Document Retrieval for BioASQ Challenge

By

Anil Dadwal

A thesis submitted to Macquarie University
in partial fulfilment of the degree of
Master of Research
Department of Computing
May 2023

MACQUARIE
University
SYDNEY·AUSTRALIA

Typeset in LaTeX 2_ε.

Statement of Originality

This work has not previously been submitted for a degree or diploma in any university. To the best of my knowledge and belief, the thesis contains no material previously published or written by another person except where due reference is made in the thesis itself.

(Signed) _____ Date: _____

Anil Dadwal

Dedication

I pay my respect to Wallumattagal clan of the Dharug Nation, and their Aboriginal Elders past, present and future.

Acknowledgements

I thank Dr. Molla-Aliod, whose guidance and supervision helped me complete my research. The computing resources for this research were provided by National computational Infrastructure, Australia.

Abstract

This study investigates the use of Natural Language Processing Transformer generated dense vectors for identifying a subset of documents that may have answers to a biomedical question. The thesis describes the current research in the Open domain question answer field and its use for this study. The study investigates if dense vectors generated by Transformers trained on biomedical data have a performance edge over dense vectors generated by the generic Transformers. The study further investigates if combining document embeddings generated by Transformers in simple operations can be used to enhance the performance of a document search algorithm. Based on the experiments conducted, it is observed that in general, without further fine-tuning, the use of dense vectors generated from biomedical Transformers do not deliver superior performance as compared to general Transformers. However, document embeddings created by Transformers have useful information that can be used in simple algorithms such as SBERT and ColBERT to identify answer bearing documents. These operations are comparatively simpler and consume less resources than building a complex Transformer using extensive biomedical data and special methods to add biomedical knowledge to these complex Transformers.

Contents

List of Figures

List of Tables

"Noblest pleasure is the joy of understanding"

Leonardo da Vinci

1 Introduction and background

This study investigates the use of dense vectors, deep neural networks, and Natural Language Processing (NLP) based technology to identify PubMed documents that are likely to contain answers to a medical question. The process for identification of documents proposed in this study can be used in the BioASQ challenge Phase B. This study builds a two-stage pipeline (figure 4.1) to search for documents. In the first stage, lexical analysis techniques such as BM25 are used to retrieve a large number of documents (500) from the PubMed database. In the second stage, these documents are matched against the query using dense vectors generated by deep neural networks-based Transformers.

It is envisaged that the research methods proposed in this research can be used for an efficient document search for the BioASQ challenge. The methods proposed in this study improves upon the classical search methods. These methods use lexical search where documents are searched based on the key word matching. The key algorithms for this term based search are Term Frequency - Inverse Document Frequency (TF-IDF) and BM25. However, The use of dense vectors has introduced a paradigm shift and it facilitates semantics based document search. This study proposes a document pipeline approach where, in a first phase, term based lexical search is used to retrieve numerous documents from the corpus and then, in the second phase, these retrieved documents are matched with the query using dense vector based algorithms. The research in this study is concentrated on the second phase of the pipeline.

The main aim of the study is to seek data to inform, if word embeddings produced by language models such as BERT and simple operations on these embedding have a comparative advantage, for biomedical Question Answering task, to word embedding obtained from complex and expensive large language models that are trained with biomedical data and knowledge bases. In other words the study evaluates if embeddings obtained from simple language models but further processed using simple operations to extract attention (figure 1.2 and 1.3) can deliver comparable results against the embeddings obtained from very complex and large language models that are trained using biomedical text (figure

1.1). Do complex domain models internally capture the knowledge and relationships and therefore word embeddings generated from them are better than the embeddings obtained from simple general language models but processed for self and cross term attention using simple operations outside the language models ? Are simple arithmetic post processing of embedding from simpler model more effective(cost and run time) than the embedding from a large and complex model trained on domain data but with no post processing ?

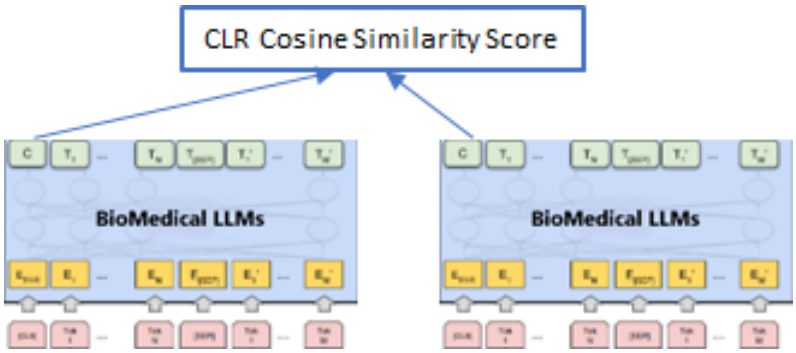

FIGURE 1.1: Query and Document embeddings from Biomedical LLMs and their similarity using CLR token

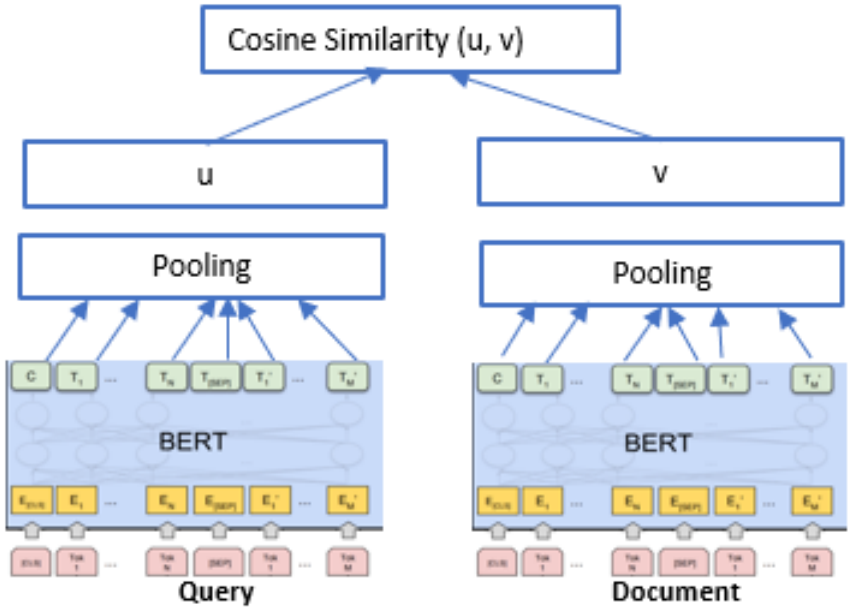

FIGURE 1.2: Query and Document embeddings from BERT and their similarity using SBERT Post processing

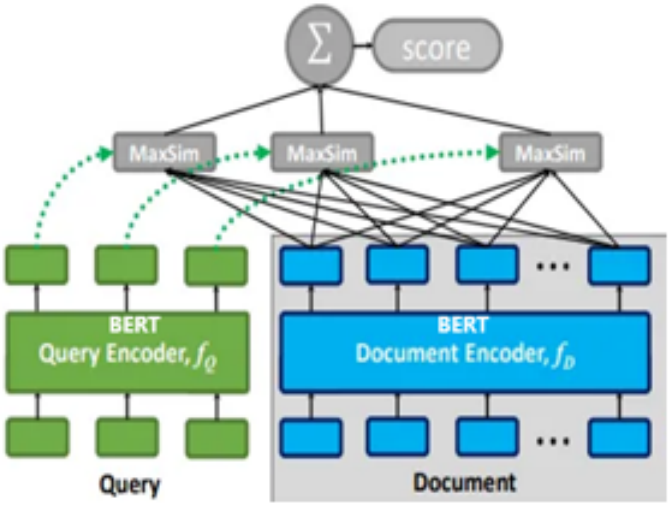

FIGURE 1.3: Query and Document embeddings from BERT and their similarity using ColBERT Post processing, Source (Khattab et al., SIGIR2020) [1]

I review the current literature about use of NLP to search a corpus to answer a question in chapter two and provide more details about this key research question in chapter 3 section 3.3.

The study tests dense vector representations created by progressively complex dense vector matching and Transformer technologies such as CLS token matching, Dense Passage Retrieval (DPR) (Karpukhin, V, 2020 [22] , sentence BERT (Reimer, N et al. [16]), and ColBERT (Khattab et al.,2020 [23]). All these representations are based on different variations of the BERT Transformer technology and dense vector matching. Additionally, the study evaluated retrieval performance using publicly available large language models trained on biomedical domain data. The study also used biomedical data to fine-tune a publicly available standard models to ascertain if word embedding generated by domain data trained BERT can capture domain knowledge, such as entities and their relationships, and can therefore improve vector matching performance. However, these biomedical standard models and general models fine-tuned on biomedical data did not provide competitive results or substantive improvements over the other publicly available BERT Transformers. The study finds that contextualized late interaction of the vectors representing terms as implemented in ColBERT and cross-architecture knowledge distillation (Hofstatter, S et al., 2021 [19]) model provides better results. The results and explanation of superior results of these two methods are discussed in the section 4.7 of this thesis. In the conclusion and further research chapter, I discuss the recent performance of large language models and their use for document retrieval. However, the contextual late interaction of terms as introduced by CoLBERT is still an active research topic that can be used efficiently for simple query and passage matching and answer document identification.

The data for this study is sourced from BioASQ. BioASQ (Tsatsaronis et al., 2015 [24]) conducts a question answer competition in the area of biomedical semantics and question answering. It generally runs two challenges called as Task a and Task b. Task a

is a classification task where a machine learning model performs a classification of a new PubMed document. The classification is based on the MeSH (Medical subject heading) and this classification is evaluated against human annotations. MeSH is a hierachical-organised vocabulary created by National Library of Medicine (USA). MeSH is used for cataloging, searching and indexing of biomedical and health information. In task b, phase A, the participants are given a test set (of 100 questions) and they are "expected to identify and submit relevant elements from designated resources, including PubMed/MedLine articles" (Nentidis et al., 2021 [2]). As part of the phase B of task b, participants can further use these retrieved documents to extract the answer to the question. Please see figure 2.1 on page 6.

"If I have seen further it is by standing on the shoulders of Giants"
Isaac Newton, 1675, In a letter to Robert Hooke

2 Literature Review

2.1 Introduction

This section is the literature review of the current state of the research in the field of **Open Domain Question Answering** (ODQA) systems and its use for the **document retrieval part of the Biomedical Question Answering BioASQ challenge (task b, phase A)**.

There has been active research in the field of ODQA for several decades, but the current advances in Neural Network (NN) based Natural Language Processing (NLP) methods have renewed interest and have facilitated breakthrough results. The purpose of this section is to present a brief overview of the current state of the research in Open Domain Question Answering (ODQA) field, with special focus on its retrieval component. The recent retrieval research from ODQA can be used for retrieving relevant documents from a biomedical corpus and used as answer source for the BioASQ challenge.

2.2 History and recent trends in QA technology

2.2.1 Open Domain Question Answering

Open domain Question Answering (ODQA) was originally defined as 'finding answers in collections of unstructured documents' (Chen-et al., 2017 [4]). These questions serve as a representation of the information need of the user as described by the "Cranfield paradigm" (Rodriguez et al., 2021 [25], page 1 paragraph 2). The Cranfield paradigm contrasts the Manchester paradigm that uses question answering to perform the Turing test that tests intelligence of the system providing the answers. The Open Domain here refers to answering any question with no predefined context. In recent past, the development of Knowledge Bases (KB) has improved the accuracy of question answering systems

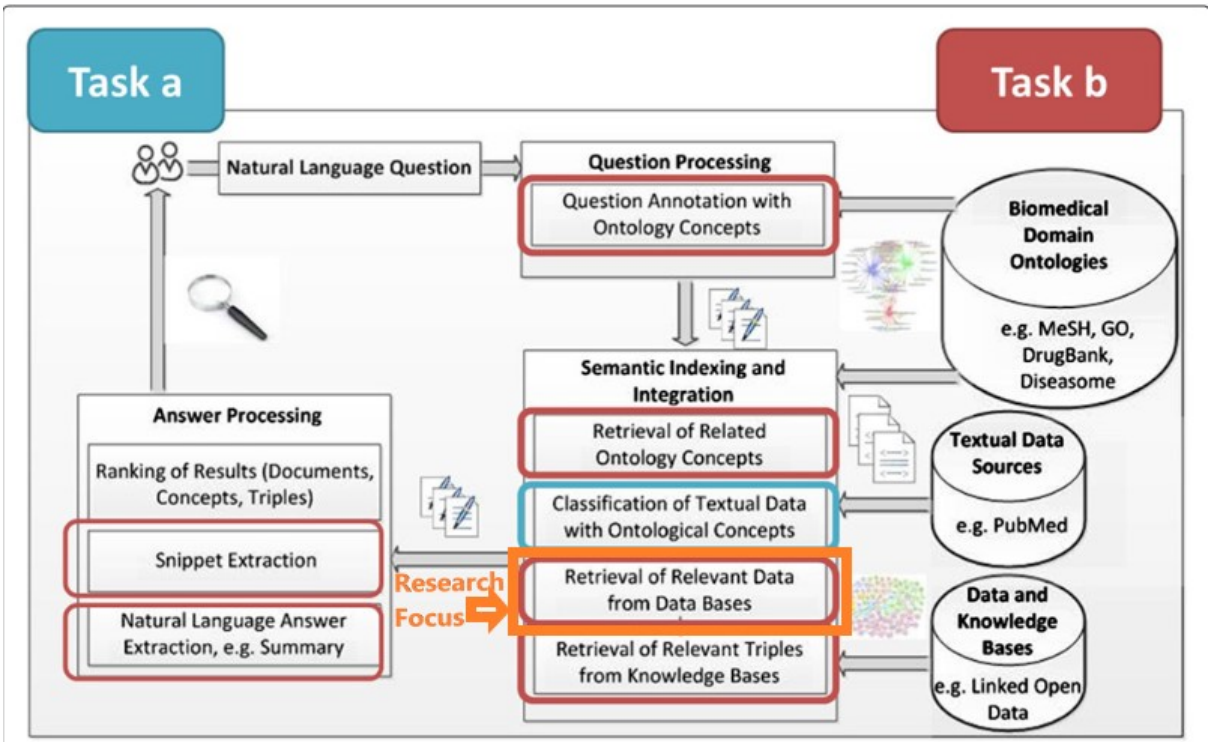

FIGURE 2.1: Semantics Indexing and Question Answering for BioASQ Challenge (Source: [2])

by using restrictive scope of search and "using predefined structured KB that is often manually constructed" (Zhu, 2021, page 1, paragraph 2 [3]). However, KB are "inherently limited due to incompleteness and fixed schema" (Chen et al., 2017 [4]). The recent developments in deep learning have improved machine comprehension of text and have therefore improved the accuracy of the ODQA based on raw unstructured text.

The ODQA architecture has been in constant evolution and has evolved from a complex and complicated multi-component system to a dual component pipeline system and recently to one component system only. This section describes the two later architectures and explores the research for improving the context search and retrieval component of a two-component pipeline system.

2.2.2 Move towards architectural simplification

Historically, there has been considerable research in the field of Question Answering. However, the current phase of the open domain Question Answering architecture relies on the use of deep learning to simplify the QA architecture. The current architecture has gradually evolved from the old multi-stage multi-component architecture to a dual component and is finally evolving towards a single component.

Traditional QA systems had limited success due to limited and closed domains. They were mostly rule-based and had a fixed schema. They were also not data-driven and suffered from syntactic and semantic disambiguation (Simmons, 1965 [26]). These systems had rigid information encoding, fixed information schema, and a uniquely structured

retrieval system, and required manual updates to add information. A data-driven system that can read unstructured information is more scalable and maintainable and therefore more useful. As shown in Figure 2.1, the current biomedical information data sources are not open domain but they do not suffer from rigid information encoding and uniquely structured retrieval systems. Hence, techniques learnt from ODQA can be used for BioASQ document search in task b. **It is envisaged that the well defined ontology hierarchies, classification (based on MesH) and Resource Description Framework (RDF) has discriminating information, that can be potentially used for search contextualization, and combined with ODQA methods to search and retrieve answers to a question**. However, this embedded information may not be easily accessible for performing an efficient search.

Three stage QA Architecture — Question Analysis, Retriever and Reader

The pipeline based development phase in this domain was instigated by the Text Retrieval Conference (TREC) QA Tracks (1997-2007). A typical TREC-QA pipeline is displayed in Figure 2.2.

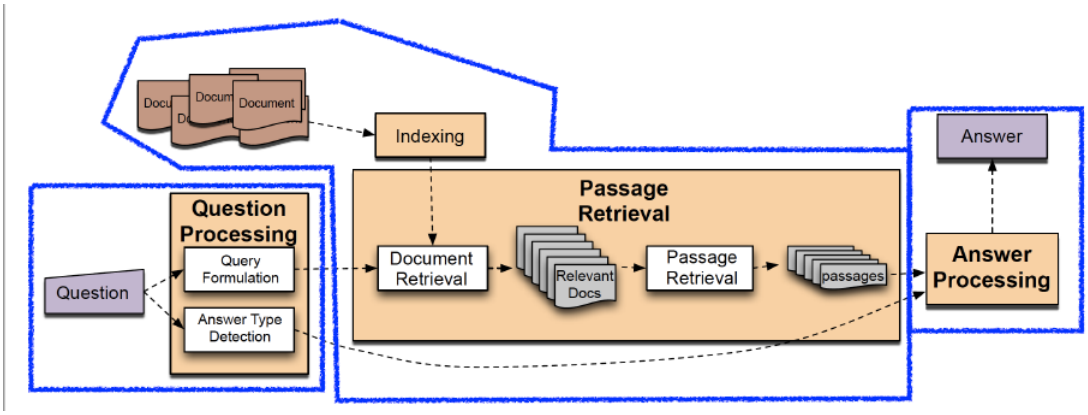

FIGURE 2.2: A TREC QA Pipeline (source: Jurafsky and Martin. Speech and Language Processing. 3rd edition)

These solutions based on pipelines were an improvement on the old systems and were built using a pipeline that had three stage components, namely Question Analysis, Document Retrieval, and Answer extraction (Zhu 2021 [3]).

The first component, namely Question Analysis, used proprietary algorithms and heuristics to restructure the query that could be used by the retrieval component. Linguistic techniques such as tagging, stemming and classification methods (Zhu et al. 2021 [3]) were used to analyze the question. The second component called Document Retrieval uses a re-structured query to search the corpus for the documents that are likely to contain the answer and therefore contextualize the answer search. The third component uses various methods to extract the answer from the context documents retrieved from the retriever. However, these tightly dependent components are difficult to maintain and

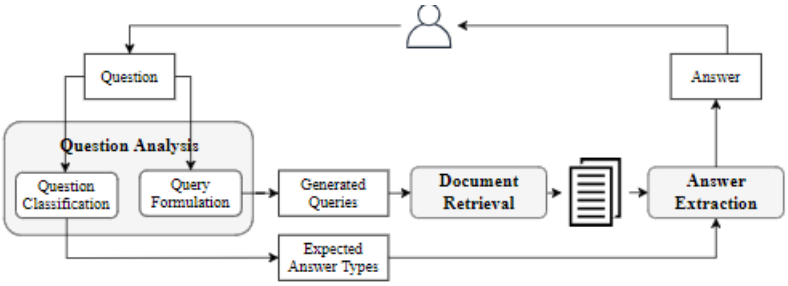

FIGURE 2.3: A three stage Open QA architecture(source Zhu et al. 2021 [3])

modern systems have replaced them with systems that are loosely coupled and use well known industry standard search and retrieval methods such as TF-IDF/BM25, semantic search and deep learning based reading comprehension.

Two-stage Architecture - Retriever and Reader

The three-component based tightly coupled and complicated QA pipeline further evolved into a simplified two-component namely **Retriever** and **Reader**. The first component, Retriever, searches the text corpus for the set of relevant documents, and the second component, Reader, extracts the answer from these contextualized retrieved document using deep learning methods such as LSTM and Transformers.

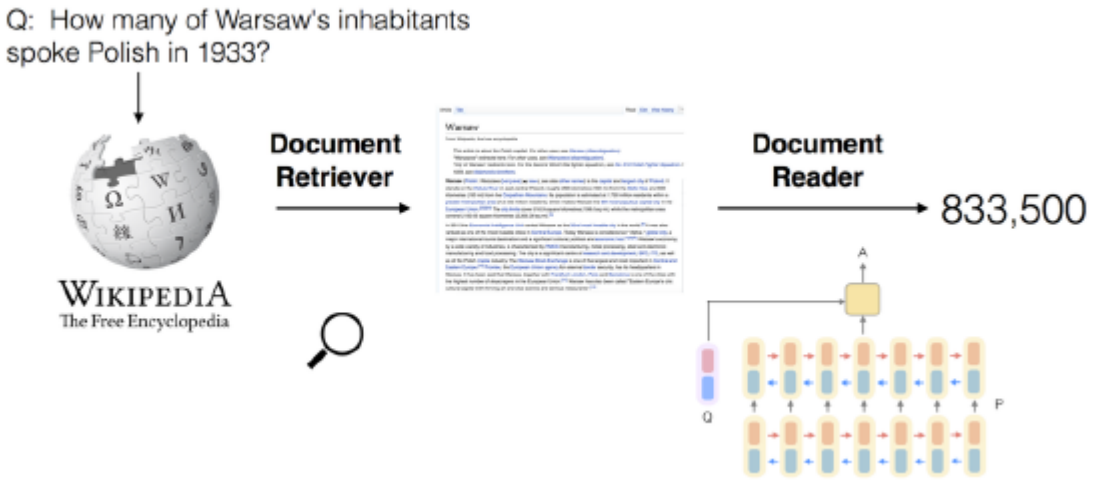

FIGURE 2.4: A two stage (Retriever - Reader) QA Architecture Source(Chen et al., 2017 [4])

Various techniques (models and systems) such as TF-IDF/BM25, probabilistic Okapi BM25, Anserini (Yang et al., 2019 [5]) are used to retrieve documents that have high likelihood of containing the answer.

Chen et al., 2017 [4] pioneered an Open Domain Question Answer (DrQA) architecture that used neural networks for the reader component. DrQA uses a novel distant supervision technique (Mintz et al., 2009 [27]) to provide training examples for this reader. Distant

supervision technique is used to overcome the lack of associated training documents for the question answering training data in CuratedTREC, WebQuestions and WikiMovies datasets. These training datasets only have question and answer pairs. Distant supervision technique was originally used to extract the entity relationship of a word pair by using sentences that have these word pairs. These selected sentences from a large unlabeled corpus are used to extract features that are later combined in a logistical regression classifier (Mintz et al., 2009 [27]) to extract the entity relationship. Chen et al., 2017 [4] used distant supervision technique in the domain of question answering to find associated training documents for a question answer pair. Subsequent research has discovered many other new novel methods to increase overall performance by using enhanced retrieval techniques such as BERTSerini (Yang et al., 2019 [5]), multiple passage training (Clark et al., 2018 [28] and Wang-et al., 2019 [29]), retrieved passage and answer re-ranker (Wang et al., 2018 [30]), and finally using expectation maximization methods to select an appropriate answer amongst the multiple candidates (min et al., 2019 [31]). The ranking of the retrieved document set can be further enhanced by Iterative Document Ranking (Zhang et al., 2021 [32]).

These enhanced retrieval techniques start with simple but better retrieval methods and progressively introduce other enhancement methods to increase the retrieval performance. As shown in Figure 2.5, BERTSerini uses a two-component architecture that uses an Anserini (Yang et al., 2017 [5]) retriever and BERT reader, hence the name BERTSerini. Anserini provides API wrappers for Apache Lucene. Lucene is an industry standard platform for developing search applications.

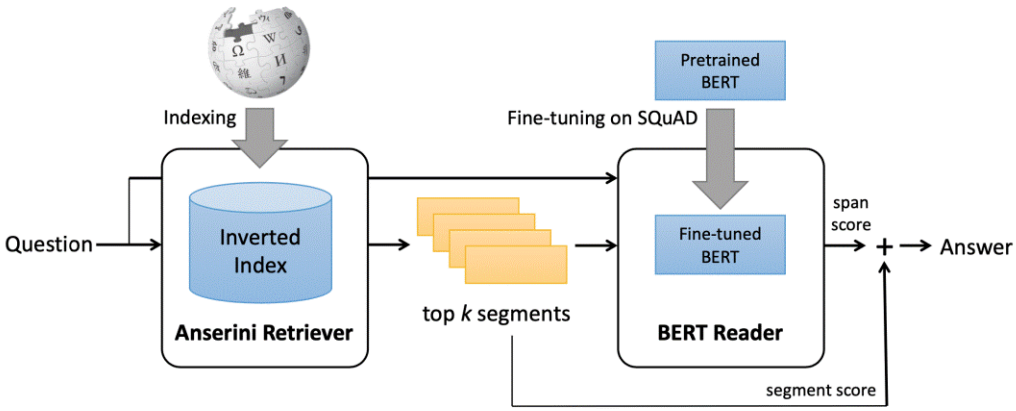

FIGURE 2.5: BERTSerini Architecture (Source Yang et al., 2019 [5])

Clark et al., 2018 [28] used a paragraph-level question answering model instead of document-level models. They improved the retrieval results by training the model to ignore paragraphs that do not contain answers.

Wang et al, 2018 [30] introduced Reinforced Ranker-Reader method to improve retrieval results. The retriever ranker and the reader components compare the question to the documents in the document set to ascertain the match quality of these retrieved documents. These documents are then ranked based on the match quality.

To overcome the weak supervision provided by various question answering tasks (such as many candidate answers in a document), Min et al., 2019 [31] suggested expectation minimization methods to select the most likely answer.

One stage, Retriever free architecture based on Transformers

The recent enhances in deep learning, semantic search based on dense vectors and language models based Transformers have further simplified the QA architecture by proposing the use of pretrained language models as Knowledge Bases (Petroni et al., 2020 [33]). It is assumed that these models such as GPT 2 and GPT 3 have learned all the information in the corpus and stored it in the billions of the model parameters. It is assumed that the information stored in the model parameters can be retrieved (as an answer to a question) without any further fine-tuning of the model. However, the results of zero-shot (no additional fine-tuning) are not competitive yet. Brown et al., 2020 [6] , Figure 2.6 have achieved better results by using few-shot training on the GPT-3 language model. This "over-reliance on parametric knowledge" (Longpre et al., 2021, page 1, paragraph 1 [34]) fails when the contextual information provided for Questions Answering contradicts the memorized parametric knowledge of large language models. These Zero shot or few shot Question Answer systems are also computationally very expensive. However, recent research about non-answerable questions can "effectively trade off lower computation cost of QA systems for lower Recall, e.g., reducing computation by approx. 60 percent, while only losing approximately 3 to 4 percent of Recall."(Garg et al., 2021 [35]).

There is considerable research going on in this field and, if successful, this will be the culmination of the simplification of the architecture, where two components, namely, retriever and reader are finally reduced to a single component that is neither a retriever nor a reader.

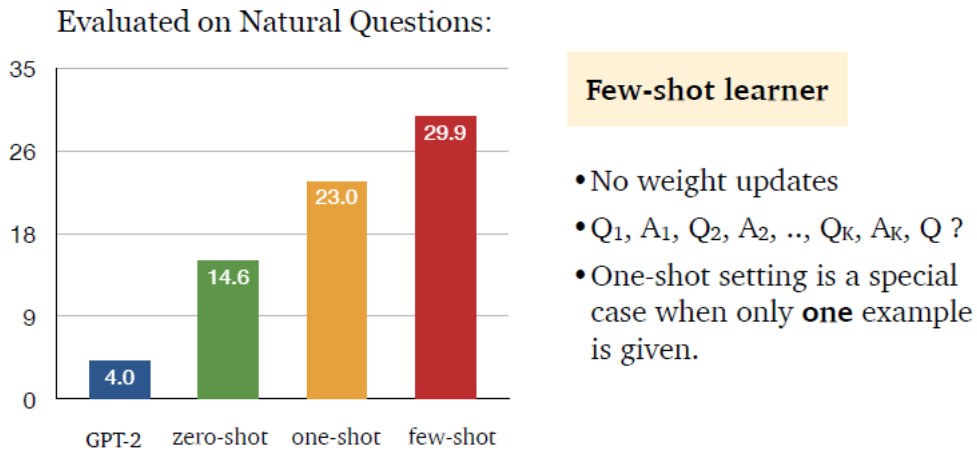

FIGURE 2.6: Fine-tuning improves the performance of GPT-3 Large Language Model. Source: Brown et al. 2020, language Models are Few-Shot Learners [6]

Roberts et al. 2020 [36] used Salient Span Masking (Guu, et al., 2020 [37]) to

fine-tune the Text-to-Text Transfer Transformer (T5) (Raffel et al., 2019 [38]) model to demonstrate that a large T5 model can match other deep learning based QA models. However, the feasibility of language model as KBs is still not confirmed (Wang et al., 2021 [39]) and it remains a challenge for these models "to remember training facts in high precision" (Wang et al., 2021, page 1 paragraph 1 [39]).

2.3 Recent deep neural networks based developments in Retriever Reader Architecture

2.3.1 Using Dense Representation for Retrieval

The two-component ODQA pipeline described in Section 2.2 relies on a traditional retriever that uses keyword-based search such as TF-IDF or BM25 weighted term vector model over unigrams and bigrams of the document or paragraph. These n-gram models are based on bag-of-words document representation. In bag-of-words, each word is represented as a vector of zeros with only single 1 that identifies the word of that vector. Hence this vector length is as long as the length of the finite vocabulary. This one-hot vector representation produces orthogonal vectors and does not capture semantic similarities between words. It is computationally expensive to train NLP models based on one-hot vector representation. These sparse one hot encoded vectors do not encode the words and the NLP features efficiently. These keyword-based methods using one-hot vector are inefficient to store and process semantically similar but different keywords. Mikolov et al., 2013 [40] proposed another coding method that encodes words in a low dimensional vector where each value of this vector is a real number value and not a binary value (0 or 1). This representation of each word as a dense vector with real values is learnt using neural network training. The dense vector representation of the word uses distributional semantics and exploits the representational power of the neural networks that captures similarities and syntactical features of words. This dense vector representation of a word is called word embedding. Dense vectors also allow information composition where multiple dense vectors can be processed and combined by a neural network to aggregate the information in the same dimensional vector. This aggregation can be used to represent several words in a single dense vector.

Research has established that dense vectors can be used to establish text similarity (Yih et al., 2011 [41]). As discussed in the sections above, the representation of words in a dimensional space allow better semantic and relational representation of the word and it is therefore easier to search words with similar meaning and semantic similarity. The similarity scores based on dense vectors can be used for identifying answers in the corpus where keywords do not match. However, it is very expensive to create dense representations of the whole corpus by using labeled question and answer data. The availability of language model training methods (such as cloze and inverse cloze task), pre-trained models and tools (e.g. FAISS) for fast maximum inner product (MIPS) (Johnson et at. 2021 [42]) and transfer learning have made it feasible to use these dense vector and neural networks for training the Retriever component.

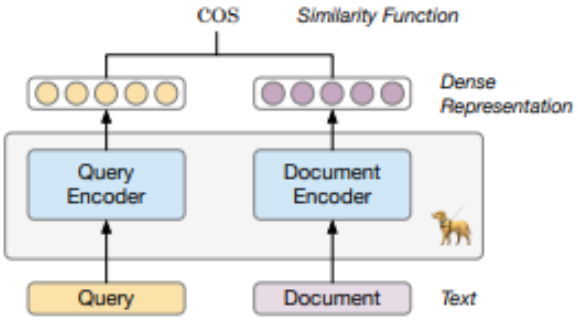

FIGURE 2.7: Dual tower architecture for text retrieval: Source (Xu et al., 2022)[7]

Lee et al., 2019 [43] proposed and built an Open-Retriever-Question-Answering (ORQA) system by using the novel technique of Inverse Cloze Task (ICT) for training dense vectors. ICT predicts the context (paragraph) given a sentence. It is different from the cloze tasks in language models where the model predicts the missing language item (such as a word). ORQA uses ICT to pretrain dense vectors and then uses question answer pairs to fine-tune these pre-trained vectors. ORQA was further enhanced by the Retrieval Augmented Language Model (REALM) (Guu et al., 2020 [37]) that uses the well-known word mask Language model (MLM) technique. This MLM technique is used to provision more data and therefore scales up the available data for training.

As mentioned above, it is expensive to train document embeddings for dense vectors. Dense Passage Retrieval (DPR) by Karpukhin et al., 2020 [22] proposed and successfully demonstrated that it is possible to "Successfully train a dense retrieval only from small number of Q/A, without any pre-training" (Karpukhin et al., 2020, page 1, paragraph 1 [22]). ORQA uses ICT to pretrain the dense vector and then use question answer pairs to fine-tune the trained vectors. DPR improves upon this by leveraging the standard BERT pre-trained model and the industry standard dual encoder architecture. In a dual architecture model , (Figure 2.7), one encoder e.g. a BERT Transformer, encodes a question and other encoder encodes a paragraph and a similarity score is calculated by vector dot product between these two vectors. DPR fine-tunes the standard BERT-based based dual encoders with an existing limited number of question answer pairs. Hence, DPR saves training data and resources by not doing the ICT-based pre-training as performed in ORQA. DPR further reduces the fine-tuning data requirement by using a novel technique of in-batch training that reuses the already provided data for negative examples. Every paragraph has an associated correct answer and this paragraph can be used as negative data for all other answers. They further discovered that a two-phased pipeline with a separate retriever and a reader training delivers better results than the joint training of retriever and reader.

As another method to overcome the lack of training data for domains that do not have large training sets, Wang et al., 2021 , [44] proposed a generative pseudo labeling for domain adaptation of dense retrieval. This delivers a dense retrieval method that can have high tolerance for domain shifts.

2.3.2 Other Deep Neural Networks methods to improve Retrieval performance

The usefulness of dense representation mentioned in the above sections has spurred further research in the field. This research explores the use of dense networks for improving the performance of the retriever component of the retriever-reader pipeline. This stream of research recognizes that the dual encoder BERT architecture does not deliver high performance due to "query-agnostic answer encoding and its over-simplified matching function" (zhao et al., 2021 [45]). Zhao et al. 2021 proposed a new method that exploits BERT's cross attention features (instead of simple dot product of the query and answers) at the terms level (instead of attention at sequence level) and retrieval ranking. It also uses inverted index (using Lucene) for fast retrieval at runtime. It is likely that this approach of using deep neural networks for retrieving candidate documents and ranking them can be used for improving the efficiency of document retriever for the BioASQ challenge as well.

As mentioned in the joint training section below, the use of expectation-maximization (Sachan et al., 2021 [46]) in an end-to-end trainable system also delivers a better retriever component that finds more relevant documents.

The Retriever can also be improved by using context in the calculation of term frequency. Term frequency calculation is generally independent of the question context. Calculating term frequency in cognizance of the context is expected to provide more information to the document matching algorithm. Dai et al., 2020 [47] defined this as "Deep Contextualized Term Weighting framework that learns to map BERT's contextualized text representations to context-aware term weights for sentences and passages" (Dai et al. 2020 page 1, paragraph 1 [47]). It further enhances the passage retrieval efficiency by storing term weights in an inverted index database. This can also be evaluated for increasing the efficiency of the retriever component for the BioASQ challenge

Xu, 2021 [48] has demonstrated that the pre-trained language model can be enhanced by using external lexical and syntactical features. Xu F, 2022 further demonstrated that adding structural hierarchies and localities to the learned parameters also improve retrieval results (Xu F, 2022) [24].

A Joint Passage Retrieval model (min et al., 2021 [49]) has also delivered improved retrieval results by using an autoregressive reranker, that ranks the multiple candidate passages retrieved from a corpus with dense representation and retrieval based on DPR (karpukhin-et al. 2020 [22]) and REALM (Guu, et, al. 2020 [37]).

A multi-hop Dense retrieval method proposed by Wenhan et al., 2021 [8] has also demonstrated good performance for retrieving documents that have a higher probability of containing answers from a dense representation corpus. This method is very successful in retrieving documents that answer complex questions that involve "aggregating information from multiple documents, requiring logical reasoning or sequential (multi-hop) processing" (Wenhan et al., 2021 [8], section 1 paragraph two).

Zhang et al. [50] proposed an Answer Support-based Reranker (ASR) to identify the best answer candidate amongst the set of K top documents provided by the retriever. It builds a three-way classifier that determines "if an answer supports, refutes, or is neural

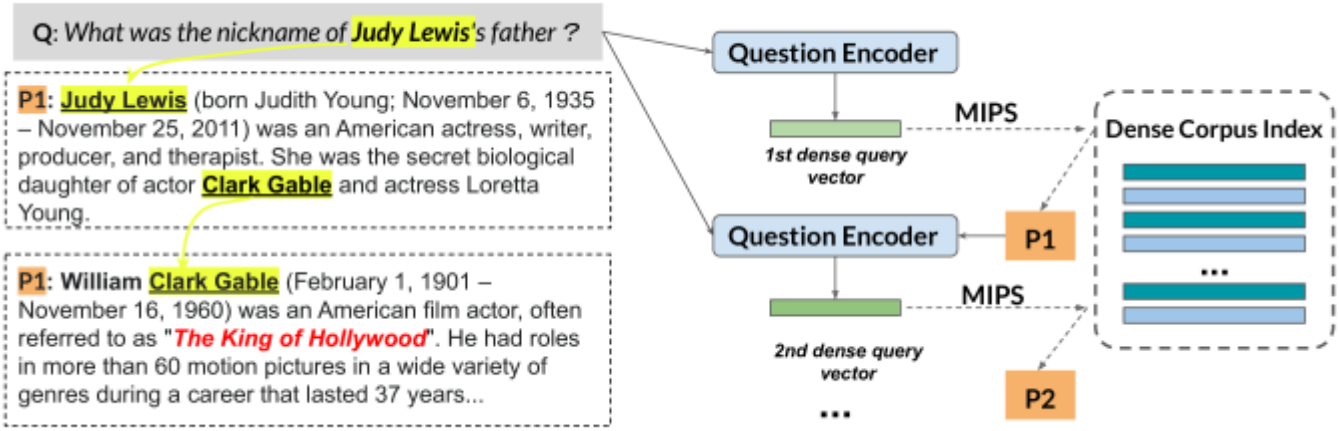

FIGURE 2.8: Multi-Hop Dense Retrieval Approach - Source: Wenhan et al., 2021 [8]

with respect to another one" (Zhang et al., 2021 [50]).

Another method called Generation-Augmented Retrieval (GAR) (mao-etal-2021 [3]) enriches the query by combining the query text with additional text generated from Pretrained language models (PLM), has also demonstrated good results. It "expands it with heuristically discovered relevant contexts, which are fetched from PLMs and provide richer background information" (mao-etal-2021 [3]).

Xu et al., [7] proposed novel training methods for dense method based retrieval to create a state of the art zero shot text retrieval.

2.3.3 Joint Retriever-Reader training and Generative Answer

Retrieval-Augmented Generation (RAG) (Lewis et al., 2020 [10]) proposed a recently developed text generation technique for generating answers for the questions that are encoded in a query encoder. This novel technique does not use reading comprehension neural networks to extract the answer span from a passage, but generates an answer using the recent text generation techniques. The authors demonstrated a joint training of retriever and reader and used DPR for retrieving text and used Bidirectional encoder and BERT based Auto regressive decoder for pretraining. Alhough Dense encoding and new Language model techniques are recent technological breakthroughs, they still do not deliver competitive performance in all domains (Thakur et al., 2021b, [51]). Seo et al., 2019 [9] demonstrated that using both Sparse and Dense (DenSPI) encoding at the phrase level can deliver better results. This is achieved by phrase level encoding and phrase indexing of the Information. It uses nearest neighbor search and encoded question to locate the answer in the encoded phrase space. Though it provides better performance, it requires sparse and dense embedding of all the phrases in the Corpus. Lee et al., 2020 [52] further improved the DenSPI by using contextualized sparse representation. This method provided better context to the sparse representation of the phrase level text. It uses a new method that, instead of calculating high weight for infrequent terms in TF-IDF, Sparse computation is driven by semantically related n-grams. This technique "leverage

rectified self-attention to indirectly learn sparse vectors in n-gram vocabulary space" (lee et al., 2020 [52]).

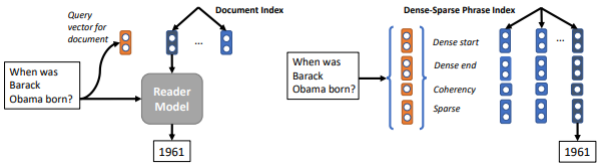

FIGURE 2.9: A Pipeline Vs. Retriever only QA Architecture proposed by Seo et al., 2019 [9]

Cheng et al, [53] proposed an ensemble model called UnitedQA (cheng et al., 2021 [53]) for combining answers from extractive and generative methods of answer generation. This model is built upon Electra and T5 pretrained models and produces results that beat the state of the art QA performance on Natural Questions and TriviaQA.

The joint Retriever-Reader training is also impeded by the lack of QA training sets that can train for unanswerable questions. Asai et al., 2021 [54] used an answerability prediction parameter that can be used to identify improvements in the current set of QA databases and QA training models.

Sachan et al., 2021 [46] used expectation maximization for an end-to-end differentiable training method. This method uses a latent variable that is used to iteratively update parameters. This enables the retriever to retrieve more relevant documents for the answer and it delivers better results compared to similar end-to-end trainable QA Systems.

2.3.4 Using Knowledge Bases for Question Answering

This chapter started with limitations of knowledge bases due to the difficulties in maintaining them and their closed nature. However, the new techniques based on Neural Networks can be used to train retrievers that can identify and retrieve documents from these knowledge bases. In language models, information is stored in the model parameters. However, this knowledge is limited by the size of the language model. Generally Knowledge stored in knowledge graphs can be retrieved efficiently using graph traversal methods. However, knowledge sourced from raw unstructured data and stored in language models is relatively difficult to retrieve.

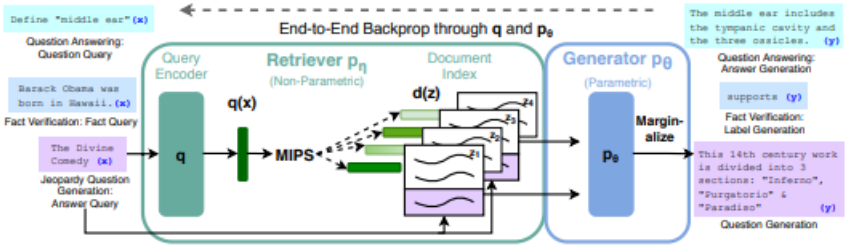

FIGURE 2.10: Retrieval-augmented (Lewis et al., 2020 [10])

The current research to add knowledge to language models can be divided into three methods, they are:

- Adding pretrained entity embedding
- Using external memory
- Training a neural retriever by using modified training data

Zhang et al., [17] pioneered the Enhanced Language Representation with Informative Entities (ERNIE) that enhance the knowledge content of the Language models. Logan et al., [55] developed a knowledge graph language model (KGLM). This feature enables selection of the information from an external knowledge graph. Khandelwal et al., [56] used Nearest Neighbour Language Models(kNN-LM) to retrieve information from a language model. The Retrieval Augmented Language Model (REALM) (Guu et al., 2020 [37]) proposed an external neural document retriever that fetches the documents from the a corpus using dense vectors. Lewis et al., [10] further extended this neural network based document retriever with retrieval-augmented answer generation(RAG).

2.4 Summary and Future Directions

The Question Answering systems have benefitted immensely from the new research in NLP. The multistage complex Question Answering system has been evolving towards simpler systems that are based on simplified processing pipelines and dense vectors.

There has been considerable research for using universal language models for NLP and their use for Open Domain Question Answering systems. Although these universal models learn information using millions of parameters, their performance is not comparable to two-component pipeline systems in Bio-Medical domain (Table 9, Wang et al., 2022, [44]). However, the recent newer Large Language Models with billions of parameters may have better performance.

Hofstatter et al., [57] identified and addressed key performance attributes of dense search. The use of dense vectors for the retriever component of the QA pipeline has delivered better results in some domains. However, these results do not translate to BioASQ questions where keyword-based retrieval delivers better results (Table 9, Wang et al., 2022, [44]). This can be attributed to medical keywords such as MeSH words embedded in the questions. The dense vectors and use of self-attention and cross-techniques as used in Transformers has also delivered good results. As part of this research, I ran several experiments to ascertain if **knowledge-enhanced dense vectors** (obtained from language models) and **attention methods, e.g. ColBERT** can deliver better document retrieval performance. The experiment used existing biomedical BERT models and term attention methods, e,g, ColBERT and recorded their performance. However, the experiment confirmed that there are considerable challenges in incorporating biomedical knowledge into language models.

2.4.1 Pictorial Representation of QA Document Retrieval Research

The figure 2.11 captures the key four pillars of the contemporary document search research as described in this chapter. As seen in the figure, the current research can be divided into four categories listed below:

- Using dense vectors to find similarity using nearest neighbour. This is in contrast to the keyword matching based search.
- Using a combination of keywords and dense vector to find a document. This is called sparse vector search. Both dense vector and sparse vector techniques use information contained in the document itself.
- query contextualization

  in contrast to dense vector and sparse vector techniques query contextualization embellish the search by using context of the search. This context can be either generated from the document or can be externally provided.
- Using external knowledge-base to assist document search.

FIGURE 2.11: Key Pillars Of Document Search Research

"Any idiot can discover such thing by accident. I was the one who discovered it by reason, which requires genuine originality"

Galileo, when questioned about the invention of the telescope, 1609 at Padua, Italy

3 Research Methods

3.1 Approach

This chapter describes the recent dense vectors based research that can be used to build a document retrieval system for BioASQ challenge.

The use of deep neural networks and dense vectors, sometimes complemented with sparse vectors, has revolutionised the information retrieval field. Semantic search uses context and relationships to search information and is more successful where there is no lexical matching of the question and the document containing the answer to the question. There have been several attempts of using dense representation of document embeddings and transformers based language models to retrieve information from a corpus. For example, Dimitriadis and Tsoumakers, 2019 [58] used document embeddings and external resources for the answer processing stage for biomedical questions.

Based on this research, I experimented to build an efficient BioASQ document retrieval system by combining the following independent ideas.

- A document retrieval pipeline that uses lexical term matching as the first phase of document retrieval from PubMed. As stated in section 2.4, terms based search e.g. BM25, delivers better results compared to dense vector based search (e.g. nearest neighbor). However, the study does not focus on this first phase of the pipeline. The study concentrates on the new emerging dense vector based research and the associated methods.

- Apply dense vector based techniques to the set of documents retrieved in the first phase of the pipeline. Different dense vector embedding and matching techniques (Sentence BERT and ColBERT) are applied to search for a good vector matching algorithm.

- Use language models trained on biomedical data to create dense vector embedding for queries and document terms. It is envisaged that vector embeddings created by models trained using biomedical Data may capture contextual biomedical knowledge that may lead to more accurate vector matching. BERT based masked language model encoders are used to create document embeddings. However, the experiment also used biomedical domain large language models to create document embeddings.

- Use query and document term level interaction (ColBERT) to select a set of documents that may contain an answer to the query.

3.2 Using BERT for Document Search

This section briefly describes the use of BERT based technologies to create word embedding and their use for document retrieval.

In general, the use of pre-trained transformers for search has delivered much superior results (see figure 3.1) as compared to non-neural and BM25 based models.

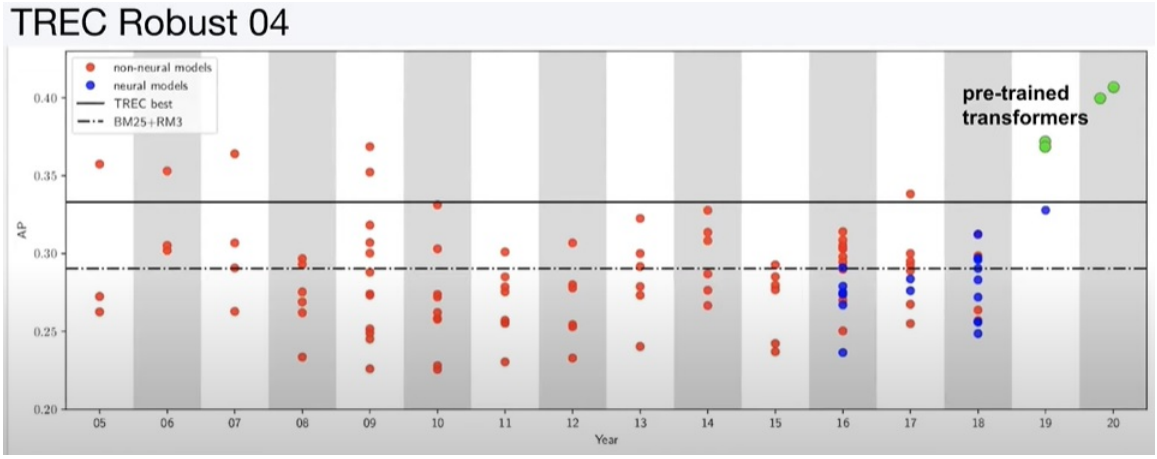

FIGURE 3.1: Effectiveness of pre-trained models: source (Yang et al., SIGIR 2019, "critically evaluating the hype of neural Hype and Pinecone presentation") [11]

Lee et al., 2020 [12] trained a BERT-based domain specific model called BioBERT (figure 3.2), that was pre-trained on biomedical corpora. This model delivered enhanced performance on biomedical text mining tasks such as entity recognition, relation extraction and biomedical question answering. However, these tasks required further fine-tuning of the BioBert model.

Chakraborty et al., 2020 [13] enhanced the BioBert model by using larger and diverse corpora (called BREATHE) of medical documents and fine tuned the resulting model called as BioMedBERT (see figure 3.3) and BM25 based ElasticSearch method and a hierarchical search ranker to enhance results. The second phase of this hierarchical search uses the BioMedBERT to extract the embedding of the document returned earlier and computed

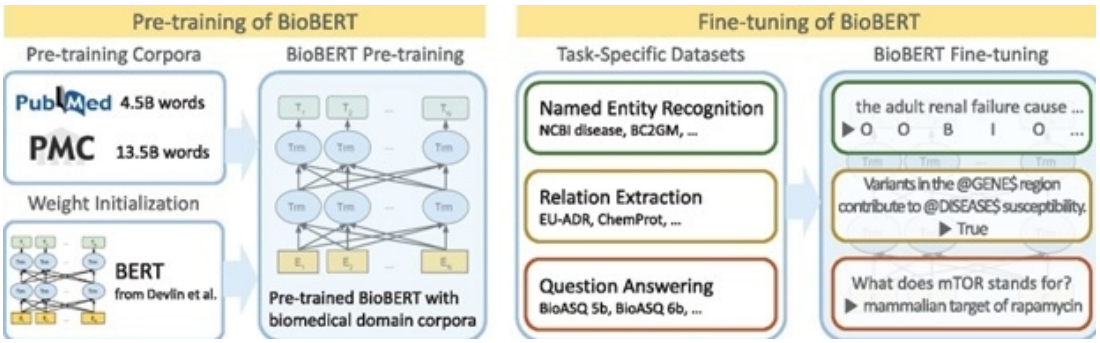

FIGURE 3.2: bioBERT and its fine-tuning for biomedical question answering: Source (Lee et al.,2020) [12]

cosine similarities of the query and the documents to return the most relevant documents.

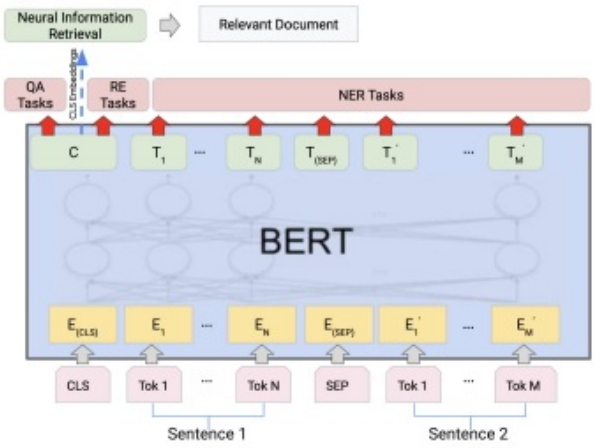

FIGURE 3.3: bioMedBERT Architecture: Source (Chakraborty et al., 2020) [13]

These methods mentioned above use deep neural networks based techniques and indeed deliver better retrieve performance as measured by MRR@10, however they are slow and inefficient. There have been several other different methods that enhance search effectiveness and efficiency.

3.2.1 General Neural Re-Ranking Models

General Neural Re-Ranking models (see figure 3.4) use re-ranking techniques where a first stage ranker (generally based on BM25) is used to select a set of document and the second stage neural re-ranker further refines this set by using transformer based attention models.

Rosa et. al., [59] demonstrated that Re-ranker based models generally outperform the dense models of similar size in several tasks. All neural re-ranking models are based

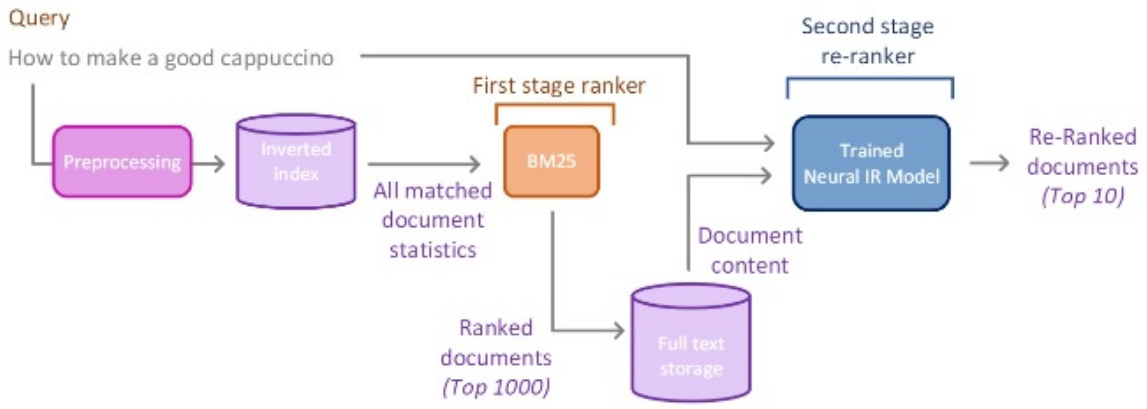

FIGURE 3.4: Neural Re-Ranking Models: Source (hofstaetter, 2022) [14]

on a matching process (see figure 3.5) that is learnt using deep neural networks.

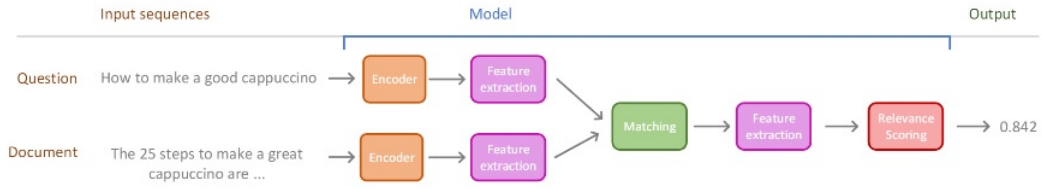

FIGURE 3.5: Neural Re-Ranking process and matching: Source (hofstaetter, 2022) [14]

The query and document terms are dense encoded and then matched using cosine similarity and this cosine similarity matrix is further processed by CNN (see figure 3.6) and multi-layered feed forward module that outputs a similarity score. The efficiency of neural ranking model can be further enhanced by using kernel based neural ranking Model (KNRM) (Xiong et al., 2017) [60] (see figure 3.7).

This combination of kernel and neural networks delivers its best performance by using Conv-KNRM (Dai et al., 2018) [61] where KNRM is performed before the match-matrix.

3.2.2 Re-Ranking using BERT

The popular transformer model BERT has also been used for Re-Ranking and this has delivered good results as measured by MRR@10. It uses concatenation of the query and the passage and its processing by BERT. The output from the final layer of BERT produces a sequence of vectors. The first vector of this sequence is called as CLS. This special vector represents aspects of the combined information contained in all the subsequent vectors in the sequence. The CLS token of the BERT is fed to a Feed Forward Neural network to train for similarity (Jiang, et al.,2021 [62]). BERT allows the use of representation-focused

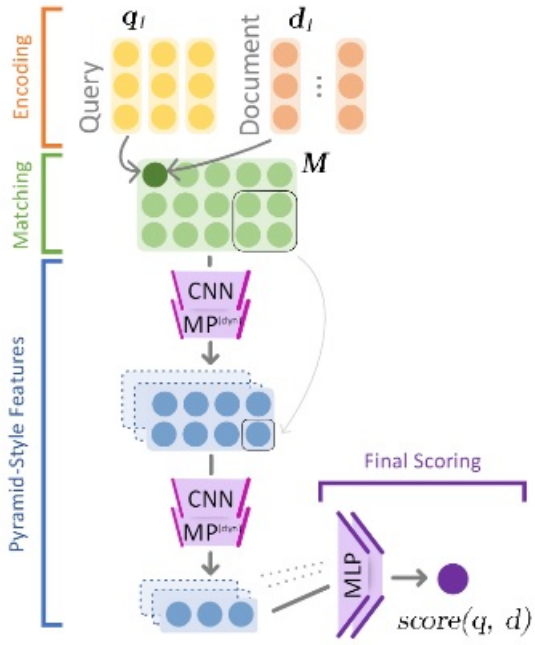

FIGURE 3.6: Query Matching and CNN: Source (Hofstaetter, 2022) [14]

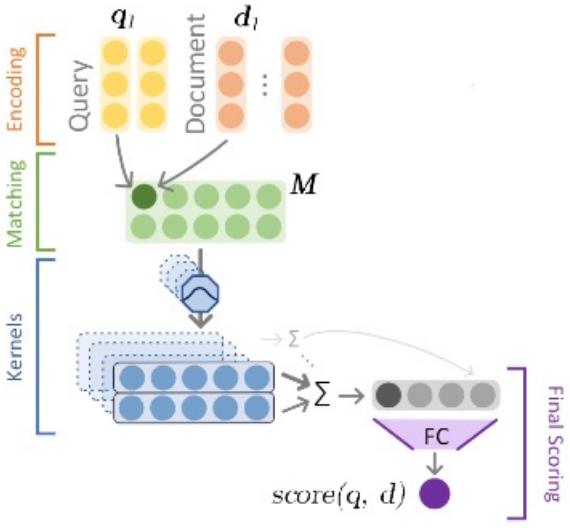

FIGURE 3.7: Neural Re-Ranking Process and matching using Kernels: Source (Hofstaetter, 2022) [14]

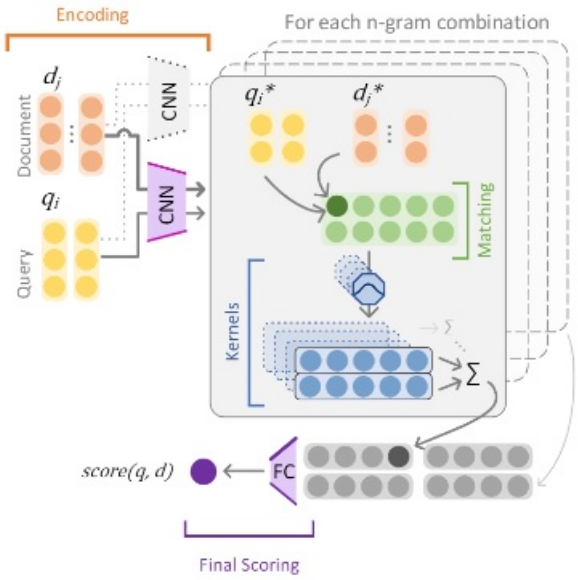

FIGURE 3.8: Conv-KNRM: Source (Hofstaetter, 2022) [14]

that represents the document and the query in a single or multiple embedding in the latent vector space (Tonellotto, 2022) [15]

Although BERT delivers very high MRR@10 scores, it has a relatively higher latency compared to several other non Re-Ranking methods.

This latency is caused by the online computation of the dense vectors for the query and all the documents in the query set returned by the first BM25 based document ranker. ColBERT (Khattab et al., 2020) [1] improves the efficiency (see figure 3.16) of the Re-Ranker by offline computation and indexing of the document corpus. ColBERT is described in section 3.2.3. Various models such as ColBERT, Sentence-BERT and Conv-KNRM described by Khattab, Reimers, and Hofstaetter respectively highlight the additional information captured by the term level interactions at the output layer of an encoder Transformer. The Transformer captures the contextual self attention information and relationships of the terms in its internal layers, but the research above points to additional value generated by calculating the interactions and matching of these document embeddings at the top layer.

3.2.3 Term Interaction Models

This section briefly describes two key models that further process the document embeddings output of a BERT layer. BERT internal layers exploit self-attention to discover and represent the lower level composition and structures present in the document. However, the embeddings output from the top layers can also be aggregated using various operations

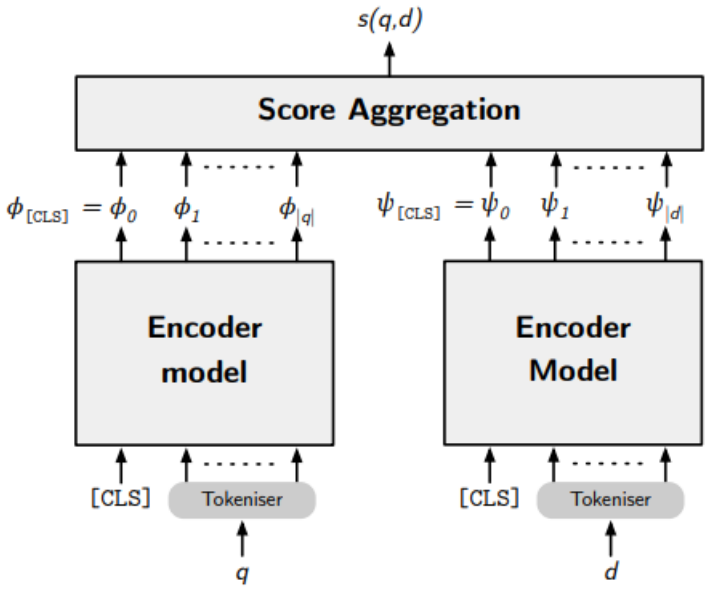

FIGURE 3.9: Representation-focused system: Source (Tonellotto, 2022) [15]

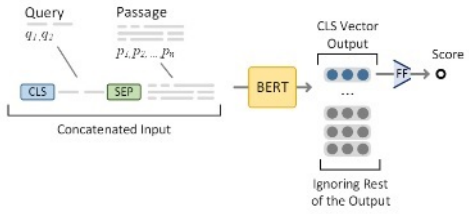

FIGURE 3.10: Conv-KNRM: Source (Hofstaetter, 2022) [14]

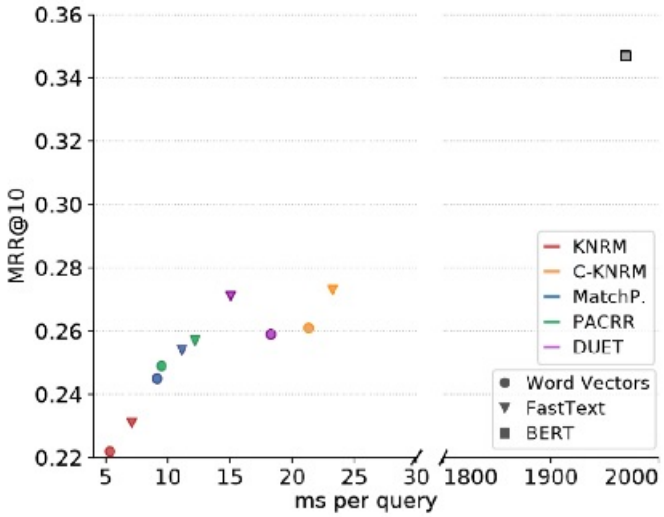

FIGURE 3.11: BERT Efficiency: Source (Hofstaetter, 2022) [14]

to obtain additional information that can be used for various NLP tasks.

Sentence BERT

SBERT enhances the word embedding output of BERT by pooling the output to create a fixed size sentence embeddings (Figure 3.12). The simple pooling methods are:

- CLS tokens
- mean of all output embeddings
- maximum of all output embeddings (Reimers et al., [16])

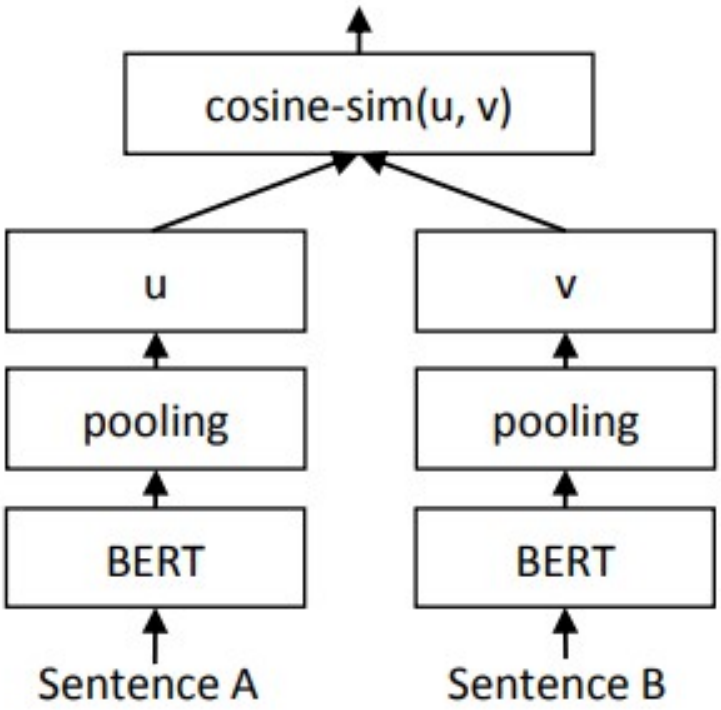

FIGURE 3.12: Sentence BERT architecture: Source (Riemers et al.) [16]

ColBERT

ColBERT (Contextualized late interaction for BERT) is another retrieval re-ranking algorithm based on term interaction model and it "introduces a late interaction architecture that independently encodes the query and the document using BERT and then employs a cheap yet powerful interaction step that models their fine-grained similarity" , (Source (Khattab et al.) [1]). ColBERT calculates a simple maximum of all terms of the query embeddings (called as MaxSim) and the document embeddings. It later creates a score

by summation of all these MaxSim terms (please see Figure 3.13 and Figure 3.14). Using this simple operation, ColBERT has delivered good performance as compared to other retrieval systems including BM25 and ConvKNRM Figure 3.7 as shown in Figure 3.15.

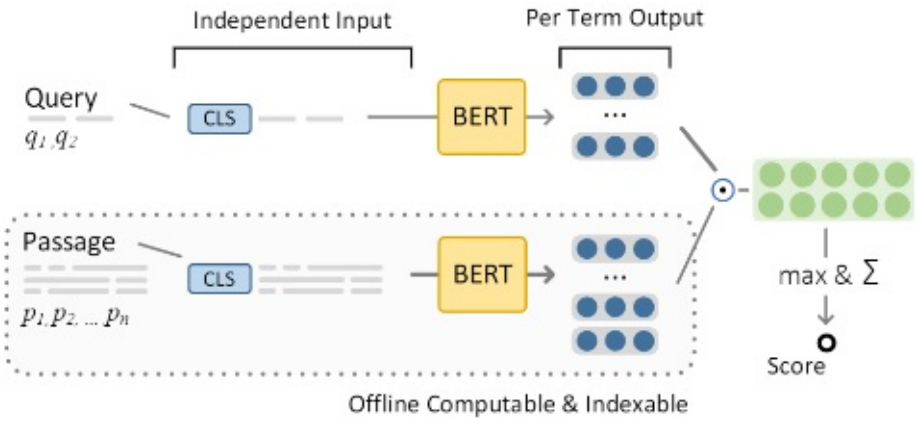

FIGURE 3.13: ColBERT Architecture: Source (Khattab et al., SIGIR2020) [1]

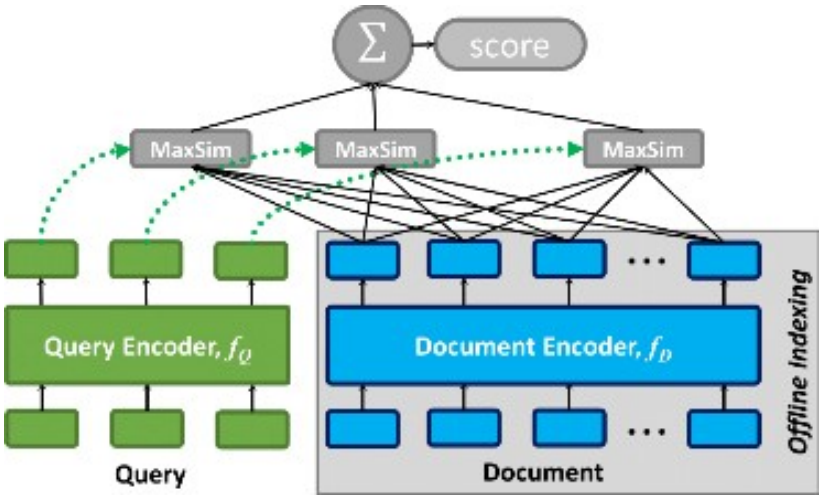

FIGURE 3.14: ColBERT MaxSlim calculations: Source (Khattab et al., SIGIR2020) [1]

3.3 Research Question and Objective

The information retrieval for question answer has received an impetus from the recent deep neural networks based vector representation of data. The recent advances in semantic

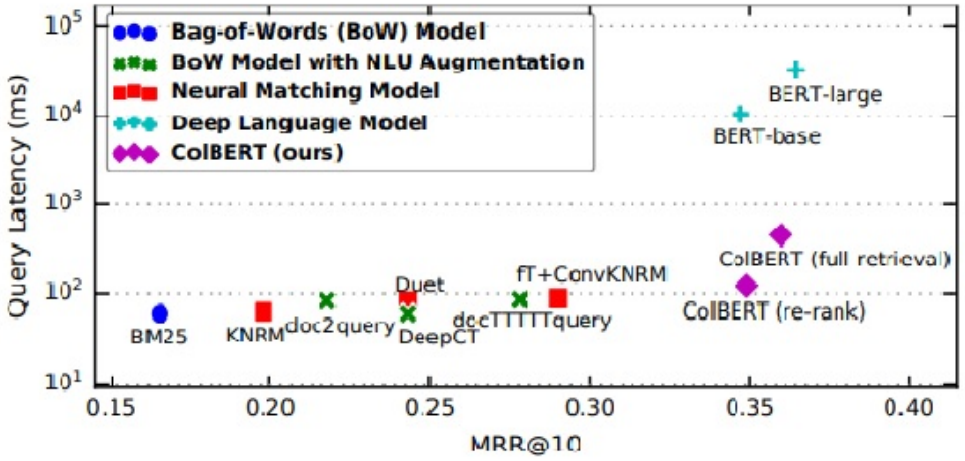

FIGURE 3.15: ColBERT performance compared to other document retrieval models: Source (Khattab et al., SIGIR2020) [1]

Model	Effect.	Query Latency	GPU Memory	Query-Passage Interaction	Passage Cache	NN Index	Storage (× Dim.)
BERT_{CAT}	1	950 ms	10.4 GB	All TF layers	–	–	–
BERT_{DOT}	× 0.87	23 ms	3.6 GB	Single dot product	✓	✓	$\|P\|$
ColBERT	× 0.97	28 ms	3.4 GB	$m * n$ dot products	✓	✓	$\|T\|$
PreTT	× 0.97	455 ms	10.9 GB	Min. 1 TF layer (here 3)	✓	–	$\|T\|$
TK	× 0.89	14 ms	1.8 GB	$m * n$ dot pr. ▫ Kernel pool	✓	–	$\|T\|$

FIGURE 3.16: ColBERT Resource Requirements and latency: Source (Khattab et al., SIGIR2020) [1]

search are based on approximate nearest neighbor in a vector databases. The corpus documents are represented as vectors in this database and the query is also encoded as a vector for searching the vector database. Similarity terms such as cosine similarity are used to measure the document and query similarity.

Based on the literature review and research methods described in this chapter, I can define the final research questions that I will explore in this thesis.

The objective of this research is to answer these two questions:

1. Do transformers trained with biomedical data generate document embeddings that deliver better results for identifying answer containing documents for BioASQ questions ?

   It is envisaged that the transformers trained with the biomedical domain data would capture the context and entity relationships and therefore generate information rich embeddings that may deliver superior vector matching results (Samarinas et. al., [63]).

2. Is identification of interaction of document embeddings for passage and query terms produced at the top layer of an encoder Transformer (BERT) useful for identifying

answer containing documents for BioASQ questions ?

These two questions essentially inform about the relative importance of self attention as captured in the internal layers of a Transformer as compared to the salient interaction between the document and questions terms observed at the top layer of the Transformer. **This can be used to inform resource allocation decision by prioritising resources for model training using the domain data or assigning resources to design various algorithms that capture the document embeddings interaction and their salience at the Transformer top layer.**

In summary, the research intends to use the recent advances in the knowledge enhanced Language models (figure 3.17 and figure 3.18) such as proposed by Zhang et al., 2019 [17], Zheng et al.,2021 [18] and contextualised late interaction (ColBERT) of document and query terms such as as proposed by Khattab et al.,2020 [1]

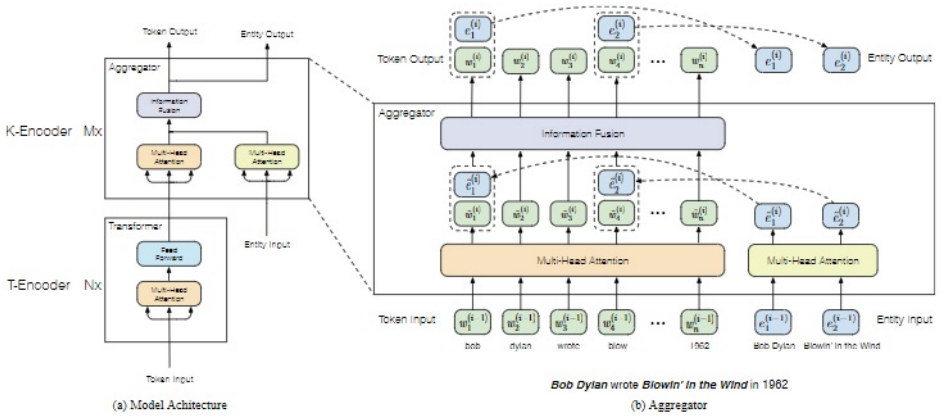

FIGURE 3.17: Enhanced Language Representation with Information Entities (source: Zhang et al.,2019 [17])

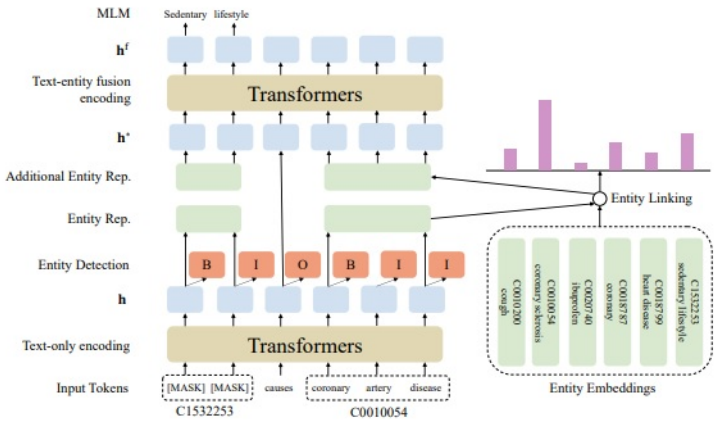

FIGURE 3.18: KeBioLM Architecture(source:Zheng et al.,2021 [18])

To conduct this research I developed a document search re-ranker pipeline (figure 4.1) that uses PubMed API (Entrez API) as first stage ranker to retrieve 500 documents. This API uses keywords based search methods to search for a document in the PubMed database. The re-ranker pipeline processes the retrieved PubMed document Ids and uses Transformers to encode the query and documents (abstracts of the PubMed documents) selected by the first stage ranker. To test if Transformers trained using biomedical data deliver better results, I use various BERT variants that incorporate the biomedical knowledge.

"What I cannot create, I do not understand. Know how to solve every problem that has been solved"

Richard Feynman's last words on his blackboard at the time of his death

4 Experiment and Results

4.1 Experiment

4.1.1 Data used for the experiment

The data used for experiments is downloaded from the BioASQ site. The data consists of 4233 records of questions answers data that was used in earlier BioASQ competitions.

The data is in a JSON file that has the following structure:

- Body: The query string, e.g. "Is Hirschsprung disease a mendelian or a multifactorial disorder?"
- documents: A list of PubMed document URLs that has the information used for collating the answers for the query , e.g. "http://www.ncbi.nlm.nih.gov/pubmed/15858239". The experiment inputs this query and a set of PubMed article abstracts and returns the top 10 matching documents.
- type: Answer type, summary, yes/No or a list
- ideal_answer: e.g. "Coding sequence mutations in RET, | GDNF, EDNRB, EDN3, and SOX10 are involved in the development of Hirschsprung disease. The majority of these genes was shown to be related to Mendelian syndromic forms of Hirschsprung's disease, whereas the non-Mendelian inheritance of sporadic non-syndromic Hirschsprung disease proved to be complex; involvement of multiple loci was demonstrated in a multiplicative model."
- concepts: a list of meta data info associated with the question.
- Snippets: A list of offset beginning and offset end of text snippets that are useful for the answer. These snippets are generally from the abstract but these can also

be from other sections of the document, e.g. from title or the main body of the document. An example of the extract from the title is shown below:


```
{"snippet" :  {
       "offsetInBeginSection": 0,
       "offsetInEndSection": 131,
       "text": "Differential contributions of rare and common,
            coding and noncoding Ret mutations to multifactorial
            Hirschsprung disease liability.",
       "beginSection": "title",
       "document": "http://www.ncbi.nlm.nih.gov/pubmed/20598273",
       "endSection": "title"
}}
```


4.1.2 Data Constraints

PubMed E-Utilities data constraints

There are some data and data processing constraints imposed by the PubMed search interface. The study used the PubMed E-Utilities interface called **Entrez**.

- Number of documents requested from PubMed

  The experiment requests PubMed to return 500 documents for each query. Several return size permutations were tried. A smaller return set generally do not deliver good results and a larger result set leads to more processing exceptions at PubMed. A request result set of 1000 actually had slightly worse performance than the request result set of 500.

- Answer text not in the abstract

  As mentioned in the data section, some snippets that has the answer are in the body and not in the abstract of the PubMed article. The experiment only reads the abstract data and therefore information missing in the abstract degrades the overall document finding performance of the experiment.

- Some PubMed documents do not have abstract

  The experiment uses PubMed and lexical search in phase one. This documents and the query are matched using semantic matching in second phase. However, It is observed that several documents returned by the PubMed in phase only have body but no abstract. The experiment only reads the abstract and hence it is likely that the experiment missed the query matching information in the document returned by PubMed.

- No PubMed data for some questions

  For some queries PubMed does not return any data. I have also manually tried submitting these queries and confirmed that PubMed indeed does not return any

data. It is not clear why PubMed fails to match the terms in the query with any document in the database. For a sample of one hundred queries PubMed did not return any data for nineteen questions. An example of such query is:

"List the human genes encoding for the dishevelled proteins?"

- PubMed processing errors for some questions

  Sometimes a query submitted to PubMed returns an processing exception. This could be due to unacceptable loads on the PubMed servers or it could be related to exceeding the data transfer rate limit for a subscriber. Out of one hundred questions submitted to PubMed for answers only one had a processing exception.

- PubMed response to a question does not have any document listed in the documents sections of the question. This can be attributed to the failure of the lexical search key words terms used at PubMed. The Key words derived from the question returns documents that are not present in the list of documents associated with that question. In a sample of 100 queries, documents returned for fourteen queries did not have any documents common with the document set of the question.

There are also some other processing constraints of the computing environment.

- Abstract size limited to 200 characters Processing more than 200 bytes of abstract data causes "out of memory" error for the NCI CUDA(GPU). Hence the abstract text is limited to first 200 characters only.

Hence for a sample of 100 queries:

- One had a processing error.
- Nineteen did not return any documents for the PubMed.
- Fourteen did not return any document common to the documents associated with the question. Please see section 4.1.1 for document details.

Additionally, several documents returned by PubMed did not have abstract. Hence these documents could not be used for the experiment.

In summary out of 100 only 66 queries were eventually useful for the experiment.

4.2 Experiment Objective

The objective of the study is to run experiments that provide decision making data for the objective listed in 3.3. The experiment uses various Transformer models(BERT and other Large Language Models) and term matching techniques(CLR, SBERT, ColBERT) to compare their performance for identifying a document that may have an answer to a BioASQ query (as described in BioASQ challenge Part B, Phase A).

The BioASQ data used for the experiment is described in the section 4.1.1. Some salient points of the experiment and search for documents are listed below:

- The data is obtained from BioASQ challenge site. This is the training data made available for the BioASQ challenge explained in the introduction section. This has the input(a query) and the output (a set of documents that can be used to answer the input query)

- Each data element has a query and the associated documents that can be used to answer the query.This is the training data i.e. for a query variable the program should return those PubMed documents as response.

- The experiment uses various Transformers and vector similarity techniques to return a set of ten documents from the original set of max 500 documents retrieved from PubMed using the PubMed API(using the input query). These documents are ranked on the basis of a similarity score. The top 10 documents from this set are selected and the experiment returns the fraction of this top 10 documents that are in the BioASQ solution set. Hence this is fraction of the true positive documents in the top 10 documents returned by the experiment. This fraction is the precision score in the top ten documents.

The documents are selected by using a processing pipeline. The first phase of the pipeline uses the query to select a set of 500 documents from PubMed. PubMed uses keyword based algorithm to select these documents. These documents returned by are further processed by using query and document dense vectors. These dense vectors are generated by Transformers.

The experiment returns the top 10 documents that have highest matching score with the query. The precision of this 10 documents is used as score (the fraction of true positive documents) is used to rank the dense vector matching algorithms (ref 4.2)

R = Number of positive documents returned by the experiment.

T = total number of documents associated with the question in the BioASQ data. Please refer 4.1.1

$$Pscore = \frac{R}{T}$$

4.3 Experiment Setup - Computation and Storage

The experiment users several Transformer models to generate vector embedding of the query and the PubMed article abstract.

A set of one hundred BioASQ questions as described in 4.1.1 and **National computing Infrastructure AKA GADI**, with up to 12 CPU and a GPU were used to run the long running experiments. To ensure a consistent baseline, all the PubMed Data required for the selected set one hundred questions was downloaded from the PubMed and saved on

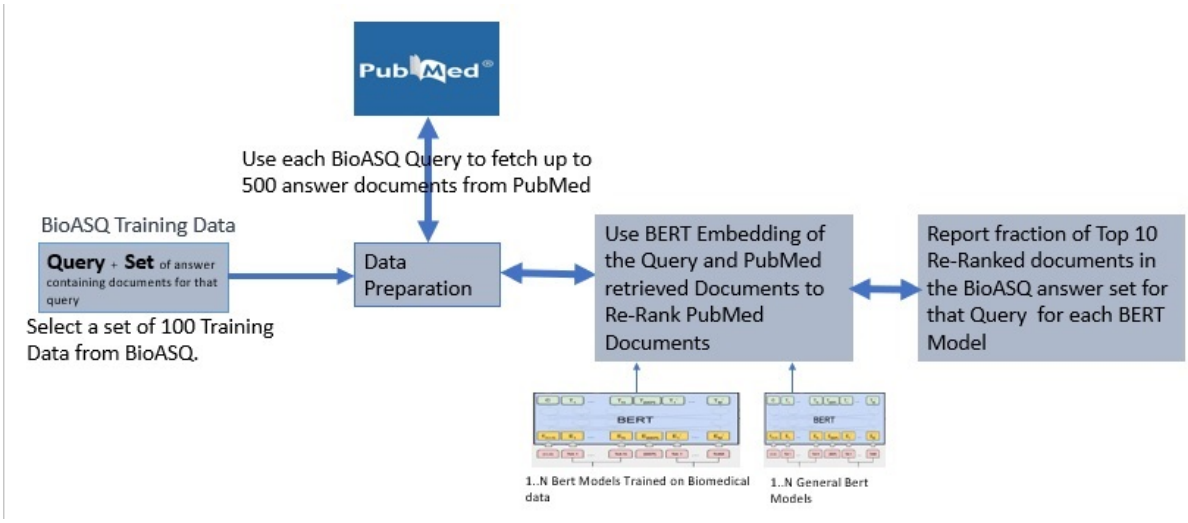

FIGURE 4.1: Data Pipeline

disk. Some of these experiments ran more than 12 hours. Several attempts were made to fine tune these models using BioASQ QA data.

Using CLR Token For Query and Document Similarity

The first token of every BERT generated embeddings sequence, called as CLS (classification token) is used to represent the sentence. "The final hidden state corresponding to this token is used as the aggregate sequence representation" (Delvin et al., [64]. It captures part of the semantics essence of the sentence and hence has been successfully used in classification and feature extraction

The CLS token at the final layer represents self and cross attention relationships in the sequence. It is expected that CLS token generated from the biomedical trained BERT would capture salient biomedical domain information such as entities and their relationships. The experiment use the cosine similarity of query and document CLR tokens to calculate a similarity score. It is assumed that a question and answer pair would have high similarity score.

4.4 Models

The BERT models are divided into two key categories, they are:

4.4.1 BERT models trained on biomedical domain data

This study used several publicly available, biomedical data trained, BERT based models to generate vector embedding for query and document abstract passages. The CLR tokens of the query and the passage embeddings are used to calculate the similarity scores.

- Bio_ClinicalBERT (Alsentzer E, et. al, [65])

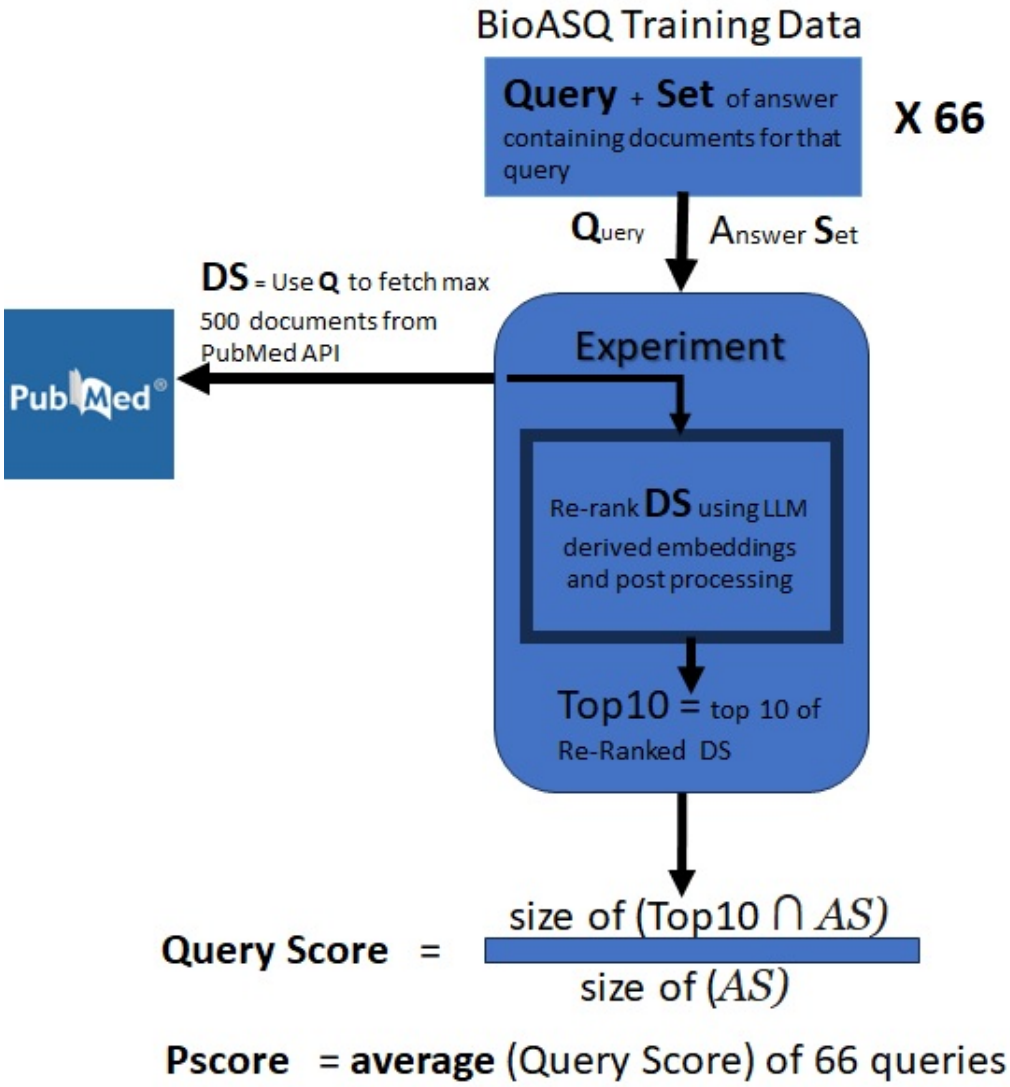

FIGURE 4.2: Experiment and the Score

Bio_ClinicalBERT was one of the first models developed for the biomedical domain. It built and released an embedding models for clinical biomedical Text. The model was trained on 2 million clinical notes and medical discharge summary. The query and passage embedding produced by this model delivered a document finding Pscore of 0.1981. As the precision was much lower than the other models no attempts were made to fine-tune this with PubMed QA data.

- BiomedNLP-PubMedBERT (GU et al. [66]) BiomedNLP-PubMedBERT was the first BERT model trained with PubMed data. It was trained using Masked Language model and it also used a small amount of BioASQ data as well. This model has the highest score for the biomedical Language Understanding and Reasoning Benchmark (BLURB). However, this model also did not produce vector embedding that could

deliver superior results for query/passage vector matching. Its Pscore using CLR matching was 0.1981.

- BioLinkBERT (Yasunaga, M et. al, [67]

  BioLinkBert captures the dependencies and knowledge across the medical documents. It is trained using masked language model and another objective of document relation prediction. It was envisaged that this will capture the relationship between biomedical entities and therefore capture the biomedical knowledge. However, this model also did not deliver good precision results. The experiment achieved a Pscore of 0.1973 with this model.

- Stanford-crfm/BioMedLM

  Stanford-crlm/BioMedLM is a large GPT-2 based autoregressive language model with 2.7B parameters. However, for this experiment it did not produce vectors that could deliver good performance for the BioASQ question answer set. It delivered P@10 of 0.1650. This model required lot more compute resources than other models evaluated for this study. The experiment ran for more than 20 hours on using 12 NCPUs. This could be a good model to fine-tune for BioASQ question answer. However, I could not get enough PubMed Q/A data to fine tune this. Plus fine tuning such a large model would take a lot of resources. Hence, after considerable effort this fine-tuning effort was abandoned. However, this remains a good research goal to fine-tune a GPT-2 or GPT-3 based large language model built using biomedical data.

- Microsoft/BioGPT

  BioGPT is also a generative pre-trained Transformer for biomedical text generation. It's authors claim good performance and 78.2 percent accuracy for PubMedQA. However PubMedQA is a limited QA dataset with only Yes/No as answers. Similar to BioMedLM it also did not deliver good results for the BioASQ questions. It delivered a Pscore of 0.1926 only. This may also be a good model for further fine-tuning.

  Additionally, Yuan et al.,2021 [18] has published research and has released a model (KeBioLM) that incorporates biomedical Unified Medical Language System (ULMA) knowledge into BERT. However, this model could not be easily downloaded, required lot of support from the authors and could not be successfully deployed in the development environment.

4.4.2 Models using SBERT pooling and ColBERT term interaction re-ranking

This section investigates the use of BERT-based models built using general data. However, these models use top layer word embedding created by the Transformer. These models

TABLE 4.1: BERT based biomedical models

BERT models trained on biomedical data		
Number	Model Name	Vector matching method
1	Bio_ClinicalBERT	CLR
2	BiomedNLP-PubMedBERT	CLR
3	BioLinkBERT	CLR
4	stanford-crfm/BioMedLM	CLR
5	microsoft/biogpt	CLR

TABLE 4.2: BERT based general models

BERT models trained for dense vector matching		
Number	Model Name	Vector matching method
1	Colberter-128-32-msmarco	ColBERT
2	Distilbert-dot-margin_mseT2-msmarco	CLR pooling
3	multi-qa-MiniLM-L6-cos-v1	SBERT
4	multi-qa-MiniLM-L6-cos-v1-finetuned	SBERT
5	Facebook DPR model	SBERT

pool embedding (SBERT (Reimer N, et. al, [16])), or use contextualised late interaction method called ColBERT (Khattab, O et. al, [1]). The experiment shows that in general these models deliver better performance than the models built with biomedical data.

- DPR for Open-Domain Question Answer.

  Dense Passage Retrieval(Karpukhin, V [22]) uses dense vector and BERT trained on Google's Natural Questions database. I used SBERT encoding to evaluate its performance for bioASQ QA dataset. It did provide better results (Pscore 0.2262) than the general models built with biomedical data.

- Multi-qa-MiniLM-L6-cos-v1 and SBERT

  This model is trained on 215M question answer pairs and it maps the passages and query to a 384 dimensional dense space. This provides document Pscore 0.2343.

- Multi-qa-MiniLM-L6-cos-v1 fine-tuned with BioASQ data

The publicly available model was further fine-tuned with BioASQ questions answer data and it improved Pscore to 0.2383.

- distilbert-dot-margin_mse-T2-msmarco

  This model is based on the Cross-Architecture knowledge distillation as proposed by Hofstatter S et. al, [19]. It is trained on MSMARCO-Pasage and is used to re-rank the passages retrieved from the first phase of the retrieval pipeline.

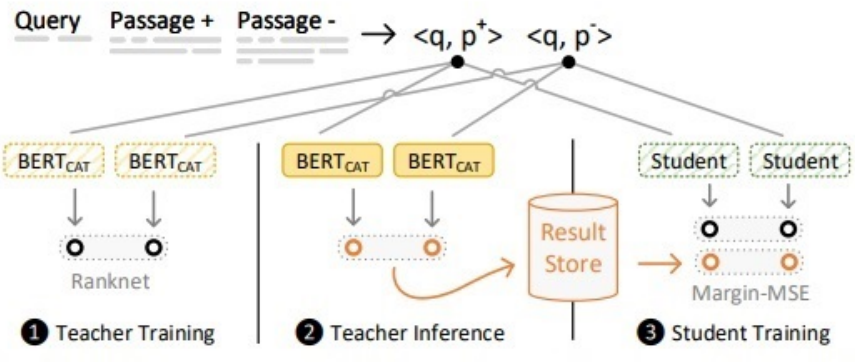

FIGURE 4.3: Knowledge Distillation Process (Source, Hofstaetter et al [19])

This model pools the CLS vectors of the query and passage vectors. It uses a three teacher concatenated scoring knowledge distillation (please see figure 4.3). The CLS pooling method provide a P@10 of **0.2501**.

- ColBERT-128-32-MSMARCO for Passage Retrieval

  This model is based on an innovative end-to-end retrieval and ranking model called COLBERTer (please see figure 4.4).

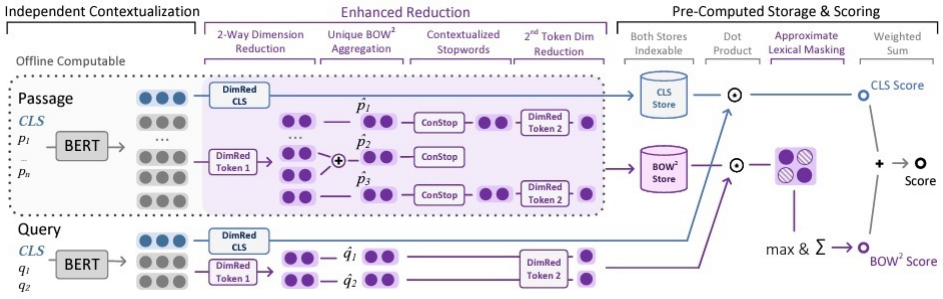

FIGURE 4.4: ColBERT Encoding Architecture (Source, Hofstaetter et al [20])

"ColBERTer combines single-vector retrieval, multivector refinement,and optional lexical matching components into one model" (Abstract , Hofstatter S, et, al. [20]). This model delivers the best performance amongst all the models considered in the this study. This model delivers a P@10 of **0.2555**.

4.5 Model Fine-tuning

The study used standard publicly available models to create vector embeddings and used different vector matching techniques (CLS, SBERT and ColBERT) to compare their performance. One model was further fine-tuned with PubMed Question Answer data leading to a better performance. The question answer retrieval performance was tested with a data set different than the one used for finetuning. The model (Multi-qa-MiniLM) using SBERT matching did improve slightly (accuracy increased from 0.2343 to 0.2383). However, attempts to fine-tune other models were not successful. The ColBERT-ms macro model data format requirements were tedious and were hard to satisfy.

4.6 Results

The table 4.3 below shows the retrieval Pscore as defined in figure 4 of various models.

TABLE 4.3: Models and their Pscore results

Model Performance			
Number	Model Name	Pscore	Vector matching method
1	Stanford-crfm/BioMedLM	0.1650	CLS similarity matching
2	BiomedNLP	0.1698	CLS similarity matching
3	Microsoft/biogpt	0.1926	CLS similarity matching
4	BioLinkBERT	0.1973	CLS similarity matching
5	Bio_Clinical Bert	0.1981	CLS similarity matching
6	Facebook-dpr-ctx_encoder-single	0.2262	SBERT
7	multi-qa-MiniLM-L6-cos-v1	0.2343	SBERT
8	multi-qa-MiniLM-L6-cos-v1-finetuned with bioMed Data	0.2383	SBERT
9	Distilbert-dot-margin	0.2501	CLS pooling and matching
10	ColBERT-ms macro	0.2555	ColBERT

4.7 Discussion of Results

The goal of this experiment was to ascertain if large language models that are trained using biomedical data capture enough entity and relationship information that they can deliver superior performance against the top layer term matching algorithms ? The term matching algorithms used in SBERT and ColBERT use the term interactions to elicit relationship between a question and a document. However, it was expected that a large language model that is trained using biomedical data would capture essence of entities

and their deeper relationships in the embedding vectors created by these language model. Can this be translated into better similarity scores between a query and an answer bearing text ? However, results of the experiment negate this hypothesis.

The SBERT use three different pooling methods to create an "aggregated" vector and this vector is used to further used in similarity operations. This is similar to using kernel and simple arithmetic operations to extract hierarchical graphic features in an image in CNN. The ColBERT captures the interaction of terms in more detailed and granular method.

4.7.1 Statistical Significance of Results

The table 4.3 collates the Pscore results of using embeddings created using biomedical BERT models (1-5) and general BERT models (6-10). The Pscore range for biomedical data trained Transformer models is 0.1650 to 0.1981. However, the Pscore range for embedding generated from general BERT Transformers and post processed using SBERT, CLS Pooling and COLBERT is 0.2262 to 0.2555

The F ratio for one way ANOVA test for this two sets is 40.28 and the p value is 0.000221. This result is significant at a very low $p < 0.01$. On the basis of very low p value we can assert that there is indeed a different in performance of two methods. It proves that the **mathematical operations on the top layer embeddings obtained from the general large language models capture relationships better than the embeddings produced by the complex and domain trained models.**

The use of one way ANOVA for this experiment has made an assumption that we can divide the group of Transformers used into two distinct but homogeneous groups. This assumption can only be true if there is strong similarity amongst the models within a group. The dispersion of results within these two groups is relatively low. This can be confirmed using Chi square test of independence within these two groups. However, I do not have access to computation resource and data to re-run the experiment and gather statistics for each model.

"The bravest are surely those who have the clearest vision"
Thucydides

5 Conclusion and further research

5.1 Key observations

In the Research Methods chapter, section 3.3, I stated that the research objective is to seek data to answer following questions:

1. Do transformers trained with biomedical data generate document embeddings that deliver better results for identifying answer containing documents for BioASQ questions ?

2. Is identification of interaction of document embeddings for passage and query terms produced at the top layer of an encoder Transformer (BERT) useful for identifying answer containing documents for BioASQ questions ?

In response to the questions above, the two key observations of this study are:

1. Based on experiment results, it is observed that Transformers trained with biomedical data do not deliver better vector matching performance for finding answer document in a document set.

   The experiment used several BERT based Transformers and two Large Language Model based transformers. There was no additional finetuning performed on these models. The term matching was done by using simple CLS token similarity matching. The low performance of very Large Language Models (BioMedLM and BioGPT) is not surprising because they are trained as decoders for text generation. Hence document embeddings generated by the LLMs may not be comparable to one generated by the BERT based transformers that are encoders in nature.

   Wang et al., also reported that, for the biomedical domain, information retrieval based on dense vector representation only does not deliver competitive results as

achieved by classical term matching techniques such as BM25 (Wang et al., 2021 [44]) Table 9, Section E).

2. Based on experiment results, it is observed that word embedding interaction information at the top layer can be used to enhance document embeddings matching performance. This enhanced performance can be used to identify documents that may have answers to a query.

5.2 Using word embedding for identifying documents for BioASQ challenge

The overall document retrieval performance (Pscore) as reported in my experiments is **0.2566**. My objective of my experiment was not to find the document but to understand if large language models trained with biomedical domain data have an performance edge over simpler models (not trained with biomedical data). The experiment data shows that simpler models with some simple arithmetic operations post processing of sentence embedding deliver comparatively better performance.

The Pscore as reported by the study is based on the 66 documents and not the whole data-set of 100. As reported in the section 4.1.1, PubMed did not provide any valid documents for 34 queries. To further improve the overall performance, we need to increase the performance of the first phase of the pipeline.

The search performance can be further improved by:

1. Use hybrid search for first stage of the pipeline It appears that a hybrid search method that combines both keywords based search and vector search may overcome the inadequacy of the keywords based search in the first phase of the pipeline. However, this would mean building a vector database of the whole PubMed data. This is a compute and storage intensive task and it would require far more storage and compute resources than those were allocated and used for this study.

2. Finetune the ColBERT model with the BioASQ Q/A data I noticed (in section 4.4.2) that fine-tuning of Multi-qa-MiniLM-L6-cos-v1 with bioASQ Q/A did improve the performance. It is envisaged that fine-tuning a ColBERT model with BioASQ data is likely to improve the retrieval performance.

5.3 Future Research

The document retrieval research is an active and fertile field and there is a lot of fruitful research going on in this field. Some of the research areas that appear to have potential are:

- Use zero-shot Large Language Models as Knowledge Base

  This topic was discussed in the literature review and the recent literature has documented its shortcomings. However, this field is moving very fast (Bommasani,

R et al, [68]) and there are new language models being build every day and these deep learning and transfer learning based models offer a good opportunity to explore their use for BioASQ challenge.

- New Framework - Retrieval-Enhanced Machine Learning

  There is new emerging research about retrieval-Enhanced framework (Zamani, et al, 2022 [21]). This research proposes a framework that "enables ML models to be augmented with IR capabilities". This overcomes the deficiencies observed in zero-shot large language models. A retrieval based language model uses an external data store for answering a question. This overcomes the issues of outdated parametric information retrieved from a large language model and lack of interpretablity of a traditional large language model.

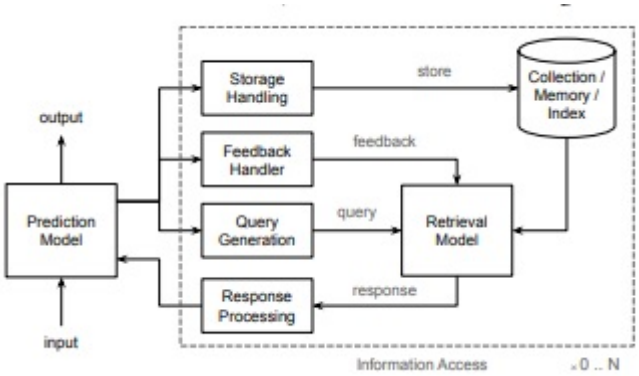

FIGURE 5.1: Retrieval Enhanced Machine Learning (Source, Zamani H et al [21])

- Further enhancements to ColBERT

  Recently, several authors have reported new enhancements to the ColBERT model, e.g. Lee et al. ([69]) have proposed a token retrieval method that improved the ColBERT document retrieval performance. It is expected that more models that refine the ColBERT or devise more operations for the term embeddings output may prove to be valuable.

References

[1] O. Khattab and M. Zaharia. *ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT*, p. 39–48 (Association for Computing Machinery, New York, NY, USA, 2020). URL https://doi.org/10.1145/3397271.3401075. xiii, xiv, 3, 24, 26, 27, 28, 29, 38

[2] A. Nentidis, G. Katsimpras, E. Vandorou, A. Krithara, L. Gasco, M. Krallinger, and G. Paliouras. *Overview of bioasq 2021: The ninth bioasq challenge on large-scale biomedical semantic indexing and question answering*. arxiv.org (2021). URL https://arxiv.org/pdf/2106.14885.pdf. xiii, 4, 6

[3] Y. Mao, P. He, X. Liu, Y. Shen, J. Gao, J. Han, and W. Chen. *Generation-augmented retrieval for open-domain question answering*. Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pp. 4089–4100 (Association for Computational Linguistics). URL https://aclanthology.org/2021.acl-long.316https://doi.org/10.18653/v1/2021.acl-long.316. xiii, 6, 7, 8, 14

[4] D. Chen, A. Fisch, J. Weston, and A. Bordes. *Reading wikipedia to answer open-domain questions*. Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 1870–1879 (Association for Computational Linguistics). URL https://www.aclweb.org/anthology/P17-1171https://doi.org/10.18653/v1/P17-1171. xiii, 5, 6, 8, 9, 18

[5] W. Yang, Y. Xie, A. Lin, X. Li, L. Tan, K. Xiong, M. Li, and J. Lin. *End-to-end open-domain question answering with bertserini*. Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations), pp. 72–77 (Association for Computational Linguistics). URL https://www.aclweb.org/anthology/N19-4013https://doi.org/10.18653/v1/N19-4013. xiii, 8, 9, 18

[6] T. Brown and B. M. Mann. *Language models are few-shot learners*. Advances in Neural Information Processing Systems 33 (NeurIPS 2020) (2020). URL https://papers.nips.cc/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf. xiii, 10

[7] C. Xu, D. Guo, N. Duan, and J. McAuley. *Laprador: Unsupervised pretrained dense*

retriever for zero-shot text retrieval (2022). URL https://arxiv.org/abs/2203.06169. xiii, 12, 14

[8] W. Xiong, X. L. Li, S. Iyer, J. Du, P. Lewis, W. Y. Wang, Y. Mehdad, W.-t. Yih, S. Riedel, D. Kiela, and B. Oğuz. *Answering complex open-domain questions with multi-hop dense retrieval* (2020). URL https://arxiv.org/abs/2009.12756. xiii, 13, 14, 18

[9] M. Seo, J. Lee, T. Kwiatkowski, A. Parikh, A. Farhadi, and H. Hajishirzi. *Real-time open-domain question answering with dense-sparse phrase index*. Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp. 4430–4441 (Association for Computational Linguistics). URL https://www.aclweb.org/anthology/P19-1436https://doi.org/10.18653/v1/P19-1436. xiii, 14, 15

[10] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Y. Yih, T. Rocktäschel, S. Riedel, and D. Kiela. *Retrieval-augmented generation for knowledge-intensive nlp tasks*. Accepted at NeurIPS 2020 (2020). xiii, 14, 15, 16, 18

[11] W. Yang, K. Lu, P. Yang, and J. Lin. *Critically examining the "neural hype"*. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval* (ACM, 2019). URL https://doi.org/10.1145%2F3331184.3331340. xiii, 20

[12] J. Lee, W. Yoon, S. Kim, D. Kim, S. Kim, C. H. So, and J. Kang. *BioBERT: a pre-trained biomedical language representation model for biomedical text mining*. Bioinformatics **36**(4), 1234 (2019). https://academic.oup.com/bioinformatics/article-pdf/36/4/1234/32527770/btz682.pdf, URL https://doi.org/10.1093/bioinformatics/btz682. xiii, 20, 21

[13] S. Chakraborty, E. Bisong, S. Bhatt, T. Wagner, R. Elliott, and F. Mosconi. *Biomedbert: A pre-trained biomedical language model for qa and ir*. Proceedings of the 28th International Conference on Computational Linguistics, pp. 669–679 (International Committee on Computational Linguistics). URL https://www.aclweb.org/anthology/2020.coling-main.59https://doi.org/10.18653/v1/2020.coling-main.59. xiii, 20, 21

[14] A. Yates, R. Nogueira, and J. Lin. *Pretrained Transformers for Text Ranking: BERT and Beyond*, p. 2666–2668 (Association for Computing Machinery, New York, NY, USA, 2021). URL https://doi.org/10.1145/3404835.3462812. xiii, xiv, 22, 23, 24, 25

[15] N. Tonellotto. *Lecture notes on neural information retrieval* (2022). URL https://arxiv.org/abs/2207.13443. xiv, 24, 25

[16] N. Reimers and I. Gurevych. *Sentence-BERT: Sentence embeddings using Siamese BERT-networks*. 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. 3982–3992 (Association for Computational Linguistics, Hong Kong, China, 2019). URL https://aclanthology.org/D19-1410. xiv, 3, 26, 38

[17] Z. Zhang, X. Han, Z. Liu, X. Jiang, M. Sun, and Q. Liu. *ERNIE: Enhanced language representation with informative entities*. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 1441–1451 (Association for Computational Linguistics, Florence, Italy, 2019). URL https://aclanthology.org/P19-1139. xiv, 16, 29

[18] Z. Yuan, Y. Liu, C. Tan, S. Huang, and F. Huang. *Improving biomedical pretrained language models with knowledge*. In *Proceedings of the 20th Workshop on Biomedical Language Processing*, pp. 180–190 (Association for Computational Linguistics, Online, 2021). URL https://aclanthology.org/2021.bionlp-1.20. xiv, 29, 37

[19] S. Hofstätter, S. Althammer, M. Schröder, M. Sertkan, and A. Hanbury. *Improving efficient neural ranking models with cross-architecture knowledge distillation* (2021). 2010.02666. xiv, 3, 39

[20] S. Hofstätter, O. Khattab, S. Althammer, M. Sertkan, and A. Hanbury. *Introducing neural bag of whole-words with colberter: Contextualized late interactions using enhanced reduction* (2022). 2203.13088. xiv, 39

[21] H. Zamani, F. Diaz, M. Dehghani, D. Metzler, and M. Bendersky. *Retrieval-enhanced machine learning*. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval* (ACM, 2022). URL https://doi.org/10.1145%2F3477495.3531722. xiv, 45

[22] V. Karpukhin, B. Oguz, S. Min, P. Lewis, L. Wu, S. Edunov, D. Chen, and W.-t. Yih. *Dense passage retrieval for open-domain question answering*. Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 6769–6781 (Association for Computational Linguistics). URL https://www.aclweb.org/anthology/2020.emnlp-main.550https://doi.org/10.18653/v1/2020.emnlp-main.550. 3, 12, 13, 18, 38

[23] O. Khattab and M. Zaharia. *ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT*, p. 39–48 (Association for Computing Machinery, New York, NY, USA, 2020). URL https://doi.org/10.1145/3397271.3401075. 3

[24] G. Tsatsaronis, G. Balikas, P. Malakasiotis, I. Partalas, and M. Zschunke. *An overview of the bioasq large-scale biomedical semantic indexing and question answering competition*. BMC Bioinformatics (2015). URL https://bmcbioinformatics.biomedcentral.com/track/pdf/10.1186/s12859-015-0564-6.pdf. 3, 13

[25] P. Rodriguez and J. Boyd-Graber. *Evaluation paradigms in question answering*. Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pp. 9630–9642 (Association for Computational Linguistics). URL https://aclanthology.org/2021.emnlp-main.758. 5

[26] R. F. Simmons. *Answering english questions by computer: A survey*. Communications of ACM **8**(1), 17 (1965). URL https://doi.org/10.1145/363707.363732. 6

[27] M. Mintz, S. Bills, R. Snow, and D. Jurafsky. *Distant supervision for relation extraction without labeled data*. Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP, pp. 1003–1011 (Association for Computational Linguistics). URL https://aclanthology.org/P09-1113. 8, 9

[28] C. Clark and M. Gardner. *Simple and effective multi-paragraph reading comprehension*. Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 845–855 (Association for Computational Linguistics). URL https://www.aclweb.org/anthology/P18-1078https://doi.org/10.18653/v1/P18-1078. 9, 18

[29] Z. Wang, P. Ng, X. Ma, R. Nallapati, and B. Xiang. *Multi-passage bert: A globally normalized bert model for open-domain question answering*. 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. 5878–5882 (Association for Computational Linguistics). URL https://www.aclweb.org/anthology/D19-1599https://doi.org/10.18653/v1/D19-1599. 9

[30] S. WANG, M. YU, X. GUO, Z. WANG, and T. KLINGER. *R3: Reinforced ranker-reader for open-domain question answering*. Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence **5981-5998** (2018). URL https://ink.library.smu.edu.sg/cgi/viewcontent.cgi?article=5240&context=sis_research. 9, 18

[31] S. Min, D. Chen, H. Hajishirzi, and L. Zettlemoyer. *A discrete hard em approach for weakly supervised question answering*. 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. 2851–2864 (Association for Computational Linguistics). URL https://www.aclweb.org/anthology/D19-1284https://doi.org/10.18653/v1/D19-1284. 9, 10, 18

[32] Z. Zhang, T. Vu, and A. Moschitti. *Joint models for answer verification in question answering systems*. Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pp. 3252–3262 (Association for Computational Linguistics). URL https://aclanthology.org/2021.acl-long.252https://doi.org/10.18653/v1/2021.acl-long.252. 9, 18

[33] F. Petroni, T. Rocktaschel, L. Patrick, A. Bakhtin, Y. Wu, A. Miller, and S. Riedel. *Language models as knowledge bases?* (2020). URL https://www.aclweb.org/anthology/D19-1250.pdf. 10

[34] S. Longpre, K. Perisetla, A. Chen, N. Ramesh, C. DuBois, and S. Singh. *Entity-based knowledge conflicts in question answering*. Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pp. 7052–7063 (Association for Computational Linguistics). URL https://aclanthology.org/2021.emnlp-main.565. 10

[35] S. Garg and A. Moschitti. *Will this question be answered? question filtering via answer model distillation for efficient question answering*. EMNLP 2021 (2021). 10

[36] A. Roberts, C. Raffel, and N. Shazeer. *How much knowledge can you pack into the parameters of a language model?* Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 5418–5426 (Association for Computational Linguistics). URL https://www.aclweb.org/anthology/2020.emnlp-main.437https://doi.org/10.18653/v1/2020.emnlp-main.437. 10

[37] K. Guu, K. Lee, Z. Tung, P. Pasupat, and M.-W. Chang. *Realm: Retrieval-augmented language model pre-training*. Pre-Print; NeurIPS 2020 (2020). URL https://arxiv.org/abs/2002.08909. 10, 12, 13, 16, 18

[38] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. Liu. *Exploring the limits of transfer learning with a unifiedtext-to-text transformer*. Journal of Machine Learning Research **21**(1-67) (2021). 11

[39] C. Wang, P. Liu, and Y. Zhang. *Can generative pre-trained language models serve as knowledge bases for closed-book qa?* Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pp. 3241–3251 (Association for Computational Linguistics). URL https://aclanthology.org/2021.acl-long.251https://doi.org/10.18653/v1/2021.acl-long.251. 11

[40] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean. *Distributed representations of words and phrases and their compositionality*. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'13, p. 3111–3119 (Curran Associates Inc., Red Hook, NY, USA, 2013). 11

[41] W.-t. Yih, K. Toutanova, J. C. Platt, and C. Meek. *Learning discriminative projections for text similarity measures*. Proceedings of the Fifteenth Conference on Computational Natural Language Learning, pp. 247–256 (Association for Computational Linguistics). URL https://www.aclweb.org/anthology/W11-0329. 11

[42] J. Johnson, M. Douze, and H. Jégou. *Billion-scale similarity search with gpus*. IEEE Transactions on Big Data **7**(3), 535 (2021). 11

[43] K. Lee, M.-W. Chang, and K. Toutanova. *Latent retrieval for weakly supervised open domain question answering*. Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp. 6086–6096 (Association for Computational Linguistics). URL https://www.aclweb.org/anthology/P19-1612https://doi.org/10.18653/v1/P19-1612. 12, 18

[44] K. Wang, N. Thakur, N. Reimers, and I. Gurevych. *Gpl: Generative pseudo labeling for unsupervised domain adaptation of dense retrieval*. arXiv (2021). URL https://arxiv.org/abs/2112.07577. 12, 16, 44

[45] T. Zhao, X. Lu, and K. Lee. *Sparta: Efficient open-domain question answering via sparse transformer matching retrieval*. Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 565–575 (Association for Computational Linguistics). URL https://aclanthology.org/2021.naacl-main.47https://doi.org/10.18653/v1/2021.naacl-main.47. 13

[46] D. S. Sachan, S. Reddy, W. Hamilton, C. Dyer, and D. Yogatama. *End-to-end training of multi-document reader andretriever for open-domain question answering*. arxiv (2021). 13, 15

[47] Z. Dai and J. Callan. *Context-Aware Document Term Weighting for Ad-Hoc Search*, p. 1897–1907 (Association for Computing Machinery, New York, NY, USA, 2020). URL https://doi.org/10.1145/3366423.3380258. 13, 18

[48] G. Xu, W. Rong, Y. Wang, Y. Ouyang, and Z. Xiong. *External features enriched model for biomedical question answering*. BMC Bioinformatics (2021). 13, 18

[49] S. Min, K. Lee, M.-W. Chang, K. Toutanova, and H. Hajishirzi. *Joint passage ranking for diverse multi-answer retrieval*. Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pp. 6997–7008 (Association for Computational Linguistics). URL https://aclanthology.org/2021.emnlp-main.560. 13, 18

[50] Z. Zhang, T. Vu, and A. Moschitti. *Double retrieval and ranking for accurate question answering* (2022). URL https://arxiv.org/abs/2201.05981. 13, 14

[51] N. Thakur, N. Reimers, A. Rücklé, A. Srivastava, and I. Gurevych. *Beir: A heterogenous benchmark for zero-shot evaluation of information retrieval models* (2021). 2104.08663. 14

[52] J. Lee, M. Seo, H. Hajishirzi, and K. Jaewoo. *Contextualized sparse representations for real-time open-domain question answering* (2020). 14, 15, 18

[53] H. Cheng, Y. Shen, X. Liu, P. He, W. Chen, and J. Gao. *Unitedqa: A hybrid approach for open domain question answering*. Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference

on Natural Language Processing (Volume 1: Long Papers), pp. 3080–3090 (Association for Computational Linguistics). URL https://aclanthology.org/2021.acl-long.240https://doi.org/10.18653/v1/2021.acl-long.240. 15, 18

[54] A. Asai and E. Choi. *Challenges in information-seeking qa: Unanswerable questions and paragraph retrieval*. Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pp. 1492–1504 (Association for Computational Linguistics). URL https://aclanthology.org/2021.acl-long.118https://doi.org/10.18653/v1/2021.acl-long.118. 15

[55] 16

[56] U. Khandelwal, O. Levy, D. Jurafsky, L. Zettlemoyer, and M. Lewis. *Generalization through Memorization: Nearest Neighbor Language Models*. In *International Conference on Learning Representations (ICLR)* (2020). 16

[57] S. Hofstatter, N. Craswell, B. Mitra, H. Zamani, and A. Hanbury. *Are we there yet? a decision framework for replacing term based retrieval with dense retrieval systems*. ArXiv **abs/2206.12993** (2022). 16

[58] D. Dimitriadis and G. Tsoumakas. *Word embeddings and external resources for answer processing in biomedical factoid question answering*. Journal of Biomedical Informatics **92**, 103118 (2019). URL https://www.sciencedirect.com/science/article/pii/S153204641930036X. 19

[59] G. M. Rosa, L. Bonifacio, V. Jeronymo, H. Abonizio, M. Fadaee, R. Lotufo, and R. Nogueira. *No parameter left behind: How distillation and model size affect zero-shot retrieval* (2022). URL https://arxiv.org/abs/2206.02873. 21

[60] S. Hofstätter, N. Rekabsaz, C. Eickhoff, and A. Hanbury. *On the effect of low-frequency terms on neural-ir models*. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR'19, p. 1137–1140 (Association for Computing Machinery, New York, NY, USA, 2019). URL https://doi.org/10.1145/3331184.3331344. 22

[61] C. Xiong, Z. Dai, J. Callan, Z. Liu, and R. Power. *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*, SIGIR '17, p. 55–64 (Association for Computing Machinery, New York, NY, USA, 2017). URL https://doi.org/10.1145/3077136.3080809. 22

[62] Z. Dai, C. Xiong, J. Callan, and Z. Liu. *Convolutional neural networks for soft-matching n-grams in ad-hoc search*. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, WSDM '18, p. 126–134 (Association for Computing Machinery, New York, NY, USA, 2018). URL https://doi.org/10.1145/3159652.3159659. 22

[63] C. Samarinas, A. Dharawat, and H. Zamani. *Revisiting open domain query facet extraction and generation*. Proceedings of the 2022 ACM SIGIR International Conference on Theory of Information Retrieval (2022). 28

[64] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. *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, Volume 1 (Long and Short Papers)*, pp. 4171–4186 (Association for Computational Linguistics, Minneapolis, Minnesota, 2019). URL https://aclanthology.org/N19-1423. 35

[65] E. Alsentzer, J. R. Murphy, W. Boag, W.-H. Weng, D. Jin, T. Naumann, and M. B. A. McDermott. *Publicly available clinical bert embeddings* (2019). 1904.03323. 35

[66] Y. Gu, R. Tinn, H. Cheng, M. Lucas, N. Usuyama, X. Liu, T. Naumann, J. Gao, and H. Poon. *Domain-specific language model pretraining for biomedical natural language processing*. ACM Transactions on Computing for Healthcare **3**(1), 1 (2021). URL https://doi.org/10.1145%2F3458754. 36

[67] M. Yasunaga, J. Leskovec, and P. Liang. *Linkbert: Pretraining language models with document links* (2022). 2203.15827. 37

[68] R. Bommasani, D. A. Hudson, E. Adeli, R. Altman, S. Arora, S. von Arx, M. S. Bernstein, J. Bohg, A. Bosselut, E. Brunskill, E. Brynjolfsson, S. Buch, D. Card, R. Castellon, N. Chatterji, A. Chen, K. Creel, J. Q. Davis, D. Demszky, C. Donahue, M. Doumbouya, E. Durmus, S. Ermon, J. Etchemendy, K. Ethayarajh, L. Fei-Fei, C. Finn, T. Gale, L. Gillespie, K. Goel, N. Goodman, S. Grossman, N. Guha, T. Hashimoto, P. Henderson, J. Hewitt, D. E. Ho, J. Hong, K. Hsu, J. Huang, T. Icard, S. Jain, D. Jurafsky, P. Kalluri, S. Karamcheti, G. Keeling, F. Khani, O. Khattab, P. W. Koh, M. Krass, R. Krishna, R. Kuditipudi, A. Kumar, F. Ladhak, M. Lee, T. Lee, J. Leskovec, I. Levent, X. L. Li, X. Li, T. Ma, A. Malik, C. D. Manning, S. Mirchandani, E. Mitchell, Z. Munyikwa, S. Nair, A. Narayan, D. Narayanan, B. Newman, A. Nie, J. C. Niebles, H. Nilforoshan, J. Nyarko, G. Ogut, L. Orr, I. Papadimitriou, J. S. Park, C. Piech, E. Portelance, C. Potts, A. Raghunathan, R. Reich, H. Ren, F. Rong, Y. Roohani, C. Ruiz, J. Ryan, C. Ré, D. Sadigh, S. Sagawa, K. Santhanam, A. Shih, K. Srinivasan, A. Tamkin, R. Taori, A. W. Thomas, F. Tramèr, R. E. Wang, W. Wang, B. Wu, J. Wu, Y. Wu, S. M. Xie, M. Yasunaga, J. You, M. Zaharia, M. Zhang, T. Zhang, X. Zhang, Y. Zhang, L. Zheng, K. Zhou, and P. Liang. *On the opportunities and risks of foundation models* (2022). 2108.07258. 45

[69] J. Lee, Z. Dai, S. M. K. Duddu, T. Lei, I. Naim, M.-W. Chang, and V. Y. Zhao. *Rethinking the role of token retrieval in multi-vector retrieval* (2023). 2304.01982. 45