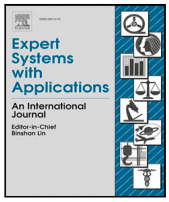
Expert Systems With Applications 238 (2024) 121629

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa



Transformer based contextual text representation framework for intelligent information retrieval

Amol P. Bhopale [a,b,*], Ashish Tiwari [a]

[a] Department of Computer Science and Engineering, Visvesvaraya National Institute of Technology, Nagpur, 440010, India
[b] Department of Computer Science and Engineering, Indian Institute of Information Technology, Nagpur, 441108, India

ARTICLE INFO

Keywords:
Neural information retrieval
BERT
Transfer learning
Query expansion
Document summarization

ABSTRACT

With the advent of transformer-based architectures, the contextual representation of text data has leveraged the query and the document to be represented in low-dimensional dense vector space. These vectors are learned embeddings of fixed sizes, resulting in deeper text understanding. In this study, we designed a pipeline for effectively retrieving documents from a large search space by combining the deeper text understanding capabilities of the transformer-based BERT model and a phrase embedding-based query expansion model. To learn the contextual representations, we fine-tuned a deep semantic matching model by separately encoding the document and the query. The encoder model is based on the Sentence BERT (SBERT) architecture, which separately generates dense vector representations of documents and queries. The study has also addressed the maximum token length limitation of transformer-based models through the summarization of lengthy documents. In addition, to improve the clarity and completeness of short queries and reduce the semantic gap, a phrase embedding-based query expansion model is employed. The documents and their dense vectors are indexed using the Elasticsearch engine, and matched them with query vectors for retrieving query-specific documents. Finally, the BERT-based cross-encoder model is used to re-rank the relevant records for each query. It performs full self-attention over the inputs, and yields richer text interactions to produce the final results. To assess performance, experiments are conducted on two well-known datasets, TREC-CDS-2014 and OHSUMED. A comparative analysis is carried out, which clearly demonstrates that the proposed framework produced competitive retrieval results.

1. Introduction

In recent years, a large number of successful deep learning algorithms have been developed to avoid the usage of manually engineered features in the field of information retrieval (IR).

Many users perform searches with the short keywords for their information need. The widespread use of term matching approaches across various search engines prevent such users from conducting short keyword-based searches. Users may not use natural language search for variety of reasons, including a lack of clarity in their information requirements, an inability to formulate information needs in non-native languages, or simply because they are ignorant of search engine's ability to recognize natural languages.

A search engine typically receives user query and returns a list of relevant documents in few milliseconds, and satisfies user information need instantly. On the other hand, the language used in a query and a document differs in terms of verbosity, correctness, and format. Finding the most relevant documents among tens of millions of records is difficult due to the semantic gap that exists between search queries and documents. The studies have shown that relevance matching and semantic matching are two different tasks with shared common characteristics. Relevance matching is heavily reliant on exact match signals, whereas semantic matching is concerned with the context of the query.

With the release of the Word2vec model in 2013 (Mikolov, Chen, Corrado, & Dean, 2013), word embedding has become a popular method for constructing term-based embeddings. However, the contextual representations obtained with the introduction of BERT (Devlin, Chang, Lee, & Toutanova, 2018) in late 2018 have overshadowed the term-based embedding. The study (Dai & Callan, 2019) discovered that questions phrased in natural language using BERT model provide better search results than keyword-based approaches. In BERT architecture, embedding vectors are generated using the surrounding context, therefore allows users to articulate their information need more effectively. As one of the contributions of this work, We propose that query expansion may benefit BERT-based retrieval models, whereas conventional approaches may not be viable due to a lack of inherent word order and natural language structure connecting them.

* Corresponding author.
E-mail addresses: amolpbhopale@gmail.com (A.P. Bhopale), at@cse.vnit.ac.in (A. Tiwari).

https://doi.org/10.1016/j.eswa.2023.121629
Received 23 June 2022; Received in revised form 15 August 2023; Accepted 13 September 2023
Available online 23 September 2023

A common approach for dealing with semantic search is to map each query vector to a document vector, which reduces the distance between semantically similar sentences and brings them closer together. The most common method is to take the average of all token vectors in the BERT output layer (also known as BERT embeddings) or to use the first token's output (i.e. the [CLS] token). However, this technique yields poor sentence embeddings. To tackle this issue, a Siamese network architecture is employed in the Sentence BERT architecture to generate fixed-size vectors of the input sentences separately. The model allows storing document vectors, and semantically related vectors are matched using a similarity measure such as cosine similarity or Manhattan/Euclidean distance. Although BERT has proven to be a successful model for retrieving passages, its maximum input length limitation makes retrieving long documents difficult. This restricts the transformer to inputs of a specific length. Therefore, in this study we incorporated BERT-based document summarization technique to extract and process most descriptive sentences from long documents.

More specifically, we assumed in this work that deep linguistic contextual representation of text can reduce the semantic gap between documents and queries in asymmetric search, thereby improving retrieval relevance. As a result, this paper presents a transformer-based document retrieval framework as an alternative to conventional inverted index-based retrieval systems. The key contributions of this paper are as below-

- For Information Retrieval, an end-to-end document retrieval pipeline is designed which considers the contextual representation of the query and the document.
- To learn the contextual representation of text and separately generate the dense vectors of documents and the query, a transformer-based encoder architecture is used.
- A BERT-based document summarization approach is used to overcome the maximum token sequence length limit and avoid descriptive feature loss in document encoding.
- To contextually expand short queries for better interpretation, a query expansion is performed using the phrase embedding model.
- The Elasticsearch framework is used to index the query and document vectors. It is also used to retrieve documents based on cosine function matching scores.
- A BERT-based cross-encoder model is employed to re-rank the findings and generate final results.
- Extensive experiments are carried out on two well-known large datasets in order to assess the performance of the proposed work.

These contributions help to find out solutions of the following research questions-

I. **(RQ1)** Is it possible to improve retrieval performance by encoding both documents and queries in the same vector space using the sentence BERT model?
II. **(RQ2)** To what extent can summarizing lengthy documents help retain key descriptive features and improve retrieval performance?
III. **(RQ3)** Does query expansion using the phrase embedding technique improve the interpretation of user queries?

The paper is organized as follows: In Section 2, we discussed studies conducted in the IR domain. Section 3 provides a comprehensive overview of the proposed framework. Section 4 discussed the experimental details, results, and analysis, followed by the conclusion and the future work in Section 5.

2. Related work

Over the last decade, IR models have gained a promising emphasis in learning query-document relevance representations. Text presentations are learned by models using variety of methodologies including click log signals, pseudo-relevance feedback, neural networks and so on. The most prominent method of learning text representations has been demonstrated to be exact matching at the document and passage levels achieved using neural network architecture. The majority of neural IR models have employed word embeddings, such as the Word2vec (Mikolov et al., 2013) approach. However, word embedding models are incapable of dealing with polysemy words and unseen words. The use of pre-trained neural language models based on transformer architecture, such as Devlin et al. (2018), Radford et al. (2019) and Peters et al. (2018) has resulted in a rapid advancement in text understanding approaches. In comparison to traditional word embedding models, these models are contextual in the sense that the representation of a word is a function of the entire input text. The papers (Devlin et al., 2018; Peters et al., 2018) show that transformer-based models outperform standard word embedding-based techniques in a wide range of NLP tasks. The rich text interpretation provided by these contextual neural language models has opened up new opportunities for IR.

We conducted a brief review of the IR literature in three categories: conventional term matching-based approaches, word embedding-based approaches, and contextual text representation-based approaches.

2.1. Conventional term matching based approaches

The practise of experimenting with various strategies to improve retrieval results in IR has a long history, and it is still ongoing. The keyword matching and a VSM for document representation were two approaches used to determine the relevance of data (Wiemer-Hastings, Wiemer-Hastings, & Graesser, 2004) in early years. As shown below, this section discusses existing IR approaches that are being studied as conventional approaches.

VSM is one of the most fundamental methods for representing data in algebraic form. Salton, Wong, and Yang (1975) invented it in 1975, document vectors are generated by treating the distinct vocabulary words as dimensions. The weights of vectors are determined by the frequency of words in a document, with some variations in weighting strategies based on word frequency and normalized frequency (Salton & McGill, 1983). For many years, document-based VSM and word-based VSM techniques have been utilized successfully in a number of text processing techniques such as text classification, clustering, sentiment analysis, information extraction, information retrieval, word sense disambiguation, and so on. Despite having various applications, the size of vector dimension grows in thousands and millions for large datasets. Thus the VSM-based methods suffer from the curse of dimensionality. Latent Semantic Analysis (LSA) (Deerwester, Dumais, Furnas, Landauer, & Harshman, 1990) is a significant dimensionality reduction technique applied in many studies. It describes document vectors in the space of words using singular value decomposition to establish word-to-word relationships. These methods have a fundamental flaw in that they ignore word co-occurrences at the collection level and only generate word correlations at the document level.

Improvements to the VSM paradigm have been observed using various word weighting approaches, such as the term frequency-inverse document frequency (TFIDF) model (Salton & McGill, 1983) and the Okapi BM25 model. VSMs are also incapable of handling the semantic association between words and documents. The following years saw the development of latent semantic indexing (LSI) and probabilistic language models in an attempt to address this constraint. The paper (Kontostathis, 2007) examined different components of the LSI technique and designed a mechanism for maintaining word relationships. The investigation found that an optimized LSI system collects connections between terms in the first few dimensions, also known as the essential dimensions. The paper (Zamani & Croft, 2017) has developed an architecture that transforms text into a sparse representation. It treats non-zero values as virtual terms in order to retrieve documents using Lucene's term-based indexing.

In recent years, several probability distribution-based studies have been conducted. da Silva and Maia (2019) investigated a query expansion technique based on the distribution semantic approach, whereas (Song, Hu, He, & Dou, 2019) proposed a self-adapting Saturated Density-based probabilistic IR model, the paper has investigated the impact of density function on proximity-based IR. To address the diverse distribution influences created by different terms in the healthcare domain, a self-adaptive technique based on Saturated Density Function is built with a gamma function. The authors have developed a probabilistic weighting model by estimating the distribution of each term depending on its density. The study, however, only looked at the impact of term density distribution at the document level, not the sentence or passage level. The study (Bhopale & Tiwari, 2020) has proposed a swarm optimized cluster based IR approach where, authors first applied K-flock clustering technique to divide dataset into small clusters. Then, on each cluster, the frequent pattern mining technique is used to generate patterns and match them with query keywords for document retrieval. The approach is computationally expensive as it involves rule mining for identifying cluster patterns.

Probabilistic language models are designed to characterize term probability distributions in relevant topical content. The term weights are determined by the distribution of the term across the collection of documents. Several neural IR-based experiments have been carried out in recent years to design semantic representations of words using low-valued dense vectors computed by taking into account the neighboring context (Marchesin, Purpura, & Silvello, 2019; Ran, He, Hui, Xu, & Sun, 2017; Yang, He, Li, & Xu, 2017). The primary goal of these models was to establish semantic connections between words and improve document retrieval relevance.

2.2. Embedding based approaches

Embedding is a neural network-based approach for processing large amounts of text input that involves the representation of word semantics in a low-dimensional vector space. In recent years, embedding techniques have accelerated development in NLP and IR studies. Embedding is not a new concept; (Bengio, Ducharme, Vincent, & Janvin, 2003) constructed a neural probabilistic language model in 2003 by learning the distributed representations of words to overcome the challenges posed by high dimensionality which is induced by one-hot vectors. The authors discovered that by using contextually related concepts for target word, they could semantically capture the relationship with improved compositionality. In paper (Collobert & Weston, 2008), the benefits of distributed word representation, i.e. word embeddings, as a feature of NLP tasks are first demonstrated. As stated previously, embedding is not a new concept; ; however, it was not widely used until 2013, when Mikolov et al. (2013) presented an open-source Word2vec embedding model to generate the vector representations of words.

In IR tasks, word embedding has been widely used to solve the problem of vocabulary mismatch by producing semantically comparable concepts. El Mahdaouy, El Alaoui, and Gaussier (2018) proposed a neural word embedding model in which words are represented in a low-dimensional vector space model to deal with feature inconsistency. They demonstrated a method for extracting Arabic information that makes use of word embedding similarity and a probabilistic IR model. In the paper, Zheng and Callan (2015) authors described a distributed semantics representation, specifically a term re-weighting technique based on the embedding approach for language modeling and BM25 retrieval models. The paper (Nalisnick, Mitra, Craswell, & Caruana, 2016) used a hybrid compositional model with word embedding to describe queries and documents, and the cosine similarity function to score documents. A probabilistic translation language model based on scores calculated between the embeddings using cosine similarity measure are demonstrated in Ganguly, Roy, Mitra, and Jones (2015) and Zuccon, Koopman, Bruza, and Azzopardi (2015).

The inability of word vector representation techniques is to represent multi-word units i.e. phrases. Even before Word2vec became prominent, the best-known studies presented in Blacoe and Lapata (2012) and Socher, Manning, and Ng (2010), where the individual word vectors are used to compute the low-dimensional vectors of phrases or sentences. Using statistical methodology influenced by word embedding techniques (Yin & Schütze, 2014), the continuous and non-contiguous units of words, i.e. phrases, are built and processed as a single unit to represent it in distributed vectors (Bhopale & Tiwari, 2021). Authors have used the phrase embedding model for query expansion in IR task. In Yu and Dredze (2015), authors have developed a feature-rich compositional transformation (FCT) model for phrase vector encoding. The vectors for each phrase are created using the weighted sum of word vectors based on the feature list. To learn phrase embeddings, a hybrid technique based on the linear combination of distributional and compositional components with compositionality constraints for each phrase was developed in paper (Li, Lu, Xiong and Long, 2018). The papers (Bai et al., 2018; Grbovic et al., 2016) established a consistent query and document embedding framework, by generating n-gram embedding from a user session and click logs. It is then generalized to arbitrary text using mean average pooling of n-gram embedding.

Traditional query expansion methods add extra terms from relevant documents to the original query, which are typically part of bag-of-words (Carpineto & Romano, 2012; Lavrenko & Croft, 2001). Query expansion technique has improved the effectiveness of neural ranking models when the query terms are softly matched with the document terms (Li, Sun et al., 2018). In ALMasri, Berrut, and Chevallet (2016), the authors introduced the VEXP technique, an embedding vector-based heuristic method for expanding query phrases by adding the most related terms in the embedding space. They built a model by fine-tuning parameters and investigated term frequencies by weighing it in the expanded query. User queries are typically too specific and short to accurately express information needs. To improve retrieval efficacy, several researchers have investigated query expansion studies by adding contextually related terms generated using embedding techniques. However, the shallow neural network-based embedding approaches such as Word2vec (Mikolov et al., 2013) and GloVe (Pennington, Socher, & Manning, 2014) represent word vectors by considering word meanings derived from the occurrences of context window words. Each word is mapped to a vector representation using a language model. For vector representation, these embedding models consider only the context window words of a few sizes and not the entire document. As a result, the same vector is generated for polysemy words, i.e. words found in different contexts with different meanings. For example, the word “cold” has a different meaning in “cough & cold”, “cold sensation”, and “cold blooded”.

2.3. Contextual text representation based approaches

Recently studied transformer based models such as BERT (Devlin et al., 2018), RoBERTa (Liu et al., 2019), and XLNET (Yang et al., 2019), provide different ways of building document representations by generating distinct vectors of same word in the different contexts. These models are pre-trained with self-supervised learning tasks such as next sentence prediction and masked language modeling tasks. Due to the self-supervised learning approach, a large amount of data is used for training without the necessity of expensive annotations. The pre-trained models are used to encode general language patterns in contextual representations and can be easily fine-tuned to outperform on several state-of-the-art models in a variety of NLP tasks. Inspired from the performances of BERT-based contextual models on various downstream NLP tasks, authors in paper (Nogueira & Cho, 2019) explored BERT’s effectiveness on passage re-ranking tasks using MS MARCO and TREC-CAR datasets. They presented a monoBERT text retrieval model that is formulated as a binary classification task. The model estimated whether

Table 1
Summary of relevant literature studied in the recent years.

Study	Model	Dataset	Dataset source
Ran et al. (2017)	Document neural relevance model (DNRM)	TREC CDS 2014 - 733,138 long scientific articles	PubMed Central Repository
Yang et al. (2017)	Feedback-Based Semantic Relevance model	TREC CDS 2014, 2015 - 733,138 long scientific articles	PubMed Central Repository
da Silva and Maia (2019)	Local Context Analysis (LCA) with Semantic Distributional Model	MED- 1033 documents, LISA- 6004 documents, NPL- 11 429 documents	Collection of short articles from various domains.
Song et al. (2019)	Self-adaptive approach on SDF model	CLEF's eHealth 2013, 2014, 2015 - 1.1M documents; TREC CDS 2014& 15 - 0.73M	CLEF eHealth; PubMed Central.
Bhopale and Tiwari (2020)	Cluster Based Optimization model	TREC CDS 2014-15 - 0.73M scientific articles; OHSUMED- 0.34M abstracts.	PubMed Repository; MEDLINE literature.
Zheng and Callan (2015)	Distributed representations of words based on NN Language Model.	ROBUST04- 0.5M articles; WT10g- 1.7M articles; GOV2- 25M articles; ClueWeb09B- 29M articles.	News articles and Web pages collected from TREC.
Nalisnick et al. (2016)	Dual Embedding Space Model (DESM)	Bing's large scale search logs with 2.7M words	Bing search engine
Yu and Dredze (2015)	Phrase Embedding Model	Subset of The Gigaword dataset from NYT portion news articles with 515M words	News Articles
Li, Lu et al. (2018)	Phrase Embedding Model	Wikipedia dump - 3.2 billion words; Baidu Baike Chinese Dataset - 1.8 billion words	General English & Chinese Datasets
El Mahdaouy et al. (2018)	Word Embedding based Probabilistic IR model	Arabic Standard TREC 2001–2000 articles & TREC 2002 - 1200 articles.	Arabic newswire dataset
Bhopale and Tiwari (2021)	Phrase Embedding based QE model for IR	TREC CDS 2014-15 - 0.73M scientific articles; OHSUMED- 0.34M abstracts.	PubMed Repository; MEDLINE literature.
Nogueira and Cho (2019)	monoBERT model & duoBERT model	MS MARCO- 8.8M passages from 3.6M web pages; TREC CAR - 29M documents	Web pages & Wikipedia dump
Dai and Callan (2019)	BERT model	Robust04- 0.5M documents; ClueWeb09-B - 50M web pages	News articles & web pages
Nogueira et al. (2019)	BERT model	MS MARCO- 8.8M passages from 3.6M web pages; TREC CAR - 29M documents	Web pages & Wikipedia dump
Wu et al. (2020)	Passage-level Cumulative Gain Model & BERT model	TianGong-PDR Dataset; THUCNews - a chinese news corpus with 1050 documents	Search engine logs

or not the given query-document pair is relevant to the query. The model has outperformed compare to the other ranking models based on shallow neural network architectures such as Co-PACRR (Hui, Yates, Berberich, & De Melo, 2018) and KNRM (Xiong, Dai, Callan, Liu, & Power, 2017).

The transformer-based models enable deep neural ranking models to capture previously difficult-to-capture latent language traits in addition to query-text contextualization. In paper (Dai & Callan, 2019), authors designed a BERT-based model to evaluate the significance of each passage independently and demonstrated better performance on longer queries than on the shorter keyword queries. They divided each document into individual passages and used BERT re-ranker to obtain relevance scores. The document score is determined by the score of the first passage, the best passage, or the sum of all passage scores. Authors in paper (Yilmaz, Yang, Zhang, & Lin, 2019) performed document ranking by aggregating sentence-level evidence across different domains using transfer learning models. In paper (Nogueira, Yang, Cho, & Lin, 2019), the authors investigated a multi-stage ranking architecture that fine-tunes BERT to compare the relevance of the query-document pair. The paper (Wu et al., 2020) has presented a passage level Cumulative Information Gain model that uses BERT architecture to consider the contextual representation of each passage and aggregates the relevance score of passages without formally splitting the document into separate passages. Inspired by the classic query likelihood model, in the paper (Ma et al., 2021) authors employed the BERT model to optimize the most representative words in a document. The present BERT applications in the search framework are confined to re-ranking since they rely on its next sentence prediction method for a regression score.

Table 1 provides the brief summary of IR approaches studied in recent years.

The proposed framework aims to improve the IR problem by utilizing transformer-based contextual representations of the document and query. However, as discussed earlier, BERT has a limitation on the maximum allowed sequence length of the input document. The computational complexity of the self-attention mechanism grows quadratically with the increasing document length. Therefore, a summarization technique is integrated with the proposed framework. We also integrated the phrase embedding technique to improve the clarity and completeness of short queries. The following section describes each component of the proposed framework.

3. The methodology

This study has assumed that the deep linguistic contextual representation of text can reduce the semantic gap between documents and queries in asymmetric search, and thus improve the relevance in the retrieval. Therefore, we designed a transformer-based contextual representation model for IR. Fig. 1 shows the flow diagram of the proposed model and each component in the flow-diagram is discussed in detail as below:

3.1. Generating semantic dense vector representation

This section presents the document formatting and the dense vectors i.e. embeddings generation of the collected documents using the encoder architecture.

3.1.1. Document formatting

Document formatting consists of applying natural language processing techniques such as case conversion, special characters, and noise removal and storing the documents in one single text file for processing.

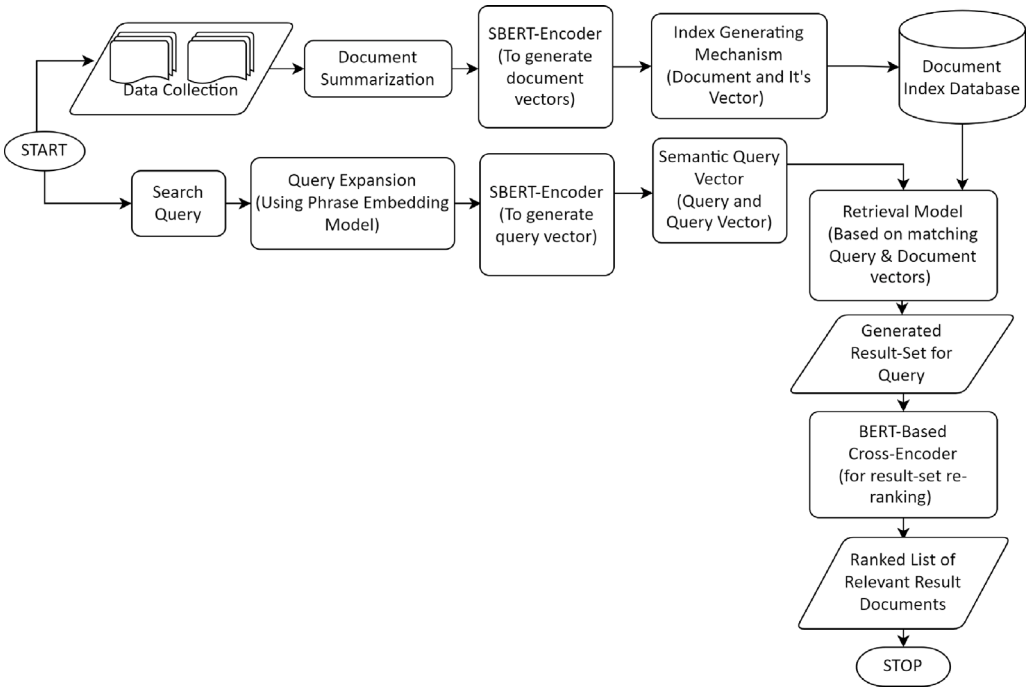


Fig. 1. Flow diagram of the proposed framework.

Table 2
Trade-off between sequence length and obtained results on OHSUMED dataset with SBERT Model.

Maximum allowed sequence length	Time required to generate representations	Retrieval result (MAP)
128 (Default)	49 mins	0.2887
300	87 mins	0.3461
512	249 mins	0.3504

The text file is further used by the document summarization technique to generate summaries for each document.

In transformer-based architectures, the run-time and the memory requirement grow quadratically with the input length. Therefore to balance the usage of resources, the maximum input sequence length allowed is 512 Wordpiece tokens, which approximately correspond to 100–200 English words depending on the size of words. BERT uses Wordpiece embeddings with a vocabulary size equal to 30,000 to generate input tokens. Following example shows how the WordPiece tokens look like for the given English word: Eg:

English Word- '*Embeddings*'

Wordpiece Token Representation- ['em', '##bed', '##ding', '##s']

In the above example, the word "*Embedding*" is converted into 4 tokens using the WordPiece embedding. BERT-based encoders truncate sequences of more than 512 tokens and therefore, may lose important descriptive features of the document. Following Table 2 gives the trade-off between the sequence length, the retrieval performance of the model, and the time required to generate text representation. The results are generated on the original documents without applying summarization and truncated the tokens from the documents having token length more than the allowed sequence length.

The above Table 2 depicts that, increasing the sequence length increases resource consumption, whereas decreasing the sequence length decreases the quality of outcomes. Therefore in this study, we applied the summarization technique to extract important sentences and thus satisfy the maximum token limit of BERT architecture.

3.1.2. Document summarization

There are two types of summarization techniques widely used in the literature- Extractive Summarization and Abstractive Summarization. The extractive summarization technique identifies the important content of the document and predicts the scores for sentences that are to be considered. The abstractive summarization on the other hand reproduces the sequences in an altogether new way after interpretation of the text using the advanced NLP techniques and conveys important information from the original text. However, the purpose of the end-user decides which summarization is better to consider. Though abstractive summarization is more advanced and closer to human-like interpretation, it causes to lose the originality of the content, thus to retain document in its original content format, we preferred to apply the extractive summarization in this study. The Extractive summarization selects the top N sentences that best represent the document. It is set up as binary classification problems where an encoder encodes sentences and generates representations, and the classifier predicts the scores to select the sentences used for summary.

The BERTSUM (Liu & Lapata, 2019) model is employed to perform extractive summarization as shown in Fig. 2, which uses a BERT encoder to generate scores of sentences. BERTSUM model takes document sentences as input and allows to interact through multiple transformer layers of BERT architecture. The value of [CLS] tokens for multiple sentences is used to aggregate features from sentences. After obtaining the [CLS] vector values, the final prediction score Y_i is calculated for each sentence S_i using the simple classifier layer which uses a sigmoid function to generate the predicted score as shown in Eq. (1). The top-scoring sentences are considered as the descriptive sentences for a document. Y_i can be calculated as below:

$$Y_i = \sigma(W_o T_i + b_o) \tag{1}$$

where σ is the sigmoid function, T_i is the vector of the ith [CLS] token, W_o is the initial weights of the linear layer and b_o is the bias value.

Following Table 3 gives the statistics of average length of document of OHSUMED (Robertson & Hull, 2000) and TREC CDS dataset before and after applying summarization.

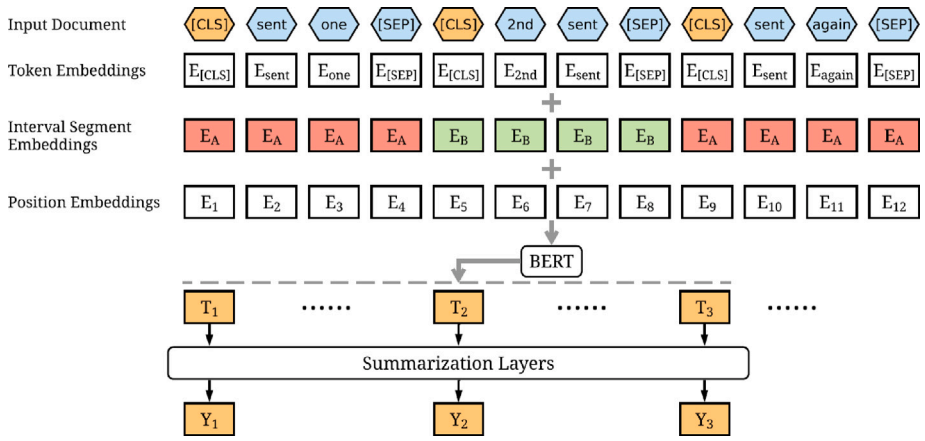


Fig. 2. BERTSUM model architecture (Liu & Lapata, 2019).

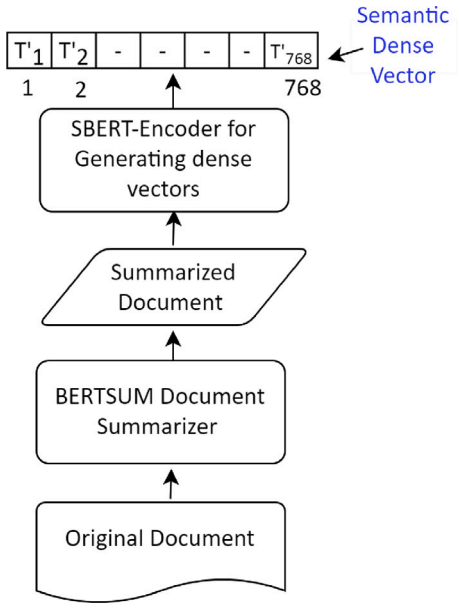


Fig. 3. SBERT-Encoder model with summarization layer.

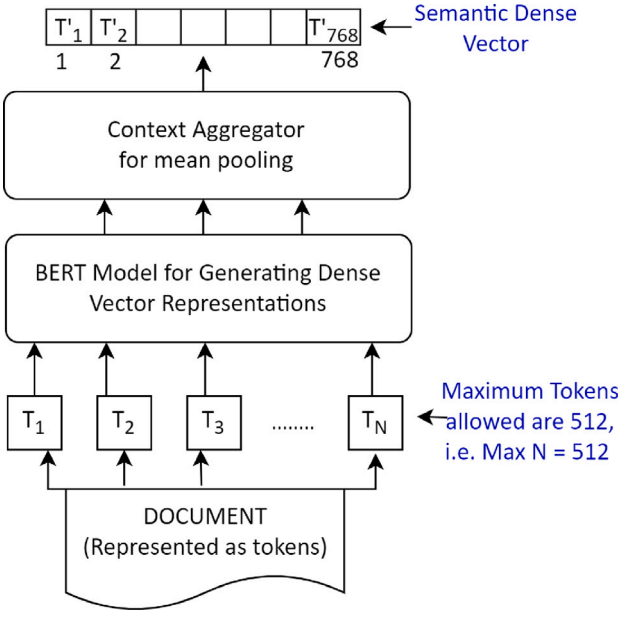


Fig. 4. SBERT encoder architecture.

Table 3
Average document length.

	OHSUMED	TREC CDS 2014
Avg. length without summarization	141	306
Avg. length with summarization	47	149

3.1.3. Dense vectors generation using SBERT-Encoder

BERT architecture generates rich interaction between the candidate and context input tokens, however, it is unable to produce separate embeddings of inputs. This makes it difficult to independently represent inputs as a dense vector. Several approaches are studied in the past to address this limitation by passing the single input to the BERT model and deriving the fixed size embeddings by averaging the outputs or by using the output of the special token [CLS] (May, Wang, Bordia, Bowman, & Rudinger, 2019; Zhang, Kishore, Wu, Weinberger, & Artzi, 2019).

In this study, we employed an SBERT-encoder architecture devised by Reimers and Gurevych (2019) which separately generates embeddings for the input sentences. As shown in Fig. 3, a document summarization layer is added at the bottom of the SBERT-Encoder model. As discussed in the above subsection, documents are summarized using the extractive summarization technique to get the descriptive sentences as a summary of the document. Each summarized document is then considered as one single sentence and used as input to the SBERT-encoder. The output of the encoder is the fixed size dense vectors of the corresponding inputs. Fig. 4 gives the steps followed in the SBERT-encoder model to generate dense vectors.

As shown in Fig. 4, the document of size less than 512 WordPiece tokens are given as input to the encoder. The pre-trained BERT-base model generates a sequence of vectors of size 768 each learned through 12 layers of transformers. The output of the BERT is passed through the context aggregator layer to perform mean-pooling and generate the single representative dense vector for each input document. The model works in an offline mode i.e. before the arrival of the user query and generates the semantic dense vectors of each document in the dataset. These embeddings are stored apriori in the indexed dataset. When the IR system receives a user query, it passes through the same model to encode in the online mode and generate the query vector. This query vector is then matched against the stored document vectors using the cosine similarity function to retrieve the semantically closed documents.

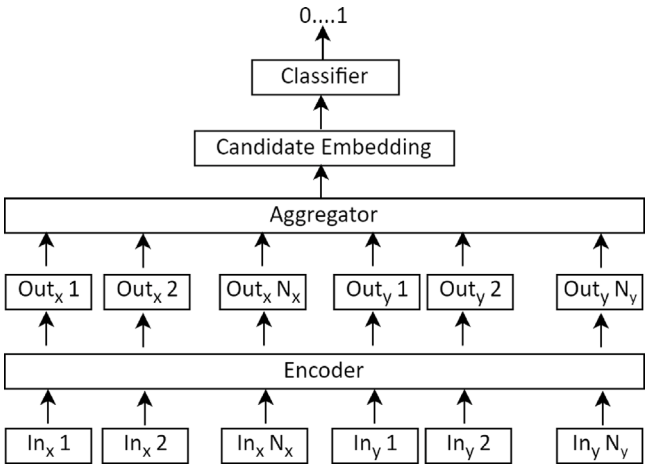


Fig. 5. Cross-Encoder model architecture.

3.2. Query expansion using phrase embedding model

In asymmetric search, the query used for retrieval usually is very short and thus, unable to clearly explain the information required. It becomes difficult for the retrieval model to make sense using the short query and retrieve relevant documents. Therefore, to enhance the clarity of the short queries this study used a Phrase Embedding model to expand the query with contextually similar words and phrases. As discussed in the paper (Bhopale & Tiwari, 2021), the model learns numerical representation of words and phrases in the same vector space using the Word2vec model trained on the desired dataset. The model generates embedding vectors for different words and phrases and uses a cosine similarity function to compare it against query term vectors. It generates the list of most similar words or phrases. In this study, we have used top three contextually similar words or phrases to expand query terms as mentioned in the original paper.

3.3. Re-ranking mechanism

The SBERT-encoder model generates embeddings of documents in a few minutes without rich interaction between the context input and the candidate inputs, as it takes only one input at a time. On the other hand, the BERT-based cross-encoder model takes 2 inputs at a time and performs all-to-all attention across tokens in the input sequence thereby yielding a richer interaction between the context and candidate input sequences. As a result, the cross-encoder architecture takes a longer duration to generate the representations and predict the similarity score. Also, cross-encoders need to recompute encoding each time for the input candidate and the context, as it does not generate separate embeddings of inputs. Therefore, in real-time when it is required to retrieve query-specific documents from the large collection in online mode, it takes a longer time compared to other models which store candidate embeddings apriori.

The SBERT-encoder model generates separate embeddings for documents and a query, and allows retrieval of the query-specific documents from a large document collection. The query and retrieved relevant document pair are then provided as input to BERT based cross-encoder model to predict scores and rank the final result set. Following Fig. 5 shows the architecture of the cross encoder model.

As shown in Fig. 5, the model uses the same encoder for inputs, therefore it performs self-attention between the candidate and context sequence tokens jointly, resulting in a richer text representation mechanism. As the candidate label can attend to the input context during the layers of the transformer, thereby the model produces candidate-sensitive input representation. An aggregator is used to perform pooling to convert multiple token vectors into a single vector known as a candidate embedding vector. The embedding result of the two inputs is passed to a linear classifier layer, which provides an output value between 0 and 1, indicating the degree of semantic similarity as:

$$\hat{y} = T(Q, RelDoc)W \tag{2}$$

where, $\hat{y}$ is the predicted value of semantic similarity degree, W is the linear layer weight, and T(Q, RelDoc) is the embedding result of the input query and relevant document retrieved for the query.

In the training process, the model is fine-tuned using the relevant query-document pairs and returns a score 1 if the semantically similar document is retrieved else returns 0. The study has used the classic cross-entropy loss function as shown in Eq. (3) to fine-tune the model.

$$Loss = -(y * log(\hat{y}) + (1 - y) * log(1 - \hat{y})) \tag{3}$$

where, y indicates the ground truth values about the relevancy of the retrieved result.

4. Experimental details

This section provides the experimental details and presents the findings of the proposed IR framework. We briefly discuss the dataset preparation, hyper-parameter setting, and the IR measures used for the performance evaluation. The findings of the proposed work and the comparative analysis are presented in the latter part of the section.

4.1. Experimental setup

All experiments are performed on an Intel(R) Xeon CPU with 32 GB RAM. All modules are implemented using the Python programming language. We utilized Elasticsearch[1] to index documents and its dense vector representations. A query vector is matched against the document vector to retrieve documents based on the similarity scores. The performance of the proposed work is evaluated by conducting experiments on two well-known datasets viz, TREC-CDS 2014 and OHSUMED, and the results are compared against the state-of-the-art methodologies.

4.2. Dataset collection and pre-processing

This study has used two well-known biomedical article datasets for experimentation and performance evaluation:

1. *TREC-Clinical Decision Support 2014 Dataset*[2] TREC is a document collection having 733,138 full-text articles in NXML format. The documents are extracted from PubMed Central open access subset repository. The dataset is provided with a set of 30 queries to evaluate the retrieval performance.
2. *OHSUMED Dataset*[3] OHSUMED is a collection of 348,566 records (54,710 training records and 293,856 testing records) obtained from Medline, an online medical information database. It includes title, abstract, and MeSH indexing words from 270 medical publications published during five years (1987–1991). The dataset is provided with 63 queries and the corresponding relevant judgments.

In both datasets, articles are available in the NXML format. Using the XML parser, we extracted the content of each document, including the article title, abstract, and keywords, and converted it to a text document for further use. Each dataset is provided with the set of queries and the corresponding relevant records i.e. *qrels* prepared by experts. The following Table 4 provides a summary of datasets.

[1] https://pypi.org/project/elasticsearch/.
[2] http://www.trec-cds.org/2014.html.
[3] https://trec.nist.gov/data/t9_filtering.html.

Table 4
Dataset statistics.

Name of the dataset	OHSUMED	TREC-CDS 2014
Type of the document	Medline articles	Pubmed articles
Document count	348,566	733,138
Query count	63	30
Total number of words	50,778	162,876

4.3. System effectiveness measures

The primary goal of IR evaluation measures is to quantify a user's satisfaction with the information required. It is believed that the focus of IR evaluation should be only on relevance. For the experimental evaluations, we used the *trec_eval* script which contains all necessary metrics. It compares the results with the provided query relevant document files, i.e., *qrels* prepared by corresponding experts. In the following, we discuss each metric in detail that is used in this study.

- *Precision (P):* The success of a system in the IR model is assessed by its ability to retrieve all and only relevant information. Precision assesses the system's ability to generate relevant records and the value is estimated based on how effectively the system eliminates irrelevant documents. It is defined as the portion of the relevant documents from the retrieved documents. Therefore, it specifies the ability of the system to generate relevant items. It is given as below:

$$P = \frac{\#\ Relevant\ Retrieved}{\#\ Total\ Retrieved} = \frac{tp}{tp + fp} \tag{4}$$

  where *tp* denotes true positive and *fp* denotes false positive. The precision metric can be applied to the ranked list by dividing it into the sets of top k documents, allowing to measure the quality of the ranked list using the simple metric P@k, which denotes precision at top k documents or relevant documents found in the top k retrieved documents. Calculating precision at multiple k values can be used to determine the capability of the system to efficiently retrieve objects at differing volumes. However, it fails to distinguish the quality of retrieval runs based on the ranks of relevant documents; for example, a system that retrieves a ranked list of relevant documents at 2,3,4 positions should be rated higher than a system that retrieves relevant documents at 6,7,8 positions. Therefore, in both cases, P@10 would be the 3/10.
- *Recall (R):* It assesses a system's accuracy and tracks how effectively a system retrieves what the user wants. It is the ratio of the number of relevant records retrieved to the total number of relevant records in the dataset. It is represented as below-

$$R = \frac{\#\ Relevant\ Retrieved}{\#\ Total\ Relevant\ in\ the\ Dataset} = \frac{tp}{tp + fn} \tag{5}$$

- *Mean Average Precision (MAP):* Precision and recall are single-value metrics that produce scores based on the entire document list returned by the system. It is preferable to consider the order in which the system has returned documents for ranking systems. As a result, MAP is a widely used metric in IR quality assessment. It combines precision on a ranked list and recall to provide a more reliable and robust statistic. After each relevant document is retrieved, the average precision (AP) for a particular query is generated by taking the average of precision values considered for the collection of top-k documents. The mean average precision (MAP) is calculated by averaging the AP over the query set. For every relevant document recovered at rank k, the precision at k_{th} rank is added, which tends to favor the retrieval outcomes with higher recall and relevant documents retrieved at lower (better) ranks.
  For a query q_i from the set of queries Q, let R_i be the relevant documents retrieved by the system.
  Let, $P(R_i[k])$ is the precision calculated until $R_i[k]$ is observed in the ranking. If the kth-ranked document is not retrieved, then $R_i[k] = 0$ for that k value.
  Hence the average precision score for query q_i is calculated as-
  Let $P(R_i[k])$ is the precision calculated until $R_i[k]$ is observed in the ranking. If the kth-ranked document is not retrieved, then $R_i[k] = 0$ for that k value.
  Hence the average precision score for query q_i is calculated as-

$$AvgP_i = \frac{1}{|R_i|} \sum_{k-1}^{|R_i|} P(R_i[k]) \tag{6}$$

  And the mean average precision is given as-

$$MAP = \frac{1}{|Q|} \sum_{i}^{|Q|} AvgP_i \tag{7}$$

  Coming back to the example discussed in *Precision*, the AP of the ranked list 2,3,4 is 1/3(1/2 + 2/3 + 3/4) = 0.638, whereas the AP of ranked list 6,7,8 is 1/3(1/6 + 2/7 + 3/8) = 0.276. These two AP values clearly indicates that, MAP prefers ranked list with more relevant documents that are ranked early.
- *Rprec:* After obtaining R relevant records, this metric produces the precision number for any particular query. In general, both precision and recall are equal at the *R*th point. Rprec and MAP are frequently significantly linked. The following is how Rprec is defined:

$$R - prec = \frac{|r|}{R} \tag{8}$$

  where, $|r|$ - is the number of relevant documents retrieved amongst the top R documents. R – is the total number of relevant documents for any specific query q.
- *Binary Preference (bpref):* This score is determined by the number of relevant documents obtained so far. It is solely based on the relative ranks of judged documents and does not consider the rank of retrieved documents. It is calculated as follows:

$$bpref = \frac{1}{R} \sum_{r} 1 - \frac{n_r}{R} \tag{9}$$

  Here, n_r is the rank of the first irrelevant retrieved document amongst judged documents, which is greater than the rank of a relevant and retrieved document r. R is the total number of judged documents. For example, when *bpref* = 70% then 30% irrelevant documents are retrieved before relevant document. Both *bpref* and *MAP* are highly correlated when used with entire results. The *bpref* returns the correct ranking for a single query as well as incomplete judgments, whereas *MAP* does not.
- *Normalized Discounted Cumulative Gain (nDCG):* It measures the usefulness of ranked documents based on the ranking positions. The relevant documents ranked at a higher rank increase the score of the metric. This metric also allows introducing non-binary relevance measures such as not relevant, relevant, and very relevant. In this research, we only considered the binary relevance measure. The nDCG@k can be calculated as below:

$$nDCG_k = \frac{DCG_k}{IDCG_k} \tag{10}$$

  where DCG is the discounted cumulative gain and IDCG is the Ideal Discounted Cumulative Gain calculated as below:

$$DCG_k = \sum_{i=1}^{k} \frac{relevance_i}{log_2(i+1)} \tag{11}$$

$$IDCG_k = \sum_{i=1}^{k} \frac{truth_i}{log_2(i+1)} \tag{12}$$

Table 5
Quality of retrieved documents in terms of P@5, P@10, MAP, Rprec, and nDCG@100 metrics for TREC-CDS 2014 dataset. W/o-Summ = Without Summarization, W-Summ = With Summarization, LS = Lexical Search, SS = Semantic Search, QE = Query Expansion using Phrase Embedding model.

Approach		P@5	P@10	MAP	bpref	Rprec	nDCG
W/o-Summ	LS (BM25F)	0.3533	0.3600	0.1446	0.3866	0.3636	0.3230
	QE + BM25F	0.5133	0.5333	0.2899	0.5462	0.5416	0.4486
	SS	0.5333	0.5533	0.3042	0.5614	0.5534	0.4602
	QE + SS	0.5400	0.5700	0.3368	0.5755	0.5511	0.4789
W-Summ	SS	0.5200	0.5533	0.3145	0.5614	0.5570	0.4666
	QE + SS	**0.5400**	**0.5667**	**0.3480**	**0.5676**	**0.5617**	**0.4991**

Table 6
Quality of retrieved documents in terms of P@5, P@10, MAP, Rprec, and nDCG@100 metrics for OHSUMED dataset. W/o-Summ = Without Summarization, W-Summ = With Summarization, LS = Lexical Search, SS = Semantic Search, QE = Query Expansion using Phrase Embedding model.

Approach		P@5	P@10	MAP	bpref	Rprec	nDCG
W/o-Summ	LS (BM25F)	0.3714	0.3651	0.1762	0.3901	0.3745	0.3775
	QE + BM25F	0.5079	0.5175	0.3095	0.5494	0.5342	0.5344
	SS	0.5206	0.5333	0.3313	0.5640	0.5489	0.5492
	QE + SS	0.5365	0.5476	0.3520	0.5780	0.5609	0.5645
W-Summ	SS	0.5238	0.5349	0.3346	0.5669	0.5523	0.5518
	QE + SS	**0.5333**	**0.5492**	**0.3551**	**0.5819**	**0.5663**	**0.5663**

4.3.1. Parameter selection

The study has used the pre-trained stsb-mpnet-base-v2 model which is a part of Sentence Transformer models. It is available on the Huggingface transformer package which is a python library hosts pre-trained models. The stsb-mpnet-base-v2 model has used Wikipedia and BookCorpus datasets of size 160 GB for model training. The model follows the architecture and the parameters of 12 layers BERT-base model. As discussed in Section 3.1.1, we conducted experiments to fix the size of the input document length to 300 Wordpiece tokens. Each document and query will be represented in the dense vector of size 768, and the BERT-base model uses 12 transformer layers and 12 attention heads.

4.4. Experimental results and discussion

As discussed in the above subsection, we used the pre-trained model and fine-tuned it on the desired datasets. The experiments are conducted to mainly investigate how the deep semantic text representation technique enhances the performance of the IR model. It is also interesting to evaluate the performance of the model when integrated with the phrase embedding-based query expansion approach. We therefore experimented and compared the performance of the presented model in different scenarios such as the performance analysis with and without document summarization along with the original and the expanded query. The presented model has generated results in two phases. In the first phase, the relevant documents w.r.t. query are retrieved which are then re-ranked in the next phase. To perform a lexical search a BM25 retrieval function is used.

The Tables 5 & 6 present the experimental results on two varying size datasets in terms of P@5, P@10, bpref, Rprec, MAP, and nDCG measures. Values generated by the proposed framework are highlighted in bold. Retrieval results are measured in terms of the mean average precision value generated by each approach. It is observed that the proposed approach has shown significant improvement over the other approaches. From the results it can be depicted that the summarization technique has retained the important descriptive features, therefore has shown improvements over without summarization approaches. Fig. 6 gives the graphical comparison of average retrieval time taken by different approaches to retrieve the top 100 documents per query. It is observed that, the time taken by query expansion is approximately 3x times more compared to the time for the original query. Also, it shows that the retrieval time is directly proportional to the size of the dataset.

Table 7
Ranking performance at different nDCG values for TREC-CDS 2014 dataset. W/o-Summ = Without Summarization, W-Summ = With Summarization, LS = Lexical Search, SS = Semantic Search, QE = Query Expansion using Phrase Embedding model.

Approach		nDCG@10	nDCG@20	nDCG@30	nDCG@100
W/o-Summ	LS (BM25F)	0.2212	0.2414	0.2653	0.3230
	QE + BM25F	0.3288	0.364	0.3924	0.4486
	SS	0.3414	0.3751	0.4046	0.4602
	QE + SS	0.3490	0.3874	0.418	0.4789
W-Summ	SS	0.3392	0.377	0.4062	0.4666
	QE + SS	**0.3675**	**0.3927**	**0.4301**	**0.4991**

Table 8
Ranking performance at different nDCG values for OHSUMED dataset. W/o-Summ = Without Summarization, W-Summ = With Summarization, LS = Lexical Search, SS = Semantic Search, QE = Query Expansion using Phrase Embedding model.

Approach		nDCG@10	nDCG@20	nDCG@30	nDCG@100
W/o-Summ	LS (BM25F)	0.3618	0.3592	0.3647	0.3775
	QE + BM25F	0.5067	0.5206	0.5233	0.5344
	SS	0.5222	0.5350	0.5380	0.5492
	QE + SS	0.5386	0.5523	0.5554	0.5645
W-Summ	SS	0.5240	0.5373	0.5390	0.5518
	QE + SS	**0.5382**	**0.5503**	**0.5511**	**0.5663**

Ranking Performance:

As in the proposed work, the framework not only retrieves the relevant documents but also performs ranking on the result set generated by the retrieval model. Therefore, to measure the ranking performance the study has used the nDCG metric. The Tables 7 & 8 gives the nDCG scores measured at different values such as 10, 20, 30, and 100 to determine the performance of the ranking model in the top 10, 20, 30, and 100 documents respectively.

Impact of Query Expansion Technique:

In the presented study, a phrase embedding-based query expansion is incorporated. We used top-3 phrases/words to expand the original query. Significant improvements in the quality of results are observed using the expanded query as shown in the Tables 5 & 6. It is observed that the quality is improved on the cost of more time compare to the original query. The query expansion technique enhances short queries with more contextually similar words/phrases which are extracted from the same dataset using the phrase embedding model, thereby helping to clearly interpret the short queries.

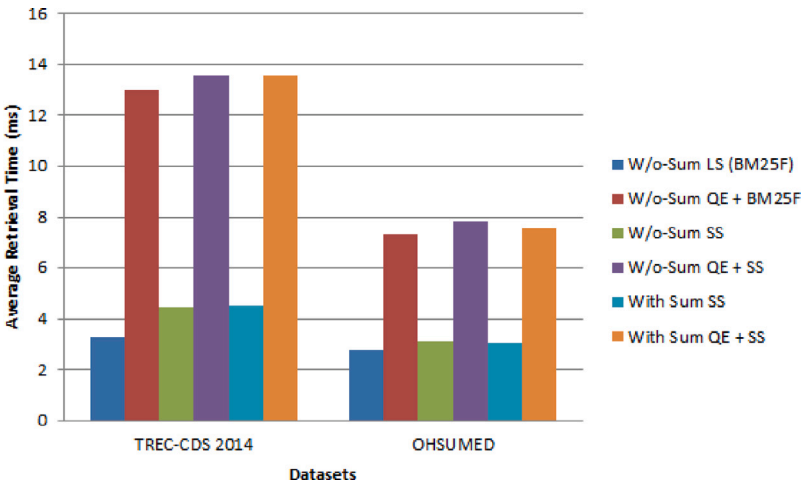


Fig. 6. Average retrieval time in ms taken by each query to retrieve top 100 documents.

The results demonstrate that the proposed BERT-based IR model can capture different nuance of word meaning with the given surrounding text. As a result, it creates different embedding representations of the word in various contexts. This BERT model feature facilitates in the resolution of polysemy word issues. In the proposed work, we also used a query expansion model to address the synonymy word problem by expanding short queries with contextually similar words or phrases. As a result, they can help to bridge the semantic gap between short queries and long documents.

As discussed above, the proposed work has several advantages over the state-of-the-art work in various aspects; however, it has a few limitations due to the length of the documents to be considered in the retrieval process. The summarization technique used in this work considers only the top sentences in each document, which may impede performance by omitting some important descriptive features. Improvements could be made further by utilizing various parameters such as click logs, user behavior, and page ranking mechanisms, which are not taken into account in this work. In the following section, we compare the proposed work with the state-of-the-art IR techniques.

4.5. Comparative analysis with state-of-the-art IR frameworks

We used MAP and nDCG metrics for comparing the performances of all approaches studied in the past. The Tables 9, and 10 gives the MAP values of various IR approaches against the proposed study. The bold value in each table indicates the values generated by the proposed approach.

As shown in the Table 9 for TREC-CDS 2014 dataset, the MAP value generated by the proposed model is better compared to other studies. However, the nDCG value is competitive as compared to studies presented in Bhopale and Tiwari (2021) ad Malik, Shoaib, El-Sayed, and Khan (2020). For the OHSUMED dataset in Table 10, the proposed approach has outperformed over all other approaches in terms of MAP and nDCG metrics. In general, the proposed transformer-based contextual representation model has contributed in the modeling of polysemy words by generating distinct representations in different contexts and therefore helps in ranking relevant documents at the top positions in the list. While modeling synonymy using the phrase embedding query expansion model helps to retrieve a higher number of relevant documents which contain synonyms of the query terms. Therefore, the comparison study inferred that the proposed transformer-based contextual representation model produced superior results than the embedding-based and data mining approaches when integrated with the phrase embedding model and the summarization technique.

5. Conclusion and future scope

In this paper, we presented a transformer-based representation framework for the deeper text understanding in the IR task. The framework includes BERT based two-tower transformer encoder model for generating query and document embeddings separately and retrieving query-relevant documents. A BERT-based document summarization model is used to address the maximum sequence length limitation while retaining important features, and summarized documents are provided as input to the encoder model. We integrated the Phrase embedding-based query expansion technique to expand short and unclear queries with contextually similar words and phrases. The semantic indexing is built using the Elasticsearch engine to retrieve the query-relevant documents. The results are then re-ranked using the BERT-based cross-encoder model. We conducted extensive experiments on two well-known varying-size datasets to evaluate the performance of the proposed framework. The proposed methodology has shown significant improvements in the retrieval performance in both offline and online experiments.

For future work, we would like to study a more generic approach that will encode the entire document for embeddings rather than the top few descriptive sentences. It is also interesting to explore the impact of recent year encoding models in the field of IR.

CRediT authorship contribution statement

Amol P. Bhopale: Conceptualization, Methodology, Software, Validation, Investigation, Data curation, Writing – original draft, Visualization. **Ashish Tiwari:** Conceptualization, Methodology, Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article

Table 9
Comparison of Mean Average Precision (MAP) and Normalized Discounted Cumulative Gain (nDCG) value generated by the proposed approach and other existing approaches on the TREC-CDS 2014 dataset.

Approach	MAP	nDCG@100
Document-based Neural Relevance Model (DNRM) (Ran et al., 2017)	0.1578	0.2558
PRF + Embedding-based technique (Yang et al., 2017)	0.1661	0.2830
Self-adaptive SDF for Probabilistic IR (Marchesin et al., 2019)	0.1883	0.3329
NLP-based feature weighting clustering (Song et al., 2019)	0.2807	0.4269
Semantic-Aware Neural Framework for IR (SAFIR) (Agosti, Marchesin, & Silvello, 2020)	–	0.1772
QE using clinical diagnosis information (CDI) & MeSH (Malik et al., 2020)	0.2409	0.4975
K-means IR (Bhopale & Tiwari, 2020)	0.2230	0.3606
Swarm Optimized Cluster-based (SOCB) IR (Bhopale & Tiwari, 2020)	0.2652	0.4209
CDI + Word Embedding based QE (Malik, Shoaib, Bukhari, El Sayed, & Khan, 2022)	0.1258	0.3497
Phrase Embedding-based QE for IR (Bhopale & Tiwari, 2021)	0.3095	0.5067
W-Summ + QE + SS	**0.3480**	**0.4991**

Table 10
Comparison of Mean Average Precision (MAP) and Normalized Discounted Cumulative Gain (nDCG) value generated by the proposed approach and other existing approaches on OHSUMED Dataset.

Approach	MAP	nDCG@100
K-medoids cluster-based Query Expansion (Khennak, Drias, Kechid, & Moulai, 2019)	0.1850	–
PSO-based BM25 IR (Thakur, Mehrotra, Bansal, & Bala, 2019)	0.1900	–
Fuzzy K-means & Firefly + APSO based QE (Khennak & Drias, 2018)	0.1840	–
Deep Relevance Matching Model DRMM (Marchesin, Purpura, & Silvello, 2020)	0.2720	0.4070
Neural Vector Space Model NVSM (Marchesin et al., 2020)	0.2140	0.3350
Firefly based PRF (Khennak & Drias, 2016)	0.1540	–
APSO based QE (Khennak & Drias, 2017)	0.1840	–
Semantic-Aware Neural Framework for IR (SAFIR) (Agosti et al., 2020)	–	0.4948
K-means IR (Bhopale & Tiwari, 2020)	0.2705	0.4977
Swarm Optimized Cluster-based (SOCB) IR (Bhopale & Tiwari, 2020)	0.3026	0.5286
Phrase Embedding-based QE for IR (Bhopale & Tiwari, 2021)	0.3095	0.5344
W-Summ + QE + SS	**0.3551**	**0.5663**

References

Agosti, M., Marchesin, S., & Silvello, G. (2020). Learning unsupervised knowledge-enhanced representations to reduce the semantic gap in information retrieval. *ACM Transactions on Information Systems (TOIS)*, *38*(4), 1–48.

ALMasri, M., Berrut, C., & Chevallet, J.-P. (2016). A comparison of deep learning based query expansion with pseudo-relevance feedback and mutual information. In *European conference on information retrieval* (pp. 709–715). Springer.

Bai, X., Ordentlich, E., Zhang, Y., Feng, A., Ratnaparkhi, A., Somvanshi, R., et al. (2018). Scalable query n-gram embedding for improving matching and relevance in sponsored search. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining* (pp. 52–61).

Bengio, Y., Ducharme, R., Vincent, P., & Janvin, C. (2003). A neural probabilistic language model. *The Journal of Machine Learning Research*, *3*, 1137–1155.

Bhopale, A. P., & Tiwari, A. (2020). Swarm optimized cluster based framework for information retrieval. *Expert Systems with Applications*, *154*, Article 113441.

Bhopale, A. P., & Tiwari, A. (2021). Leveraging neural network phrase embedding model for query reformulation in ad-hoc biomedical information retrieval. *Malaysian Journal of Computer Science*, *34*(2), 151–170.

Blacoe, W., & Lapata, M. (2012). A comparison of vector-based representations for semantic composition. In *Proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural language learning* (pp. 546–556).

Carpineto, C., & Romano, G. (2012). A survey of automatic query expansion in information retrieval. *ACM Computing Surveys (CSUR)*, *44*(1), 1–50.

Collobert, R., & Weston, J. (2008). A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on machine learning* (pp. 160–167).

da Silva, F. T., & Maia, J. E. (2019). Query expansion in text information retrieval with local context and distributional model. *Journal of Digital Information Management*, *17*(6), 313.

Dai, Z., & Callan, J. (2019). Deeper text understanding for IR with contextual neural language modeling. In *Proceedings of the 42nd international ACM SIGIR conference on research and development in information retrieval* (pp. 985–988).

Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., & Harshman, R. (1990). Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, *41*(6), 391–407.

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

El Mahdaouy, A., El Alaoui, S. O., & Gaussier, E. (2018). Improving arabic information retrieval using word embedding similarities. *International Journal of Speech Technology*, *21*(1), 121–136.

Ganguly, D., Roy, D., Mitra, M., & Jones, G. J. (2015). Word embedding based generalized language model for information retrieval. In *Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval* (pp. 795–798).

Grbovic, M., Djuric, N., Radosavljevic, V., Silvestri, F., Baeza-Yates, R., Feng, A., et al. (2016). Scalable semantic matching of queries to ads in sponsored search advertising. In *Proceedings of the 39th international ACM SIGIR conference on research and development in information retrieval* (pp. 375–384).

Hui, K., Yates, A., Berberich, K., & De Melo, G. (2018). Co-PACRR: A context-aware neural IR model for ad-hoc retrieval. In *Proceedings of the eleventh ACM international conference on web search and data mining* (pp. 279–287).

Khennak, I., & Drias, H. (2016). A firefly algorithm-based approach for pseudo-relevance feedback: Application to medical database. *Journal of Medical Systems*, *40*(11), 240.

Khennak, I., & Drias, H. (2017). An accelerated PSO for query expansion in web information retrieval: application to medical dataset. *Applied Intelligence*, *47*(3), 793–808.

Khennak, I., & Drias, H. (2018). Data mining techniques and nature-inspired algorithms for query expansion. In *Proceedings of the international conference on learning and optimization algorithms: Theory and applications* (pp. 1–6).

Khennak, I., Drias, H., Kechid, A., & Moulai, H. (2019). Clustering algorithms for query expansion based information retrieval. In *International conference on computational collective intelligence* (pp. 261–272). Springer.

Kontostathis, A. (2007). Essential dimensions of latent semantic indexing (LSI). In *2007 40th annual hawaii international conference on system sciences (HICSS'07)* (p. 73). IEEE.

Lavrenko, V., & Croft, W. B. (2001). Relevance-based language models: Estimation and analysis. *Croft and Lafferty [2]*, 1–5.

Li, M., Lu, Q., Xiong, D., & Long, Y. (2018). Phrase embedding learning based on external and internal context with compositionality constraint. *Knowledge-Based Systems*, *152*, 107–116.

Li, C., Sun, Y., He, B., Wang, L., Hui, K., Yates, A., et al. (2018). NPRF: A neural pseudo relevance feedback framework for ad-hoc information retrieval. arXiv preprint arXiv:1810.12936.

Liu, Y., & Lapata, M. (2019). Text summarization with pretrained encoders. arXiv preprint arXiv:1908.08345.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., et al. (2019). Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692.

Ma, X., Guo, J., Zhang, R., Fan, Y., Li, Y., & Cheng, X. (2021). B-PROP: bootstrapped pre-training with representative words prediction for ad-hoc retrieval. In *Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval* (pp. 1513–1522).

Malik, S., Shoaib, U., Bukhari, S. A. C., El Sayed, H., & Khan, M. A. (2022). A hybrid query expansion framework for the optimal retrieval of the biomedical literature. *Smart Health*, *23*, Article 100247.

Malik, S., Shoaib, U., El-Sayed, H., & Khan, M. A. (2020). Query expansion framework leveraging clinical diagnosis information ontology. In *2020 14th international conference on innovations in information technology (IIT)* (pp. 18–23). IEEE.

Marchesin, S., Purpura, A., & Silvello, G. (2019). Focal elements of neural information retrieval models. An outlook through a reproducibility study. *Information Processing & Management*, Article 102109.

Marchesin, S., Purpura, A., & Silvello, G. (2020). Focal elements of neural information retrieval models. An outlook through a reproducibility study. *Information Processing & Management*, *57*(6), Article 102109.

May, C., Wang, A., Bordia, S., Bowman, S. R., & Rudinger, R. (2019). On measuring social biases in sentence encoders. arXiv preprint arXiv:1903.10561.

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781.

Nalisnick, E., Mitra, B., Craswell, N., & Caruana, R. (2016). Improving document ranking with dual word embeddings. In *Proceedings of the 25th international conference companion on world wide web* (pp. 83–84).

Nogueira, R., & Cho, K. (2019). Passage re-ranking with BERT. arXiv preprint arXiv:1901.04085.

Nogueira, R., Yang, W., Cho, K., & Lin, J. (2019). Multi-stage document ranking with bert. arXiv preprint arXiv:1910.14424.

Pennington, J., Socher, R., & Manning, C. D. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)* (pp. 1532–1543).

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., et al. (2018). Deep contextualized word representations. arXiv preprint arXiv:1802.05365.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al. (2019). Language models are unsupervised multitask learners. *OpenAI Blog*, *1*(8), 9.

Ran, Y., He, B., Hui, K., Xu, J., & Sun, L. (2017). A document-based neural relevance model for effective clinical decision support. In *2017 IEEE international conference on bioinformatics and biomedicine (BIBM)* (pp. 798–804). IEEE.

Reimers, N., & Gurevych, I. (2019). Sentence-bert: Sentence embeddings using siamese bert-networks. arXiv preprint arXiv:1908.10084.

Robertson, S., & Hull, D. A. (2000). The TREC-9 filtering track final report. In *TREC, Vol. 10* (pp. 344250–344253). Citeseer.

Salton, G., & McGill, M. J. (1983). *McGraw-Hill computer science series*, *Introduction to modern information retrieval*. McGraw-Hill, URL: https://books.google.co.in/books?id=7f5TAAAAMAAJ.

Salton, G., Wong, A., & Yang, C.-S. (1975). A vector space model for automatic indexing. *Communications of the ACM*, *18*(11), 613–620.

Socher, R., Manning, C. D., & Ng, A. Y. (2010). Learning continuous phrase representations and syntactic parsing with recursive neural networks. In *Proceedings of the NIPS-2010 deep learning and unsupervised feature learning workshop, Vol. 2010* (pp. 1–9).

Song, Y., Hu, W., He, L., & Dou, L. (2019). Enhancing the healthcare retrieval with a self-adaptive saturated density function. In *Pacific-Asia conference on knowledge discovery and data mining* (pp. 501–513). Springer.

Thakur, N., Mehrotra, D., Bansal, A., & Bala, M. (2019). A novel multi-parameter tuned optimizer for information retrieval based on particle swarm optimization.

Wiemer-Hastings, P., Wiemer-Hastings, K., & Graesser, A. (2004). Latent semantic analysis. In *Proceedings of the 16th international joint conference on artificial intelligence* (pp. 1–14). Citeseer.

Wu, Z., Mao, J., Liu, Y., Zhan, J., Zheng, Y., Zhang, M., et al. (2020). Leveraging passage-level cumulative gain for document ranking. In *Proceedings of the web conference 2020* (pp. 2421–2431).

Xiong, C., Dai, Z., Callan, J., Liu, Z., & Power, R. (2017). End-to-end neural ad-hoc ranking with kernel pooling. In *Proceedings of the 40th international ACM SIGIR conference on research and development in information retrieval* (pp. 55–64).

Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R. R., & Le, Q. V. (2019). Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in neural information processing systems, Vol. 32*.

Yang, C., He, B., Li, C., & Xu, J. (2017). A feedback-based approach to utilizing embeddings for clinical decision support. *Data Science and Engineering*, *2*(4), 316–327.

Yilmaz, Z. A., Yang, W., Zhang, H., & Lin, J. (2019). Cross-domain modeling of sentence-level evidence for document retrieval. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)* (pp. 3490–3496).

Yin, W., & Schütze, H. (2014). An exploration of embeddings for generalized phrases. In *Proceedings of the ACL 2014 student research workshop* (pp. 41–47).

Yu, M., & Dredze, M. (2015). Learning composition models for phrase embeddings. *Transactions of the Association for Computational Linguistics*, *3*, 227–242.

Zamani, H., & Croft, W. B. (2017). Relevance-based word embedding. In *Proceedings of the 40th international ACM SIGIR conference on research and development in information retrieval* (pp. 505–514).

Zhang, T., Kishore, V., Wu, F., Weinberger, K. Q., & Artzi, Y. (2019). Bertscore: Evaluating text generation with bert. arXiv preprint arXiv:1904.09675.

Zheng, G., & Callan, J. (2015). Learning to reweight terms with distributed representations. In *Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval* (pp. 575–584).

Zuccon, G., Koopman, B., Bruza, P., & Azzopardi, L. (2015). Integrating and evaluating neural word embeddings in information retrieval. In *Proceedings of the 20th australasian document computing symposium* (pp. 1–8).