ranking model that focuses on clinical terms to improve the screening efficiency for medical systematic review. Williams et al. [50] describe a deployed QBSD search system on academic documents by combining multiple similarity functions and show its applicability as well as scalability on larger collections. The legal case retrieval task in the Competition on Legal Information Extraction/Entailment (COLIEE) [41] is an instance of QBSD in the legal domain where the input queries are long law cases. TLIR [32] as the top team in the COLIEE 2021 competition, proposes a cross-encoder model, concatenating query document and the candidate document. It has multiple stages of building the interaction between paragraphs in the query document and candidate document. Abolghasemi et al. [2] improve the performance of a BERT-based cross-encoder re-ranker by adding a document-level representation learning objective in the fine-tuning step and evaluate their model on the legal case and scientific QBSD benchmark datasets. Further, they investigate the effectiveness of the deep contextualized term-based retrieval models such as TILDE [56] and TILDEv2 [55] in the QBSD problem [1]. Althammer et al. [3] propose a paragraph-aggregation retrieval that adapts the Dense Passage Retrieval (DPR) [20] models to the QBSD problem by addressing the limited input size of DPR models.

In terms of query by multiple documents, there are a few modern studies addressing the problem. Wang et al. [48] concatenate the multiple documents into a single example document. Lissandrini et al. [26] study a special QBMD case where query examples are in the form of a graph and the collection to perform the search is a knowledge graph. El-Arini and Guestrin [13] propose a solution based on modeling query documents using a concept graph for the scientific publication search domain. Zhang and Lee [52] formulate the QBMD problem as a one-class text classification and utilize support vector machines for their solution. Zhu and Wu [54] argue that assuming the entire collection as an unlabeled example

results in poor retrieval performance and extend their approach by reducing the unlabeled set from the entire collection to a small subset of documents. Different from these approaches, we encode each document into a bag of vectors based on paragraphs and then use a hierarchical architecture to model the connections between them. Because the interactions exist in paragraphs across and *within* query documents, our proposed method naturally generalizes to the QBSD task where the query is a single document.

Multi-Document Summarization. Compared to Single Document Summarization (SDS), Multi-Document Summarization (MDS) must address the cross-document relations and redundancy. Researchers pursue two common approaches for the MDS task [31]: 1) concatenating all the input documents and creating a flat sequence representation which transforms MDS to SDS [28, 31], 2) hierarchical concatenation of input documents [4, 5, 35, 47]. Following the idea of Liu and Lapata [29], we incorporate the hierarchical attention mechanism in our proposed model. In Section 6, we explore using an unsupervised MDS method [53] to address QBMD and find it does not work well.

Relevance Feedback. The QBMD is also related to relevance feedback [42] in that the query examples can be regarded as the feedback documents from a user. However, the performance of a relevance feedback system is highly dependent on the quality of the original query submitted by the user [54]. In a relevance feedback system, a list of search results is first returned based on the initial user query. Then, the top-ranked documents on the list are selected as the feedback documents: either manually by the user [12] or automatically by the retrieval model (pseudo-relevance feedback) [18, 25, 34]. Smucker and Allan [45] study the "find-similar" feature, provided by some commercial search engines, as a form of manual feedback and explore user behavior and its possible effect on retrieval performance. QBMD is motivated by settings where a query is hard to formulate or not needed – for example legal, patent, literature or other example-based search settings.

3 PROBLEM DEFINITION

The search queries in most standard ad-hoc information retrieval datasets are short sentences or keywords which reflect the users' information needs. However, in a QBMD system, a user expresses their information need by providing multiple examples (query documents). A formal problem statement is as follows:

Given a set of example documents related to a query topic, $Q = \{QD_1, ..., QD_i, ..., QD_m\}$ where QD_i refers to i-th example document, i.e., query-document, retrieve a ranked list of documents $R = [d_1, ..., d_i, ..., d_n]$ that are relevant to the query, where d_i is the i-th document in the ranked list and is retrieved from a collection of documents C.

4 DATASETS

Since the available QBMD benchmark datasets are limited (to the best of our knowledge only BETTER [39] dataset is available and it is in cross-language retrieval setting with corpus in a language other than English), to train and evaluate our proposed approach and baselines we build a large-scale weakly-supervised training dataset, Wiki-QBMD, as well as two evaluation datasets Robust04-QBMD and Multi-News-QBMD.

Table 1: Statistics of the QBMD datasets. Avg #d^+/q denotes the average number of relevant documents per query.

	Wiki-QBMD	Robust04-QBMD	Multi-News-QBMD
Document count	2.4M	528,155	135,980
Query count	183,837	233	1,036
Query documents	3	{1, 2, 3, 4, 5}	{1, 2, 3, 4, 5}
Avg. #d^+/q	35.10	70.08	6.69

We construct our datasets on the principle that the query documents and the retrieval targets in the data collection are both relevant to the users' information needs. We build the input examples for the QBMD task by sampling the documents relevant to the same query in a keyword-based information retrieval dataset. Given a keyword query, we randomly sample N (defined as the query length) relevant documents as the query and leave the rest of the relevant documents as the retrieval targets. The sampled query-documents are then removed from the collection. The relevance judgments in our constructed datasets are in 3 strength levels in terms of annotation: 1) The relevancy in Wiki-QBMD is synthetic. 2) In Multi-News-QBMD only the relevant documents are judged and non-relevancy is implicit. 3) Robust04-QBMD has explicit human annotation for both relevant documents and non-relevant documents. Table 1 shows the statistics of our datasets[1].

Weakly Supervised QBMD Dataset. Most standard ad-hoc retrieval datasets (e.g., Robust04, ClueWeb09) do not have enough annotated queries to develop an effective deep neural retrieval model. Large passage ranking datasets such as MS MARCO [36], do not have multiple relevant documents for each query to sample from. Therefore, we build a QBMD retrieval dataset to support deep learning methods using relevant document sampling technique described above. Following the idea of WikIR [15], we build a large, weakly-supervised QBMD dataset from Wikipedia, named Wiki-QBMD. We assume that if an article a contains an internal link to another article a_t in its first sentence and the anchor text exactly matches the title of article a_t, then the content of article a is a query document for which a_t is relevant. The intuition behind this assumption is that the first sentence of most Wikipedia articles is a good descriptive sentence of the article's content [43]. If a link is present, it points to a topic that is semantically relevant to the considered article [19]. Finally, for articles with more than five content examples, we randomly sample 3 examples as query documents and use the other ones as relevant documents. Because the query documents and relevant documents are exchangeable, we generate 3 query-document pairs for QBMD from the same title. For model evaluation purposes, we randomly separate 300 articles to build a synthetic test dataset. Since all the information we use to build the examples for the title is contained in the first sentence, and we do not want the models to take word order into account to use this bias to their advantage, we removed the title and the first sentence of each article when constructing Wiki-QBMD dataset.

QBMD Evaluation Datasets. To build these datasets, we select query topics with more than five relevant documents for each evaluation dataset and sample datasets with the number of query documents from 1 to 5. Defining the number of query documents

[1] Our datasets are available at https://github.com/zhiqihuang/Hint/

as the query length, we sample the datasets such that the query documents in the dataset with query length k is the subset of query documents in the datasets with query length *more than* k. Because there are variations in the utility of relevant documents, we repeat the sampling process five times and build 5 collections to compensate for the possible selection of relevant documents with low-grade relevancy for a query topic. In Section 6 and 7, we report the mean and standard deviation among the collections. Further, since the number of queries is limited in our evaluation dataset we adopt 5-fold cross validation for fine-tuning and testing for each collection. For each fold, the training, validation, and test data are 60%, 20%, and 20% of the query set, respectively.

- *Robust04-QBMD.* We build Robust04-QBMD based on the standard ad-hoc retrieval dataset, Robust04, which the corpus consist of 528K newswire articles. Robust04 has 250 query topics, among them 233 have more than five relevant documents.
- *Multi-News-QBMD.* We build Multi-News-QBMD based on the Multi-News [14] dataset, a large-scale multi-document summarization dataset consisting of human-written summaries of multiple news articles. The human-written summaries are the target sequence and the news articles linked in the summary articles are source sequence in the summarization task. In QBMD, we create our corpus by collecting all source sequences, each of which we define as relevant to the summary and serving as a query-document for a specific topic. To avoid misguiding our retrieval approaches and to decrease noise, we delete query topics where at least one of its source documents appeared as a source document in other query topics.

QBSD Evaluation dataset. For a comprehensive evaluation, we further employ a benchmark QBSD dataset, COLIEE 2021. The dataset is in the legal retrieval domain, and the search task is a QBSD retrieval task where the query only contains one single example law case. This collection contains 4415 legal cases with a training and a test set of 650 and 250 query cases, respectively. And each query case has 5 relevant documents on average.

5 HINT MODEL

5.1 Neural Retrieval Approach

We employ a two-stage retrieval approach for addressing the QBMD problem, where first we obtain an initial set of candidate documents using a lexical matching retrieval technique and then re-rank the initial set of candidate documents using a neural re-ranker.

For the neural re-ranker, we propose HINT, a bi-encoder retrieval architecture based on hierarchical attention. Figure 1 depicts the architecture of HINT, which comprises a query encoder, a document encoder, and a scoring function. In general, given a set of example documents as the query Q and a candidate document d, the query encoder converts Q into a bag of contextualized vectors E_Q, while the document encoder maps d into another bag E_D. The vectors in E_Q are contextualized based on the content from the multiple documents. Then HINT computes a relevance score using the similarity matrix between E_Q and E_D. Despite the length of the documents, the scoring function weighs query documents equally by selecting the top-K "matched" parts between each query document and candidate document. Next, we introduce each model component in detail.

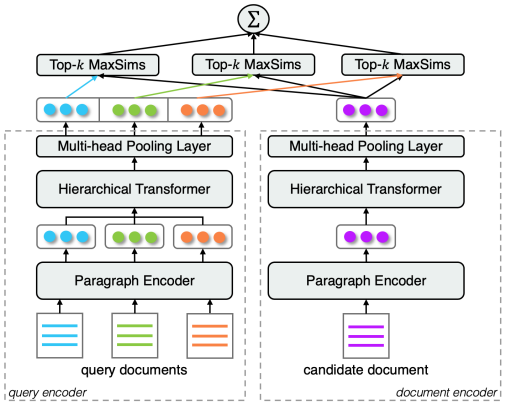

Figure 1: Overview of HINT architecture.

5.2 Query Encoder

Unlike keyword-based retrieval tasks, where most query words contribute to the relevance matching process, the information need is scattered throughout the query documents in QBMD. As a document often belongs to various topics, some parts of the query could be irrelevant to the user's intent. Therefore, instead of treating the query documents as a long sequence, we break query documents into paragraphs and build a hierarchical encoder based on the Hierarchical Transformer.

Interactions within paragraph. We parse query documents into a set of paragraphs by moving a sliding window over each query document with a pre-defined stride size. Each step of this sliding window is taken to be a paragraph. To mark the boundary of a document, we prepend a special token <d> to the first paragraph indicating the beginning of a document (BOD) and append token </d> to the last paragraph as the end of a document (EOD) for each query document. Let q_i be the i-th query document with L_i paragraphs, the input to the query encoder is in the format of <d>$p_{i,1} \cdots p_{i,L_i}$</d><d>$p_{(i+1),1} \cdots$</d>, where p_{ij} represents the j-th paragraph in the i-th query document.

We employ a Transformer-based module to encode tokens within a paragraph into the hidden vector space. Tokens are first represented by their corresponding embeddings. In QBMD task, we need to consider two positional embeddings: the position of the paragraph (e_{pop}) and the position of the token within the paragraph (e_{pot}). To distinguish between query documents and candidate documents, a type embedding (e_{typ}) is also applied to each token. Thus, a given token's input representation is the sum of the corresponding token, position, and type embeddings.

The token input representations are then passed into a multi-layer Transformer. We use the same network architecture as the vanilla transformer layer [46]. The multi-head attention mechanism allows each token to attend to other tokens within the same paragraph through different attention distributions. The output of this module is contextualized token-level representations for each paragraph. Let $\mathcal{D}^N \in \mathbb{R}^{L \times S \times d}$ denote the output of the paragraph encoder for a query with N documents, then $\mathcal{D}^N$ contains a set of token representations in each paragraph from query documents:

$$\mathcal{D}^N = \{\mathcal{P}_{ij} \in \mathbb{R}^{S \times d} \mid 1 \leq i \leq N, 1 \leq j \leq L_i \text{ and } i, j \in \mathbb{Z}^+\}$$

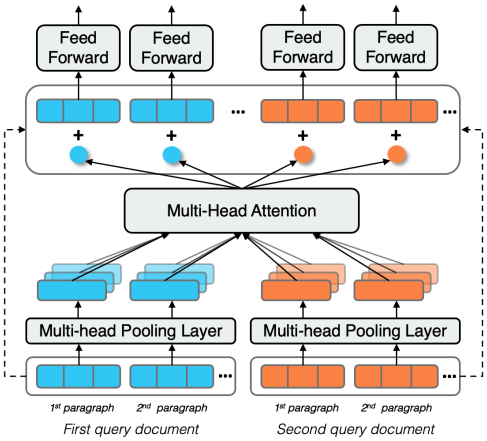

Figure 2: The architecture of hierarchical transformer layer.

where S is the paragraph length (stride). L_i is the number of paragraphs in query document QD_i and $L = \sum_i L_i$. Because this step aims to build the token-level semantic representation, we can take advantage of a pre-trained Transformer-based semantic search model. In fact, the token embeddings and Transformer layers are initialized by *all-mpnet-base-v2*[1], a pre-trained model using a billion sentence pairs. We keep the beginning of sequence (BOS) and end of sequence (EOS) tokens from this pre-trained model as they now indicate the beginning and the end of a paragraph.

Since the interaction at this stage is between tokens within a paragraph, the encoder does not have to process all paragraphs in query documents simultaneously. We can sequentially feed paragraphs into the encoder with a small batch size, making this encoding module memory efficient.

Interactions across paragraphs. We use the hierarchical transformer (HT) layers [29] to model the interactions across paragraphs. The complete architecture of an HT layer is shown in Figure 2. We first obtain a group of fixed length representations for each paragraph by applying a multi-head pooling layer to each $\mathcal{P}_{ij}$ in $\mathcal{D}^N$. Suppose $T_{ijk} \in \mathbb{R}^d$ is the k-th token representation in p_{ij}, let $W_a \in \mathbb{R}^{h \times d}$ be a trainable weight vector, $W_a T_{ijk}$ projects token representation into h different scores. We consider h as the number of heads. Then for the z-th head, we calculate a probability weight distribution over all tokens within the paragraph by applying the softmax layer to the corresponding score vector.

$$A^z_{ijk} = \frac{\exp(W^z_a T_{ijk})}{\sum_{k=1}^{S} \exp(W^z_a T_{ijk})} \tag{1}$$

where W^z_a is the z-th row of W_a. For each head, we also introduce a value vector $W^z_b \in \mathbb{R}^{d_{head} \times d}$ which projects T_{ijk} into a subspace with dimension of $d_{head} = d/h$.

$$B^z_{ijk} = W^z_b T_{ijk} \tag{2}$$

We compute a new pooled representation for each head using weighted summation followed by another linear transformation $W^z_p \in \mathbb{R}^{d_{head} \times d_{head}}$ and layer normalization, denoted as $\mathrm{LN}(\cdot)$.

$$H^z_{ij} = \mathrm{LN}(W^z_p \sum_{k=1}^{S} A^z_{ijk} B^z_{ijk}) \tag{3}$$

This multi-head pooling approach allows the model to extract different kinds of information over different regions of representation hyperspace. Now for each paragraph, the layer outputs h fixed-length vectors representing the semantics within a paragraph. Next, we apply the multi-head attention mechanism to learn the interaction across the paragraphs.

$$Q^z_{ij} = W^z_q H^z_{ij}; \quad K^z_{ij} = W^z_k H^z_{ij}; \quad V^z_{ij} = W^z_v H^z_{ij}$$

$$C^z_{ij} = \mathrm{softmax}(\frac{Q^z_{ij} K^{z\,\mathrm{T}}_{ij}}{\sqrt{d_{head}}}) V^z_{ij}$$

C^z_{ij} is the the context vector generated by dot-product self-attention over all paragraphs on head z. And the output of the multi-head attention, C_{ij}, is a linear transformation of the concatenated context vectors from all heads.

$$C_{ij} = W_o \left[C^1_{ij}, C^2_{ij}, \cdots, C^h_{ij}\right]$$

where $W^z_q, W^z_k, W^z_v \in \mathbb{R}^{d_{head} \times d_{head}}$ and $W_o \in \mathbb{R}^{d \times d}$ are trainable vectors. Since this attention is paid to all paragraphs, C_{ij} contains information from multiple query-documents. The output of a HT layer is a combination of the token-level and paragraph-level representations with the residual connection. Suppose T'_{ijk} is the output of HT layer with respect to T_{ijk},

$$T'_{ijk} = \mathrm{LN}(T_{ijk} + \mathrm{FFN}(T_{ijk} + C_{ij}))$$

where $\mathrm{FFN}(\cdot)$ is the the position-wise feed-forward networks with rectified activation function: $\mathrm{FFN}(x) = \max(0, xW_1 + b_1)W_2 + b_2$. This way, each token can collect information across queries and in a hierarchical and efficient manner.

Query document representation. Because in QBMD, there could be several parts of the query document that are relevant to the query intent, instead of generating one single representation vector for each query document, we choose to represent the query documents by a bag of vectors. The output layer of the query encoder consists another multi-head pooling layer (MHP) and a fully connected ranking head. Suppose T'_{ijk} is the output from the HT layer and there are h heads in MHP:

$$\mathrm{MHP}(T'_{ijk}) = \{E^z_{ij} \mid 1 \le z \le h\}$$

$$E_{ij} = W_r \left[E^1_{ij}, E^2_{ij}, \cdots, E^h_{ij}\right]$$

E^z_{ij} is the result of multi-head pooling on z-th head, see equations (1) to (3). We finally fuse information from all heads by concatenating all pooling results and applying the ranking head:

$$E_{ij} = W_r \left[E^1_{ij}, E^2_{ij}, \cdots, E^h_{ij}\right]$$

where $W_r \in \mathbb{R}^{d \times d}$ is the ranking head. The outputs of the query encoder, E_q, is then a set of contextualized paragraph-level representations: $E_Q = \{E_{ij} \mid 1 \le i \le N,\ 1 \le j \le L_i\}$.

In QBMD, the order of the query-documents should not affect the relevance judgement. Note that the query encoder of HINT is *permutation equivariant* at document level (indicated by i). Suppose π is a permutation of the query documents, then $E_{\pi q} = \pi E_q$.

[1] https://huggingface.co/sentence-transformers/all-mpnet-base-v2/

5.3 Document Encoder and Scoring Function

As a bi-encoder architecture, the document encoder in HINT has a similar architecture as the query encoder. We first segment a candidate document d into L_d consecutive paragraphs, to which we prepend BOD token (<d>) to the first paragraph and append EOD token (</d>) to the last paragraph. Unlike queries, the document token type embedding (e^d_{typ}) is added to each token in the paragraph encoder. We build the query encoder and document encoder as a Siamese neural network where parameter weights are shared across two encoders. After passing L_d paragraphs through the document encoder, we obtain a bag of contextualized representations E_D for the candidate document: $E_D = \{E_l \mid 1 \leq l \leq L_d\}$.

Inspired by the late-interaction scoring function [21, 22], we propose a **stratified late-interaction** that considers each query document equally and following the ColBERT [22] model we employ maximum cosine similarity (MaxSim) in our scoring function. First, we find the MaxSim of each vector in E_q with vectors in E_D. Intuitively, this MaxSim operation softly searches each query paragraph against all the paragraphs in the candidate document for the best matching in terms of the maximum similarity:

$$M_{ij} = \Big\{ \max_{l=1}^{L_d}(E_{ij} \cdot E_l) \mid 1 \leq i \leq N,\ 1 \leq j \leq L_i \Big\}$$

We first sum only the top-K highest similarity by MaxSim for each query document as a single document matching score and then combine the matching score from multiple query documents via summation. In this way, despite the length of the document, only K most significant paragraphs can contribute to the relevance score between q and d.

$$F(q,d) := \sum_{i=1}^{N} \sum_{j=1}^{K} \text{top}_K\{M_{ij} : 1 \leq j \leq L_i\} \tag{4}$$

Because the query encoder is permutation equivariant, under a permutation of indices of the query documents, the relevant score of (q,d) remains the same, that is $F(q,d) = F(\pi q, d)$. Therefore, HINT is *permutation invariant*.

6 FIRST-STAGE RETRIEVAL

We investigate and benchmark different strategies to form the query for our initial ranking stage.

Keyphrase. We concatenate query documents into one long sequence and extract key phrases as the query. We select 1-gram to 3-grams as the candidate phrases and rank based on their TF-IDF scores calculated using an unsupervised keyphrase extraction model with multipartite graphs [8, 9] and select the top 100 key phrases with their corresponding weight. We use Galago's query language model[2] and its implementation of the query likelihood model with the default parameter to retrieve the documents.

SummPip. We exploit a multi-document summarization method to generate a summary from query documents and use it as the query. Since the explicit query (the ground truth summary) is not available, the fine-tuning of a deep supervised summarization model in the QBMD domain is not possible. Therefore, we select SummPip [53], an unsupervised extractive multi-document summarization method based on sentence graphs and spectral clustering, to summarize the query documents. Then we use the summarization as the textual query input for Galago's query likelihood model with default parameters.

docT5query. We leverage doc2query [38], which generate questions for a given passage, to identify the latent information need (i.e., query) of the query-documents. Since the input of doc2query is passage length we break down our query documents into passages and using doc2query we generate questions for each passage. We select the top 10 questions generated for each passage and concatenate them to each other and build a mid-point representation for the query-documents. Then, similar to the Keyphrase approach we extract the key phrases from the mid-point representation and rank them based on their TF-IDF scores. Finally, we select the top 100 keyphrases and their corresponding weights and use Galago's query likelihood implementation with its default parameters to retrieve documents.

Table 2 shows the results of our first-stage retrieval methods with a focus on Recall as our primary evaluation metric. We can see that the Keyphrase approach outperform SummPip and docT5query across all measure for all datasets. In particular, the Keyphrase recall is 19%, 0.8% and 4.5% above the docT5query, the second best-performing approach, for Wiki-QBMD, Robust04-QBMD and Multi-News-QBMD respectively. Therefore, we select Keyphrase as our approach for first-stage retrieval and obtaining the initial set of candidate documents.

7 EXPERIMENTS

7.1 Baselines

We compare HINT with the following methods:

- **Keyphrase**: This is the method we use for the first stage of retrieval which is explained in Section 6 in details. The compared neural models (including HINT) are re-ranking the top 100 retrieved documents by the Keyphrase method.
- **Rocchio**: Since the user already provides several relevant documents as the query, we design an unsupervised re-ranking method based on the Rocchio feedback approach. We first convert all documents into TF-IDF vectors. Then, we consider the query documents as the positive feedback and compute the average of their document vectors. For the negative feedback, we use the average of the bottom 5 documents from the initial rank list. After a grid-search on training data, we set $\beta = 0.75$, $\gamma = -0.30$ for positive and negative vectors respectively for Wiki-QBMD; $\beta = 1.0$, $\gamma = -0.25$ for Robust04-QBMD and $\beta = 0.8$, $\gamma = -0.25$ for Multi-News-QBMD. Finally, we use the weighted sum of the feedback vectors to re-rank the documents.
- **CD-Longformer**: Abolghasemi et al. [2] adopts a BERT-based cross-encoder architecture for re-ranking documents in the QBD task. However, since the input length of concatenation of multiple documents in QBMD is generally longer than the input sequence length of conventional transformer models such as BERT, we employ Longformer [7], a long-sequence transformer model. We take advantage of a model introduced by Caciularu et al. [10], which is pre-trained on a Multi-Document Summarization corpus [14] with the goal of capturing cross-text relationships, particularly aligning or linking matching information elements across

[2]https://www.lemurproject.org/galago.php/

Table 2: Query extraction methods for the first stage retrieval. For each column, the highest value is marked with bold text. At this stage, we select R@100 as the primary evaluation metric. Subscripts refer to the standard deviation of 5 corpuses.

Method	Wiki-QBMD			Robust04-QBMD			Multi-News-QBMD		
	MAP	MRR	R@100	MAP	MRR	R@100	MAP	MRR	R@100
Keyphrase	$\mathbf{0.1803}_{(0.0056)}$	$\mathbf{0.3970}_{(0.0095)}$	$\mathbf{0.4812}_{(0.0090)}$	$\mathbf{0.1358}_{(0.0060)}$	$\mathbf{0.5377}_{(0.0202)}$	$\mathbf{0.3127}_{(0.0101)}$	$\mathbf{0.4080}_{(0.0032)}$	$\mathbf{0.6089}_{(0.0085)}$	$\mathbf{0.7738}_{(0.0035)}$
SummPip	$0.1367_{(0.0066)}$	$0.3298_{(0.0094)}$	$0.3688_{(0.0116)}$	$0.1002_{(0.0064)}$	$0.4464_{(0.0210)}$	$0.2405_{(0.0088)}$	$0.3917_{(0.0060)}$	$0.5987_{(0.0120)}$	$0.7538_{(0.0073)}$
docT5query	$0.1502_{(0.0043)}$	$0.3516_{(0.0049)}$	$0.4110_{(0.0068)}$	$0.1353_{(0.0056)}$	$0.5191_{(0.0175)}$	$0.3088_{(0.0076)}$	$0.3674_{(0.0030)}$	$0.5590_{(0.0093)}$	$0.7424_{(0.0028)}$

documents. We refer to it as **C**ross-**D**ocument **Longformer** (CD-Longformer) in our table of results. Following Caciularu et al. [10], we tagged sentences of each document with begin (<s>) and end of sentence(</s>) tokens as well as labeling begin and end of documents with the special tokens of begin (<doc-s>) and end of document (</doc-s>). Further, we differentiate the query-documents input and the candidate document by special tokens of <query> and <cand>. Exceeding the transformers input size is inevitable, therefore for query-documents we set a limit of 2600 tokens and truncate the longest document in case of passing it. Further, since the candidate document is only one document, we set the maximum token length to 1400 tokens.

- **KW-BERT**: We design another cross-encoder re-ranking architecture by leveraging the extracted bag-of-phrases in the first stage retrieval as the queries which converts QBMD into a keyword based retrieval problem. Following Nogueira and Cho [37], we use BERT as the re-ranker and feed the query phrases as the first sequence and the candidate document as the second sequence. The query sequence is truncated at 64 tokens and the complete input sequence is truncated at the maximum length of 512 tokens. We use the [CLS] token vector as input to a fully connected layer to obtain the relevance score. The BERT parameters are also initialized from a cross-encoder retrieval model, *ms-marco-MiniLM-L-12-v2*[3], which is pre-trained on the MS MARCO [36] passage retrieval dataset. Note that this is not a QBMD approach because it explcitly has the latent query. It thus serves as an a upper bound of sort.
- **EQ-BERT**: We can utilize the explicit query as a reference to evaluate the performance of models in QBMD setting. For queries in Wiki-QBMD, the title and first sentence of the Wikipedia article can be treated as the explicit query. In Robust04-QBMD, the explicit query is the corresponding description field of the topic in Robust04. And for Multi-News-QBMD, we use the ground truth summary as the explicit query. Replacing key phrase query in KW-BERT by the explicit query, we create another cross-encoder model which serves as an upper bound for other methods.

7.2 Implementation Details

Model configuration. We parse the input sequence into paragraphs with the fixed length of 128 tokens and the stride between paragraph is 50 tokens. If there are fewer than 128 tokens, padding tokens ([PAD]) are added to the last paragraph. We employ 3 HT layers to model the interactions between paragraphs. We choose 12 as the number of heads for all multi-head mechanism in HINT. We select $K = 5$ most significant paragraphs from each query document for the scoring function in equation (4). We use the positive documents and randomly sample negative documents from the ranked list of the first stage retrieval to form training triplets. We follow the same training schema for all neural re-ranking models except CD-Longformer: First, we train on the Wiki-QBMD dataset for 10 epochs with a batch size of 256. For the CD-Longformer model, the decrease in loss trajectory is slower compared to other models, so we train it on Wiki-QBMD with a batch size of 24 for 22 epochs. Then the models are fine-tuned on Robust04-QBMD, Multi-News-QBMD, and COLIEE 2021 datasets. All models are trained using AdamW optimization algorithm [30] with a learning rate of 2e-5 for CD-Longformer and 5e-6 for all the other models.

Evaluation. For evaluating retrieval effectiveness at the re-ranking stage, we report mean average precision (MAP) and mean reciprocal rank (MRR) of the top 100 ranked documents and precision of the top 10 retrieved documents (P@10). For QBSD, we adopt the official metric used in the COLIEE competition [41], to report precision, recall, and F-score at the cut-off of 5, as well as ndcg@10 for comparison with Althammer et al.'s proposed model [3]. We determine statistical significance using the two-tailed paired t-test with a p-value less than 0.05 (i.e., 95% confidence level).

7.3 Results

Main Results. Table 3 lists the evaluation results of queries that have 3 relevant documents on three QBMD datasets. As a neural re-ranker, HINT significantly improves upon Keyphrase and Rocchio. Moreover, regarding MAP and MRR, we observed substantial improvements from HINT compared with CD-Longformer and KW-BERT across all datasets. However, we can see that EQ-BERT performs better than HINT by a large margin. This performance gap shows that the model using the explicit query still has the advantage of precise information needs. In contrast, because KW-BERT and EQ-BERT have the same model architecture, the sub-optimal results of the KW-BERT are mainly due to the low quality of the key phrase extracted as the query. The CD-Longformer encodes the concatenation of all query documents and the candidate document. With the increase of the query documents, it still faces the limitation of the number of maximum tokens. Further, we observe that the CD-Longformer fine-tuned model on Wiki-QBMD is not performing well on the other two datasets. We hypothesize the model might overfit on the Wikipedia articles and fine-tuning on the limited number of queries does not help. In general, Robust04-QBMD has a larger standard deviation than the other two datasets indicating it has more variations in the utility of relevant documents

Ablation Studies. To evaluate the design purpose of HINT, we consider the following model variations.

[3]https://huggingface.co/cross-encoder/ms-marco-MiniLM-L-12-v2/

Table 3: Model performance on QBMD datasets. Note that the EQ-BERT is reported as an upper bound reference. For each column (except EQ-BERT), the highest value is marked with bold text. Subscripts refer to the standard deviation of 5 corpuses. For HINT, statistically significant improvements are marked by ▲ (over KW-BERT), ♦ (over Longformer) and ⋆ (over Rocchio).

Model	Wiki-QBMD			Robust04-QBMD[4]			Multi-News-QBMD		
	MAP	MRR	P@10	MAP	MRR	P@10	MAP	MRR	P@10
Keyphrase	$0.1803_{(0.0056)}$	$0.3970_{(0.0095)}$	$0.1635_{(0.0043)}$	$0.1358_{(0.0060)}$	$0.5377_{(0.0202)}$	$0.2957_{(0.0089)}$	$0.4080_{(0.0032)}$	$0.6089_{(0.0085)}$	$0.1871_{(0.0005)}$
EQ-BERT	0.2868	0.6451	0.2355	0.1687	0.6591	0.3901	0.5077	0.7609	0.2214
Rocchio	$0.1898_{(0.0077)}$	$0.4346_{(0.0115)}$	$0.1717_{(0.0030)}$	$0.1315_{(0.0062)}$	$0.5670_{(0.0163)}$	$0.3026_{(0.0137)}$	$0.4210_{(0.0051)}$	$0.6422_{(0.0075)}$	$0.1928_{(0.0015)}$
CD-Longformer	$0.2234_{(0.0059)}$	$0.4715_{(0.0148)}$	$0.2019_{(0.0054)}$	$0.0963_{(0.0031)}$	$0.3994_{(0.0125)}$	$0.2231_{(0.0044)}$	$0.3143_{(0.0044)}$	$0.4882_{(0.0031)}$	$0.1629_{(0.0017)}$
KW-BERT	$0.2229_{(0.0070)}$	$0.4837_{(0.0165)}$	$0.2000_{(0.0067)}$	$0.1228_{(0.0052)}$	$0.5010_{(0.0148)}$	$0.2715_{(0.0139)}$	$0.4541_{(0.0030)}$	$0.6449_{(0.0054)}$	$0.2102_{(0.0023)}$
HINT	$0.2580^{\star\blacklozenge\blacktriangle}_{(0.0099)}$	$0.5303^{\star\blacklozenge\blacktriangle}_{(0.0156)}$	$0.2258^{\star\blacklozenge\blacktriangle}_{(0.0056)}$	$0.1522^{\star\blacklozenge\blacktriangle}_{(0.0077)}$	$0.6048^{\star\blacklozenge\blacktriangle}_{(0.0200)}$	$0.3330^{\star\blacklozenge\blacktriangle}_{(0.0119)}$	$0.4775^{\star\blacklozenge\blacktriangle}_{(0.0019)}$	$0.6700^{\star\blacklozenge\blacktriangle}_{(0.0026)}$	$0.2164^{\star\blacklozenge}_{(0.0009)}$

Table 4: Ablation study of our model components. For each column, the highest value is marked with bold text. We report the performance drop on MAP (in percentage) of each ablation from the complete HINT architecture.

Model	Wiki-QBMD			Robust04-QBMD			Multi-News-QBMD		
	MAP	MRR	P@10	MAP	MRR	P@10	MAP	MRR	P@10
HINT	**0.2580**	**0.5303**	**0.2258**	**0.1522**	**0.6048**	**0.3330**	**0.4775**	0.6700	**0.2164**
HINT-Flat	0.2247(−12.9%)	0.4849	0.1990	0.1367(−10.2%)	0.5716	0.3020	0.4593(−3.8%)	0.6668	0.2064
HINT-One	0.2423(−6.0%)	0.5050	0.2140	0.1446(−5.0%)	0.5926	0.3215	0.4665(−2.3%)	**0.6713**	0.2108
HINT-Separate	0.2420(−6.2%)	0.5024	0.2155	0.1444(−5.1%)	0.5894	0.3272	0.4647(−2.6%)	0.6641	0.2080

- **HINT-FLAT**: To evaluate the effect of the hierarchical architecture of HINT, we remove the HT and MHP layers so that paragraphs are encoded independently by the paragraph encoder, and tokens can only attend to each other within the same paragraph. We apply the mean pooling to the output of the paragraph encoder to generate a representation per paragraph.
- **HINT-One**: Next, we add hierarchical layers back to the model. However, we consider multiple documents as one document by removing the BOD and EOD tokens that indicate the boundaries. The model still parses the long sequence into paragraphs, but the positional embedding now only reflects the order of the current paragraph in the whole input sequence. This variant explores the effect of multiple documents as the input. Note that under this setting, the model is no longer permutation invariant.
- **HINT-Separate**: Finally, we explore the effect of the interactions between query documents. In this variation, we pair the candidate document with each query document. Thus, the mode only scores between one query document and the candidate document at a time. The final relevant score is the sum of all pairs. Like ColBERT [22], the interaction between query documents is delayed to the scoring function.

The results in Table 4 suggest that removing any of these design features causes a drop in the model performance. The most significant drop in HINT-Flat shows that the most critical component is the hierarchical layers that allows interaction between paragraphs.

7.4 Number of Query Documents

Intuitively, in QBMD, more query documents should lead to a richer context of the query topic [48, 52]. While HINT is at first trained on Wiki-QBMD dataset with 3 query documents, it does not assume the number of query documents, and it can easily adapt to a different number of query documents in the fine-tuning stage. Figure 3 shows the effect of different number of query documents in QBMD for our two evaluation datasets that have 1 to 5 query documents. We observe that regardless of the number of query documents, HINT consistently improves first-stage retrieval and outperforms both KW-BERT and Rocchio significantly. Surprisingly, on Robust04-QBMD, its MAP matches EQ-BERT given 4 and 5 query documents, while as a re-ranking model, KW-BERT fails to outperform the first stage retrieval method, Keyphrase. As expected, the retrieval performance first increases with more query documents provided then plateaus. Such trends suggest that the content related to the information needs in query documents can be saturated. After the turning point, increasing the number of query documents provides more noise than the query-relevant context, causing the performance to drop.

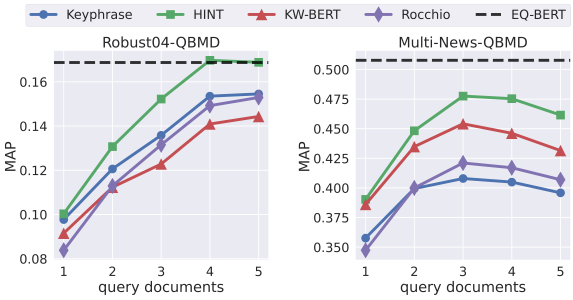

Figure 3: Effect of different numbers of documents in query.

Table 5: Model performance on COLIEE 2021 test set. We select the first-stage method based on R@100 for all rerankers. The highest value is marked with bold text.

	Method	COLIEE 2021				
		R@100	nDCG@10	P@5	R@5	F1
First Stage Retrieval	$BM25_{optimized}$ [6]	**0.6651**	0.2753	0.1528	0.2756	0.1966
	Keyphrase	0.6123	0.2614	0.1512	0.2509	0.1887
	SummPip	0.5627	0.2135	0.1352	0.2397	0.1728
	docT5query	0.5127	0.2083	0.1160	0.2084	0.1490
Reranker	Rocchio	-	0.2512	0.1404	0.2587	0.1817
	TLIR [32]	-	NR	0.1533	0.2556	0.1917
	KW-BERT	-	0.3189	0.1720	0.3252	0.2250
	MTFT-BERT [2]	-	0.3137	0.1744	0.2999	0.2205
	HINT	-	**0.3410**	**0.1800**	**0.3329**	**0.2334**
DPR	PARM VRRF [3]	0.6396	0.1280	NR	NR	NR

Table 6: Performance comparison between query terms extracted by Keyphrase method and re-ranked by HINT weight matrix on Robust04-QBMD using 3 query documents.

Query Likelihood	MAP	MRR	P@10	R@100
+ Keyphrase	0.1358	0.5377	0.2957	0.3127
+ Keyphrase re-ranked by HINT	**0.1445**	**0.5598**	**0.3180**	**0.3318**

7.5 Query by Single Document

To explore the generalizability of our approach to QBSD, we use COLIEE 2021 [41] a QBSD benchmark dataset in in the legal retrieval domain, where both the query and documents are law cases. We first fine-tune the model checkpoints trained on Wiki-QBMD on the COLIEE training set and then evaluate them on the test set. Table 5 shows the results of first-stage retrieval, re-rankers, and the Dense Passage Retrieval (DPR)-based [3] models for COLIEE 2021. We use numbers reported in prior studies and mark any missing measures as "Not Reported (NR)" since they were not present in the original study. $BM25_{optimized}$ [6] use a term extraction method tuned for legal case collection to achieve the best recall during first-stage retrieval. Thus, we adopt it as the initial retrieval method for reranking. Both TLIR [32] and MTFT-BERT [2] use a cross-encoder architecture for re-ranking. It can be seen that HINT achieves the best performance among all compared methods. Moreover, limited by the model's maximum input sequence length, cross-encoder based methods, like TLIR and MTFT-BERT, are hard to generalize from QBSD to QBMD.

7.6 Token Weights Analysis

To understand the query encoding process, we analyze the token weight matrix in the MHP layer. Taking the weight matrices from the final MHP layer (equation 1), we first obtain a single weight per token by averaging multiple heads. Then for a word split by BPE tokenization, we select the most significant weight among all subwords to represent the original one. Thus, a score based on pooling weights is assigned to each word. This score approximately reflects the impact of words on the query representations learned by HINT. We re-rank the query phrases extracted by the Keyphrase method according to the average word score within each phrase, run the query likelihood model using re-ranked phrases, and compare the results in Table 6. Although HINT is a neural model primarily designed for semantic matching, using the weights learned from the model to re-rank query phrases can improve performance on a lexical-based retrieval model, indicating it focuses on words close to the information needs. Table 7 shows one success and one failure case of phrases re-ranked by HINT. Compared with top phrases extracted based on the TF-IDF score, HINT tend to find terms that are more related to the explicit query.

Table 7: Case study of top query phrases extracted by Keyphrase and re-ranked by HINT weight matrix.

Example (Robust04 Query Description)	**Top-3 by Keyphrase**	**Top-3 re-ranked by HINT**	**Change of P@10**
What role does blood-**alcohol** level play in automobile **accident** fatalities?	pleads guilty, guilty, sentenced	**alcohol**, **accident**, state	+0.2567
What is the status of the Three Gorges Project?	resettlement, Beijing, three gorges	development, China, three gorges	−0.1259

8 CONCLUSION

In this work, we investigate the problem of Query-by-Multiple-Document (QBMD), where the users' information need is stated by multiple relevant documents. We present HINT, a neural bi-encoder retrieval architecture based on hierarchical attention for the task of QBMD. HINT is capable of capturing the interaction at different levels of semantics. We construct three QBMD datasets that support both the training and evaluation of deep neural retrieval models. For either query-by-single or multiple documents, our comprehensive experimental results demonstrate that HINT significantly outperforms other baselines, including the neural baselines with cross-encoder architecture, showing the importance of combining within and across query documents interaction.

ACKNOWLEDGMENTS

This work was supported in part by the Center for Intelligent Information Retrieval and This research is based upon work supported in part by the Center for Intelligent Information Retrieval, and in part by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via Contract No. 2019-19051600007 under Univ. of Southern California subcontract no. 124338456. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of ODNI, IARPA, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation therein. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the sponsor.

REFERENCES

[1] Amin Abolghasemi, Arian Askari, and Suzan Verberne. 2022. On the Interpolation of Contextualized Term-Based Ranking with BM25 for Query-by-Example Retrieval. In *Proceedings of the 2022 ACM SIGIR International Conference on Theory of Information Retrieval* (Madrid, Spain) *(ICTIR '22)*. Association for Computing Machinery, New York, NY, USA, 161–170. https://doi.org/10.1145/3539813.3545133

[2] Amin Abolghasemi, Suzan Verberne, and Leif Azzopardi. 2022. Improving BERT-based Query-by-Document Retrieval with Multi-task Optimization. In *Advances in Information Retrieval: 44th European Conference on IR Research, ECIR 2022*.

[3] Sophia Althammer, Sebastian Hofstätter, Mete Sertkan, Suzan Verberne, and Allan Hanbury. 2022. PARM: A Paragraph Aggregation Retrieval Model for Dense Document-to-Document Retrieval. In *ECIR 2022, Stavanger, Norway*. Springer.

[4] Reinald Kim Amplayo and Mirella Lapata. 2021. Informative and Controllable Opinion Summarization. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*. 2662–2672.

[5] Diego Antognini and Boi Faltings. 2019. Learning to Create Sentence Semantic Relation Graphs for Multi-Document Summarization. *EMNLP-IJCNLP* (2019).

[6] AA Askari and SV Verberne. 2021. Combining lexical and neural retrieval with longformer-based summarization for effective case law retrieva. In *Proceedings of the second international conference on design of experimental search & information REtrieval systems*. CEUR, 162–170.

[7] Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150* (2020).

[8] Florian Boudin. 2016. pke: an open source python-based keyphrase extraction toolkit. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: System Demonstrations*. Osaka, Japan, 69–73. http://aclweb.org/anthology/C16-2015

[9] Florian Boudin. 2018. Unsupervised Keyphrase Extraction with Multipartite Graphs. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*. 667–672.

[10] Avi Caciularu, Arman Cohan, Iz Beltagy, Matthew E Peters, Arie Cattan, and Ido Dagan. 2021. CDLM: Cross-document language modeling. *arXiv preprint arXiv:2101.00406* (2021).

[11] Arman Cohan, Sergey Feldman, Iz Beltagy, Doug Downey, and Daniel S Weld. 2020. SPECTER: Document-level Representation Learning using Citation-informed Transformers. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 2270–2282.

[12] W. Bruce Croft. 2019. The Importance of Interaction for Information Retrieval. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval* (Paris, France) *(SIGIR'19)*. Association for Computing Machinery, New York, NY, USA, 1–2.

[13] Khalid El-Arini and Carlos Guestrin. 2011. Beyond keyword search: discovering relevant scientific literature. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*. 439–447.

[14] Alexander Richard Fabbri, Irene Li, Tianwei She, Suyi Li, and Dragomir Radev. 2019. Multi-News: A Large-Scale Multi-Document Summarization Dataset and Abstractive Hierarchical Model. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. 1074–1084.

[15] Jibril Frej, Didier Schwab, and Jean-Pierre Chevallet. 2020. WIKIR: A Python Toolkit for Building a Large-scale Wikipedia-based English Information Retrieval Dataset. In *Proceedings of the 12th Language Resources and Evaluation Conference*. European Language Resources Association.

[16] Atsushi Fujii, Makoto Iwayama, and Noriko Kando. 2007. Overview of the Patent Retrieval Task at the NTCIR-6 Workshop.. In *NTCIR*.

[17] Viet Ha-Thuc, Yan Yan, Xianren Wu, Vijay Dialani, Abhishek Gupta, and Shakti Sinha. 2017. From query-by-keyword to query-by-example: LinkedIn talent search approach. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. 1737–1745.

[18] Xiao Han, Yuqi Liu, and Jimmy Lin. 2021. *The Simplest Thing That Can Possibly Work: (Pseudo-)Relevance Feedback via Text Classification*. Association for Computing Machinery, New York, NY, USA, 123–129.

[19] Jaap Kamps and Marijn Koolen. 2008. The importance of link evidence in Wikipedia. In *Advances in Information Retrieval: 30th European Conference on IR Research, ECIR 2008, Glasgow, UK, March 30-April 3, 2008. Proceedings 30*. Springer, 270–282.

[20] Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense Passage Retrieval for Open-Domain Question Answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Online, 6769–6781. https://doi.org/10.18653/v1/2020.emnlp-main.550

[21] Omar Khattab, Christopher Potts, and Matei Zaharia. 2021. Baleen: Robust Multi-Hop Reasoning at Scale via Condensed Retrieval. In *Advances in Neural Information Processing Systems*, M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (Eds.), Vol. 34. Curran Associates, Inc.

[22] Omar Khattab and Matei Zaharia. 2020. Colbert: Efficient and effective passage search via contextualized late interaction over bert. In *Proceedings of the 43rd International ACM SIGIR*.

[23] Mi-Young Kim, Juliano Rabelo, and Randy Goebel. 2019. Statute law information retrieval and entailment. In *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Law*. 283–289.

[24] Grace E Lee and Aixin Sun. 2018. Seed-driven document ranking for systematic reviews in evidence-based medicine. In *The 41st international ACM SIGIR conference on research & development in information retrieval*. 455–464.

[25] Canjia Li, Yingfei Sun, Ben He, Le Wang, Kai Hui, Andrew Yates, Le Sun, and Jungang Xu. 2018. NPRF: A Neural Pseudo Relevance Feedback Framework for Ad-hoc Information Retrieval. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*.

[26] Matteo Lissandrini, Davide Mottin, Themis Palpanas, and Yannis Velegrakis. 2018. Multi-example search in rich information graphs. In *2018 IEEE 34th International Conference on Data Engineering (ICDE)*. IEEE, 809–820.

[27] Matteo Lissandrini, Davide Mottin, Themis Palpanas, and Yannis Velegrakis. 2019. Example-based search: A new frontier for exploratory search. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1411–1412.

[28] Peter J Liu, Mohammad Saleh, Etienne Pot, Ben Goodrich, Ryan Sepassi, Lukasz Kaiser, and Noam Shazeer. 2018. Generating Wikipedia by Summarizing Long Sequences. In *International Conference on Learning Representations*.

[29] Yang Liu and Mirella Lapata. 2019. Hierarchical Transformers for Multi-Document Summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*.

[30] Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101* (2017).

[31] Congbo Ma, Wei Emma Zhang, Mingyu Guo, Hu Wang, and Quan Z Sheng. 2020. Multi-document summarization via deep learning techniques: A survey. *arXiv preprint arXiv:2011.04843* (2020).

[32] Yixiao Ma, Yunqiu Shao, Bulou Liu, Yiqun Liu, Min Zhang, and Shaoping Ma. 2021. Retrieving legal cases from a large-scale candidate corpus. *Proceedings of the Eighth International Competition on Legal Information Extraction/Entailment, COLIEE2021* (2021).

[33] Sheshera Mysore, Tim O'Gorman, Andrew McCallum, and Hamed Zamani. 2021. CSFCube-A Test Collection of Computer Science Research Articles for Faceted Query by Example. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.

[34] Shahrzad Naseri, Jeffrey Dalton, Andrew Yates, and James Allan. 2021. Ceqe: Contextualized embeddings for query expansion. In *European Conference on Information Retrieval*. Springer, 467–482.

[35] Mir Tafseer Nayeem, Tanvir Ahmed Fuad, and Yllias Chali. 2018. Abstractive unsupervised multi-document summarization using paraphrastic sentence fusion. In *Proceedings of the 27th International Conference on Computational Linguistics*.

[36] Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. MS MARCO: A human generated machine reading comprehension dataset. In *CoCo@ NIPS*.

[37] Rodrigo Nogueira and Kyunghyun Cho. 2019. Passage Re-ranking with BERT. *arXiv preprint arXiv:1901.04085* (2019).

[38] Rodrigo Nogueira, Wei Yang, Jimmy Lin, and Kyunghyun Cho. 2019. Document Expansion by Query Prediction. *arXiv preprint arXiv:1904.08375* (2019).

[39] NIST (National Institute of Standards and Technology). 2022. IARPA BETTER. https://ir.nist.gov/better/

[40] Florina Piroi and Allan Hanbury. 2019. Multilingual patent text retrieval evaluation: CLEF–IP. In *Information Retrieval Evaluation in a Changing World*. Springer.

[41] Juliano Rabelo, Randy Goebel, Mi-Young Kim, Yoshinobu Kano, Masaharu Yoshioka, and Ken Satoh. 2022. Overview and Discussion of the Competition on Legal Information Extraction/Entailment (COLIEE) 2021. *The Review of Socionetwork Strategies* 16, 1 (2022), 111–133.

[42] Gerard Salton and Chris Buckley. 1990. Improving retrieval performance by relevance feedback. *Journal of the American society for information science* (1990).

[43] Shota Sasaki, Shuo Sun, Shigehiko Schamoni, Kevin Duh, and Kentaro Inui. 2018. Cross-Lingual Learning-to-Rank with Shared Representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*.

[44] Yunqiu Shao, Jiaxin Mao, Yiqun Liu, Weizhi Ma, Ken Satoh, Min Zhang, and Shaoping Ma. 2020. BERT-PLI: Modeling Paragraph-Level Interactions for Legal Case Retrieval.. In *IJCAI*. 3501–3507.

[45] Mark D Smucker and James Allan. 2006. Find-similar: similarity browsing as a search tool. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*. 461–468.

[46] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30 (2017).

[47] Danqing Wang, Pengfei Liu, Yining Zheng, Xipeng Qiu, and Xuan-Jing Huang. 2020. Heterogeneous Graph Neural Networks for Extractive Document Summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 6209–6219.

[48] Shuai Wang, Harrisen Scells, Ahmed Mourad, and Guido Zuccon. 2022. Seed-Driven Document Ranking for Systematic Reviews: A Reproducibility Study. In

European Conference on Information Retrieval. Springer, 686–700.

[49] Linkai Weng, Zhiwei Li, Rui Cai, Yaoxue Zhang, Yuezhi Zhou, Laurence T Yang, and Lei Zhang. 2011. Query by document via a decomposition-based two-level retrieval approach. In *Proceedings of the 34th international ACM SIGIR conference on research and development in information retrieval*. 505–514.

[50] Kyle Williams, Jian Wu, and C Lee Giles. 2014. Simseerx: a similar document search engine. In *Proceedings of the ACM symposium on Document engineering*.

[51] Yin Yang, Nilesh Bansal, Wisam Dakka, Panagiotis Ipeirotis, Nick Koudas, and Dimitris Papadias. 2009. Query by document. In *Proceedings of the Second ACM International Conference on Web Search and Data Mining*. 34–43.

[52] Dell Zhang and Wee Sun Lee. 2009. Query-By-Multiple-Examples using Support Vector Machines. *Journal of Digital Information Management* 7, 4 (2009), 202.

[53] Jinming Zhao, Ming Liu, Longxiang Gao, Yuan Jin, Lan Du, He Zhao, He Zhang, and Gholamreza Haffari. 2020. Summpip: Unsupervised multi-document summarization with sentence graph compression. In *Proceedings of the 43rd International ACM SIGIR*.

[54] Mingzhu Zhu and Yi-Fang Brook Wu. 2014. Search by Multiple Examples. In *WSDM '14* (New York, New York, USA). Association for Computing Machinery, New York, NY, USA.

[55] Shengyao Zhuang and Guido Zuccon. 2021. Fast passage re-ranking with contextualized exact term matching and efficient passage expansion. *arXiv preprint arXiv:2108.08513* (2021).

[56] Shengyao Zhuang and Guido Zuccon. 2021. TILDE: Term Independent Likelihood MoDEl for Passage Re-Ranking. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Virtual Event, Canada) *(SIGIR '21)*. Association for Computing Machinery, New York, NY, USA, 1483–1492. https://doi.org/10.1145/3404835.3462922