MINISTRY OF EDUCATION AND SCIENCE OF UKRAINE
V.N. Karazin Kharkiv National University
Faculty of Mathematics and Informatics
Department of Theoretical and Applied Informatics

Master's Qualification Thesis

On the topic

Managing Web based Information Resources Under Uncertainty- Using pre trained models to enhance text retrieval and reduce search uncertainty

Executed by: _-year student, group MCS-64
specialty 122 «Computer Science»
Educational and research program
«Informatics»

_____LI YUNZHEN_____
(surname and initials)
Supervisor _____ Yurii Parfeniuk_
(surname and initials)
Reviewer __Dmytro Chumachenko_____
(surname and initials)

Kharkiv – 2024

CONTENT

INTRODUCTION

Research Background and Significance

In the digital age, the internet has become an immense ocean of information resources. However, the uncertainty inherent in these resources poses challenges for users in efficiently and accurately retrieving and identifying key information. With the rapid increase in the number of internet users, the rapid growth and management of information resources have become particularly important. Information Retrieval (IR) technology, which originated from library resource search and retrieval methods, has become a significant research area within Natural Language Processing (NLP) and Artificial Intelligence (AI), especially in managing and retrieving web-based information resources under uncertainty.

Purpose of the Study

The purpose of this study is to explore and develop effective information retrieval technologies to manage and retrieve web-based information resources, especially under conditions of uncertainty. Our goal is to enhance the accuracy and efficiency of information retrieval, enabling users to find relevant information more quickly, thereby improving user experience and contributing positively to social and economic development.

Object and Subject of the Research

The object of the research is web-based information resources, and the subject focuses on how to manage and retrieve these resources under conditions of uncertainty. This includes studying how to utilize advanced information retrieval technologies, such as statistical retrieval methods, machine learning and deep learning ranking methods, and pre-trained model ranking methods, to handle and

retrieve vast amounts of uncertain and dynamically changing web information resources.

Research Objectives

The main objectives of this study include:

1. Analyzing and comparing the performance of different information retrieval technologies in handling uncertain information resources.

2. Developing and optimizing pre-trained models to effectively handle long-text information resources and improve retrieval accuracy.

3. Exploring methods such as paragraph aggregation and abstract summary semantic enhancement to improve the retrieval effectiveness of long-text information resources.

4. Assessing the practical application value and contributions of the proposed methods to the field of information retrieval.

Information Basis

The information basis for this study includes the latest research findings, literature reviews, and the current state of research in the field of information retrieval both domestically and internationally. We will refer to and analyze existing information retrieval models and technologies, as well as their limitations and challenges in dealing with uncertain information resources.

Practical Significance of the Results

The practical significance of the research results lies in:

1. Enhancing the accuracy and efficiency of information retrieval systems to improve user experience.

2. Providing new technologies and methods for the field of information retrieval, especially in handling long-text and uncertain information resources.

3. Promoting the development of related technologies and disciplines, such as bioinformatics and medical informatics, in interdisciplinary fields.

4. Offering theoretical support and practical guidance for the design and development of information retrieval systems.

Through this study, we expect to provide new perspectives and solutions for the field of information resource management, especially under conditions of uncertainty, which is of significant theoretical and practical importance for Managing Web-based Information Resources Under Uncertainty. By improving the accuracy and efficiency of information retrieval, we can not only enhance the user experience in searching for information but also provide a scientific basis for the effective management and utilization of information resources, thereby achieving the maximization of information resource utilization in an environment of uncertainty.

CHAPTER 1 RELATED TECHNOLOGIES

Since the advent of the Transformer architecture, information retrieval has made significant strides over the past five years. Pre-trained models, such as BERT, have achieved remarkable success in the field of information retrieval. This chapter will describe the basic concepts and definitions of information retrieval and briefly review the success of pre-trained models in this domain.

1.1 Managing Web-based Information Resources Under Uncertainty And Long Text Retrieval Method

Managing Web-based Information Resources Under Uncertainty refers to the collection of strategies, techniques, and methodologies designed to effectively handle, organize, and retrieve information from web-based sources despite the presence of uncertainty. This uncertainty can arise from inaccurate data, imprecise or incomplete information, and unreliable results that may lead to irrational decisions. The objective is to develop systems capable of considering all uncertain information and managing web services and resources based on the best possible answer rather than relying solely on the quality of exact answers.

Long-Text Retrieval Methods

Long-text retrieval methods focus on processing and retrieving information from lengthy textual resources, especially within open-domain question-answering systems. These methods typically involve segmenting long texts into smaller, more manageable units, employing sophisticated algorithms to match query relevance

with the text, and extracting accurate and comprehensive answers from long-form content. For instance, the LongRAG (Long Retrieval--Augmented Generation) framework leverages large language models (LLMs) with long contextual capabilities to enhance retrieval-generation effectiveness. Key innovations of LongRAG include long retrieval units, long retrievers, and long answer generators that work in concert to achieve efficient answer extraction without relying on complex re-ranking mechanisms.

Relationship Between Long-Text Retrieval Methods and Managing Web-based Information Resources Under Uncertainty

Long-text retrieval methods are closely related to Managing Web-based Information Resources Under Uncertainty as they address the core issue of how to effectively manage and retrieve web-based information resources in an environment of uncertainty. Long-text retrieval techniques provide the technological means to handle long-form data, which is crucial for managing web-based information resources, especially when faced with incompleteness and inconsistencies in information.

In an environment of uncertainty, information resource management needs to be able to process and integrate heterogeneous, contradictory, or incomplete information from various sources. The development of long-text retrieval technologies, such as LongRAG, which utilizes long-context LLMs, can better understand and extract key information from long texts. This is significant for reducing uncertainty and improving decision quality. Moreover, advancements in long-text retrieval technologies also drive the development of theoretical frameworks for information resource management, especially when dealing with large-scale, complex web datasets.

In summary, long-text retrieval methods not only improve the efficiency and accuracy of information retrieval but also have important theoretical and practical significance for managing and retrieving web-based information resources under conditions of uncertainty. Through these methods, we can better understand and utilize web-based information resources, thereby making more reasonable decisions amidst uncertainty.

Core Objective of Information Retrieval

The core objective of an information retrieval system is to provide users with information relevant to their needs. This raises a fundamental question: how to assess the relevance between a query q and a document d. In the actual retrieval process, a user's search intent initiates the action of search, followed by the user issuing a query to the search engine to obtain results. This query q can be seen as representing the user's search intent. The search engine's task is then to return the most relevant results for the given query q and present these results in a ranked list to the user. Thus, the better a search engine can assess the relevance between query q and document d, the higher the user's satisfaction. To evaluate the relevance score between q and d, various models have been proposed, such as traditional statistical models, machine learning and deep learning models, and the pre-trained model-based retrieval models discussed in this paper. Their core goal is to compute the relevance between queries and documents.

Basic Framework of Information Retrieval

Here, we provide a more detailed definition of information retrieval: given an information need and a document collection $D = \{ d_i \}$, the objective of retrieval is to return a result list r, where the results are ranked from highest to lowest relevance score with respect to the query q and each candidate document d_i in the collection. Typically, the system returns the top k results, so this process

can also be considered a top-\(k \) ranking process. The quality of retrieval results is evaluated using metrics such as MAP@k, P@k, and NDCG@k. In theory, the higher the evaluation score, the better the retrieval results meet user needs.

Due to the large scale of document collections, information retrieval systems must consider efficiency as well as effectiveness. A multi-stage information retrieval architecture is commonly used in industry, as it is both effective and practical. In traditional retrieval architectures, different functional modules are responsible for efficiency and effectiveness, specifically in the recall and reranking stages. The recall stage usually employs statistical models with the objective of returning as many relevant documents as possible to prepare for the subsequent ranking stage, also known as initial ranking. The reranking stage aims to rank more relevant documents higher in the list. Since this stage prioritizes effectiveness over efficiency, pre-trained language models such as BERT are often used. A typical information retrieval framework is illustrated in Figure 1-1.

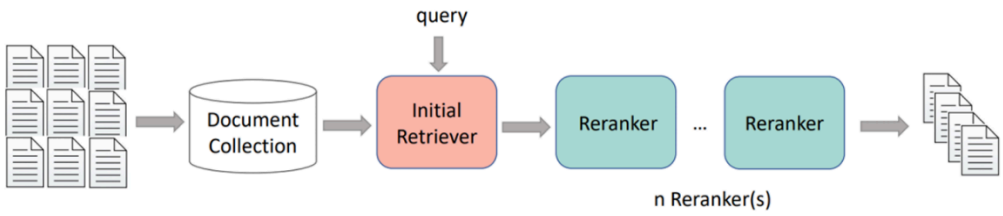


Figure 1.1 Modern Information Retrieval Architecture (Image from Fan et al. [21])

(1) Preparation

The initial preparation work for information retrieval includes query parsing, document parsing, and document indexing. First, user queries are processed to avoid issues such as typos or overly short or long queries. This process includes a

dedicated query parser that performs spell checks and query expansion. Given the massive scale of document collections in information retrieval, it is not feasible to dynamically process documents as with queries. Thus, document collections are usually pre-indexed to reduce computational load during subsequent query processing.

(2) Recall Stage

The recall stage quickly filters and retrieves relevant documents from a vast pool of candidate documents. Often called the first retrieval round, this stage typically uses precise term-frequency matching models such as BM25. Based on the relevance scores generated by the model, the initial results are sorted to form an initial result list. This stage is necessary because deep learning models and pre-trained models have high computational costs. If these models were used directly to perform relevance matching for every document in the candidate pool, the computational delay would not meet the requirements of real-time information retrieval systems. Therefore, the recall stage quickly matches potentially relevant documents to be forwarded to the next, finer ranking stage. Recall rate is the most crucial evaluation metric in this stage. Although the recall and reranking stages both aim to estimate relevance scores between query and document pairs, they typically employ different models. In the era of pre-trained models, methods have also emerged to apply them to the efficiency-focused recall stage. For example, ColBERT uses a BERT-based dual encoder to generate embeddings for queries and documents and then compares their relevance using cosine similarity. Compared to other baseline models, each query can be processed up to two orders of magnitude faster.

(3) Reranking Stage

After obtaining the initial ranking list, the number of documents in the reranking stage is usually significantly smaller than the original document collection, typically around 1,000. This number ensures that enough relevant documents are retrieved in the recall stage while allowing the reranking stage to compute and return results quickly. In this stage, complex deep learning or pre-trained models are typically employed for document reranking. This reranking process can be iterative to generate a final ranked list.

During this process, each reranking model receives the ranking list from the previous stage and provides a re-ranked list with the same or fewer results. Depending on the model's role in each stage, different models may be used. Generally, models with higher performance but lower efficiency are used in the later stages. Traditional reranking models include machine learning ranking models (e.g., RankNet and LambdaMart) and neural models (e.g., DRMM and Duet). Due to their fuzzy semantic matching capabilities, pre-trained models can capture finer-grained relevance signals, making them especially effective in the reranking stage. By learning contextual embeddings and modeling complex interactions between queries and documents, these pre-trained models have achieved significant success in reranking applications.

Based on the number of models used in the reranking stage, retrieval systems can be categorized into single-stage, two-stage, and multi-stage retrieval. Single-stage retrieval directly returns the initial ranked list from the recall stage to the user without further reranking. This type of retrieval is typically suitable for early retrieval frameworks, such as Boolean retrieval and exact matching scenarios. Two-stage retrieval further improves ranking accuracy by utilizing a reranking model, considering features that the recall stage did not address, such as multi-modal features, user behavior, and knowledge graph or semantic

information. This approach allows a more comprehensive consideration of various features, improving the quality of search results. A multi-stage retrieval system includes multiple reranking stages. Different reranking models can adopt various architectures and utilize different information sources to yield complementary results, enhancing overall performance. This flexible multi-stage structure better adapts to diverse search needs, providing more accurate and comprehensive search results.

1.2 Pre-trained Model: BERT

BERT is a milestone in the development of pre-trained models and is considered one of the most outstanding models in natural language processing. Its design and structure are based on the Transformer model, which enables it to adapt to various NLP tasks through unsupervised learning on large-scale text data.

Pre-trained models are a type of model trained on extensive amounts of text data through unsupervised learning, resulting in rich language representations. BERT's pre-training tasks include the Masked Language Model (MLM) task and the Next Sentence Prediction (NSP) task. In the MLM task, portions of the input text are randomly masked, and the model must predict the masked words. In the NSP task, the model must determine whether two sentences are consecutive in the original text. Through these tasks, BERT stores basic common-sense knowledge in its network, acquiring the ability to capture semantic information.

BERT's architecture is based on the Transformer model, a deep neural network framework that utilizes self-attention mechanisms. The Transformer captures contextual information within sentences, enabling the model to consider both preceding and succeeding words. This bidirectional modeling capability is

one of BERT's significant improvements over previous models, such as those based on recurrent neural networks.

Through unsupervised training, BERT learns rich representations at both the sentence and word levels, as the model must understand the context and perform language reasoning. This makes BERT an ideal choice for various NLP tasks. When applying BERT to downstream tasks, it can either be used as a feature extractor or fine-tuned by adding a task-specific output layer on top of BERT.

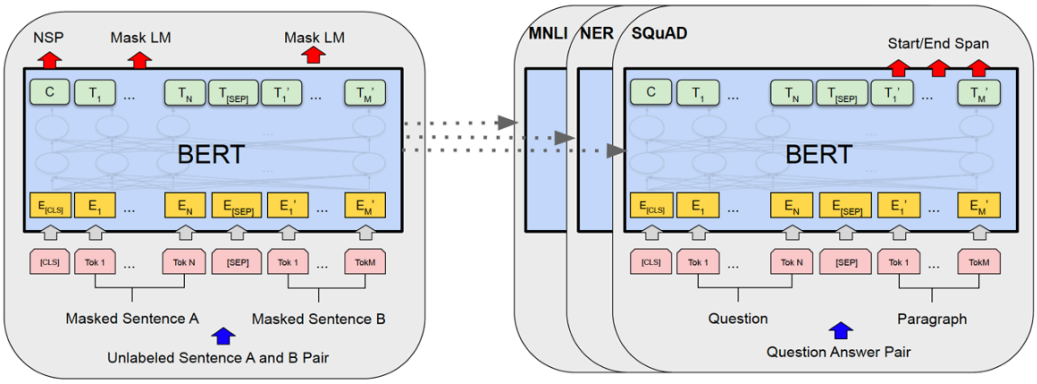


Figure 1-2: Pre-training and Fine-tuning Process of BERT Model (Image from Devlin et al. [2])

1.3 Text Ranking Methods Based on BERT Pre-trained Model

Representation-Based Retrieval Methods

Modern search engines use a multi-stage architecture to effectively deliver accurate results to users. Due to the high computational cost of Transformer-based pre-trained models, they are usually employed only in the final reranking stage, where a subset of documents is re-ranked. Deep learning ranking models can be classified into two types based on the feature assumptions for evaluating document relevance: representation-based retrieval architectures and interaction-based retrieval architectures. "Representation" refers to the model's representation of the

document or query, while "interaction" refers to the model's ability to capture relevance information by directly modeling interactions between documents and queries.

Representation-based retrieval architectures assume that encoding documents and queries yields rich representations that can directly estimate relevance. Interaction-based architectures, on the other hand, assume that interactions between documents and queries contain additional information about their relationship, necessitating more complex interactions between them. Pre-trained models can be integrated into both architectures, effectively replacing traditional deep learning models to enhance query and document representations or enable deeper query-document interactions, thereby improving retrieval performance. The structure of both types is shown in Figure 1-3.

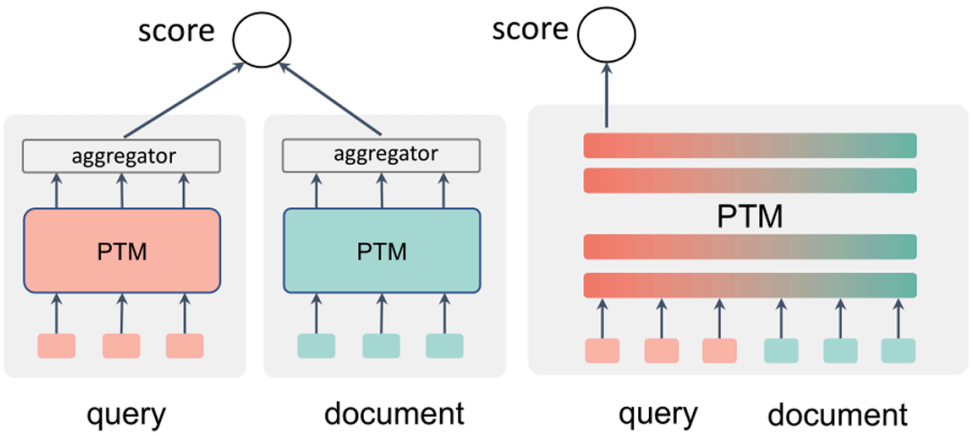


Figure 1-3: Representation-Based (Left) and Interaction-Based (Right) Retrieval Methods (Image from Fan et al. [21])

Representation-based models typically use a dual-encoder architecture that independently computes representations of queries and documents, followed by a similarity function (e.g., cosine or dot product) to measure their relevance. The

application of pre-trained models in representation-based methods can be expressed as:

$$relevance = f(PLM(Q), PLM(D))$$

where Q and D represent the query and document, respectively, and PLM is the pre-trained model, such as BERT. The function f calculates the relevance score based on their representations. The results are derived from PLM (Q/D) , the representations of the query or document through the pre-trained model.

For example, BERT's input text format is usually "\[CLS\] Q \[SEP\] D \[SEP\]," and it outputs a 768-dimensional vector. In fine-tuning, BERT is designed to use a special classification token, the \[CLS\] token, to represent the entire input sequence. The model can also apply mean or max pooling on the sequence's embeddings to obtain a single 768-dimensional vector. Various similarity functions can calculate relevance scores, such as cosine similarity, dot product, and feedforward networks. Qiao et al. used BERT to independently encode queries and documents, utilizing the final layer's \[CLS\] embedding for ranking scores based on cosine similarity. While representation-based retrieval systems generally lack the precision of interaction-based models, they offer faster computation and enable approximate nearest neighbor search from precomputed representations, making them suitable for the recall stage.

Interaction-Based Retrieval Methods

Interaction-based retrieval methods aim to capture term-level interactions between the query and document and use these semantic interactions to generate relevance scores. A typical interaction-based BERT model structure is shown in Figure 2-4.

Applying BERT and other pre-trained models to information retrieval tasks can be achieved directly by using an interaction-centric architecture for reranking. BERT's architecture, based on the attention mechanism, models complex interactions between queries and documents. To capture these complex semantic interactions, the interaction-based method does not independently compute representations for the query and document but instead inputs them together to obtain interaction representations. This results in high computational costs, so interaction-based methods are only applicable to the reranking stage, where relevance can only be calculated for a limited number of documents.

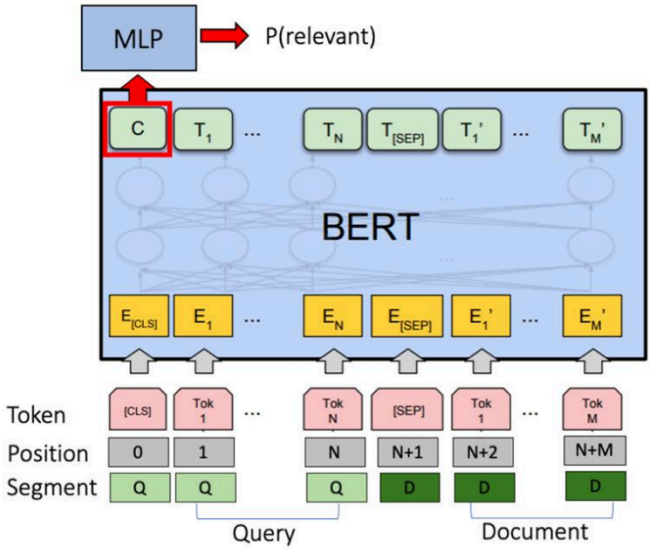


Figure 1-4: Interaction-Based BERT Model Structure (Image from Dai et al. [9])

In general, interaction-based methods using pre-trained models can be represented as:

$$relevance = f(PLM(Q, D))$$

where Q and D represent the query and document, \text{PLM} is the pre-trained model, and f is the evaluation function for relevance scores based on interactions. PLM(Q, D) represents the result obtained by jointly inputting the query and document into the pre-trained model.

Taking BERT as an example, in interaction-based methods, BERT's input is the concatenation of the query and document, separated by special tokens in the format "\[CLS\] Q \[SEP\] D \[SEP\]." The output vector for \[CLS\] can be regarded as the interaction between the query and document, which is then fed into a multi-layer feedforward neural network, generating the final relevance score.

Building upon machine learning-based retrieval methods, researchers have found that pointwise and pairwise loss functions can fine-tune pre-trained models. Due to the 512-token input limit of models like BERT, loss functions involving document lists are unsuitable. MonoBERT reformulates the retrieval ranking task as a paragraph relevance matching problem, applying interaction-based methods to obtain relevance scores for related paragraphs. This model uses a pointwise loss function, specifically cross-entropy loss, to fine-tune BERT. Even with BERT's simplest usage in the reranking stage, this approach yields highly effective results. CEDR concatenates query and document as input to BERT, then separates the resulting query and document vectors to compute a similarity matrix, which is input into existing interaction-centered deep learning ranking models, such as DRMM and KNRM, to obtain further results. DuoBERT applies a pairwise loss function by creating a sequence embedded with special tokens, combining a query and two text segments as input, in the format "[CLS] Q [SEP] D+ [SEP] D- [SEP]," where D+ and D- are positive and negative samples, respectively. During inference, similar to MonoBERT, relevance scores are derived from the \[CLS\] representation. By training the model through contrastive learning, DuoBERT

achieves improved results. However, due to BERT's length constraint, the query and concatenated text segments cannot exceed 512 tokens, which significantly impacts performance when applied to document ranking tasks.

1.4 Chapter Summary

This chapter introduces the relevant work related to this study. It first outlines the core objectives and basic framework of modern information retrieval, detailing the processes of the recall and reranking stages. Next, it provides an overview of the pre-trained language model BERT. Finally, it describes BERT-based ranking methods, covering both representation-based and interaction-based ranking methods and analyzing their limitations.

CHAPTER 2 PARAGRAPH-LEVEL LATENT SEMANTIC-BASED LONG-TEXT RANKING METHOD

2.1 Introduction

In recent years, pre-trained language models have achieved breakthrough results across various natural language processing tasks. The successful application of these large-scale language models may drive a new era of search engine development. Information retrieval frameworks that utilize pre-trained language models consist of two key stages: recall and reranking. This multi-stage retrieval approach is not only effective for search engines but also performs well in other downstream NLP tasks such as question answering and recommendation systems.

BERT's effectiveness is largely due to its complex Transformer-based architecture and its ability to compute deep contextual semantic interactions within input sequences. However, the computational complexity of BERT is quadratic in relation to input sequence length, making efficiency a concern with longer sequences. Moreover, BERT's input sequence length was fixed during pre-training, with 90% of training texts capped at 128 tokens and the remaining 10% at 512 tokens. During inference, the model must adhere to the input-output configuration set in training. Here, "tokens" represent the smallest units in the text, which can be a word, letter, or subword. Consequently, BERT's maximum sequence length limit is 512 tokens, far below the length of most candidate documents in real retrieval systems, which cannot directly be used to compute document relevance scores in BERT. If a document exceeds 512 tokens, the model only captures the semantic information at the beginning of the document, resulting in inaccurate relevance judgments.

To apply BERT in document retrieval tasks, the most popular and effective approach is to break documents into natural paragraphs or independent sentences, aggregating paragraph-level or sentence-level relevance scores to represent the overall relevance between the query and the document. This method's success suggests that BERT was originally designed for handling shorter texts. Some improved Transformer-based architectures can process thousands of tokens in a single inference, such as Longformer and BigBird. However, due to the demands of real-time retrieval, their response times limit their practical application.

Numerous studies have shown that paragraph-level relevance significantly impacts document-level relevance and that integrating fine-grained relevance signals can improve retrieval performance. In the early 2000s, researchers used the highest paragraph relevance score as the overall document score. Birch argued that the most relevant or top-\(k relevant paragraph or sentence scores could effectively replace document-level relevance, suggesting that the most relevant paragraphs or sentences in a document can, to some extent, represent the overall document relevance.

However, determining document relevance based on individual isolated text blocks can lead to semantic mismatches. Common paragraph-level relevance aggregation methods cannot capture important semantic information such as document structure. For example, using general aggregation methods, a document with disordered paragraphs may yield the same result as a naturally structured document. These methods may therefore perform poorly in certain real-world scenarios, as they assume that a few paragraphs can substitute for the entire document while neglecting the semantic information in the remaining paragraphs.

Further analysis reveals that a document comprises multiple paragraphs with different characteristics that contribute differently to relevance judgments. Wu et

al. created a dataset of human relevance judgments, studying how people assess relevance. They found that highly relevant documents often contain a higher proportion of relevant paragraphs. Kong et al. proposed the principle of overall relevance, suggesting that the proportion of relevant paragraphs within a document determines its relevance score. Li et al. conducted an eye-tracking study, concluding that introductory paragraphs attract more attention and significantly impact document-level relevance. In most candidate documents, the beginning or end usually contains the main ideas and the author's viewpoint, making these sections more relevant to the topic. These findings highlight the need to consider the contributions of different paragraphs to more accurately reflect overall document relevance.

To capture this document-level relevance information, this paper proposes a paragraph-level latent semantic-based retrieval method named KnBERT. Built on an interaction-based BERT retrieval method, KnBERT uses kernel functions to capture latent semantic relationships between paragraph context positions. To enhance retrieval performance, this study introduces a reward mechanism that integrates paragraph-level relevance semantic distribution information. Considering the real-time response requirements of information retrieval, this paper does not train an additional CNN or Transformer-based classifier based on kernel functions, as this would exponentially increase computational complexity. Instead, kernel function properties are embedded in the retrieval process's weight distribution. Specifically, a weighting method based on different kernel functions describes paragraph positions. Additionally, this paper conducts an in-depth analysis of paragraph-level relevance, positing that documents with paragraph distributions matching the query are more relevant and will rank higher in the final list. Using this approach, KnBERT effectively captures and utilizes document-level relevance information without compromising retrieval performance.

Experiments were conducted on the Robust04 and GOV2 datasets, using MAP@1000, MAP@100, P@20, and NDCG@20 metrics to evaluate accuracy. Results indicate that KnBERT significantly improves retrieval precision compared to interaction-based BERT retrieval baselines at the same computational complexity. Additionally, KnBERT achieves better retrieval precision than several recent BERT-based improved retrieval methods. This research provides a feasible approach and effective solution for addressing long-text retrieval issues in industrial applications, offering practical technical support for enhancing retrieval system performance.

2.2 Paragraph-Level Latent Semantic-Based Long-Text Ranking Method (KnBERT)

This section introduces the architecture of KnBERT, including the score aggregation strategy based on kernel functions and the reward mechanism algorithm for enhancing positive samples. The main steps of KnBERT are as follows: First, the long document is divided into a set of fixed-length paragraphs using a sliding window method. Next, the query is concatenated with each paragraph to create query-paragraph pairs, which are encoded by the pre-trained BERT model to generate interaction representations between the query and each paragraph. These representations are then converted to relevance scores for each paragraph via a trained fully connected layer, with each score multiplied by a paragraph weight derived from a kernel function. Finally, the paragraph scores are aggregated to produce the overall document score. If the computed paragraph scores match the analyzed relevant paragraph distribution, the document score is further boosted with a weight enhancement. Based on the relevance scores of the candidate documents, a ranked list is generated. By considering paragraph-level interactions and weight distribution, KnBERT can more accurately evaluate

document relevance. This paper designs four model variants based on different kernel functions. The following sections detail the model design and experimental results. The architecture of the model, shown in Figure 3-1, consists of four stages: input, encoding, aggregation, and enhancement.

Score Aggregation Strategy Based on Kernel Functions

The score aggregation strategy based on kernel functions covers the input, encoding, and aggregation stages of the overall model architecture. Each part is explained in detail below.

(1) Input Stage

As previously mentioned, applying BERT to real-time information retrieval primarily involves re-ranking the document list returned from the recall stage. Thus, the initial inputs for this stage include query q_i and a list of candidate documents D = \{d_1, d_2, ..., d_{1000}\} from the recall stage. The input stage involves segmenting long documents, concatenating the query with each paragraph, and obtaining encodings.

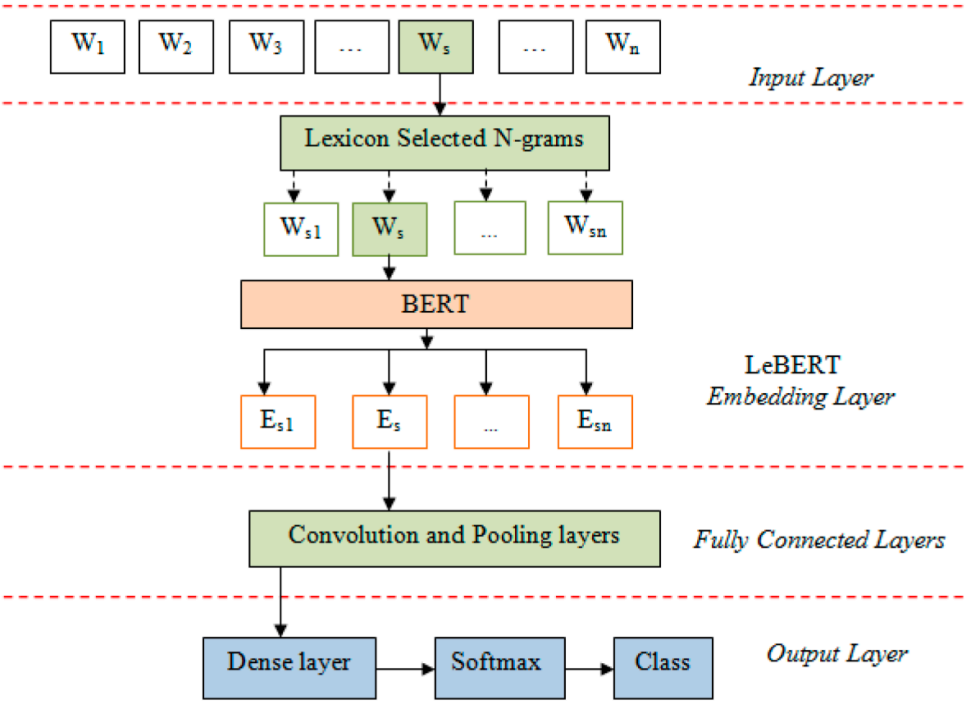


Figure 2-1: KnBERT Model Architecture

Since BERT's input length is limited to 512 tokens, candidate documents are divided into paragraph sets P_i with lengths not exceeding this limit. This paper uses a sliding window to segment documents. Each document is divided into paragraphs using a sliding window of 100 words, with an overlap of 50 words between adjacent paragraphs. The size of the sliding window affects the smoothness of the kernel function during score aggregation, so a shorter window length is chosen. Each candidate document varies in length, resulting in a different number of paragraphs. Assuming n paragraphs, this process can be formalized as:

$$P_i = \{p_1, p_2, \ldots, p_i\} = \{d_{i1}, d_{i2}, \ldots, d_{in}\}$$

The query is then concatenated with each paragraph to form the input format for BERT. This concatenation follows the format used in interaction-based retrieval

methods, represented as "\[CLS\] Q \[SEP\] D \[SEP\]." Here, \[CLS\] is the designated starting token from BERT's pre-training, and \[SEP\] is the separator token. Assuming the query has length N and the paragraph has length M , the total length of the embedded sequence is N + M + 3 , which remains below the 512-token limit.

Finally, the input text is tokenized by the pre-trained model's tokenizer and converted into vectors for subsequent encoding. Besides text embeddings, BERT's input includes positional embeddings, marking the position of each token, and segment embeddings, distinguishing between the query and paragraph. Positional embeddings increase sequentially from 0, with the last position embedding at N + M + 2 . Segment embeddings assign values of 1 for the query and 0 for the paragraph.

(2) Encoding Stage

The sum of the text embedding, positional embedding, and segment embedding is input into the BERT model, which produces semantic representations for each text encoding. The final layer's \[CLS\] token encodes the interaction between the query and paragraph. If there are n paragraphs, the encoding stage yields n \[CLS\] semantic vectors, represented as:

$$P_{cls} = \{p_1^{cls}, p_2^{cls}, ..., p_n^{cls}\}$$

(3) Aggregation Stage

The set P_{\text{cls}} , representing encoded interactions, is fed into the aggregation layer, where each interaction representation generates a relevance score between 0 and 1 through a fully connected layer. This can be expressed as:

$$S_{p_i} = W_d p_i^{cls}$$

where S_{p_i} is the relevance score for paragraph pi, and Wd is a weight tuned through the retrieval task.

Previous methods have aggregated paragraph-level relevance into document relevance scores using strategies such as max, sum, and average pooling. However, relying solely on local paragraph relevance may disregard partial semantic information in the rest of the document. Unlike common aggregation methods, this paper proposes a method that aggregates position-based local relevance into overall document relevance using kernel functions.

Kernel functions are widely used in information retrieval. For instance, Pan et al. used kernel functions to calculate term frequency or co-occurrence rates in documents. This paper presents a novel approach that uses various kernel functions to capture the semantic position information of paragraphs within each candidate document. This process provides both overall document relevance and deep contextual information. Gaussian, circular, and cosine kernels are commonly used kernel functions with shared properties such as continuity and symmetry. Inspired by Pan et al., this paper uses different kernel functions to describe position-based weights, as shown in Equation 2-3:

$$S_D = \alpha \times \log(S_{P\max}) + (1-\alpha) \times \log(\frac{\sum_{i=1}^{n} K(i,n) \cdot S_{p_i}}{\sum_{i=1}^{n} \frac{1}{i}})$$

where S_{p_i} is the relevance score between the query and the i -th paragraph; K(i, n) is the paragraph weight based on the kernel function; S_{max} is the highest paragraph score; and \alpha is a parameter controlling the relative weights of the components.

Position weights are modeled by capturing each paragraph's relative position within the document and applying three different kernel functions, assigning higher

weights to paragraphs near the beginning and end of the document. In this study, the weight is denoted as K(i, n) . To assess the effectiveness of these kernel functions, a simple decay function is also provided for comparison. The calculations are as follows:

1. Decay Function (KnBERT-D)

$$K(i,n) = \frac{1}{i}$$

2. Gaussian Kernel (KnBERT-Gs)

$$K(i,n) = 1 - \exp\left(\frac{-(i-u)^2}{2\sigma^2}\right)$$

3. Circular Kernel (KnBERT-C)

$$K(i,n) = 1 - \sqrt{1 - \left(\frac{(i-u)}{\sigma}\right)^2}$$

4. Cosine Kernel (KnBERT-Cos)

$$K(i,n) = 1 - \frac{1 + \cos\left(\frac{(i-u)\cdot\pi}{\sigma}\right)}{2}$$

where u controls phase alignment, ensuring equal weights for the first and last paragraphs, and sigma dynamically adjusts kernel scale.

Reward Mechanism Algorithm for Enhancing Positive Samples

After aggregating paragraph scores with kernel functions, a reward mechanism algorithm is used in the final enhancement stage to boost relevant information. Studies suggest that the number of relevant paragraphs affects document relevance, with highly relevant documents containing more relevant paragraphs. This study proposes a reward mechanism inspired by reinforcement learning to improve retrieval performance by integrating paragraph-level relevance distribution information. Specifically, two reward strategies are introduced: one rewards documents with continuous high-relevance subsequences, and the other rewards documents with a high proportion of highly relevant paragraphs.

Three hyperparameters are involved in these reward mechanisms: \phi , \gamma , and S_b . \phi is a threshold for determining paragraph relevance (e.g., \phi = 10 means the top 10% of paragraphs by BERT-calculated scores are considered relevant); \gamma is a threshold for the number of continuous relevant paragraphs needed to trigger a reward; and S_b is the score boost, set as a dynamic value. For example, a 10% boost moves a document's relevance into the top 10%. If reward conditions are unmet, S_b = 0 , leaving document weight unchanged. The reward score is calculated as shown in Equation 3-8:

$$S_B = S_D \times (1 + \frac{S_b}{S_D})$$

where S_D is the document weight from Equation 2-3.

Since the initial recall stage includes rich statistical and semantic information crucial for relevance judgment, the retrieval and reranking models should be complementary. Therefore, weighted scores from both are combined for

better results. Before interpolation, scores should be weighted, resulting in KnBERT's final formula:

$$S_{final} = (1-\beta) \times S_{ini} + \beta \times (S_B))$$

where S_{\text{ini}} is the initial document score (e.g., DPH+KL), and \beta is a parameter adjusting the relative weight of the components. S_B is the document weight calculated in Equation 2-8.

2.3 Experiment Design and Result Analysis

Experiment Design

(1) Datasets

This study uses the Robust04 and GOV2 text retrieval datasets for experimentation. Robust04 contains 528,155 articles, including news data from sources such as The Financial Times and The Los Angeles Times. The GOV2 dataset includes 25,205,179 documents, consisting of web pages collected from U.S. government websites. The Robust04 dataset contains 250 queries, while GOV2 includes 150 queries. Each query consists of a topic, description, and narrative, as shown in the example in Figure 3-2.

To simulate realistic retrieval scenarios, only topic keywords are used in the recall and reranking stages of the experiments. Both datasets contain numerous relevance judgments between queries and documents, which helps validate the reliability of the experimental results. Yang et al. analyzed these datasets, noting that over 100 papers have conducted extensive research on them, underscoring their broad and deep relevance in the information retrieval field.

Title	air traffic controller
Description	What are working conditions and pay for U.S. air traffic controllers?
Narrative	Relevant documents tell something about working conditions or pay for American controllers. Documents about foreign controllers or individuals are not relevant.

Figure 2-2: Query Example from the Datasets

(2) Evaluation Metrics

The evaluation metrics used in this study are divided into two parts: MAP@100 and MAP@1000 for evaluating overall ranking accuracy, and NDCG@20 and P@20 for evaluating accuracy in the top-ranked documents. These metrics serve different purposes. MAP reflects the overall accuracy improvement from the reranking model compared to the recall stage, while P@20 and NDCG@20 focus on the top 20 candidate documents, typically displayed on the first two pages of search results, which are the documents users are most likely to click on. Detailed descriptions of MAP, P@20, and NDCG@20 calculations are provided below.

Precision in information retrieval tasks is defined as the proportion of relevant documents in the ranked result list. Mean Average Precision (MAP) is the mean of the Average Precision (AP) across queries. Normalized Discounted Cumulative Gain (NDCG) is a graded relevance metric for evaluating retrieval quality, differing from precision in that documents are assigned varying levels of relevance, rather than simply being relevant or irrelevant. In NDCG, the higher a relevant document is ranked, the higher the score, making it especially useful for evaluating the quality of top-ranked results, which are most likely to be clicked by users.

(3) Experimental Setup

The experiments were conducted in two stages: a recall stage and a reranking stage. The DPH+KL model was used for the recall stage to retrieve the top 1000 candidate documents related to each query from all documents in both datasets. Then, the proposed KnBERT was used to rerank these lists.

The recall stage was indexed with the Anserini information retrieval framework [51] to quickly obtain recall results. The training and inference for the reranking stage model were carried out using the open-source TensorFlow framework, version 1.15. For training and inference in the reranking stage, only the top 1000 documents from the initial ranking were used, not all candidate documents. During inference, the maximum sequence length for the model was set to 384; any text exceeding this length was truncated, while shorter texts were padded with the \[MASK\] token. For the Robust04 dataset, document titles containing overall semantic information were added before each paragraph during inference.

A five-fold cross-validation setup was used, dividing the data into five equal parts, with three folds for training, one for validation, and one for inference. To ensure fairness, the Robust04 dataset was divided following the same setup as in other studies [9]. The final accuracy was averaged over five runs.

For model training, cross-entropy loss was used to fine-tune the model, similar to the method used by Nogueira et al. [8]. First, a round of fine-tuning was conducted on the MS MARCO paragraph ranking dataset to capture retrieval-related semantic information. Then, further fine-tuning was performed on the datasets used in this study (Robust04 and GOV2). This fine-tuning stage utilized only the most relevant paragraphs, concatenating them with the query to form query-document pairs as input. This fine-tuning strategy enables the model to

better adapt to the target domain's semantics and relevance. The cross-entropy loss function is defined in Equation 3-10:

$$L = -\sum_{i \in I_{pos}} \log(p_i) - \sum_{i \in I_{neg}} \log(1 - p_i)$$

where I_{\text{pos}} and I_{\text{neg}} represent sets of documents relevant and irrelevant to the query, respectively, and p_i is the probability of document relevance.

The experiments were conducted on a server equipped with two NVIDIA GeForce RTX 3090 GPUs (24GB VRAM each) and 128GB of RAM. The model was trained over two epochs with a batch size of 32, using the Adam optimizer with an initial learning rate of 1e-6.

(4) Baseline Models

KnBERT was compared with the following representative models, including both traditional and BERT-based methods. All BERT-based reranking models used the DPH+KL model for the recall stage, reranking the initial list and interpolating the results with the first-stage ranking scores to produce the final ranking. Document segmentation was standardized across models, with documents divided using a 100-token sliding window with a 50-token overlap.

- DPH+KL: Used as the initial ranking model in the recall stage to generate the top 1000 documents. DPH [52] is a traditional retrieval model based on randomness deviation. Using Kullback-Leibler divergence for Rocchio pseudo-relevance feedback, it generates the final document list.

- BM25+RM3: A strong baseline that incorporates pseudo-relevance feedback, with RM3 [53] used for query expansion. The default settings in the Anserini framework were used to obtain results.

- BERT-Base: The original BERT-Base model fine-tuned on MS MARCO, achieving impressive performance in document retrieval due to its strong transfer learning capabilities. Document relevance was determined by the highest paragraph score, the average of all paragraphs, or the score of the first paragraph.

- Co-BERT [18]: A context-aware BERT model capturing local ranking context among different candidate documents and query-specific information. Unlike other BERT-based models, Co-BERT is an end-to-end framework. Document embeddings and interaction representations of query-document pairs are stacked into a sequence, capturing semantic information specific to each query, with final results produced through list-based scoring.

- PARADE-Avg [13]: Another BERT-based model that aggregates paragraph representations to capture full-document relevance signals. The Avg variant used for comparison has similarities with the proposed method. PARADE-Avg assumes each paragraph contributes differently based on its position, with paragraph relevance normalized with document length to produce the final weight.

- BERT-QE [14]: A BERT-based reranking model using query expansion to improve retrieval accuracy. For a fair comparison, the same BERT model size was used for expansion, referred to as BERT-QE-BBB in this study.

Experiment Results and Analysis

This section presents the experimental results of KnBERT, comparing it with other state-of-the-art retrieval baselines and conducting an in-depth analysis. Experiments were conducted on the two classic TREC datasets, Robust04 and GOV2, using four evaluation metrics: MAP@1K, MAP@100, P@20, and NDCG@20. The baseline models used for comparison include two unsupervised ranking models (BM25+RM3 and DPH+KL), three interaction-based BERT

retrieval models with different score aggregation methods (BERT-MaxP, BERT-SumP, and BERT-FirstP), and three improved pre-trained retrieval models (PARADE-Avg, BERT-QE, CO-BERT). For more details on these models, see Section 3.3.1.

Since DPH+KL is used in the recall stage of this study, comparing KnBERT with this model clearly shows the improvement brought by the reranking stage. BERT-MaxP, which uses the score of the most relevant paragraph in each document as the document score, serves as a strong baseline for interaction-based BERT retrieval models. The improvement of KnBERT over these two models is emphasized in the experimental results tables, with improvements shown in parentheses. The Wilcoxon matched-pairs signed-rank test [50] was also used, with “” and “+” indicating statistically significant differences compared to DPH+KL and BERT-base models, respectively. The results are shown in Tables 2-1 and 2-2.

Model	**MAP@1K**	**MAP@100**	**P@20**	**NDCG@20**
DPH+KL	0.3046	0.2528	0.3924	0.4397
BERT-MaxP	0.3443	0.2919	0.4303	0.4876
BM25+RM3	0.2903	0.2451	0.3821	0.4407
BERT-SumP	0.3245	0.2730	0.4122	0.4679
BERT-FirstP	0.3228	0.2715	0.4092	0.4687

PARADE-Avg	0.3352	0.2857	0.4464	0.5124
BERT-QE	0.3571	0.3057	0.4532	0.5177
CO-BERT	0.3631	0.3103	0.4629	0.5213
KnBERT-D	0.3695	0.3186	0.4663	0.5318
KnBERT-Gs	0.3714	0.3203	0.4681	0.5358
KnBERT-C	0.3709	0.3206	0.4721	0.5412
KnBERT-Cos	0.3709	0.3206	0.4721	0.5412

Table 2-1: Experimental Results of KnBERT on the Robust04 Dataset

Model	**MAP@1K**	**MAP@100**	**P@20**	**NDCG@20**
DPH+KL	0.2182	0.5896	0.5122	0.5122
BERT-MaxP	0.2410	0.6178	0.5392	0.5392
BM25+RM3	0.2022	0.5634	0.4851	0.4851

BERT-SumP	0.2233	0.6017	0.5283	
BERT-FirstP	0.2243	0.6044	0.5310	
PARADE-Avg	0.1976	0.6225	0.5741	
BERT-QE	0.2312	0.5997	0.5206	
CO-BERT	0.2470	0.6668	0.5781	
KnBERT-D	0.2580	0.6651	0.6044	
KnBERT-Gs	0.2601	0.6691	0.5979	
KnBERT-C	0.2622	0.6705	0.5978	
KnBERT-Cos	0.2608	0.6762	0.5986	

Table 2-2: Experimental Results of KnBERT on the GOV2 Dataset

Among the kernel-based methods, KnBERT-C performs best on the MAP metric compared to KnBERT-Gs and KnBERT-Cos on both datasets. For the P@20 metric, KnBERT-Gs outperforms the other methods on the GOV2 dataset. On GOV2, kernel-based methods consistently outperform the simple decay function

on NDCG@20, although a monotonic decay function occasionally yields better results on NDCG@20. All proposed models consistently outperform unsupervised baselines and recent BERT-based reranking models, confirming the effectiveness of the kernel function-based paragraph score aggregation method.

Compared to the unsupervised DPH+KL model, the proposed methods show significant improvement in all metrics across both datasets. BM25+RM3, a widely used unsupervised ranking model, highlights the superiority of multi-stage retrieval frameworks over single retrieval models. These results, even considering computational costs, demonstrate the potential and stability of reranking models.

KnBERT also easily surpasses the BERT-Base model pre-trained on MS MARCO. With kernel-based methods, KnBERT performs significantly better than BERT-MaxP on Robust04 and GOV2 datasets. Specifically, on the NDCG@20 metric, KnBERT improves by 11.0% on Robust04 and 12.1% on GOV2 compared to BERT-MaxP. In terms of MAP@1000, KnBERT outperforms BERT-Base by 7.9% on Robust04 and 4.8% on GOV2, validating the effectiveness of the kernel function-based architecture over a single BERT ranking model.

Compared to recent BERT-based reranking models that use full-document relevance signals or pseudo-relevance feedback, KnBERT maintains its advantage, with all four variants outperforming these improved models. Although PARADE

-Avg shares the same core concept as KnBERT-D, KnBERT’s framework adds a normalization term to the paragraph scores and combines position-based scores with MaxP scores, revealing that simple score aggregation can sometimes be more effective than representation aggregation strategies. Comparisons with the BERT-QE pseudo-relevance feedback ranking model further confirm KnBERT’s framework's effectiveness. The best KnBERT variant improves MAP on the Robust04 dataset by 4.0%, with lower computational costs than BERT-QE.

The reward mechanism proves effective in improving framework performance. While the accuracy improvement is moderate, it demonstrates the mechanism's effectiveness on both TREC datasets. Table 2-3 shows improvements with this mechanism applied to one KnBERT variant, confirming its value.

Model	Metric	MAP@1K	MAP@100	P@20	NDCG@20
BERT-MaxP		0.3875	0.2410	0.6178	0.5392
KnBERT-D		**0.4037** (+4.2%)	**0.2580** (+7.1%)	**0.6651** (+7.7%)	**0.6044** (+12.1%)
+ Consecutive Relevant Paragraphs		0.4035 (+4.1%)	0.2590 (+7.5%)	0.6581 (+6.5%)	0.5915 (+9.8%)
+ High Proportion of Relevant Paragraphs		0.4068 (+5.0%)	0.2608 (+8.2%)	0.6671 (+8.0%)	0.5917 (+9.7%)
+ Both Mechanisms		0.4050 (+4.5%)	0.2603 (+8.1%)	0.6674 (+8.0%)	0.5996 (+11.2%)

Table 2-3: Experimental Results of Reward Mechanism on the GOV2 Dataset

Previous research on score aggregation and representation aggregation has confirmed the effectiveness of aggregation-based retrieval methods, but these methods have yet to be fully compared. In this study, KnBERT is compared with these complex methods in terms of effectiveness and time complexity. Methods compared include score aggregation using max pooling, sum pooling, first pooling, and k-max pooling of relevant sentences, as well as representation aggregation methods using max, sum, and mean pooling. For score aggregation, time complexity remains constant, with only paragraph weight positions differing based on the pooling method. Representation aggregation, however, requires an additional neural network to aggregate representations. As shown in Table 3-4,

kernel-based score aggregation outperforms other score aggregation methods given the same time complexity.

Model	**Robust04**				**GOV2**	
	MAP	P@20	NDCG@20	MAP	P@20	NDCG@20
BERT-MaxP (score max pooling)	0.3443	0.4303	0.4876	0.3875	0.6178	0.5392
BERT-SumP (score sum pooling)	0.3245	0.4122	0.4679	0.3671	0.6017	0.5283
BERT-FirstP (score first pooling)	0.3228	0.4092	0.4687	0.3672	0.6044	0.5310
Birch (score k-max pooling)	0.3691	0.4669	0.5325	0.3985	0.6554	0.5894
PARADE-Max	0.3711	0.4723	0.5442	0.3352	0.6228	0.5636

PARADE-Sum	0.3526	0.4711	0.5385	0.3268	0.6218	0.5747
PARADE-Avg	0.3352	0.4464	0.5124	0.3174	0.6225	0.5741
PARADE-Transformer	0.3803	0.4920	0.5659	0.3628	0.6651	0.6093
KnBERT (kernel pooling)	0.3714	0.4725	0.5413	0.4045	0.6762	0.5986

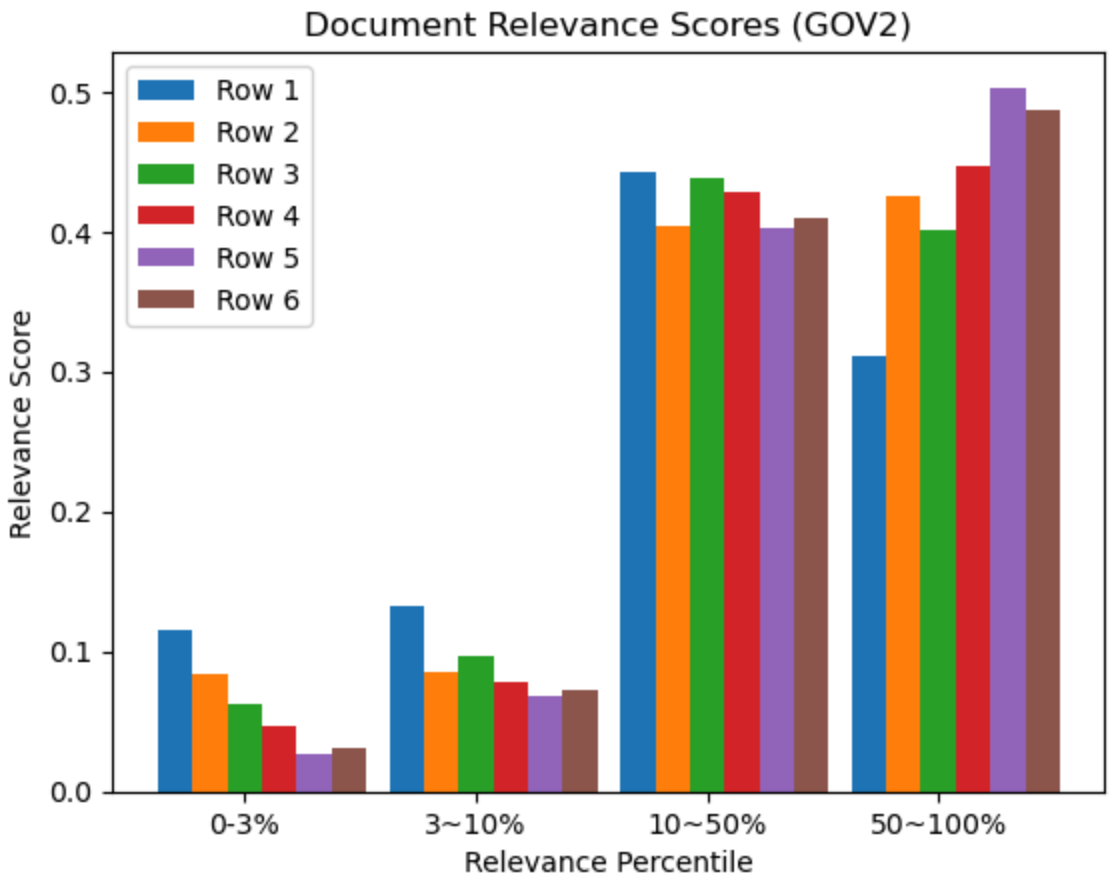
Table 2-4: Experimental Results of Different Pooling Methods on the Robust04 Dataset

While kernel-based score aggregation is effective on both datasets, unpredictable situations may arise in real-world applications. It is hypothesized that query specificity may reduce the effectiveness of complex score aggregation in KnBERT, where highly relevant paragraphs per document are fewer than with typical queries. Wu et al. [42] studied distribution across datasets, concluding that query types impact the number of highly relevant paragraphs per document. Score aggregation strategies perform better on datasets with complex, fine-grained relevance distributions. When a dataset has only 1–2 relevant paragraphs per document, relevance judgments are simplified as a single highly relevant paragraph can represent document relevance. MS MARCO is an example of such a dataset.

Analyzing the GOV2 dataset, this hypothesis was validated through paragraph-level relevance judgments and distributional analysis. Previous studies [54-55] provide paragraph-level relevance judgments for GOV2, enabling the inclusion of both paragraph and document relevance distributions. In the GOV2 dataset, 38% of documents have only one relevant paragraph, while 40% have three or more. More relevant paragraphs per document correlate positively with the effectiveness of complex aggregation.

Further analysis of the results is shown in Figure 3-3, which illustrates the joint distribution of paragraph-level relevance and overall document relevance in the proposed framework. Based on TREC relevance judgments, document relevance has three levels: non-relevant (0), relevant (1), and highly relevant (2). As document-level relevance increases, the proportion of highly relevant paragraphs in the document also rises sharply, while the proportion of non-relevant paragraphs remains steady with minor differences.

Even when documents have fewer relevant paragraphs, simple score aggregation performs well, but complex score aggregation, as proposed here, retains advantages. By combining optimal paragraph and position-based semantic information through kernel functions during inference, the proposed framework balances robustness and effectiveness.


Document Relevance Scores (GOV2)
Row 1
Row 2
Row 3
Row 4
Row 5
Row 6
Relevance Score
0.0
0.1
0.2
0.3
0.4
0.5
0-3%
3~10%
10~50%
50~100%
Relevance Percentile

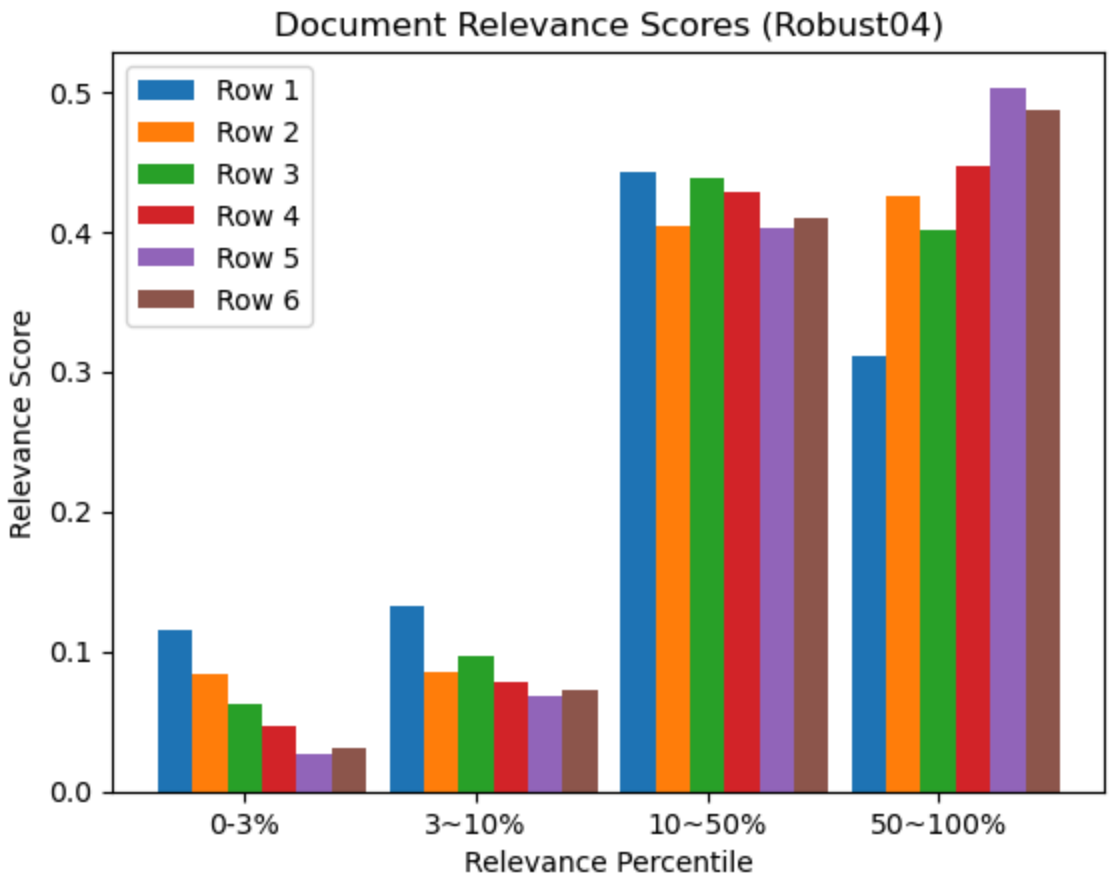


Figure 2-3: Distribution of Relevant Paragraphs in the Dataset

Although BERT-based rerankers provide ranked lists enriched with deep contextual information, supplementing semantic information missed by traditional frequency-based models, the computational cost is challenging in real applications. Retrieval systems must rank and return documents within seconds after a user query, making efficiency crucial. To improve KnBERT's efficiency, smaller BERT variants were considered.

To guide real-world retrieval applications, this study examined the reranking performance of KnBERT with various BERT model sizes. The experiment varied hyperparameters, including the number of hidden layers and attention heads, while

keeping other framework settings constant. Table 3-5 shows the performance of different BERT sizes on Robust04, along with parameter counts and inference times per document, helping to balance effectiveness and efficiency in various applications.

Model Size	Layers/Heads	MAP@1K	MAP@100	P@20	NDCG@20	Params	Inference Latency (ms/doc)
Base	12/768	0.3714	0.3203	0.4681	0.5358	123M	78.2
Small	4/512	0.3458	0.2946	0.4430	0.5090	35M	18.1
Mini	4/256	0.3363	0.2852	0.4309	0.4928	13M	8.41
Tiny	2/128	0.3256	0.2738	0.4189	0.4761	5M	2.85

Table 2-5: Experimental Results of Different Model Sizes on the Robust04 Dataset

As detailed in Section 3.2, a simple function was applied to normalize kernel-based paragraph position weights based on paragraph count to balance document length. Normalization effects varied across datasets. With identical hyperparameters across both datasets, Table 3-6 shows normalization effects. For Robust04, normalization positively impacted all metrics, while GOV2 saw slight decreases in MAP@1000, P@20, and NDCG@20. Average document length may explain the differences, as GOV2's average length is about three times that of Robust04. In GOV2, longer documents tend to have higher relevance but lower scores due to length, affecting relevance judgment accuracy. Document length should be considered in relevance assessment.

Model	**Robust04**			**GOV2**		

	MAP	P@20	NDCG@20	MAP	P@20	NDCG@20
BERT-Base	0.3443	0.4304	0.4876	0.3875	0.6178	0.5392
KnBERT-D (with normalization)	0.3697	0.4701	0.5351	0.4063	0.6644	0.5938
KnBERT-D (without normalization)	0.3706	0.4725	0.5363	0.4032	0.6540	0.5883

Table 2-6: Experimental Results on Robust04 and GOV2 Datasets with Different Normalization Settings for KnBERT-D

The hyperparameters in the kernel functions and reward mechanism have a substantial impact on final results. This study thoroughly investigates these parameters to examine their influence on the retrieval framework. The effect of parameters within the kernel functions is illustrated in Figure 3-4, while the impacts of reward mechanism hyperparameters are shown in Table 3-7.

The sensitive parameter\sigma potentially affects the robustness of the KnBERT method, and it is determined byu, which is controlled by the number of

paragraphs in a document and is set asi + (1)/2, wherei represents the number of paragraphs.

The figure reveals that the values of\sigma andu significantly influence model performance. When using a circular kernel function with\sigma set to 0.5\(u, KnBERT achieves the best results in MAP@1000. For P@20 and NDCG20, a kernel function value of 0.7\(u performs better on both datasets. Therefore, different kernel scales require different optimal parameter values. The dynamic hyperparameter setting method in this study initially resolves sensitivity issues.

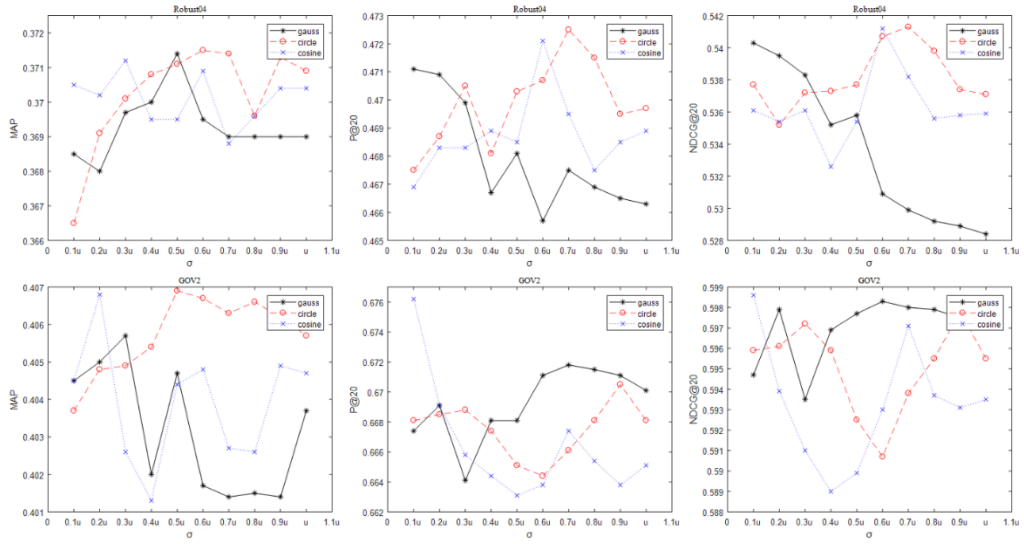


Figure 2-4: Impact of Hyperparameters in Kernel Functions

	φ	γ	S_{bonus}	指标 MAP@1K	MAP@100	P@20	NDCG@20
BERT-Base	-	-	-	0.3875	0.2410	0.6178	0.5392
Impacts of φ	5	2	30	0.4050	0.2603	0.6674	0.5996
	10	2	30	0.4049	0.2594	0.6664	0.5988
	15	2	30	0.4049	0.2594	0.6678	0.5969
	20	2	30	0.4046	0.2595	0.6664	0.5979
Impacts of γ	5	2	30	0.4050	0.2603	0.6674	0.5996
	5	3	30	0.4045	0.2589	0.6685	0.6001
	5	4	30	0.4047	0.2590	0.6668	0.6001
Impacts of S_b	5	2	3	0.4058	0.2612	0.6624	0.5865
	5	2	10	0.4050	0.2610	0.6641	0.5956
	5	2	30	0.4050	0.2603	0.6674	0.5996

Table 2-7: Experimental Results on the Reward Mechanism with Different Hyperparameters

2.4 Summary of This Chapter

Existing BERT-based reranking methods focus on ranking lists based on the most relevant paragraphs, often disregarding other parts of the document, potentially losing some context information. The primary difference between the kernel-based reranking framework proposed here and other models is the use of kernel functions to capture document-level relevance through paragraph-level relevance aggregation. The proposed framework, named KnBERT, naturally incorporates paragraph-level kernel functions into a BERT-based reranking approach, offering a promising direction for developing a generalized retrieval-reranking framework.

Through experiments on two TREC datasets, the results show that the proposed framework is effective, outperforming BERT-based reranking models and baseline models across MAP@1000, MAP@100, P@20, and NDCG@20 metrics. Further discussions and analyses covered the applications of score aggregation and hyperparameter settings.

CHAPTER 3. ABSTRACT SUMMARY-BASED SEMANTIC ENHANCEMENT FOR LONG DOCUMENT RANKING

3.1 Introduction

Recent research efforts have focused on applying BERT models [2] to document ranking tasks, primarily adhering to the Probabilistic Ranking Principle (PRP) and independently considering query-document pairs [56-58]. The main goal of ranking tasks is to improve the accuracy of the ranked list based on the given query. BERT's architecture, characterized by multiple Transformer encoders with multi-head attention and feedforward layers, enables it to capture semantic information within documents, unlike traditional statistical ranking methods. BERT's input sequence format is crucial for its effectiveness, and the complexity of input text semantics greatly affects retrieval performance. Many successful studies have enhanced retrieval systems by extracting contextual features from documents and integrating them into relevance calculations, thus incorporating local context.

Sequence-to-sequence generation models have demonstrated excellent performance in question answering and text generation tasks. Abstract text summarization [59-60], which involves generating concise summaries to capture the main ideas of the source text, is particularly effective in distilling document semantics. Using these model-generated summaries as BERT input can enhance the model's capacity to capture semantic information.

Unlike traditional statistical ranking methods, BERT can capture semantic relationships between texts. Its fuzzy matching ability means that input sequence content, format, and semantic complexity significantly impact BERT's retrieval performance. To capture the cross-paragraph interactions within documents, this study proposes a summary-enhanced BERT-based ranking method called SEBERT. By utilizing abstract summaries to enrich input text semantics, SEBERT improves retrieval accuracy. An example of text summary utilization is shown in Figure 4-1. To ensure the inclusion of rich semantic information within BERT's input constraints, a generative summarization model is used to summarize each candidate paragraph's context, which is then concatenated into the input sequence.

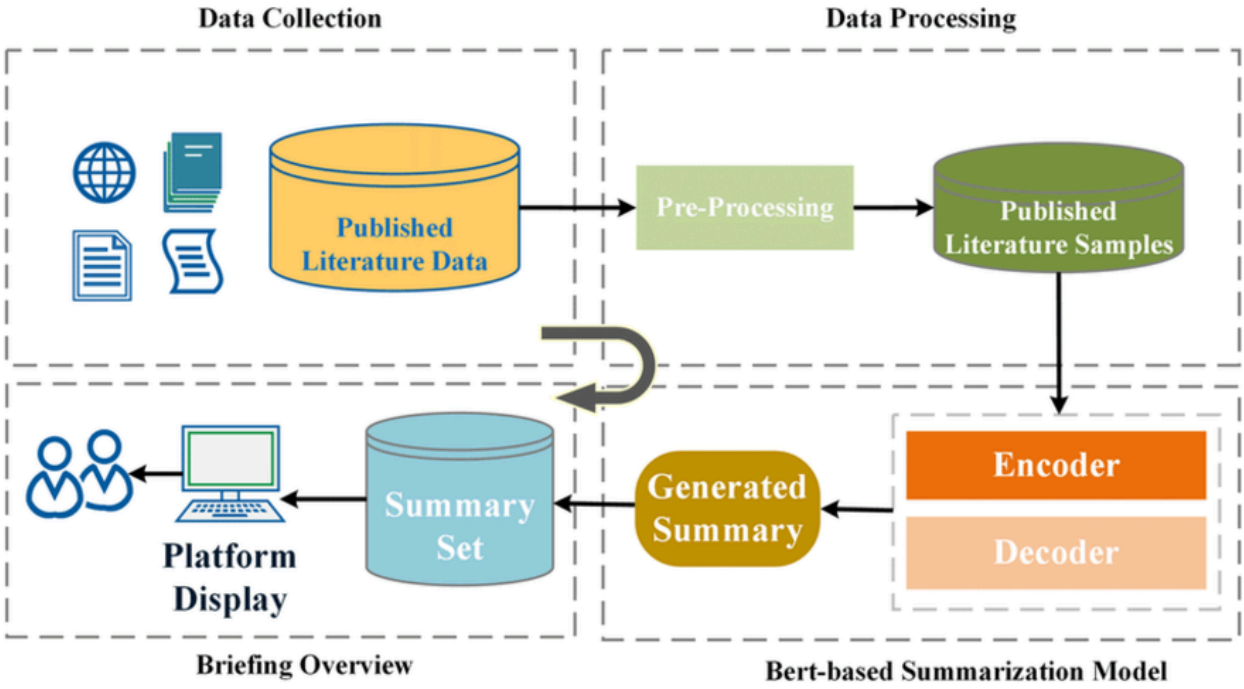


Figure 3-1: An example of using text summarization in a BERT based retrieval framework

Experiments were conducted on the Robust04 and GOV2 datasets, with performance evaluated using MAP@1000, MAP@100, P@20, and NDCG@20 metrics. Results showed that adding summaries as extra semantic input

significantly enhanced retrieval accuracy, with SEBERT outperforming several recent BERT-based enhanced retrieval methods. This study offers a feasible approach to improve long-document retrieval in industry applications, providing practical support for enhancing retrieval system performance.

3.2 Abstract Summary-Based Semantic Enhancement for Long Document Ranking (SEBERT)

This study employs two methods for integrating generative models into the retrieval framework, illustrated in Figures 3-2 and 3-3, named SEBERT and SEBERT-LA, respectively.

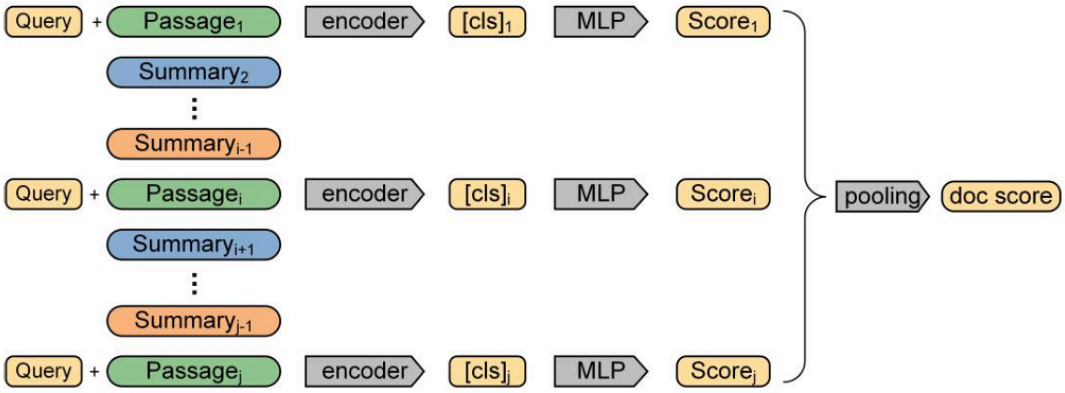


Figure 3-2: Retrieval Framework of SEBERT

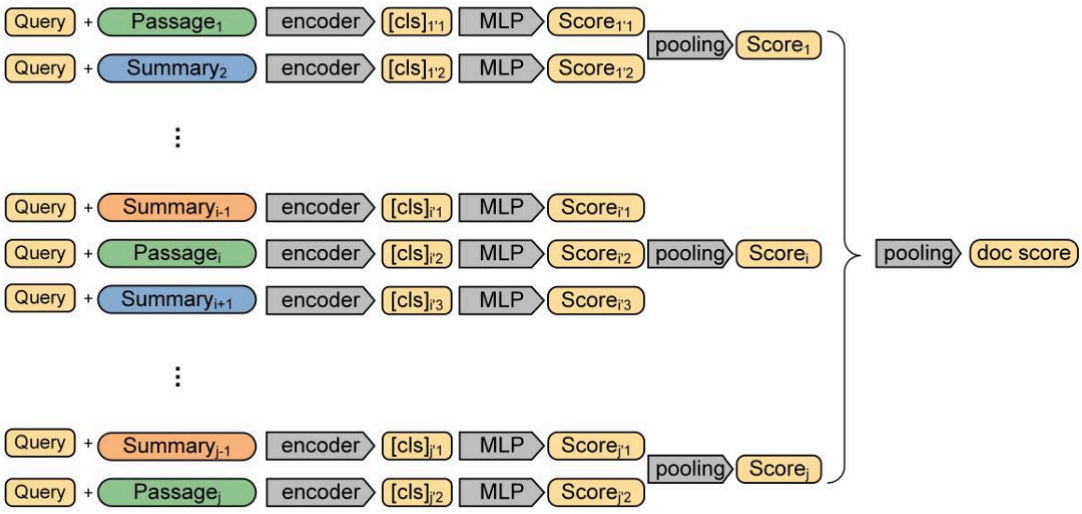


Figure 3-3: Retrieval Framework of SEBERT-LA

SEBERT's approach to determining query-paragraph relevance is similar to the interaction-based BERT retrieval method, and it involves four stages: summary generation, input, encoding, and aggregation.

(1) Summary Generation Stage

To generate summaries for candidate paragraphs, a sliding window divides the document with a stride of half the window length. These sentences are then re-used to generate abstract summaries that encapsulate semantic information. Each sentence generates a corresponding summary in the offline phase, so re-calculation is unnecessary during user queries. The BERT model uses these summaries to provide semantic context for relevance determination in the next step. The window length affects the richness of semantic content, so it is analyzed during the testing phase. Each candidate document's length and paragraph count vary, which can be formalized as follows:

$$P_i = \{p_1, p_2, ..., p_i\} = \{d_{i1}, d_{i2}, ..., d_{in}\}$$

This study uses PEGASUS [59] as the default summarization model in the framework. PEGASUS has a standard Transformer encoder-decoder architecture with three sections: one section uses `[MASK1]` as a template for text generation, while the other two retain input content with some words randomly masked as `[MASK2]`. PEGASUS, known for its strong summarization capabilities, divides the documents in the dataset, generating paragraph summaries to strengthen BERT's performance. Summary generation is represented by:

$$Z_i = SUM(p_i)$$

where Z_i represents the summary of the i -th paragraph, SUM denotes the summarization model, and p_i is the paragraph text.

(2) Input Stage

The query and divided paragraphs are concatenated to form the BERT input, structured as:

$$[CLS] + Q + [SEP] + Z_{i-1} + p_i + Z_{i+1} + [SEP]$$

Here, `[CLS]` denotes the starting token, `[SEP]` is a separator, and Z_{i-1} , p_i , and Z_{i+1} are the concatenated text sequence. If the input text exceeds the allowed length due to summary content, it is truncated accordingly.

A tokenizer from the pre-trained model converts the input text into tokens, which are then encoded as vectors for subsequent encoding. In addition to text embeddings, BERT's input includes positional and segment embeddings, with the former marking the token's position in the sequence and the latter distinguishing between query and paragraph segments.

(3) Encoding Stage

The sum of text, positional, and segment embeddings is used as input to the BERT layers, which encode each token's semantic features. The final layer's `[CLS]` token encodes query-paragraph interaction. For n paragraphs, this stage generates n `[CLS]` token embeddings, represented as P_{cls} = \{ p_{cls1}, p_{cls2}, \ldots, p_{clsn} \} .

(4) Aggregation Stage

The encoding layer output P_{cls} serves as input to the aggregation layer, where a fully connected layer generates relevance scores within the range (0, 1). This process is described as follows:

$$S_{p_i} = W_d p_i^{cls}$$

where S_{p_i} is the relevance score of paragraph p_i , and W_d is the weight fine-tuned for the retrieval task.

In summary, this study defines the SEBERT process as follows:

$$S_{(q,p_i)} = BERT(q, Z_{i-1} + p_i + Z_{i+1})$$

where S_{q,p_i} represents the relevance score (0, 1) calculated by BERT, and Z_{i-1}, p_i, Z_{i+1} are the input sequence containing the target paragraph and its contextual summaries.

The final document score is obtained through max pooling:

$$S_{final} = \max(BERT(q, p_1 + Z_2), \ldots, BERT(q, Z_{i-1} + p_i))$$

For SEBERT-LA, neighboring paragraph summaries are encoded and scored by BERT first. The final paragraph score incorporates the scores of both the target and adjacent paragraphs. Document scores are pooled similarly to SEBERT using max pooling, while paragraph scores use average pooling:

$$S_{(q,p_i)} = avg(BERT(q, Z_{i-1}), BERT(q, p_i), BERT(q, Z_{i+1}))$$

$$S_{final} = \max(S_{(q,p_1)}, S_{(q,p_2)} \ldots, S_{(q,p_i)})$$

This study pre-trains BERT on the MS MARCO paragraph-ranking dataset and then fine-tunes it on the target dataset (e.g., GOV2). The cross-entropy loss function is used for model fine-tuning:

$$L = -\sum_{i \in I_{pos}} \log(p_i) - \sum_{i \in I_{neg}} \log(1 - p_i)$$

where I_{pos} and I_{neg} are the sets of relevant and non-relevant documents, respectively, and p_i is the probability of relevance for document i .

3.3 Experimental Design and Results Analysis

Experimental Design

Experiments were conducted using the standard Robust04 and GOV2 text retrieval collections. Robust04 is a news collection consisting of 528,155 documents, used in the TREC 2004 Robust track. Only short keyword queries (titles) were considered for this work, with 250 topics in Robust04 and 150 in GOV2. All rankings produced by BERT-based models, including the proposed model and baselines, were interpolated with the initial ranking scores in the same way. P@20 and NDCG@20 metrics were reported to evaluate the top-ranked

results, and MAP@100 and MAP@1000 metrics were used to assess the entire ranking list.

The experiments followed a standard five-fold cross-validation setup, with queries split into five equal parts. In each fold, hyperparameters were tuned on the validation set, and the best NDCG@20 configuration on the validation set was used to report test set performance. Final performance is the average of all results.

The BERT-Base model was used, consisting of a multi-layer bidirectional Transformer encoder with 12 hidden layers, a hidden size of 768, 12 attention heads, and approximately 110 million parameters. The maximum sequence length was set to 384, and sequences shorter than this were padded accordingly. Only the most relevant paragraph rather than all paragraphs was used for training, with summaries of the preceding and succeeding paragraphs appended to each training paragraph. This approach reduced training time, enhanced paragraph logicality, and improved contextual semantic relevance without sacrificing model performance. Additionally, the batch size (number of paragraph-query pairs per batch) was set to 32, and the model was trained for 2 epochs using the Adam optimizer with an initial learning rate of 1e-6.

Experimental Results and Analysis

This section presents the results of the BERT ranking framework that integrates paragraph context on several datasets, comparing them with unsupervised and supervised baseline models. Tables 4-1 and 4-2 display the results of BERT and comparison models on the Robust04 and GOV2 datasets for MAP@1000, MAP@100, P@20, and NDCG@20 metrics. The asterisk () indicates that the results are statistically significantly different from BERT-Base (MS MARCO) based on the Wilcoxon signed-rank test ($P < 0.05$).

From Tables 3-1 and 3-2, it is evident that the BERT ranking framework with paragraph context integration improves all four metrics compared to both unsupervised and pretrained language model baselines.

Model	**MAP@1K**	**MAP@100**	**P@20**	**NDCG@20**
DPH+KL	0.3046	0.2528	0.3924	0.4397
BERT-MaxP	0.3443	0.2919	0.4303	0.4876
BM25+RM3	0.2903	0.2451	0.3821	0.4407
BERT-SumP	0.3245	0.2730	0.4122	0.4679
BERT-FirstP	0.3228	0.2715	0.4092	0.4687
BERT-QE	0.3571	0.3057	0.4532	0.5177
CO-BERT	0.3631	0.3103	0.4629	0.5213
SEBERT	0.3706	0.3195	0.4683	0.5315
SEBERT-LA	0.3690	0.3182	0.4671	0.5299

Table 3-1: SEBERT Results on the Robust04 Dataset

Model	MAP@1K	MAP@100	P@20	NDCG@20
DPH+KL	0.3605	0.2182	0.5896	0.5122
BERT-MaxP	0.3875	0.2410	0.6178	0.5392
BM25+RM3	0.3350	0.2022	0.5634	0.4851
BERT-SumP	0.3671	0.2233	0.6017	0.5283
BERT-FirstP	0.3672	0.2243	0.6044	0.5310
BERT-QE	0.3574	0.2312	0.5997	0.5206
CO-BERT	0.4022	0.2470	0.6668	0.5781
SEBERT	0.4000	0.2560	0.6728	0.5961
SEBERT-LA	0.3991	0.2547	0.6624	0.5895

Table 3-2: SEBERT Results on the GOV2 Dataset

BERT-MaxP is an interaction-based BERT retrieval method using the score of the most relevant paragraph as the document score. SEBERT demonstrates significant improvements over BERT-MaxP in the Robust04 dataset, with P@20 and MAP@1000 increasing by 19.7% and 21.3%, respectively. In the GOV2 dataset, where document length is about three times that of the Robust04 dataset, the SEBERT framework performs considerably better than both unsupervised models and BERT-MaxP, indicating enhanced document retrieval capabilities.

To further validate the proposed methods, SEBERT was compared with other relevance feedback and BERT-based retrieval models. BERT-QE expands queries using the most relevant segments and document relevance, while CO-BERT scores multiple documents concurrently. For fairness, the same BERT-Base model size was used, with settings from the original papers. As shown in Tables 4-1 and 4-2, the BERT ranking framework with paragraph context integration achieves considerable accuracy improvements over these enhanced models.

This study also investigated the impact of the number of context paragraphs, represented by the hyperparameter Z_N , through ablation experiments where Z_N was set to 1 and 2 in both SEBERT and SEBERT-LA models. Initial experiments indicate that increasing the number of paragraphs does not necessarily improve retrieval accuracy, particularly in P@20 and NDCG@20, suggesting that excessive semantic features can obscure salient information.

The length of each candidate paragraph is also crucial. Paragraphs were analyzed at lengths of 100, 256, and 384, with a sliding window stride of half the paragraph length. When paragraph length was set to 384, input sequences often exceeded the 512-token limit, as each sequence included a query, candidate paragraph, and two paragraph summaries. In such cases, the sequence was

truncated to 512 tokens. Table 4-3 shows that both SEBERT and SEBERT-LA perform better with longer input sequences, benefiting from richer semantic representations.

Model	**Paragraph Length**	**N_Z**	**Robust04**				**GOV2**			
			MAP	P@20	NDCG@20	NDCG@10	MAP	P@20	NDCG@20	NDCG@10
SEBERT	100	1	0.3695	0.4697	0.5306	0.5464	0.4000	0.6728	0.5961	0.6044
SEBERT	100	2	0.3667	0.4663	0.5254	0.5413	0.4007	0.6641	0.5910	0.5991
SEBERT-LA	100	1	0.3690	0.4671	0.5299	0.5399	0.3991	0.6624	0.5895	0.5958
SEBERT-LA	100	2	0.3674	0.4661	0.5231	0.5367	0.4012	0.6578	0.5867	0.5947
SEBERT	256	1	0.3702	0.4693	0.5329	0.5501	0.4014	0.6718	0.5980	0.6064
SEBERT	384	1	0.3706	0.4683	0.5315	0.5479	0.4026	0.6658	0.5910	0.5989
SEBERT-LA	256	1	0.3681	0.4671	0.5292	0.5391	0.4013	0.6644	0.5890	0.5962

Table 3-3: Impact of Paragraph Length on Retrieval Accuracy

Additionally, this study explores how summarization models influence ranking performance. In this analysis, only the top 100 documents from DPH+KL were reranked to reduce computational demands. All other hyperparameters (e.g.,

summarized paragraph length) were kept identical to ensure fairness. Table 4-4 reports MAP@100, P@20, and NDCG@20. Three summarizers were used:

- PEGASUS[59]: The default summarization model in this framework, PEGASUS is a transformer-based encoder-decoder. During pretraining, PEGASUS masks key sentences in the input document, predicting them through other sentences, similar to summarization tasks.

- BART[60]: BART combines the bidirectional encoding of BERT and the left-to-right decoding of GPT, built on a standard sequence-to-sequence model. It captures more bidirectional context than GPT, advancing text generation and maintaining SOTA in text understanding tasks.

- T5[61]: A text-to-text transformer model, T5 operates with text inputs and outputs, achieving a ROUGE-1 score of 44.17 on the CNN/Daily Mail dataset, making it suitable for abstractive summarization.

Model	**Parameter Size**	**ROUGE-1 Score on CNN/DM**	**Robust04**			**GOV2**		
			MAP@100	P@20	NDCG@20	MAP@100	P@20	NDCG@20
PEGASUS-BASE	223M	41.79	0.3195	0.4697	0.5306	0.2560	0.6728	0.5961
PEGASUS-LARGE	568M	43.90	0.3209	0.4725	0.5383	0.2601	0.6691	0.5979
BART-BASE	121M	42.16	0.3184	0.4687	0.5322	0.2569	0.6711	0.5983
T5-BASE	220M	42.05	0.3193	0.4705	0.5341	0.2583	0.6658	0.5910

Table 3-4 Impact of Different Summarization Models on Experimental Results

From Table 3-4, it is evident that using these base-size models for summarization yields similar results in reranking the top 100 documents. Although larger summarization models perform better, they increase the framework's computational load, which conflicts with task requirements. This study suggests that appropriately-sized summarization models better suit retrieval tasks. The purpose of summarization is to condense text semantics, and the base neural models align well with this need.

Moreover, a specific example illustrates why SEBERT outperforms BERT for ranking. Table 3-5 provides a query example where bold text denotes semantic relevance:

Query	**Model**	**Document**	**DPH+KL Rank → New Rank**
Ship losses	BERT	AN INQUIRY into the loss of a Greek **oil tanker** off the Western Australian coast has revealed 'substantial and graphic' evidence of deficiencies in international checks on shipping, the Australian government said yesterday.	296 → 61
Ship losses	BERT+PEGASUS (SEBERT)	Original passage + The 22-year-old Kirki was not at sea, as it **lost its bow** in heavy seas. An inquiry found evidence of deliberate efforts to make the ship look seaworthy, with **plates being so thin** that they could be shaved with. Sen. Collins said the **corrosion** was also deliberate.	296 → 41

Table 3-5 Query Example Using Abstractive Summarization Method

In this example, the query is “Ship losses.” The candidate document includes the phrase “the loss of a Greek oil tanker,” clearly related to the query. However, the exact-match DPH+KL model struggles to determine the document’s relevance, as it cannot recognize “ship” and “oil tanker” as similar entities. Using BERT for reranking significantly improves the ranking because of its ability for fuzzy matching, interpreting “ship” and “oil tanker” as related. With SEBERT’s additional semantic context, more precise terms appear in the input sequence, alongside numerous terms related to “loss.” Thus, SEBERT provides a ranking boost over DPH+KL and BERT.

3.4 Chapter Summary

The abstractive summarization-enhanced long-text ranking method leverages abstraction to enrich text semantics. By using a text generation model to summarize the context of candidate paragraphs, these summaries are concatenated as part of the input sequence, allowing BERT to capture more semantic information despite sequence length limitations. Experiments on two TREC datasets reveal the effectiveness of the proposed method in aggregating contextual semantic relevance.

The proposed paragraph-context-based BERT ranking framework enhances relevance judgment and model performance. Future experiments will consider expanding the dataset range to further validate the model’s capabilities.

CONCLUSION

In recent years, text ranking methods based on pretrained models have achieved groundbreaking advances, becoming a central focus in information retrieval research. Applying BERT-based pretrained models to long-text ranking tasks presents two main challenges: (1) the input sequence length for models like BERT is capped at 512 tokens, truncating longer sequences and thereby leading to inaccurate relevance assessments; (2) existing methods for handling long-text ranking do not fully consider the structural semantic information of documents, resulting in semantic loss that may impact ranking performance for long texts. This thesis addresses these issues by systematically exploring ranking methods tailored for long texts. The main contributions of this research include:

1. Proposing a Paragraph-Level Latent Semantic-Based Long-Text Ranking Method (KnBERT):

In paragraph aggregation-based BERT long-text retrieval methods, the choice of pooling method—how paragraphs are aggregated—is crucial. Current commonly used pooling methods neglect the semantic structure of documents, leading to potential errors in certain contexts. Analyzing further, documents are composed of multiple paragraphs, each with distinct characteristics that contribute variably to relevance assessments. This research builds upon BERT-based retrieval, leveraging kernel functions to capture latent semantic relationships among paragraph positions and introduces a reward mechanism that integrates

paragraph-level semantic distribution to enhance retrieval performance. Experiments conducted on the Robust04 and GOV2 datasets investigate the effects of different kernel aggregation strategies and reward mechanism hyperparameters, showing that these strategies effectively improve retrieval model performance.

2. Proposing an Abstractive Summary-Based Semantic Enhancement Method for Long-Text Ranking (SEBERT):

Unlike traditional statistical ranking methods, BERT can capture semantic relationships across texts. This fuzzy matching capability implies that the input content, format, and semantic complexity can greatly influence BERT's retrieval effectiveness. For relevance assessment, each candidate text's context is crucial. Current retrieval methods focus on aggregating multiple text blocks to assess overall document relevance but do not account for paragraph context and inter-paragraph semantic information. To address this, the thesis proposes using abstractive summaries to enhance input semantics, capturing cross-paragraph interactions. A generative language model was employed as a summarization module to accurately summarize the context for each candidate paragraph, and these summaries were concatenated as input sequences. Extensive experiments on the Robust04 and GOV2 datasets validate SEBERT's effectiveness, showing improved performance over traditional and neural baseline models.

Future Directions

This thesis proposes a paragraph-level latent semantic-based long-text ranking method and an abstractive summary-based semantic enhancement method for long-text ranking. Building upon insights and challenges encountered in the research process, future work can proceed in the following directions:

1. Improving Kernel-Based Ranking Method Precision Without Increasing Query Latency:

Future work can explore (1) more effective reward mechanisms for relevant documents; (2) enhanced kernel-based paragraph aggregation methods, expanding into representation aggregation. Generating more accurate document representations based on paragraph-level semantic information could further improve retrieval accuracy.

2. Integrating Generative Large Language Models with Abstractive Summary-Based Ranking Methods to Enhance Retrieval Performance:

Large models can more accurately capture textual semantics. Incorporating them into the current method could significantly improve retrieval precision by leveraging their nuanced understanding of text meaning.

REFERENCES

[1] Vaswani, A., Shazeer, N., Parmar, N., et al. Attention is All You Need. Advances in Neural Information Processing Systems, 2017, 30.

[2] Devlin, J., Chang, M. W., Lee, K., et al. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. Proceedings of NAACL-HLT, 2019: 4171-4186.

[3] Robertson, S., Zaragoza, H., Taylor, M. Simple BM25 Extension to Multiple Weighted Fields. Proceedings of the Thirteenth ACM International Conference on Information and Knowledge Management, 2004: 42-49.

[4] Robertson, S., Zaragoza, H. The Probabilistic Relevance Framework: BM25 and Beyond. Foundations and Trends® in Information Retrieval, 2009, 3(4): 333-389.

[5] Mikolov, T., Chen, K., Corrado, G., et al. Efficient Estimation of Word Representations in Vector Space. arXiv preprint arXiv:1301.3781, 2013.

[6] Palangi, H., Deng, L., Shen, Y., et al. Deep Sentence Embedding Using Long Short-Term Memory Networks: Analysis and Application to Information Retrieval. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 2016, 24(4): 694-707.

[7] Peters, M., Neumann, M., Iyyer, M., et al. Deep Contextualized Word Representations. arXiv preprint arXiv:1802.05365, 2018.

[8] Nogueira, R., Cho, K. Passage Re-ranking with BERT. arXiv preprint arXiv:1901.04085, 2019.

[9] Dai, Z., Callan, J. Deeper Text Understanding for IR with Contextual Neural Language Modeling. Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, 2019: 985-988.

[10] Yilmaz, Z. A., Yang, W., Zhang, H., et al. Cross-domain Modeling of Sentence-level Evidence for Document Retrieval. Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), 2019: 3490-3496.

[11] Nguyen, T., Rosenberg, M., Song, X., et al. MS MARCO: A Human Generated Machine Reading Comprehension Dataset. Choice, 2016, 2640: 660.

[12] Dietz, L., Verma, M., Radlinski, F., et al. TREC Complex Answer Retrieval Overview. Proceedings of The Twenty-Sixth Text REtrieval Conference, 2017.

[13] Li, C., Yates, A., MacAvaney, S., et al. PARADE: Passage Representation Aggregation for Document Reranking. ACM Transactions on Information Systems, 2023, 42(2): 1-26.

[14] Zheng, Z., Hui, K., He, B., et al. Contextualized Query Expansion via Unsupervised Chunk Selection for Text Retrieval. Information Processing & Management, 2021, 58(5): 102672.

[15] Yu, H. C., Dai, Z., Callan, J. PGT: Pseudo Relevance Feedback Using a Graph-based Transformer. Advances in Information Retrieval: 43rd European Conference on IR Research, ECIR 2021, Virtual Event, March 28–April 1, 2021, Proceedings, Part II 43. Springer International Publishing, 2021: 440-447.

[16] Pan, M., Wang, J., Huang, J. X., et al. A Probabilistic Framework for Integrating Sentence-level Semantics via BERT into Pseudo-relevance Feedback. Information Processing & Management, 2022, 59(1): 102734.

[17] Wang, J., Pan, M., He, T., et al. A Pseudo-relevance Feedback Framework Combining Relevance Matching and Semantic Matching for Information Retrieval. Information Processing & Management, 2020, 57(6): 102342.

[18] Chen, X., Hui, K., He, B., et al. Co-BERT: A Context-Aware BERT Retrieval Model Incorporating Local and Query-specific Context. arXiv preprint arXiv:2104.08523, 2021.

[19] Beltagy, I., Peters, M. E., Cohan, A. Longformer: The Long-document Transformer. arXiv preprint arXiv:2004.05150, 2020.

[20] Zaheer, M., Guruganesh, G., Dubey, K. A., et al. Big Bird: Transformers for Longer Sequences. Advances in Neural Information Processing Systems, 2020, 33: 17283-17297.

[21] Fan, Y., Xie, X., Cai, Y., et al. Pre-training Methods in Information Retrieval. Foundations and Trends® in Information Retrieval, 2022, 16(3): 178-317.

[22] Yin, D., Hu, Y., Tang, J., et al. Ranking Relevance in Yahoo Search. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016: 323-332.

[23] Liu, Y., Lu, W., Cheng, S., et al. Pre-trained Language Model for Web-scale Retrieval in Baidu Search. Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, 2021: 3365-3375.

[24] Li, H., Xu, J. Semantic Matching in Search. Foundations and Trends in Information Retrieval, 2014, 7(5): 343-469.

[25] Craswell, N., Mitra, B., Yilmaz, E., et al. MS MARCO: Benchmarking Ranking Models in the Large Data Regime. Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2021: 1566-1576.

[26] Khattab, O., Zaharia, M. ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT. Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, 2020: 39-48.

[27] Burges, C., Shaked, T., Renshaw, E., et al. Learning to Rank Using Gradient Descent. Proceedings of the 22nd International Conference on Machine Learning, 2005: 89-96.

[28] Burges, C., Ragno, R., Le, Q. Learning to Rank with Nonsmooth Cost Functions. Advances in Neural Information Processing Systems, 2006, 19.

[29] Guo, J., Fan, Y., Ai, Q., et al. A Deep Relevance Matching Model for Ad-hoc Retrieval. Proceedings of the 25th ACM International Conference on Information and Knowledge Management, 2016: 55-64.

[30] Mitra, B., Diaz, F., Craswell, N. Learning to Match Using Local and Distributed Representations of Text for Web Search. Proceedings of the 26th International Conference on World Wide Web, 2017: 1291-1299.

[31] Yates, A., Nogueira, R., Lin, J. Pretrained Transformers for Text Ranking: BERT and Beyond. Proceedings of the 14th ACM International Conference on Web Search and Data Mining, 2021: 1154-1156.

[32] Qiao, Y., Xiong, C., Liu, Z., et al. Understanding the Behaviors of BERT in Ranking. arXiv preprint arXiv:1904.07531, 2019.

[33] Liu, T. Y. Learning to Rank for Information Retrieval. Foundations and Trends® in Information Retrieval, 2009, 3(3): 225-331.

[34] MacAvaney, S., Yates, A., Cohan, A., et al. CEDR: Contextualized Embeddings for Document Ranking. Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, 2019: 1101-1104.

[35] Xiong, C., Dai, Z., Callan, J., et al. End-to-End Neural Ad-hoc Ranking with Kernel Pooling. Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2017: 55-64.

[36] Pradeep, R., Nogueira, R., Lin, J. The Expando-mono-duo Design Pattern for Text Ranking with Pretrained Sequence-to-sequence Models. arXiv preprint arXiv:2101.05667, 2021.

[37] Ren, R.,Qu, Y., Liu, J., et al. RocketQAv2: A Joint Training Method for Dense Passage Retrieval and Passage Re-ranking. arXiv preprint arXiv:2110.07367, 2021.

[38] Guo, J., Fan, Y., Pang, L., et al. A Deep Look into Neural Ranking Models for Information Retrieval. Information Processing & Management, 2020, 57(6): 102067.

[39] Yilmaz, Z. A., Wang, S., Yang, W., et al. Applying BERT to Document Retrieval with Birch. Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP): System Demonstrations, 2019: 19-24.

[40] Callan, J. P. Passage-level Evidence in Document Retrieval. SIGIR'94: Proceedings of the Seventeenth Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval. London: Springer London, 1994: 302-310.

[41] Liu, X., Croft, W. B. Passage Retrieval Based on Language Models. Proceedings of the Eleventh International Conference on Information and Knowledge Management, 2002: 375-382.

[42] Wu, Z., Mao, J., Liu, Y., et al. Investigating Passage-level Relevance and Its Role in Document-level Relevance Judgment. Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, 2019: 605-614.

[43] Kong, K., Luk, R., Ho, K., et al. Passage-Based Retrieval Using Parameterized Fuzzy Set Operators. ACM SIGIR Workshop on Mathematical/Formal Methods for Information Retrieval, 2004.

[44] Li, X., Liu, Y., Mao, J., et al. Understanding Reading Attention Distribution During Relevance Judgment. Proceedings of the 27th ACM International Conference on Information and Knowledge Management, 2018: 733-742.

[45] Pan, M., Huang, J. X., He, T., et al. A Simple Kernel Co-occurrence-based Enhancement for Pseudo-relevance Feedback. Journal of the Association for Information Science and Technology, 2020, 71(3): 264-281.

[46] Lv, Y., Zhai, C. X. A Comparative Study of Methods for Estimating Query Language Models with Pseudo Feedback. Proceedings of the 18th ACM Conference on Information and Knowledge Management, 2009: 1895-1898.

[47] Voorhees, E. M. Overview of the TREC 2004 Robust Track. Thirteenth Text Retrieval Conference. DBLP, 2004.

[48] Clarke, C. L. A., Craswell, N., Soboroff, I. Overview of the TREC 2004 Terabyte Track. TREC, 2004, 4: 74.

[49] Yang, W., Lu, K., Yang, P., et al. Critically Examining the "Neural Hype": Weak Baselines and the Additivity of Effectiveness Gains from Neural Ranking Models. Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, 2019: 1129-1132.

[50] Oyeka, I. C. A., Ebuh, G. U. Modified Wilcoxon Signed-rank Test. Open Journal of Statistics, 2012, 2(2): 172-176.

[51] Yang, P., Fang, H., Lin, J. Anserini: Reproducible Ranking Baselines Using Lucene. Journal of Data and Information Quality (JDIQ), 2018, 10(4): 1-20.

[52] Amati, G., Amodeo, G., Bianchi, M., et al. FUB, IASI-CNR and University of "Tor Vergata" at TREC 2008 Blog Track. TREC, 2008.

[53] Lavrenko, V., Croft, W. B. Relevance Based Language Models. Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 2001: 120-127.

[54] Keikha, M., Park, J. H., Croft, W. B. Evaluating Answer Passages Using Summarization Measures. Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval, 2014: 963-966.

[55] Keikha, M., Park, J. H., Croft, W. B., et al. Retrieving Passages and Finding Answers. Proceedings of the 19th Australasian Document Computing Symposium, 2014: 81-84.

[56] Robertson, S. E. The Probability Ranking Principle in IR. Journal of Documentation, 1977, 33(4): 294-304.

[57] Zhao, J., Huang, J. X., He, B. CRTER: Using Cross Terms to Enhance Probabilistic Information Retrieval. Proceedings of the 34th International ACM

SIGIR Conference on Research and Development in Information Retrieval, 2011: 155-164.

[58] Zhao, J., Huang, J. X., Ye, Z. Modeling Term Associations for Probabilistic Information Retrieval. ACM Transactions on Information Systems (TOIS), 2014, 32(2): 1-47.

[59] Zhang, J., Zhao, Y., Saleh, M., et al. Pegasus: Pre-training with Extracted Gap-sentences for Abstractive Summarization. International Conference on Machine Learning. PMLR, 2020: 11328-11339.

[60] Lewis, M., Liu, Y., Goyal, N., et al. BART: Denoising Sequence-to-sequence Pre-training for Natural Language Generation, Translation, and Comprehension. arXiv preprint arXiv:1910.13461, 2019.

[61] Roberts, A., Raffel, C., Lee, K., et al. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. arXiv preprint arXiv:1910.10683, 2019.

[62] Tian-Di, G. Research on Long Text Ranking Methods Based on Pre-trained Models. Master's Thesis, Central China Normal University, 2022.

[63] Min, P. Research on Pseudo-Relevance Feedback Query Expansion Technology Based on Latent Semantic Relationships. Master's Thesis, Central China Normal University, 2019.