A Comparative Analysis of Sentence Embedding Techniques for Document Ranking

Vishal Gupta[1,2,*], Ashutosh Dixit[1] and Shilpa Sethi[1]

[1]*J.C. Bose University of Science & Technology, YMCA, Faridabad, Haryana, India*
[2]*MMEC, MM(DU), Mullana, Ambala, Haryana, India*
E-mail: sahara.vishal@gmail.com
[*]*Corresponding Author*

Received 23 August 2022; Accepted 25 November 2022;
Publication 28 December 2022

Abstract

Due to the exponential increase in the information on the web, extracting relevant documents for users in a reasonable time becomes a cumbersome task. Also, when user feedback is scarce or unavailable, content-based approaches to extract and rank relevant documents are critical as they suffer from the problem of determining semantic similarity between texts of user queries and documents. Various sentence embedding models exist today that acquire deep semantic representations through training on a large corpus, with the goal of providing transfer learning to a broad range of natural language processing tasks such as document similarity, text summarization, text classification, sentiment analysis, etc. So, in this paper, a comparative analysis of six pre-trained sentence embedding techniques has been done to identify the best model suited for document ranking in IR systems. These are SentenceBERT, Universal Sentence Encoder, InferSent, ELMo, XLNet, and Doc2Vec. Four standard datasets CACM, CISI, ADI, and Medline are used to perform all the experiments. It is found that Universal Sentence Encoder and SentenceBERT outperform other techniques on all four datasets in terms of MAP, recall,

Journal of Web Engineering, Vol. 21_7, 2149–2186.
doi: 10.13052/jwe1540-9589.2177

F-measure, and NDCG. This comparative analysis offers a synthesis of existing work as a single point of entry for practitioners who seek to use pre-trained sentence embedding models for document ranking and for scholars who wish to undertake work in a similar domain. The work can be expanded in many directions in the future as various researchers can combine these strategies to build a hybrid document ranking system or query reformulation system in IR.

Keywords: BERT, cosine similarity, document ranking, information retrieval, sentence embedding.

1 Introduction

A huge amount of data is uploaded on the internet every year. As a result, speculating documents that meet the user's requirements becomes a time-consuming process. Moreover, the length of the user query is too small, unstructured, incomplete, or imprecise, which is not enough to explain the needs of the user. The average length of the query according to [1] was 2.30 words, which is the same as recorded several years ago [2]. Various researchers proposed machine learning models based on different techniques such as bag-of-words like BM25 [3] or query likelihood [4] and supplemented representation of text with n-grams [5], restricted vocabularies [6], and QR (Query Reformulation) [7] to extract the relevant information from the web. But the resultant information is still prone to the issues such as topic drift, vocabulary mismatch, and low precision results. The retrieved information coupled with unwanted results forces the user to either change the query terms or sift through a long list of documents resulting in the problem of Information Overkill. The main reason behind the problem is speculating the documents based upon inadequate query terms at the database level and correspondingly ranking them. Although, there exist techniques that either contextually reformulates the user query [8–10] or contextually rank the documents using sentence embeddings [11–13], but no technique combined both aspects simultaneously. Thus, there is a need for a combined approach that can express the user's information needs at the level of query submission as well as document ranking. Before making an amalgam of query reformulating and document ranking techniques, one of the main challenges is to decide which best techniques to proceed with and how the performance boosting of search results is validated. In this paper, a comparative analysis of six sentence embedding models has been done to find the best technique to solve the problem of Information Overkill.

Embedding is defined as the procedure for mapping discrete variables to a low dimension continuous vector [14]. Early embedding approaches only worked with words wherein the embedding for each word in the corpus is created [14–16]. The easiest way was the one-hot encoding of the sequence of words; assigning a 1 to each word and a 0 to the others. While this worked well for expressing words as well as various simple text-processing operations, it underperformed for complicated tasks such as inferring the different meanings of a word in different contexts, the same meaning from different words, locating related words in a sphere, etc. For example, if a user enters the query "top Chinese hotel in Agra," he is looking for results that include terms like "Chinese food," "hotels in Agra," and "top." The conventional techniques such as One-hot encoding, Bag-of-Words, TF-IDF (Term Frequency-Inverse Document Frequency), etc. will not identify the resemblance between 'top' and 'best' or the correlation between 'food' and 'hotel' and thus, he will not receive a result corresponding to "best Chinese food in Agra". In this case, word embeddings can help solve the problem. In essence, word embedding creates a vector representation of the data by not only converting the word but also recognizing its semantics. Word2Vec [17], GloVe [18], ELMo [19], and FastText [20] are a few prominent word embedding techniques to mention here.

The basic idea is to gain knowledge from the words in the neighbourhood. Researchers had discovered new ways to show incremental information on the words, resulting in ground-breaking advancements in word embedding techniques. What if we could work directly with individual phrases rather than individual words? Using merely words in a huge text would be cumbersome, and the amount of information we could retrieve from the word embeddings is restricted. Let us say we read the line "I hate packed locations," and then read "But I enjoy one of India's crowded city Maharashtra." How can we get the query engine to make the connection between "packed locations" and "crowded city"? Clearly, word embedding would be inadequate in this case, thus sentence embedding can be applied to make sensible inferences in the aforementioned scenario. Analogous to word embedding, sentence embedding systems can also use vectors to represent complete sentences and their semantics. This helps the search engine to understand the context, intent, and other aspects of the user text. Document ranking against a user's query based on sentence embedding is a favoured research topic that also offers new ways to help machines grasp our language.

In recent years, sentence encoders such as Google's BERT and USE, Facebook's InferSent, and AllenAI's SciBERT and ELMo have gotten a lot of attention. A sentence can be encoded into deep contextualized embeddings

using these pre-trained machine learning models. For several NLP (Natural Language Processing) tasks, they have been shown to outperform earlier state-of-the-art approaches such as Word2Vec, GloVe, ELMo, etc. [21–25]. Calculating semantic similarity and relatedness, which is crucial in constructing efficient Information Retrieval systems, is one of these jobs. Work on sentence encoders has previously focused on a variety of domains, including social media posts [26], news [26, 27], and web pages [27]. Our goal is to see how well some of the most popular sentence encoders do when it comes to document relatedness and ranking.

1.1 Motivation of Work

The motivation behind the study is to thoroughly demonstrate sentence embedding models to deal with the Information Overkill problem in IR (Information Retrieval). The article investigates the applicability of existing pre-trained sentence embedding models in determining document relevance with respect to a user query and their comparative analysis to best fit in different setups. The distinct motivation for this comprehensive survey is as follows:

- To study existing machine learning techniques for document ranking to deal with the Information Overkill problem in IR.
- To analyze the applicability of embedding techniques in different phases of search engines and demonstrate how the search engine results improve by using them.
- To perform the comprehensive and comparative analysis of recent pre-trained embedding models.

To understand the need for embeddings and their applications in IR, it is important to explore recent embedding techniques and perform a comparative analysis. It will open new directions for the researchers to propose new methods to deal with the Information Overkill problem in IR and improve the performance of search engines.

1.2 Research Contributions

A comprehensive review has been conducted to investigate various sentence embedding models for improving the performance of document ranking systems in IR. The research methodology in Section 2 is designed to study and compare various sentence embedding models. This is done using guidelines based on SLR (Systematic Literature Review) proposed in [28]. The

papers we have referred to are from 2001 to 2022 and were written in the English language. The main sources for paper consideration include Springer, ACM, IEEE, Google Scholar, Elsevier, and ScienceDirect. Supplemental literature from other journals has also been selected. The keywords such as "sentence embedding models for IR", "BERT", and "Document ranking in IR" have been considered in searching the relevant papers. Articles that have utilized modalities like Information Retrieval, deep learning, cosine similarity, USE etc. have also been considered. The conditions for the selection of an article include an article that must be focused on sentence embedding models for IR and used recent techniques like neural networks, evolutionary computing, and fuzzy logic for document ranking with good indexing and high citations. The conditions for rejection of an article include articles written in languages other than English, duplicate articles, articles related to other fields of IR like indexing and crawling, articles not available in full text, and low citation index. The recent machine learning techniques that deal with the Information Overkill problem in IR are studied and demonstrated thoroughly. Existing sentence embedding models, their limitations, and their benefits are summarised in tabular format. Sentence embedding helps in solving the problem of missing information related to user query context. The comparative analysis among the existing sentence embedding models based on MAP, F-Score, recall, and NDCG on four standard datasets namely, Medline, CACM (Communications of the ACM), CISI (Centre for Inventions and Scientific Information), and ADI has been carried and presented in Section 5.

1.3 Article Organization

Section 1 presents an introduction to the embedding techniques and the motivation behind them. Section 2 covers a Systematic Literature Review (SLR) related to the document relevance problem in IR. The issues and challenges in the existing literature are identified and listed in Section 3. Section 4 thoroughly covers recent sentence embedding models for improving document ranking systems in IR. A comparative analysis of six sentence embedding models on different metrics is done in Section 5. Section 6 concludes the findings of our study and future scope.

2 Literature Review

Document irrelevance is a major issue while presenting the search results in response to a user query in IR systems [29]. Conventional IR systems

were based on bag-of-words like BM25 [3] or query likelihood [4] and supplemented the representation of text with n-grams [5], restricted vocabularies [6], and query reformulation [7]. Recently, machine learning approaches such as query reformulation on bag-of-sparse-features [30, 31], balancing terms scores using BERT [32, 33], and adding terms to the document with sequence-to-sequence models [34] can significantly improve the quality of search results. However, these methods still rely on the lexical retrieval framework, and thus, they could attain only a little improvement in semantic inference [35].

Utilizing dense text representations, neural networks perform better for semantic matching. Neural network models for IR may be divided into two categories [36, 37] namely, interaction-based and representation-based models. Models built on interactions between word pairs in queries and documents are known as interaction-based models. These models primarily learn local interactions between query and document, and then employ deep neural networks to learn hierarchical interaction patterns among them. The trained model is utilized for document reranking, but it may be prohibitively expensive for initial-phase extraction. The major disadvantage of interaction-based approaches is the loss of semantic information during the transformation of documents and queries to similarity matrices. In contrast, representation-based models train a unique vector for the query or document and quantify their relevance using weighing functions such as cosine similarity, okapi BM25 (BM stands for Best Matching), dice, etc. [37]. Further, Latent Semantic Indexing (LSI) [38], Siamese networks [39], and MatchPlus [40] can be traced to representation-based neural retrieval models. The work in [41] and [42] employed BERT-based retrieval to discover transits for question-answering, while [43] proposed a deck of pre-training tasks for phrase extraction. Azad et al. [44] presented a survey paper on various query reformulation techniques for document ranking which can improve the performance of IR systems. Various sentence embedding models exist today that acquire deep semantic representations through training on a large corpus, with the goal of providing transfer learning to a broad range of NLP tasks such as document similarity, text summarization, text classification, sentiment analysis etc.

So, machine learning, as well as neural network models, can help to solve the problems of topic drift and Information Overkill in IR and many researchers have contributed significant work for improving document ranking. In Table 1, we present the contributions of researchers in a similar domain.

Table 1 Contribution of researchers in the IR domain for document ranking

Researcher(s)	Objective	Dataset Used (Size)	Methodology	Findings	Limitations
Zamani et al. [46]	The work studied the following four hypotheses: 1. To investigate the performance of ad-hoc retrieval by incorporating multiple fields in the document. 2. To compare the performance of the proposed framework with the baselines such as BM25, LTR (Learning To Rank), DSSM (Deep Structured Semantic Model), and C-DSSM (Convolutional DSSM). 3. To compare the performance of learning per field query representations with learning a single query representation. 4. To investigate the performance of the proposed framework by adding field-level masking and field-level dropout.	Bing search log ($\sim$140k)	A neural ranking model was designed to represent documents as well as query and computed the Hadamard product of these representations. The product was fed to a fully-connected neural network with a single non-linear hidden layer to calculate the final retrieval score.	1. The work investigated the ranking quality of documents using NDCG@10 by incorporating the 'title' field with other fields such as URL, Body, Anchor texts, and clicked query. It was found that the most significant improvement in document ranking is achieved using title and clicked URL. 2. The work compared the performance of the proposed framework having a single field as well as multiple fields using NDCG@10 with BM25 and LTR (Learning To Rank). The proposed framework used sparse tensors for word hashing, so it was memory-efficient and accurately learnt document representation by considering multiple document fields which leads to a high retrieval rate. 3. The work compared the performance by designing different query representations for different fields. The results achieved showed that learning per-field query representations performs better than learning a single query representation. 4. The work investigated the performance of the proposed framework in three experimental setups: (i) without masking and dropout, (ii) only masking, masking and dropout. (iii) The most significant improvement in document ranking is achieved by adding both masking and dropout.	The outcomes of the suggested framework typically decline as query length increases. Due to the rarity of long queries, models based on representation learning are expected to perform substantially better for shorter queries.

(*Continued*)

Table 1 Continued

Researcher(s)	Objective	Dataset Used (Size)	Methodology	Findings	Limitations
Wang et al. [47]	The objective of the work was to design an end-to-end Neural Pseudo Relevance Feedback (NPRF) framework, that improves the representation of user information needs from a single query to several PRF (Pseudo Relevance Feedback) documents.	TREC1-3 (741,856 documents with 150 queries) and Robust04 (528,155 documents and 249 queries)	The NPRF framework was instantiated using three cutting-edge neural retrieval models, including the unigram DRMM and KNRM models and the n-gram PACRR model. The relq (q, d) function, which creates Dq, the set of top-m documents, was used to generate the initial ranking for the input query "q". A neural IR model is then used to record and assess how each document dq ∈ Dq interacted with the target document 'd'. This resulted in m real-valued relevance scores, each of which represented the relevance of d as determined by one of the feedback documents dq. As a result, the ultimate relevance score of each target document is determined by how well it interacts with both the initial query and the feedback documents.	When the current neural IR models were integrated with their proposed framework, training and validation losses were decreased and the learned ranking functions performed better.	Semantic information is reduced while translating the document's and the query's semantic representations into a matching matrix. This will have an impact on the performance of the IR system as identifying the hidden contextual relationships between words is important for the ranking of documents.

Cao et al. [48]	The objective of the work was to propose a deep learning based automated software-specific QR technique.	Stack Overflow web server (∼7M)	Using the Transformer's attention mechanism, the suggested model SEQUER learned patterns for query reformulation using query logs supplied by Stack Overflow. SEQUER first extracted QR threads for sampling both initial queries as well as corresponding reformulated queries. The model was then trained on a large corpus of query reformulation pairs, each of which had the original query and the associated reformulated query. With a sizable parallel corpus of QR pairings, SEQUER trained a Transformer-based model to imitate the patterns of QR (such as spelling correction, expression refining, and redundant word deletion). Given the initial queries, the trained model can recommend a list of reformed candidates for selection.	The proposed method successfully located the data, improving performance by 129.33% in terms of MRR (Mean Reciprocal Rank) compared to the initial query.	Contextual data such as the users' profiles, search history, and post-visiting history is missing. Adding more contextual information will further improve the performance of the proposed model as it will help in identifying the hidden contextual relationships between words.

(*Continued*)

Table 1 Continued

Researcher(s)	Objective	Dataset Used (Size)	Methodology	Findings	Limitations
Bhopale, A.P., and Tiwari, A. [49]	The objective of the work was to present a neural network phrase embedding model to deal with semantic phrases or multi-word units for biomedical literature retrieval.	TREC-CDS (733,138 PubMed articles) and OHSUMED (348,566 articles)	The proposed technique kept the word and the phrase in the same vector space by employing the "word2vec" technique to embed multi-word units into vector representations.	The work implements a chunking technique in a distributed environment and enhances query language models by expanding hidden query terms and phrases for the semantically related query terms.	The proposed model is restricted only for phrase extraction and is slower than the baseline IR model.
Padaki et al. [50]	The objective of the work was to explore BERT's sensitivity in the understanding of document and query text content for document ranking.	Robust04 (249 queries and 0.5M documents)	The technique was passage-based BERT re-ranking which split documents into passages, evaluated the relevance between a query and a passage using BERT's two-sentence classification model, and ranked documents using their maximum passage scores.	Long, natural language queries showed higher accuracy of BERT compared to short, keyword queries, illustrating BERT's capacity to extract rich information from complex queries.	Topic drift is present as it is difficult to identify expansion terms that are both in-domain with the corpus and consistent with the initial intent.
Fan-Jiang et al. [51]	The objective of the study was to improve the retrieval efficiency of spoken documents.	Topic Detection and Tracking collection (TDT-2) (Size)	Proposed a BERT-based model using PRF and QR. In order to include information about the usage of terms induced from top documents into the BERT-based retrieval model, the top N feedback documents were first extracted in response to the original query.	The applicability of BERT on spoken document retrieval with a query reformulation was explored with limited query-document relevance information. The result showed that BERT-QR coupled with NRM (Neural Relevance-aware query Model) obtained the highest MAP of 0.731.	The performance of the proposed model may degrade with instances of irrelevant feedback documents utilized for QR.

Zheng et al. [52]	1. To handle vocabulary mismatch problem 2. To provide relevant information using query reformulation.	TREC Robust04 (249 queries and 528,155 documents) and GOV2 (150 queries and 25,205,179 documents)	The proposed query reformulation model worked in three phases. The first phase involved re-ranking documents using a fine-tuned BERT model. The second phase used the BERT model to extract pertinent text chunks from feedback documents, which were then employed in the third phase's final reranking.	The proposed model outperformed in terms of P@20, NDCG@20, and MAP@1K when compared to the baseline models such as NPRF, SNRM, CEDR, etc. with a relatively small increase in computational cost.	Low topic drift occurred as some irrelevant terms are also included in the reformulated query.
Viji, D. and Revathy, S. [53]	To investigate NLP application in detecting text similarity for question pairs/ documents.	Quora question pairs (400 thousand question pairs)	The BERT approach was used to extract text features, which was followed by weighted word embeddings. The features were applied to layers of the Siamese Network as embedded vectors along with their weight values. The Deep Siamese Bi-LSTM model was used to train the embedded vectors of the input text features in the various layers. The similarity scores given to each sentence were then used to discover the semantic text similarity.	The proposed framework has a greater accuracy of 91% when evaluating semantic text similarity when compared to industry benchmarks like the Shingling method, CNN technique, and MLP technique.	With the increase in file sizes, the computing time for calculating the Semantic-Text similarity scores has increased.

(*Continued*)

Table 1 Continued

Researcher(s)	Objective	Dataset Used (Size)	Methodology	Findings	Limitations
Lamsivah et al. [54]	The objective was to record the semantic and syntactic connections between document sentences and the components of users' queries (words, phrases).	DUC'2005 (32 documents and 50 queries), DUC'2006 (25 documents and 50 queries), and DUC'2007 (25 documents and 45 queries)	Firstly, the authors performed the splitting of every document d_i in cluster D into a group of sentences. The sentences from the clusters and the users' queries were then encoded into a fixed-length vector using Universal Sentence Embedding model. Then, using the BM25 model, a score was given to each sentence in cluster D depending on how relevant it was to the query Q. The top-k ranked sentences were then chosen based on the score. Lastly, the re-ranking of selected sentences had been done using the Maximal Marginal Relevance method and greedy search algorithm.	For the DUC'2007 dataset, the proposed technique achieves an increment of 0.71%, 0.51%, and 0.19% for R-1, R-2, and R-SU4 metrics.	The performance of the proposed approach can further be improved using the latest models such as T5, GPT3, etc.
Hassan et al. [45]	The objective was to determine the semantic similarity of texts for title-based research paper recommendations.	CiteULike dataset	The input research paper and the candidate papers in the corpus were converted into sentence embeddings using five pre-trained sentence encoders and then BM25 was used for document ranking.	Sentence embedding models when combined with BM25 outperformed BM25 alone as well as any of the encoders alone.	Sentence encoders used do not perform well in the domain of research paper recommendation as only the titles of research papers are taken into consideration.

It may be observed from Table 1, that many researchers have proposed methods to find semantic similarities between texts in different domains and used different datasets. But, still, there is a scope for improvement in this domain.

3 Issue and Challenges

This section discusses the issues and challenges faced in document ranking. From the above discussion, it has been found that numerous researches had been done for improving document ranking in IR. Still, there is a scope for improvement in this domain. Most of the research work had been focused on improving document ranking based on the relationship between queries and documents in terms of syntactic and semantic relationships. These techniques suffered from various problems such as topic drift [50, 52], vocabulary mismatch [6, 44, 47], extraction of irrelevant terms [47, 50–52, 55] etc. These problems arise because no technique alone is able to completely capture the various features of the query terms, leading to irrelevant results. Also, the user's query lacks clarity and is only vaguely expressed. So, one solution is to make a hybrid of techniques considering all the parameters which will improve the effectiveness of IR systems. Also, the recently developed machine learning techniques discussed in Table 1 like BERT, Transformers, Universal Sentence Encoder, etc., which focused on both semantics as well as context can be used along with other techniques. A new similarity measure or framework can also be designed based on the above techniques to effectively extract the documents and correspondingly rank them. In order to optimize the solution further to obtain the best values of weights for terms, genetic algorithms can also be applied. So, one can design a hybrid framework that applies sentence embedding models to obtain context-aware embeddings, and a genetic algorithm to optimize the weights of extracted terms for query reformulation. The major challenge for document ranking is to understand the intent of the users from the query and find the documents accordingly. To tackle this, we use sentence embedding models which capture the contextual features of queries as well as documents.

4 Embedding Models

Sentence embeddings are a document processing approach for mapping sentences to vectors as a way of encoding text with real numbers that can be used in machine learning. Similarity assessments, such as cosine

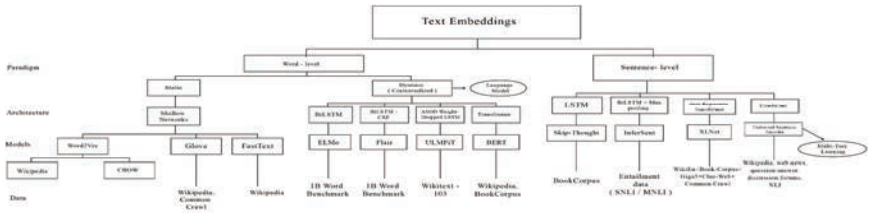

Figure 1 Embedding models.

similarity or Manhattan/Euclidean distance, assess semantic textual similarity so that the results can be used for different NLP tasks, such as Information Retrieval, paraphrasing, and text summarization. Deep learning is now the best-performing approach for every NLP task, according to benchmarks [56–58]. Deep learning advances in recent years, particularly the Transformer architecture [59], have resulted in SOTA (state-of-the-art) NLP scores that are much better than older methods like Word2Vec, GloVe, ELMo, etc. [21–25].

In recent years, sentence encoders like Google's BERT and USE, Facebook's InferSent, and AllenAI's SciBERT and ELMo have gotten a lot of attention. A sentence can be encoded into deep contextualized embeddings using these pre-trained machine learning models. For several natural language processing tasks, they have been shown to outperform SOTA approaches such as Word2Vec, GloVe, ELMo, etc. [21–25]. Calculating semantic similarity and relatedness, which is crucial in constructing efficient Information Retrieval systems, is one of these jobs. The various text embedding models are systematically organized in a hierarchical structure as shown in Figure 1. The first level represents the paradigm, the second level represents the architecture used, the third level reflects the different embedding models corresponding to previous levels and the last level represents the data on which different embedding models were pre-trained.

As the paper mainly focuses on sentence embedding models, the six most popular sentence embedding models viz. Doc2Vec, SentenceBERT, InferSent, Universal Sentence Encoder, ELMo, and XLNet are critically reviewed on the basis of four vital parameters such as architecture, working, applications, and data used and given in the following subsections.

4.1 Doc2Vec

Doc2Vec embedding, an extension of Word2Vec, is one of the most widely used approaches for sentence embedding. It was first introduced in 2014. It is

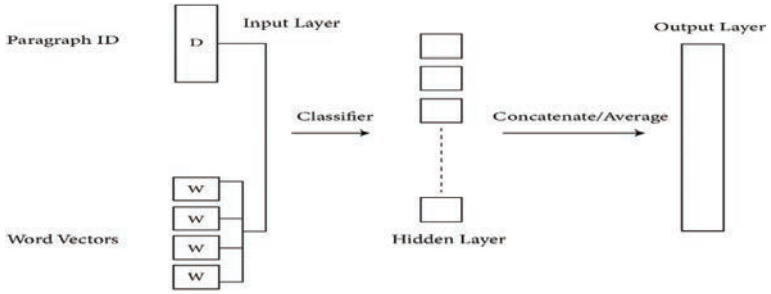

Figure 2 Doc2Vec framework for learning paragraph vector.

an unsupervised technique that extended the Word2Vec model by providing an additional 'Paragraph Vector (PV)'. PVDM (Distributed Memory version of Paragraph Vector) or PVDBOW (Distributed Bag of Words version of Paragraph Vector) was used to perform the addition of paragraph vector [60]. The term "paragraph" refers to texts of various lengths, including phrases, paragraphs, and in this case, the entire document. In the PV framework, every paragraph and every word are mapped to a unique vector, represented by columns in matrix D and matrix W, respectively (Figure 2). For obtaining the final sentence representation, either the average or conjugate of the paragraph vector and words vector is taken. Due to the input being the combined size of the tag(s) and every word in the context rather than the size of a single word vector, conjugate produces a substantially larger model [61].

In the past, researchers had proven that PV performed better than other language models on a number of tasks such as question-answering, sentiment analysis, sentence similarity, etc. [62–64] and has great potential for IR [60]. The PV model can estimate a document-level language model to jointly learn word and document embeddings. Incorporating it into the language model framework for IR tasks is therefore simpler. Authors in [63] performed an analysis of the PV model for IR. The Doc2Vec model was used by the authors of [65] to analyze linguistic features to answer re-ranking of why-questions. The continuous bag-of-word model (CBOW) and continuous skip-gram model (CSG) concepts are both used by the model to compute Doc2vec similarity and estimate the likelihood that each answer candidate positively contributed to the question.

4.2 SentenceBERT

SentenceBERT, the current leader of the pack, debuted in 2018 and quickly rose to the top of the sentence embeddings leaderboard. Sentence BERT

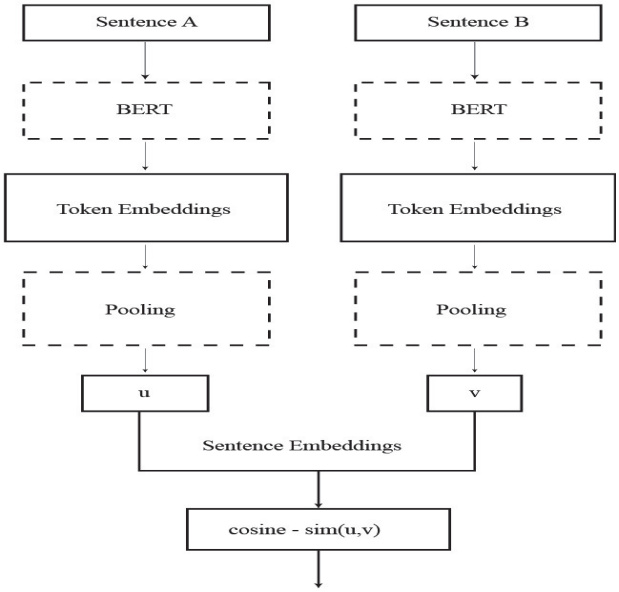

Figure 3 Sentence BERT architecture.

produces fixed-sized sentence embeddings by fine-tuning pre-trained BERT and RoBERTa networks. As shown in Figure 3, this is accomplished by utilizing a Siamese network structure. The pre-trained language model in this Siamese network can be BERT or RoBERTa, the default pooling technique is to compute the average of all output vectors, and u and v are the sentence embeddings. S-BERT adds a pooling operation to a BERT/RoBERTa model's output in order to construct a fixed-sized sentence embedding. The default pooling approach is MEAN, which was found to be superior to using the [CLS]-token output or a MAX pooling strategy. A fixed-sized sentence embedding is essential for creating embeddings that can be used quickly in downstream tasks like inferring semantic textual similarity using cosine similarity scores. S-BERT uses a regressive objective function for inference within a Siamese network that is similar to the one used for fine-tuning once it has been trained. The cosine similarity between two sentence embeddings 'u' and 'v' is computed as a score between $[-1 \ldots 1]$, as shown in Figure 3. Here, it may be noted that concatenation is not required prior to computing the cosine similarity of the sentence embeddings because the regressive objective function is optimized with mean-squared error loss. Authors in [66] demonstrated the dramatic speed increase in 2019 using SBERT for extracting the

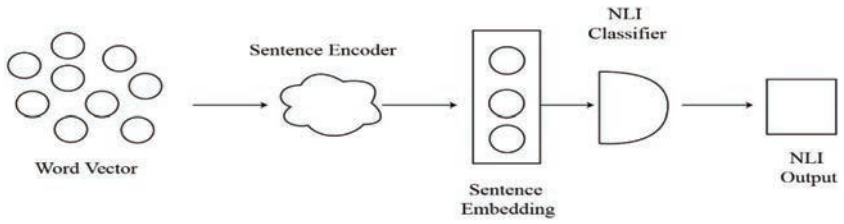

Figure 4 InferSent flow architecture.

most similar sentence pair from 10K sentences. With SBERT, embeddings are constructed in around 5 seconds and compared with cosine similarity in about 0.01 seconds as opposed to 65 hours with BERT.

4.3 InferSent

Facebook AI Research introduced InferSent, a supervised sentence embedding method in 2017 [67]. The most significant characteristic of this model is that it was created using SNLI (Stanford Natural Language Inference) dataset. The InferSent architecture consists of two stages as shown in Figure 4:

1. At the first stage, the sentence encoder takes word vectors and encodes sentences into vectors.
2. At the second stage, an NLI classifier trains the sentence vectors using the encoded vectors as input.

It may be noted that it also employs a Siamese network, much like the SentenceBERT, but instead of max-pooling only, it uses a bi-LSTM, a neural network with memory, with the max-pooling operation to recall the complete sentence to encode [68]. As a result, InferSent embeddings offer detailed semantic representations of sentences, however, their generation and training are sluggish due to the complex Bi-LSTM structure. This model also struggles with long-term context dependencies in comparison with the transformer-based models for long sentences.

4.4 Universal Sentence Encoder (USE)

USE is one of the most effective sentence embedding systems; recommended by Google. It was introduced in 2018 [69]. The most important feature of USE is that it can be used for multiple tasks, such as sentiment analysis, text categorization, sentence similarity, removal of duplicate sentences etc. This encoder is based on two different types of encoders, transformer and Deep

Averaging Network (DAN). Compared to the DAN encoder, the transformer encoder is more accurate and computationally demanding. Both models may generate embeddings for a single word or a sentence. The general workflow is as follows:

i. Firstly, sentences are tokenized once they have been converted to lowercase.
ii. The sentence is then converted into a multi-dimensional vector, depending on the type of encoder used. In the case of the transformer, it may be identical to the encoder component of the transformer structure and utilizes the self-attention mechanism as shown in Figure 5(a). The DAN calculates the bigram and unigram embeddings first, adding them together to produce the single embedding. Following the loading of the data, a deep neural network generates the 512-dimensional sentence embeddings as shown in Figure 5(b).
iii. These sentence embeddings can be applied to supervised as well as unsupervised tasks, like Skipthoughts and NLI (Natural Language Inference).

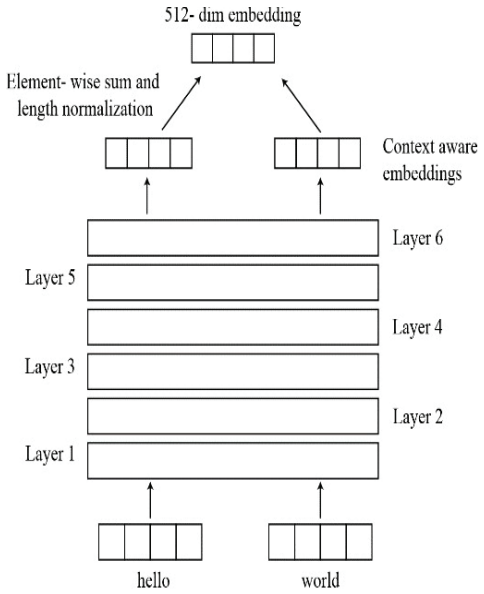

Figure 5(a) USE with transformer.

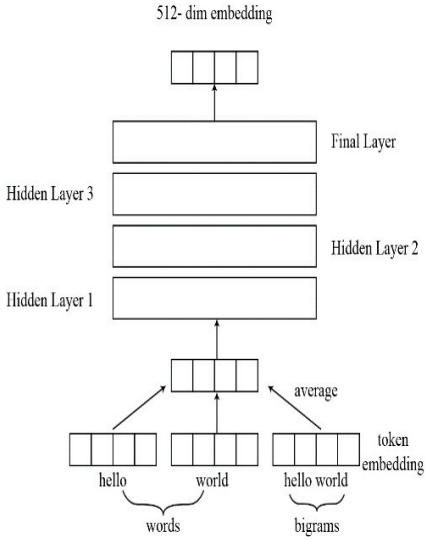

Figure 5(b) USE with deep averaging network.

4.5 ELMo

This model, which was published in early 2018, generates deeply contextualized word embeddings using Recurrent Neural Networks (RNNs) in the form of LSTM architecture, as illustrated in Figure 6 [70]. Current models may easily incorporate ELMo embeddings, which significantly improve the state of the art for many challenging NLP tasks like sentiment analysis, textual entailment, and question answering. The name "ELMo" stands for "Embeddings from Language Models" [19] and refers to how the embeddings are calculated from the internal states of a two-layer bidirectional Language Model (biLM). Each of the two biLM layers has two passes – a forward pass and a backward pass – and they are stacked together. A character-level Convolutional Neural Network (CNN) is used in the architecture to convert the words of a sentence into raw word vectors. These unprocessed word vectors are then fed to the first layer of the biLM. The forward pass contains both additional information about a word and its context (the words that come before it). The backward pass contains information about the word and the context after it. This pair of data from the forward and backward passes make up the intermediate word vectors. The following layer of the biLM is then given these intermediate word vectors. The final representation (ELMo) is

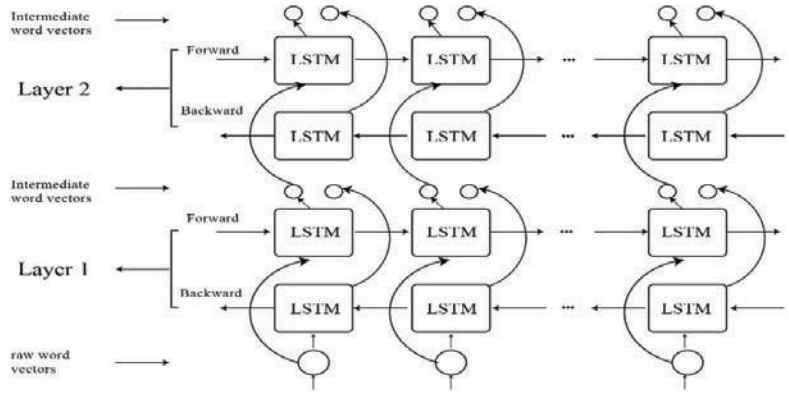

Figure 6 ELMo.

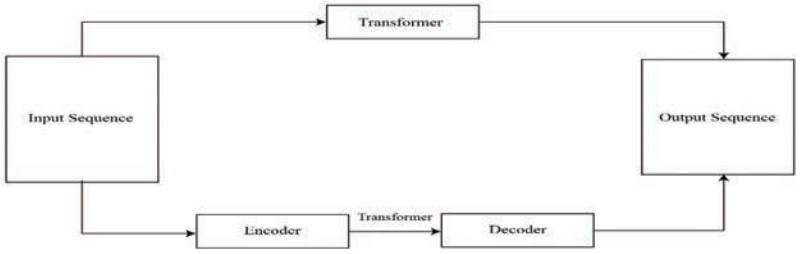

Figure 7 Basic architecture of XLNet.

the weighted sum of the two intermediate word vectors and the raw word vectors.

4.6 XLNet

This is the most recent nominee for the "Coolest New NLP Model" introduced in 2019 [71]. To achieve bidirectional dependencies, it takes a different strategy than BERT (i.e., being able to learn context by not just processing input sequentially). The Transformer-XL model's pre-trained variant, XLNet, optimizes the expected likelihood over all permutations of the input sequence factorization order in order to learn bidirectional contexts using an autoregressive approach. It combines the concepts of auto-regressive models and bidirectional context modelling while overcoming the limitations of BERT's and beating it on 20 tasks, frequently by a significant margin. These tasks include question answering, natural language inference, sentiment analysis, and document ranking. The basic architecture of XLNet is shown in Figure 7.

The key characteristics of XLNet are as follows:

- As a result of considering all possible permutations of factorization order, XLNet computes the most likely sequence. Consequently, when calculating the expectation, each position learns to capture contextual information from all positions, therefore capturing bidirectional context.
- Unlike BERT, XLNet does not depend upon data corruption, hence it does not have the pretrain finetune discrepancy.
- XLNet incorporates Transformer-XL's novelties, such as the recurrence mechanism and relative encoding. As a result, tasks requiring a longer text sequence perform better.

5 Comparative Analysis

To compare the performance of various embedding models, the experiment is carried out to address the following research questions:

RQ1: Which model will work best as the foundation for the generation of sentence embedding using transfer learning?
RQ2: What are the strengths and weaknesses of each model?
RQ3: To explore unseen issues in sentence embedding models for document ranking that require further research?

5.1 Evaluation Metrics Used

To answer RQ1, we need to determine the performance of each of the underlying models discussed in Section 4. We use Recall (R), Mean Average Precision (P), Normalized Discounted Cumulative Gain (NDCG), and F-measure as evaluation parameters. Recall, Precision and F-measure are computed using Equations (1), (2), and (3) respectively.

$$R = \frac{|REL_{EXT}|}{|REL_q|} \tag{1}$$

$$P = \frac{|REL_{EXT}|}{|RET|} \tag{2}$$

Here, REL_{EXT} refers to the retrieved documents that are relevant to the user query, RET refers to the ranked list of retrieved documents, and REL_q refers to the relevant documents that are actually useful to the user and match his search needs. For the sake of better understanding, let us

consider a test case for query, q, on document collection D, where $|\mathrm{D}| = 600$ and relevant documents for the query, $REL_q = \{$d3, d5, d9, d25, d39, d44, d56, d71, d89, d123, d499, d550$\}$. The ranked list of retrieved documents represented by RET; $RET = \{$d123, d9, d38, d84, d511, d44, d56, d129, d250, d6, d187, d8, d25, d3$\}$. Hence relevant documents in the extracted set, $REL_{EXT} = \{$d123, d44, d56, d9, d25, d3$\}$. So, from above discussion, Precision = 6/14 = 43% and Recall = 6/12 = 50%.

The datasets used in this paper also contain a list of relevant documents i.e., REL_q corresponding to each query. We assume $RET = 10$ i.e., we consider the top 10 extracted documents. The REL_{EXT} can be generated by taking the intersection of generated relevant document set RET with the given list of relevant documents REL_q. We conducted the experiment with 241 queries over 5779 documents.

The F-Measure approach, which is generated from precision and recall by taking their harmonic mean, is also used. It can be calculated as given in Equation (3).

$$F = ((\rho^2 + 1)P * R)/(\rho^2 P + R) \tag{3}$$

where P is precision, R is recall, and ρ enables us to choose the relative precision and recall while evaluating performance. In our experiment, the value of ρ is set to 1. When the metric is equal to 1, we call it the balanced metric.

MAP: The TREC (Text Retrieval Conference) community uses MAP as one of the most used measures [72]. To determine the MAP, we must first determine the precision of each relevant document. All relevant documents that are not part of the output are given a precision value of 0. The average of these precision scores is then used to determine the average precision of a single query. The average precision for each query, which is the MAP for many queries, is then calculated.

For a query q_j from the set of queries Q, let Ri be the relevant documents extracted by the system. Assume that $P(R_i[k])$ represents the precision estimated until $R_i[k]$ is noticed in the ranking. $R_i[k]$ will be 0 if the Kth ranked document cannot be retrieved. The average precision score "AP" for query q_i is therefore determined as given in Equation (4).

$$AP_i = \frac{1}{|R_i|} \sum_{k=1}^{|R_i|} P(R_i[k]) \tag{4}$$

The MAP is computed using Equation (5).

$$MAP = \frac{1}{|Q|} \sum_{i=1} |Q_i| AP_i \tag{5}$$

NDCG: The NDCG is a ranking quality metric. This metric is used to evaluate document retrieval strategies in IR. The following premise must be kept in mind in order to fully understand the notion of NDCG: "The highly relevant documents are more useful than the moderately relevant documents, which are in turn more useful than the irrelevant documents". A relevance score is given to each retrieved document called recommendation. The sum of all the relevance scores in a recommendation set is the cumulative gain (CG) determined as given in Equation (6).

$$CG = \sum_{k=1}^{n} relevance_k \tag{6}$$

CG only considers relevance scores into account and does not use the position of the document due to which sometimes it fails to accurately calculate results. In order to discount the relevance score, another metric called DCG (Discounted Cumulative Gain) divided the relevance score by the log of the corresponding position in the computation using Equation (7).

$$DCG = \sum_{k=1}^{n} \frac{2^{relevance_k} - 1}{log_2(k+1)} \tag{7}$$

The quantity of recommendations delivered to each user may vary depending on a variety of criteria. The DCG will consequently change. We need a score with appropriate upper and lower bounds so that we can use it to generate the final score, which is the average of all the recommendation scores. This normalization is brought about by NDCG. To compute NDCG for each recommendation set, we must first calculate DCG of the recommended order and DCG of the ideal order (iDCG). NDCG is the result of dividing DCG of the recommended order by DCG of the ideal order as given in Equation (8).

$$NDCG = \frac{DCG}{iDCG} \tag{8}$$

This ratio will always be in the range [0, 1].

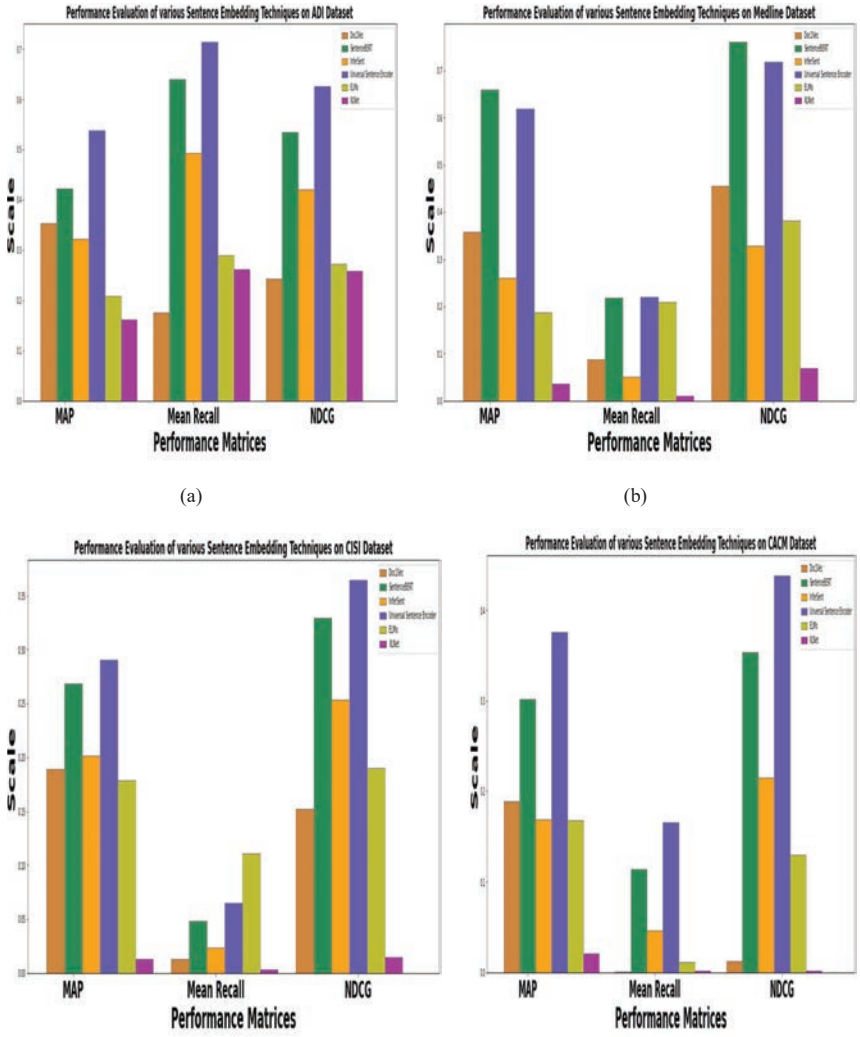

Figure 8 Performance evaluation of various sentence embedding techniques on different datasets using MAP, mean recall and NDCG.

Figures 8(a)–8(d) depict performance evaluation using the mean average precision, NDCG and mean recall matrices of underlying sentence embedding models at top 10 documents cut-off using CACM, CISI, ADI and Medline datasets.

Figures 8(a)–8(d) show the experimental results for the comparative performance of six pretrained sentence embedding models in terms of MAP, Recall and NDCG on four different datasets, which aims to answer RQ1. All the datasets are pre-processed under a similar experimental setup. It may be noted from Figures 8(a)–8(d) that USE and SentenceBERT based sentence embedding techniques outperformed other sentence embedding techniques on all the four datasets. Particularly, the USE sentence embedding model outperformed other models by 26.48 %, 8.97% and 17.6% for ADI, CISI, and CACM dataset respectively while in case of Medline dataset, SentenceBERT outperformed other models by 23.36%. The reason behind the outstanding performance of USE over rest of the models is deeply investigated. It is found that being a multi-task model, its architecture forgoes recurrence for the attention-based transformer, that adds to its performance and make it well suited with transfer-learning. Besides, the embeddings generated using USE are the most adaptable to domain-specific tasks and the simplest to implement.

F-Measure is a good choice for comparing different models because the high value of the F-Measure indicates the retrieval effectiveness of the model with respect to a user search. Therefore, we obtain the F-measure values for each of the underlying model to more precisely answer RQ1. These values are plotted using line graphs as given in Figures 9(a)–9(d).

From the graphs in Figures 9(a)–9(d), it is inferred that the USE model effectively retrieves the relevant documents with respect to a user query A possible reason for the high F-Measure obtained by the USE model on all four datasets is that it was trained on both SNLI and web question-answer pages, and there exists a similarity between these tasks and the training data used for the USE model.

The documents extracted using sentence embeddings generated by USE are further used to reformulate the initial query to improve the results further, as these documents are similar to the query raised by the user in terms of semantics as well as context. This will remove the problem of the topic draft which arises due to the extraction of irrelevant documents as discussed in Section 3 of this paper.

5.2 Strength and Weakness

Each sentence embedding approach has its some merits and demerits. There were a lot of advances in sentence embedding approaches in the last few years. But there is no sentence embedding approach made that can perform

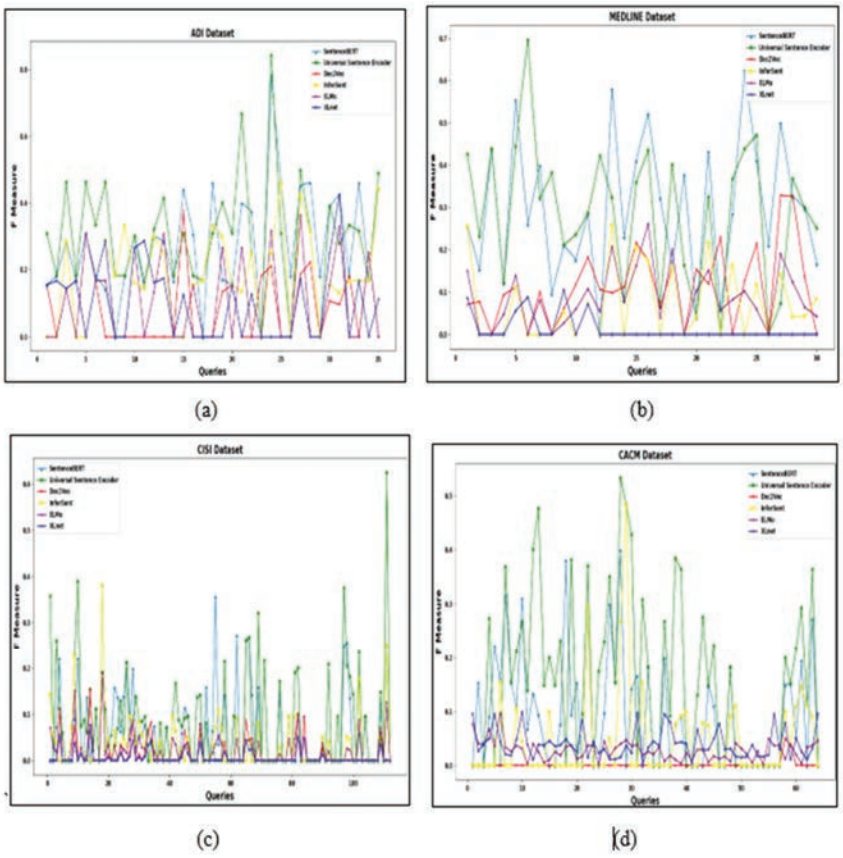

Figure 9 Plot of F-Measure over various datasets (ADI, Medline, CISI and CACM dataset).

well on every downstream NLP task. So, one should choose an embedding model keeping in mind the task to be performed using it. Out of the six discussed models, doc2vec most closely resembles the word2vec word embeddings, which merely add a learnt vector specific to every paragraph to the training and inference stages. Doc2vec offers an improvement over existing methods that average word embeddings, but it does not explicitly represent word order or take polysemy into account.

SentenceBERT embeddings are trained on the decision of whether a sentence could follow another one. In order to train a new encoder, SentenceBERT requires a large amount of continuous text. InferSent however needs annotated data and is, therefore, the most difficult option if a specialized encoder is desired. ELMO and SentenceBERT are capable of generating

different word embeddings that capture the context of a word – its location within a sentence. As the ELMo uses bidirectional LSTM it can get an understanding of both the next and previous words in the sentence. InferSent and Elmo encoders are also too narrow in predicting what embeddings can contain. In contrast, ELMo is a character-based model that handles out of vocabulary words by employing character convolutions. XLNet combines the concepts of auto-regressive models and bidirectional context modelling while overcoming the limitations of SentenceBERT's and beating it on 20 tasks, frequently by a significant margin. These tasks include question answering, natural language inference, sentiment analysis, and document ranking. By emphasizing transfer learning with multi-task learning, the universal sentence encoder differs from the other techniques by eschewing recurrence for the attention-based transformer. Embeddings generated using USE are the most adaptable to domain-specific tasks and are easy to implement.

5.3 Discussion

To compare contextually the behavior of various sentence embedding models for document ranking, it is required to effectively train them on different types of datasets. For this purpose, four datasets belonging to different domains are utilized; one containing biomedical abstracts, another containing computer science ACM abstracts, another containing information science abstracts, and the last one containing abstracts on information management. The Medline dataset contains 30 queries and 1033 documents, the CISI dataset contains 112 queries and 1460 documents, the CACM dataset contains 64 queries and 3204 documents, and the ADI dataset contains 35 queries and 82 documents. To proceed with the experiment, it is required to pre-process the queries and documents. After that, six pretrained sentence embedding models have been used to generate contextualized sentence embeddings corresponding to a given corpus of documents and queries respectively. Lastly, the cosine similarity score between each document corresponding to a user query is calculated. The cosine similarity is advantageous because even if the two similar documents are far apart by the Euclidean distance (due to the size of the document), chances are they may still be oriented closer together. A document with a higher cosine similarity rate with respect to the query is considered as more similar to the query. So, based on this score, a ranking of documents has been done to upgrade the IR system's performance. The comparison of various sentence embedding models based on the F measure is graphically presented in Figures 8(a)–8(d). It may be noted from the graphs that the F measure is high in USE.

It has also been observed from the values in Figures 8(a)–8(d), that less improvement is there in terms of precision, recall and F measure. This may be due to the fact that although these models were able to deal with context-related problems, but because they were trained on general domain corpora like Wikipedia, their applications are restricted to certain tasks or areas. So, in the future, we can further improve the performance of these models by combining them with other techniques like evolutionary algorithms which can further optimize the performance. Also, datasets used contain documents and queries written only in the English language. So, one question arises whether we can use these models for multi-lingual documents or queries, which will be the scope for future work.

6 Conclusion

Many sentence embedding techniques exist today that acquire deep semantic representations by training on a large corpus, with the goal of providing transfer learning to a wide range of NLP tasks such as text summarization, text similarity, sentiment analysis, question-answering, etc. The present study does a comparative analysis of six pre-trained sentence embedding techniques for document ranking to upgrade the IR systems performance. Sentence embedding techniques were used for converting text to appropriate vector representations. The similarity scores between the embedded vectors of queries and documents were calculated by using the cosine similarity method. The study was carried out with six sentence embedding techniques (Doc2Vec, SentenceBERT, InferSent, Universal Sentence Encoder, ELMo, and XLNet) on four datasets (CACM, CISI, ADI and Medline datasets). It has been found that Universal Sentence Encoder and SentenceBERT outperform other techniques on all four datasets in terms of MAP, recall, F-measure, and NDCG. The obtained results may be commercially very useful for industries in downstream NLP and TAR (technical assistance reports) applications to effectively inject the domain-specific semantic context into the search engine. Some of these applications include recommendation systems, chatbot systems, query auto completion, query reformulation and microblogging. Ultimately, the semantically richer sentence embedding models like Universal Sentence Encoder and SentenceBERT may be plugged into existing NLP models to dramatically improve the performance of NLP professional toolkits and aid to language data service providers. Further, it has also been observed that different sentence embedding techniques can capture the

different features of the terms, however, sentence embedding techniques are still very far from the concept of USE that can have a broad transfer quality.

In the future, the work can be expanded in many ways by combining these methodologies to build a hybrid ranking system to score the documents. Future studies will also focus on improving these strategies by including different similarity measurements to achieve better results.

Acknowledgment

I would like to express my sincere and deep gratitude to my Ph.D. supervisors, J. C. Bose University of Science & Technology, Faridabad for their continuous guidance, constructive criticism and valuable advice. I would also express gratitude to the chairperson, Computer Engineering Department, J. C. Bose University of Science & Technology, Faridabad and Head, Computer Science and Engineering Department, MMEC, MM(DU), Mullana, who helped me in collecting the necessary information for obtaining the experimental results.

Funding

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Conflict of Interest

The authors declare there is no conflict of interest.

References

[1] Crabtree, D., Andreae, P., and Goa, X.: The vocabulary problem in human-system communication. In Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, pp. 191–200 (2007).

[2] Lau, R.Y., Bruza, P.D., and Song, D.: Belief revision for adaptive information retrieval. In Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval, ACM, pp. 130–137 (2004).

[3] Robertson, S.E., Walker, S.: Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. In: Proceedings

of the 17th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval. pp. 232–241 (1994).

[4] Lafferty, J.D., Zhai, C.: Document language models, query models, and risk minimization for information retrieval. In: SIGIR 2001: Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 111–119 (2001).

[5] Metzler, D., Croft, W.B.: A markov random field model for term dependencies. In: SIGIR 2005: Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 472–479 (2005).

[6] Rajashekar, T.B., Croft, W.B.: Combining automatic and manual index representations in probabilistic retrieval. J. Am. Soc. Inf. Sci. 46(4), 272–283 (1995).

[7] Lavrenko, V., Croft, W.B.: Relevance-based language models. In: SIGIR 2001: Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 120–127 (2001).

[8] Bouramoul, A., Kholladi, M.K. and Doan, B.L.: Context based query reformulation for information retrieval on the web. In International Arab Conference on Information Technology. ACIT (2009).

[9] Jiang, J., He, D., Han, S., Yue, Z. and Ni, C.: Contextual evaluation of query reformulations in a search session by user simulation. In Proceedings of the 21st ACM international conference on Information and knowledge management, pp. 2635–2638 (2012).

[10] Torjmen-Khemakhem, M. and Gasmi, K.: Document/query expansion based on selecting significant concepts for context based retrieval of medical images. Journal of biomedical informatics, 95, p. 103210 (2019).

[11] Agbele, K.K., Ayetiran, E. and Babalola, O.: A Context-Adaptive Ranking Model for Effective Information Retrieval System (2018).

[12] Montazeralghaem, A., Rahimi, R. and Allan, J.: Relevance Ranking Based on Query-Aware Context Analysis. Advances in Information Retrieval, 12035, p. 446 (2020).

[13] Kim, J.: A Document Ranking Method with Query-Related Web Context. IEEE Access, 7, pp.150168–150174 (2019).

[14] https://towardsdatascience.com/Se-network-embeddings-explained-4d028e6f0526

[15] Naseem, U., Razzak, I., Khan, S.K. and Prasad, M.: A comprehensive survey on word representation models: From classical to state-of-the-art word representation language models. Transactions on Asian and Low-Resource Language Information Processing, 20(5), pp.1–35 (2021).

[16] Zamani, H., and Croft, W. B.: Estimating Embedding Vectors for Queries. Proceedings of the 2016 ACM International Conference on the Theory of Information Retrieval – ICTIR '16 (2016).

[17] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, Advances in Neural Information Processing Systems 26, pages 3111–3119. Curran Associates, Inc. (2013).

[18] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In Empirical Methods in Natural Language Processing (EMNLP), pages 1532–1543 (2014).

[19] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," in Proc. of NAACL (2018).

[20] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. arXiv preprint arXiv:1607.04606 (2016).

[21] Iz Beltagy, Arman Cohan, and Kyle Lo. 2019. SciBERT: Pretrained Contextualized Embeddings for Scientific Text. arXiv preprint arXiv:1903.10676 (2019).

[22] Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St John, Noah Constant, Mario GuajardoCespedes, Steve Yuan, Chris Tar, et al. 2018. Universal sentence encoder. arXiv preprint arXiv:1803.11175 (2018).

[23] Alexis Conneau, Douwe Kiela, Holger Schwenk, Loic Barrault, and Antoine Bordes. 2017. Supervised learning of universal sentence representations from natural language inference data. arXiv preprint arXiv:1705.02364 (2017).

[24] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018).

[25] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In Proc. of NAACL (2018).

[26] Wei Yang, Haotian Zhang, and Jimmy Lin. 2019. Simple Applications of BERT for Ad Hoc Document Retrieval. arXiv preprint arXiv:1903.10972 (2019).
[27] Zhuyun Dai and Jamie Callan. 2019. Deeper Text Understanding for IR with Contextual Neural Language Modeling. arXiv preprint arXiv:1905.09217 (2019).
[28] Kitchenham, B. and Brereton, P.: A systematic review of systematic review process research in software engineering. Information and software technology, 55(12), pp.2049–2075 (2013).
[29] Sethi, S., and Dixit, A. (2017). An Automatic User Interest Mining Technique for Retrieving Quality Data. International Journal of Business Analytics (IJBAN), 4(2), 62–79.
[30] Yao, X., Van Durme, B., Clark, P.: Automatic coupling of answer extraction and information retrieval. In: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics. pp. 159–165 (2013).
[31] Chen, T., Van Durme, B.: Discriminative information retrieval for question answering sentence selection. In: Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics. pp. 719–725 (2017).
[32] Dai, Z., Callan, J.: Context-aware term weighting for first-stage passage retrieval. In: The 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval (to appear) (2020).
[33] Devlin, J., Chang, M., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. pp. 4171–4186 (2019).
[34] Nogueira, R., Yang, W., Lin, J., Cho, K.: Document expansion by query prediction. CoRR abs/1904.08375 (2019).
[35] Gao L., Dai Z., Chen T., Fan Z., Van Durme B., Callan J.: Complement Lexical Retrieval Model with Semantic Residual Embeddings. In: Hiemstra D., Moens MF., Mothe J., Perego R., Potthast M., Sebastiani F. (eds) Advances in Information Retrieval. ECIR 2021. Lecture Notes in Computer Science, vol 12656. Springer, Cham (2021).
[36] Guo, J., Fan, Y., Ai, Q., Croft, W.B.: A deep relevance matching model for ad-hoc retrieval. In: Proceedings of the 25th ACM International Conference on Information and Knowledge Management. pp. 55–64 (2016).

[37] Luyu Gao, Zhuyun Dai, Tongfei Chen, Zhen Fan, Benjamin Van Durme, Jamie Callan.: "Chapter 10 Complement Lexical Retrieval Model with Semantic Residual Embeddings", Springer Science and Business Media LLC (2021).

[38] Deerwester, S.C., Dumais, S.T., Landauer, T.K., Pumas, G.W., Harshman, R.A.: Indexing by latent semantic analysis. J. Am. Soc. Inf. Sci. 41(6), 391–407 (1990).

[39] Bromley, J., Guyon, I., LeCun, Y., Sackinger, E., Shah, R.: Signature verification using a siamese time delay neural network. In: Advances in Neural Information Processing Systems 6. pp. 737–744 (1993).

[40] Caid, W.R., Dumais, S.T., Gallant, S.I.: Learned vector-space models for document retrieval. Inf. Process. Manag. 31(3), 419–429 (1995).

[41] Lee, K., Chang, M., Toutanova, K.: Latent retrieval for weakly supervised open domain question answering. In: Proceedings of the 57th Conference of the Association for Computational Linguistics. pp. 6086–6096 (2019).

[42] Guu, K., Lee, K., Tung, Z., Pasupat, P., Chang, M.: REALM: retrieval-augmented language model pre-training. CoRR abs/2002.08909 (2020).

[43] Chang, W., Yu, F.X., Chang, Y., Yang, Y., Kumar, S.: Pre-training tasks for embedding-based large-scale retrieval. In: 8th International Conference on Learning Representations (2020).

[44] Azad, H.K. and Deepak, A.: Query expansion techniques for information retrieval: a survey. Information Processing & Management, 56(5), pp. 1698–1735 (2019).

[45] Hassan, H.A.M., Sansonetti, G., Gasparetti, F., Micarelli, A. and Beel, J.: Bert, elmo, use and infersent sentence encoders: The panacea for research-paper recommendation?. In RecSys (Late-Breaking Results), pp. 6–10 (2019).

[46] Zamani, H., Mitra, B., Song, X., Craswell, N., and Tiwary, S.: Neural ranking models with multiple document fields. In: Proceedings of the eleventh ACM international conference on web search and data mining. pp. 700–708 (2018).

[47] Wang, L., Luo, Z., Li, C., He, B., Sun, L., Yu, H., and Sun, Y.: An end-to-end pseudo relevance feedback framework for neural document retrieval. Information Processing & Management, 57(2) (2020).

[48] Cao, K., Chen, C., Baltes, S., Treude, C., Chen, X.: Automated Query Reformulation for Efficient Search based on Query Logs From Stack Overflow. In 2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE), IEEE, pp. 1273–1285 (2021).

[49] Bhopale, A.P., and Tiwari, A.: Leveraging Neural Network Phrase Embedding Model for Query Reformulation in Ad-Hoc Biomedical Information Retrieval, Malaysian Journal of Computer Science, Vol. 34, Issue 2, pp. 151–170 (2021).

[50] Padaki, R., Dai, Z., Callan, J.: Rethinking Query Expansion for BERT Reranking. In European Conference on Information Retrieval, Springer, Cham, pp. 297–304 (2020).

[51] Fan-Jiang, S.W., Lo, T.H., Chen, B.: Spoken Document Retrieval Leveraging Bert-Based Modeling and Query Reformulation. In ICASSP 2020–2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, pp. 8144–8148 (2020).

[52] Zheng, Z., Hui, K., He, B., Han, X., Sun, L., Yates, A.: Contextualized query expansion via unsupervised chunk selection for text retrieval, Information Processing & Management, 58(5), p. 102672 (2021).

[53] Viji, D. and Revathy, S.: A hybrid approach of Weighted Fine-Tuned BERT extraction with deep Siamese Bi–LSTM model for semantic text similarity identification. Multimedia Tools and Applications, pp. 1–27 (2022).

[54] Lamsiyah, S., El Mahdaouy, A., El Alaoui, S.O. and Espinasse, B.: Unsupervised query-focused multi-document summarization based on transfer learning from sentence embedding models, BM25 model, and maximal marginal relevance criterion. Journal of Ambient Intelligence and Humanized Computing, pp. 1–18 (2021).

[55] Sethi, S., and Dixit, A. (2019). A novel page ranking mechanism based on user browsing patterns. In Software Engineering (pp. 37–49). Springer, Singapore.

[56] Young, T., Hazarika, D., Poria, S. and Cambria, E.: Recent trends in deep learning based natural language processing. ieee Computational intelligenCe magazine, 13(3), pp. 55–75 (2018).

[57] Tekir, S. and Bastanlar, Y.: Deep learning: Exemplar studies in natural language processing and computer vision. Data Mining-Methods, Applications and Systems (2020).

[58] Zhou, M., Duan, N., Liu, S. and Shum, H.Y.: Progress in neural NLP: modeling, learning, and reasoning. Engineering, 6(3), pp. 275–290 (2020).

[59] https://medium.com/inside-machine-learning/what-is-a-transformer-d07dd1fbec04

[60] Le, Q. and Mikolov, T.: Distributed representations of sentences and documents. In International conference on machine learning, pp. 1188–1196 (2014).
[61] https://radimrehurek.com/gensim/models/doc2vec.html
[62] Dai, A.M., Olah, C., Le, Q.V. and Corrado, G.S.: Document embedding with paragraph vectors In: NIPS Deep Learning Workshop (2014).
[63] Ai, Q., Yang, L., Guo, J. and Croft, W.B.: Analysis of the paragraph vector model for information retrieval. In Proceedings of the 2016 ACM international conference on the theory of information retrieval, pp. 133–142 (2016).
[64] Ai, Q., Yang, L., Guo, J. and Croft, W.B.: Improving language estimation with the paragraph vector model for ad-hoc retrieval. In Proceedings of the 39th international ACM SIGIR conference on research and development in information retrieval, pp. 869–872 (2016).
[65] Breja, M. and Jain, S.K.: Analyzing Linguistic Features for Answer Re-Ranking of Why-Questions. Journal of Cases on Information Technology (JCIT), 24(3), pp.1–16 (2022).
[66] Reimers, N. and Gurevych, I.: Sentence-bert: Sentence embeddings using siamese bert-networks. arXiv preprint arXiv:1908.10084 (2019).
[67] Conneau, A., Kiela, D., Schwenk, H., Barrault, L. and Bordes, A.: Supervised learning of universal sentence representations from natural language inference data. arXiv preprint arXiv:1705.02364 (2017).
[68] Reimers, N. and Gurevych, I.: Sentence-bert: Sentence embeddings using siamese bert-networks. arXiv preprint arXiv:1908.10084 (2019).
[69] Cer, D., Yang, Y., Kong, S.Y., Hua, N., Limtiaco, N., John, R.S., Constant, N., Guajardo-Céspedes, M., Yuan, S., Tar, C. and Sung, Y.H.: Universal sentence encoder. arXiv preprint arXiv:1803.11175 (2018).
[70] https://www.analyticsvidhya.com/blog/2019/03/learn-to-use-elmo-to-extract-features-from-text/
[71] Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R.R. and Le, Q.V.: Xlnet: Generalized autoregressive pretraining for language understanding. Advances in neural information processing systems, 32 (2019).
[72] Sethi, S. (2021). An optimized crawling technique for maintaining fresh repositories. Multimedia Tools and Applications, 80(7), 11049–11077.

Biographies

Vishal Gupta has more than 15 years of teaching experience. He has received his M.Tech. (CSE) from MMU, Mullana in the year 2011. Presently he is serving as Assistant Professor in the Department of Computer Science and Engineering at MMEC, MM(DU), Mullana, Ambala, Haryana and is pursuing his PhD at J.C. Bose University of Science & Technology, YMCA, Faridabad, Haryana. He has published more than thirteen research papers in various International journals and conferences. His area of research includes Information Retrieval System, Data Structures and Algorithms and Artificial Intelligence.

Ashutosh Dixit has more than 18 years of teaching experience. He has published more than 80 research papers in various International Journals and Conferences of repute. He has successfully supervised 7 PhD theses and currently supervising 3 PhD research scholars. Presently he is Professor in Department of Computer Engineering and Dean, Academic Affairs at J. C. Bose University of Science & Technology, YMCA, Faridabad, Haryana.

Earlier, he has been Former Dean, Faculty of Sciences, Former Dean, Faculty of Life Sciences, Former Chairperson, Department of Physics, Department of Chemistry and Department of Environmental Sciences in the present University. Currently, he is also working as Dean Academics Affairs and Director, IQAC. He has one ongoing research project funded by AICTE and one international patent to his credit. His area of research includes Internet and Web Technologies, Data Structures and Algorithms, Computer Networks and Mobile and Wireless communications.

Shilpa Sethi has received her Master in Computer Application from Kurukshetra University, Kurukshetra in the year 2005 and M. Tech. (CE) from MD University Rohtak in the year 2009. She has done her PhD in Computer Engineering from YMCA University of Science & Technology, Faridabad in 2018. Currently she is serving as Associate Professor in the Department of Computer Applications at J.C. Bose University of Science & Technology, Faridabad Haryana. She is also working as Director, International Affairs in the present University. She has published 3 research papers in SCI journals, 10 research papers in Scopus indexed journals and more than 30 research papers in various UGC approved journals and international conferences. Her area of research includes Internet Technologies, Web Mining, Information Retrieval System and Artificial Intelligence.