Neural Retrieval Models for Matching Patients to Clinical Trials

João da Costa Pereira
joao.costa.pereira@tecnico.ulisboa.pt

Instituto Superior Técnico, Lisboa, Portugal

November 2022

Abstract

Recruiting clinical study participants is the most critical and one of the most challenging parts of the clinical trial process. Thus, it is crucial to investigate the barriers to recruitment and define strategies to improve these processes. One way to overcome this challenge is to utilize the available patient data and information retrieval techniques to match patients to clinical trials. This study aims to experiment with different retrieval techniques for matching clinical trials and patient descriptions using the TREC 2021 Clinical Trials Track test collection. These retrieval methods receive input queries corresponding to patient descriptions and produce a ranked list of clinical trials using dense and sparse term-based retrieval approaches. A Sentence-BERT bi-encoder model was used for the dense retrieval task, trained explicitly for semantic search on a large and diverse dataset. Transfer learning methods were also explored to improve the ranking results by adapting this model to the clinical domain using complementary data. In addition, random negative sampling techniques were experimented to construct a training set for the model learning. Overall, experiments with dense retrieval methods led to better results when compared to sparse techniques. Fine-tuning the models with small in-domain datasets proved to be more effective in retrieval compared to the experiments with pre-trained models in a zero-shot scenario. The best run used a model fine-tuned on two complementary datasets reserved for training. On the other hand, the worst result were obtained when using random negative sampling.
Keywords: Information Retrieval, Neural Models for Ranking Results, Bi-Encoder, BERT.

1. Introduction

Clinical trials have been a part of the medical landscape for many years, creating opportunities for physicians to find viable treatments for a range of conditions. Since the beginning of clinical research, there have been various challenges. Through times, clinical trials have been suffering many changes and updates to how they are carried out and standards that must be followed to ensure they are ethical.

One of the biggest challenges faced just before starting clinical trials is patient recruitment. Patient recruitment is essential to the success of pharmaceutical research and, consequently, patient care. Yet, nearly 80% of clinical trials conducted in the United States fail to meet their enrollment timelines, and up to 50% of research sites enroll one or no patients [1]. This problem translates into millions in lost revenue each day a drug is delayed. More importantly, new cutting-edge medications are delayed in their journey to the patients who need them most. A possible solution to overcome these difficulties, and help the task of matching patient and clinical trials, is to use the vast amounts of patient data already available in the form of electronic health records (EHR).

In 2021, the Text Retrieval Conference (TREC) introduced the Clinical Trials Track. The primary motivation of this task lies in building an automated system to help clinical trials meet their recruitment targets by suggesting trials relevant to a given patient. These patient matching systems should generally take input queries corresponding to patient descriptions and produce a ranked list of clinical trials. The core aspect of the trial descriptions is the inclusion/exclusion criteria. These criteria are a set of characteristics that subject patients must have/not have to be suitable for the study, defining the trial eligibility. Notably, a clinical trial can be relevant for a patient based on their disease or condition, but they might be ruled out due to their age, gender, or other factors. These intricate criteria can be a challenge for retrieval systems. Another challenge of these systems is that patient and trial descriptions are often long.

This study assumed that matching clinical trials to patients could take advantage of advances in language modeling using transformer-based models with limited training data, the often case in the biomedical domain.

In order to study this premise, various exper-

iments were conducted with first-stage retrieval using neural and sparse term-based retrieval approaches with the available test collection from the TREC 2021 Clinical Trials Track. Moreover, these experiments can be divided in two parts: retrieval with sparse representations using BM25, a simple term-based model present in many researched retrieval systems, and retrieval with dense representations using transformer-based models to encode the inputs (e.g., trials and topics) in low-dimensional representation spaces. several dense retrieval approaches were investigated, including experiments with pre-trained models trained on large corpus in a zero-shot scenario, and a fine-tuned version of these models trained on small in-domain data. Random negative sampling techniques were also used to construct the training examples.

This research provides evidence to the following questions when experimenting with first-stage retrieval procedures for ranking clinical trials to patient descriptions. (1) What is the effectiveness of sparse and dense retrieval when ranking clinical trials from the entire collection? (2) Compared to sparse models, how well do BERT-based pre-trained models perform when applied in a zero-shot scenario for retrieval? (3) Does training a model on small in-domain data affect the effectiveness of the downstream task and how it performs compared to a zero-shot scenario with a pre-trained model?

The rest of this document is organized as follows: **Background and Related Work** addresses general concepts important for this study, including a brief description on document retrieval and transformer based neural ranking, followed by an overview of the 2021 edition of the TREC Clinical trials track. **Clinical Trial Data** describes the available datasets for the retrieval task and for training the neural models used for dense retrieval. **Implementation** describes the implementation of the conducted experiments. **Experimental Evaluation** introduces the obtained results for each experiments and discusses this results. **Conclusions and Future Work** concludes the study and addresses possible relevant extensions to this study.

2. Background and Related Work

This section addressed document retrieval in general, followed by an overview of the 2021 edition of TREC Clinical Trials track.

2.1. Document Retrieval

Document retrieval is a standard retrieval task in which an information retrieval system must respond to a previously unseen query by producing a ranked list of documents from a static collection, where documents at the top of the ranking are more likely to be relevant. The texts in the documents often come from a distinct set of authors ranging from several phrases to numerous paragraphs. The query comes from a user and is usually relatively succinct, ranging from a few terms to a few sentences [7][19]. In this study, the queries are represented by synthetic patient descriptions created by individuals with medical training, and the documents are clinical trial descriptions comprised in a large collection.

A significant characteristic of document retrieval is the heterogeneity of the query and the documents, leading to the critical vocabulary mismatch problem [7][4][29]. Matching terms and sentences with similar meanings could alleviate the problem, but exact matching is crucial, particularly with rare terms [7][6]. Relevance assessment is another challenge since it is a time-consuming and expensive process involving human beings [17]. It consists of creating relevance judgments for a subset of documents for each query [17].

2.2. Transformer-based Neural Ranking

In recent years, various neural ranking models (NRMs) based on BERT have been proposed in the information retrieval (IR) community. These BERT-based methods tend to be robust against the vocabulary mismatch problem because they learn semantic representations of query–document pairs in a latent space [19].

Such neural ranking models come in two varieties [8]: bi-encoder models [16][10], which learn separate embeddings for the query and document, and cross-encoder models [21], which learn a joint embedding for the query and document.

Bi-encoders are mainly used for retrieval, given that the individual representations can be indexed through methods supporting the fast execution of maximum inner product searches, such as FAISS [9]. On the other hand, both bi-encoder and cross-encoder models are applicable for re-ranking [18].

2.3. TREC 2021 Clinical Trials Track

The Text Retrieval Conference (TREC) is an annual workshop co-sponsored by the National Institute of Standards and Technology (NIST) and the Intelligence Advanced Research Projects Activity with the purpose of building the infrastructure required for extracting relevant information from large volumes of electronic documents.

TREC has researched various tasks in different domains. In 2021 TREC came up with multiple tracks, including the Clinical Trials Track. This track challanges participants to experiment and implement retrieval methods for ranking clinical trials to patient descriptions. Over hundred runs were submitted from 26 research teams.

The available papers show that several methods were experimented to address the task in hand.

Some teams decided to use sparse retrieval approaches [**?**], while other teams focus on retrieval techniques with dense representations [**?**]. A common practice was to use multi-stage architecture for retrieval, where a more simple sparse model was used for the first-stage, and a second stage using dense representations through transformer-based models [12]. The aim of the first-stage is to retrieve a list of relevant trials from the entire collection using a simple and efficient model. The next stage focuses on re-ranking the obtained list from the first-stage using more effective approaches but more computationally expensive [5].

The transformer-based models were usually pretrained models trained on biomedical domain data, such as BioBERT and ClinicalBERT [2][25][20]. These models were also used for extraction in techniques such as automatic knowledge base (KB) extraction, keyword extraction, and named entity recognition (NER) [20][25]. Query expansion methods were also tested to to enrich the patient descriptions [20].

3. Clinical Trial Data

This section describes the test collections used in this study to retrieve relevant clinical trials for patient descriptions and train a transformer-based model to improve retrieval effectiveness.

3.1. TREC-CT

For this task, a collection of 375,580 publicly accessible clinical trials XML files with descriptions was used, which was obtained from ClinicalTrials.gov [1] on April 27, 2021. These XML files correspond to current and historical clinical trials in the United States and other places.

Additionally, 75 topics were made available, consisting of synthetic patient cases in the form of an admission note created by individuals with medical training. These provided topics are unstructured text documents ranging from 5 to 10 sentences.

The clinical trials corpus includes different fields describing the trials, including the official title, a brief summary, a detailed description of the trial, eligibility criteria, gender, and others.

Each topic-trial pair is judged using three labels: eligible (relevant and the patient does not satisfy any exclusion criteria), excluded (relevant but the patient satisfies some of the exclusion criteria), and not relevant. These relevances r are comprised in a set of relevance judgments, (q, d, r) triples, where q is the topic, and d a clinical trial document. This test collection has 35,832 relevance judgments in a text file, known as qrels file.

[1] `https://clinicaltrials.gov`

3.2. Complementary Data

One of the experiments in this work focuses on improving the effectiveness of dense retrieval with a transformer-based model by fine-tuning it with in-domain data. The idea was to use the Clinical Trials Track test collection just for retrieval, find complementary data related to the track's task, and use it for fine-tuning the neural ranking model. The test collections found were the SIGIR-CT and the TREC-PM.

The SIGIR-CT [11] dataset includes 204,855 publicly available clinical trials, again crawled from ClinicalTrials.gov, and 60 patient case reports describing a patient with certain conditions and observations. The topics were adopted from the TREC Clinical Decision Support Track, 30 from 2014 and 30 from 2015. The relevance assessment file contains 3,870 relevance judgments, however, the labeling scheme differs from that of the TREC CT track. In both test collections, some trials are labeled as entirely relevant, e.g., "Highly likely to refer this patient for this clinical trial" or "eligible," and some are labeled as not relevant. However, this collection includes an intermediate label indicating, "Would consider referring this patient to this clinical trial upon further investigation." In contrast, TREC's intermediate label indicates that patients are excluded, as they meet the inclusion criteria and some exclusion criteria.

The TREC-PM 2017 dataset[2] has 241,006 clinical trials extracted again from ClinicalTrials.gov on April 2017 and 30 synthetic patient cases created by precision oncologists at the University of Texas MD Anderson Cancer Center [23]. These topics include the disease, genetic variants, demographic, and potentially other patient information. The relevance assessment file contains 13,019 relevance judgments in a three-scale labeling scheme, where the trials can be labeled as definitely relevant, partially relevant, and not relevant. The partially relevant is primarily the same as definitely relevant, but with the exception that disease can also be more general, where the form of cancer in the trial is more general than the one in the topic. Gene can also be a missing variant, where the trial does not focus on the particular gene in the topic, or a different variant, where the trial focuses on the particular gene in the topic but a different variant than the one in the topic. Table 1 shows a summary of the datasets.

4. Implementation

Various methods were explored to address the task of retrieve clinical trials for patient case descriptions within the TREC-CT test collection. The implementation of these methods can be divided into

[2] `http://www.trec-cds.org/2017.html`

Table 1: Summary of the available test collections

Dataset	Year	# topics	# trials	# judgments	# relevant
TREC-CT	2021	75	375,580	35,832	5,570
SIGIR-CT Trials	2016	60	204,855	3,870	421
TREC-PM	2017	30	241,006	13,019	436

two main parts. The first part investigated a sparse retrieval approach, whereas the second focused on exploring neural retrieval techniques and compared them with the previous approach. In addition, some techniques were tested to improve effectiveness, including fine-tuning the pre-trained model used for dense retrieval.

The implementation of the various experiments used two Python libraries. Pyserini [15], an easy-to-use toolkit that provides effective first-stage retrieval in a multi-stage ranking architecture, and FAISS[3], a tool that enables efficient similarity search.

The retrieval process can be divided into three steps. The first stage concerns pre-processing the available clinical trials and topics, while the second step indexes these clinical trials and saves them on a local disk. The last step is to search through these indexes and retrieve the top-k clinical trials for each topic.

It is necessary to evaluate the obtained results, in this case, the ranking list, to assess the effectiveness of the retrieval. For evaluation, it was used ranx[4], a python library for fast ranking evaluation.

4.1. General Architecture

All the retrieval approaches use a first-stage architecture, where a set of documents are returned from a large collection. The architecture and the test collection used are always the same throughout all the experiments. Therefore, the only change is the model used for ranking. The general architecture of the retrieval process is shown in Figure 1.

4.2. Pre-processing the Test Collection

The first step before exploring retrieval methods was to examine the test collection to understand how the clinical trials and topics are organized and process these documents if necessary. Since the sparse retrieval implementation uses Pyserini and requires a specific format for the documents to be indexed and posteriorly searched, it was necessary to process these files and save them in the required format. Pyserini (via Anserini) provides ingestors for document collections in three JSON formats. The opted format was the JSONL format, in which a JSON line defined each clinical trial document in a unique file representing the entire collection.

All the information contained in the various tags are extracted recursively and stored in a JSON representation for each trial. The result is a unique string containing information on the different fields of the trial, with no more than a single neighboring whitespace between terms and the information of each field separated by a newline character. This string can aggregate the entire information of the clinical trial, i.e., the texts contained in all the fields, or a combination of certain fields. Processing the texts and choosing the right combination of fields can improve retrieval performance and reduce the vocabulary mismatch between the trials and topics.

Each JSON representation is organized in two keys: *id* containing the trial's unique identifier, and *contents* composed of a unique text containing all the trial information.

4.3. Sparse Retrieval

Before ranking, an inverted index is created storing representations of the trials in a documents–term matrix. This part is particularly important for the efficiency of the models in the first-stage retrieval, since it ranks over the entire collection of documents. To generate the inverted index, the sparse retrieval implementation uses Pyserini. This library builds inverted indexes in the Lucene format[5] [15].

In all experiments with sparse representations, the model used was a default implementation of BM25 using Pyserini. It receives the number of documents to retrieve (e.g., 1000) and a given query used to search through the inverted index. For each query (topic), the top 1000 most similar documents (clinical trials) are fetched with the help of the inverted index computed previously and stored on disk. the final result is a list comprising a ranking with the most 1000 significant trials for each topic and its scores.

4.4. Dense Retrieval

For dense retrieval experiments, the clinical trials were indexed using FAISS. First, these documents are encoded in a low dimensional representation

[3] https://github.com/facebookresearch/faiss

[4] https://amenra.github.io/ranx/

[5] https://lucene.apache.org/core/3_6_2/fileformats.html

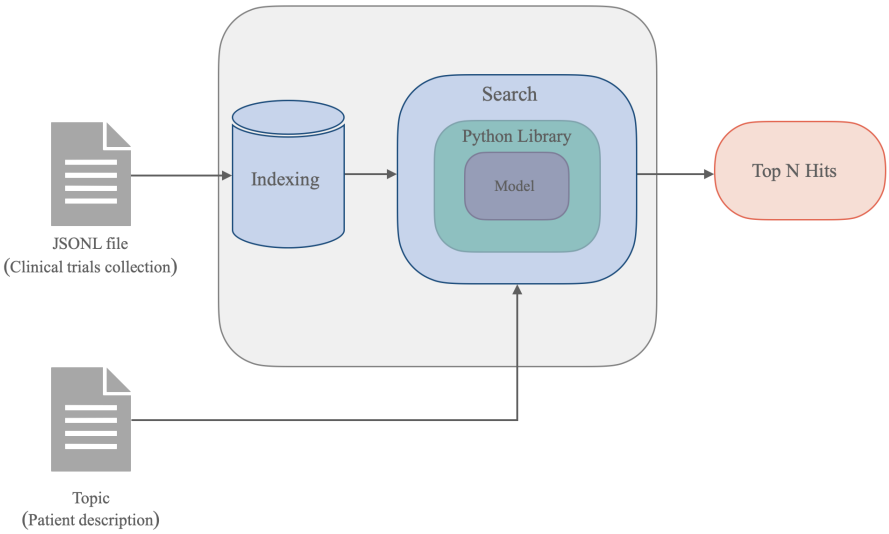

Figure 1: General representation of the retrieval architecture

space. These dense representations are stored in a FAISS index, in this case, a flat index.

For ranking these dense representations, each experiment used once more the FAISS library. Having the FAISS index containing the representation of each trial, the Faiss library is responsible for searching and retrieving the top 1000 relevant trials for each topic. During the search, all the indexed vectors are decoded sequentially and compared to the vector the query (topic) vector.

Both the clinical trials and the topics were encoded, independently, into dense representations using the same BERT-based model.

The similarity measure between vectors depends on the FAISS index that is being used. The flat index used is the IndexFlatIP, which the dot product for similarity measure.

4.5. Model Fine-Tuning

The motivation behind fine-tuning the model is the fact that it is a facto standard in state-of-the-art NLP [24] due to its effectiveness, and consequently, it would improve results obtained in previous tested experiments with pre-trained models in a zero-shot scenario.

The experimented models used were Sentence Transformers models pre-trained on large datasets using the Sentence Transformer[6] framework.

Before training the model it is necessary to prepare the data accordingly to feed it to the model and choose an appropriate loss function. The dataset preparation often consists of creating training examples in the form of pairs of sentences and labels indicating their similarities. In the case of this work, the input data would be pairs of topics and their relevant documents. The strategy used in the experiments does not required the relevance label indicating the relevance of the trial to the topic, thus, only the relevant pairs will be fed to the model as training examples.

The chosen strategy for fine-tuning the models is training with in-batch negatives [10][14][26][28]. It is an effective strategy for learning a bi-encoder model that boosts the number of training examples. It is an alternative to reduce computational cost [10][28] and surpass the problem of having a small dataset (low number of relevance judgments) and, consequently, a small number of training examples. The idea is to use documents from other query-document pairs in the same batch as negative.

Assuming that there are B queries in a batch, each one is associated with a relevant document. Let $\boldsymbol{Q}$ and $\boldsymbol{D}$ be the $(B \times d)$ matrix of query and document embeddings in a batch size of B. $\boldsymbol{S} = \boldsymbol{Q}\boldsymbol{P}^{\top}$ is a $(B \times B)$ matrix of similarity scores, where each row corresponds to a query paired with B documents. In doing so, the model is trained on $B^2(q_i, d_j)$ query-document pairs in each batch. When $i = j$, any (q_i, d_j) pair is a positive instance, and it is a negative one when $i \neq j$. It creates B training instances in each batch, where $B - 1$ are negative documents for each query. In rare cases, for a given query, some documents from other query-document pairs could be relevant.

The Sentence Transformers framework uses the Multiple Negative Ranking loss (MNR loss) for this strategy. The authors state that this loss function is a good option for training embeddings for retrieval

[6] https://www.sbert.net

setups, and its performance usually increases with increasing batch sizes. It can be formalized as:

$$L = -\frac{1}{B}\sum_{i=1}^{B}\frac{\exp\left(\text{sim}\left(q_i, d_i\right)\right)}{\sum_j \exp\left(\text{sim}\left(q_i, d_j\right)\right)} \quad (1)$$

The *sim* is the similarity function between the query and document vectors and can be the cosine similarity or dot product. The similarities are multiplied by a scaling factor of 20, making the differences larger. Having the embeddings and computed the B^2 similarity scores, i.e., the predictions within the batch, it sees the task as a classification task. It applies the cross-entropy loss between the predictions and the gold labels. In this case, the gold labels are in a square matrix, represented by 1 in the diagonal when $i = j$ (and 0 otherwise).

Each experiment with training was operated within the following framework: between each epoch (approximately 52 to 54 steps, depending on the dataset), the trained model performs a retrieval run using the TREC-CT, and the effectiveness of the ranking is evaluated. The trained model is stored to be used in the next epoch. When a trained model is less effective than its previous version, it is no longer trained. It took on average four epochs to train the model (longer fine-tuning did not yield improvement on the retrieval task).

Each model was trained with batches of size 8, a linear warmup schedule for 50% of an epoch, and Adam optimizer with learning rate 2e-5. The maximum input sequence was set to 512.

4.6. Filtering Results with Regular Expression Rules

As described previously, each topic-trial pair is judged as eligible, excluded, or not relevant. The exclusion depends on the patient's characteristics and the eligibility criteria specifying what characteristics or conditions a patient must have/not have to be suitable for the trial.

Since choosing a correct clinical trial largely depends on matching its eligibility criteria in terms of some factors (e.g., age and gender), in some of the conducted experiments, a simple strategy was used to penalize the relevance scores between a given trial and a topic when the patient meets the exclusion criteria. The idea was to extract and standardize metadata from the patient descriptions using regular expressions, as implemented in other thesis relevant to this study [3]. The extracted metadata was the age and gender of the patients. For the clinical trials, the latter is represented by *male* and *female*. However, it can have different representations for patient descriptions, for instance, *man*, *gentleman*, *boy*, or *M* in the case of a male. Therefore, these values were normalized to the values of the clinical trials.

With these rules, it was possible to compare patients' demographics with the eligibility demographics of a clinical trial and understand if a patient is eligible for the trial. When one of these patients' metadata is incongruent with the trial, the patient is considered not eligible for the given trial. In this case, the score between the patient-trial pair is penalized by discounting some factor.

The process starts with a ranking list obtained from a chosen retrieval run, and the demographics are compared for each patient-trial pair in the list. Whenever a patient does not meet the trial's age and gender information, the pair's score is decremented by a value.

5. Experimental Evaluation

Several experiments were conducted to retrieve the most relevant clinical trials for patient descriptions using traditional and more recent approaches using deep learning and state-of-the-art architectures in NLP and information retrieval. This section reports the obtained results from the executed experiments.

5.1. Sparse Retrieval

The first two run experimented with a sparse retrieval technique. The selected model was BM25 due to its simplicity and effectiveness, but also because it has been a standard in information retrieval research and widely used in all the TREC Clinical Trials track's submissions explored. In addition, some teams used BM25 as a term of comparison for more complex models that operate with dense contextualized representations. The obtained results can be seen in Table 2.

Runs

(1) The clinical trials were indexed entirely using all the available fields. In the second run, the clinical trials were indexed with the following combination of fields: brief title, official title, brief summary detailed description, and eligibility criteria. These free text fields contain relevant information about the study and its essential criteria. The results still have a lot to improve.

(2) Results show an improvement in all the evaluated metrics. In both experiments, the fraction of relevant trials retrieved is very similar, with more than one point of difference in recall. However, there is a more significant difference when comparing the NDCG@10 for both runs. This difference could prove that selecting specific fields impacts the quality of the retrieved list and can be explained by the fact that many fields are noisy and thus negatively affect the retrieval. This results also supports that term-based retrieval models are predisposed to the vocabulary mismatch problem. This result is

Table 2: Results on the retrieval task with the TREC-CT test collection using a sparse retrieval model, BM25. (5) indicates that the combination of fields included five fields: brief title, official title, brief summary, detailed description, criteria.

Run	Model	Fields	TREC-CT				
			P@10	R-Prec	MRR	Recall	NDCG@10
(1)	BM25	entire	0.1053	0.0682	0.2441	0.2431	0.2018
(2)	BM25	combination(5)	0.1640	0.0925	0.3128	0.2576	0.2917
(3)	BM25 + Regex Filter	combination(5)	**0.1920**	**0.1023**	**0.3471**	**0.2576**	**0.3145**

used as a term of comparison for the experiments with dense retrieval.

(3) The third run experimented regular expressions to create a method for penalizing the score of patient-trial pairs in a ranking list whenever a patient meet exclusion criteria for the trial in the pair. In this case, the ranked list used was the one obtained from the second run (2).

The obtained results had improved gains between 1 to 3% depending on the evaluated metric. As expected the recall did not change since the elements in the list are the same, however the quality of the ranking list improved for the top 10 as shown by the improvements on the NDCG@10. Overall, the results obtained in this run are the best ones for the sparse retrieval experiments.

5.2. Dense Retrieval

Table 3 presents the various runs based on dense retrieval and the obtained results for the different scenarios tested. These results can be divided into two parts: the two experiments with the pre-trained models in a zero-shot scenario, where the models were trained in data that is not related to the domain in this study, and four other runs where the pre-trained models were fine-tuned. These runs differ in the dataset used for training the model and in the batch construction for the case of the run (7). As a reminder, in all these runs, the clinical trials were encoded and indexed using the combination of fields used in the second run of the sparse retrieval experiments.

Pre-trained Models

(4) Ranks the clinical trials for the patient descriptions using the model msmarco-bert-base-dot-v5. Comparing the evaluated metrics with the best run for the sparse retrieval, the results show a small increasing of almost one and a half points for recall, however, they show a decreasing in the other metrics. The most significant difference is in the NDCG@10, indicating that the quality of the ranked list is worst. Moreover, it is closer to the first run of the sparse experiments where all the trials fields were used, than to the second run. This run ranks the clinical trials for the patient descriptions using the model msmarco-bert-base-dot-v5.

These results corroborate the fact that not all transformer-based models are perfect for a ready-to-use situation and BM25 could be a strong baseline for retrieval. Although BM25 is a term-based model tested in a default scenario, it outperformed the semantic-based approach with dense representations produced by a transformer-based model pre-trained with a large dataset. In defense of dense retrieval, this first model was tested in a zero-shot scenario.

(5) Another pre-trained model was tested, the all-distilroberta-v1. The results show a significant overall improvement compared to the previous run (4). Moreover, some evaluated metrics show an increase of almost 10 points (e.g., NDCG@10) and others more than 10 points (e.g., MRR and recall). For recall, it almost doubled the values of the previous runs, making it the best candidate for a first-stage retrieval so far. This improvement in the results could be explained by the data used to train the all-distilroberta-v1. According to the Sentence Transformers authors, the model was trained with all their available training data (more than 1 billion training pairs). One important factor is that the training data includes sections from scientific medicine and biology papers. Also to support these results, the authors in [13] noticed a significant performance gain when using distillation in pre-training stage for a bi-encoder when compared to a model that does not use distillation.

Fine-tuned Models

Having tested with pre-trained models in a zero-shot scenario and improved the results over the sparse retrieval experiments, the following tests focused on improving results by fine-tuning the model used in the fifth run (5). As described previously, the models were fine-tuned on small in-domain datasets in different strategies.

(6) The strategy was to fine-tune the all-distilroberta-v1 on the SIGIR-CT dataset. This dataset is very similar to the one used to evaluate the model (TREC-CT). In-batch negative sampling was used to increase the number of training exam-

Table 3: Results on the retrieval task with the TREC-CT test collection using experimented transformer-based models, pre-trained and fine-tuned. The best BM25 run is included for better comparison. [(1)] indicates that training used random negative sampling with one sample. (6) indicates that the model trained was the fine-tuned model in run (6).

Run	Model	Dataset	Encoder	TREC-CT				
				P@10	R-Prec	MRR	Recall	NDCG@10
(2)	BM25	-	-	0.1640	0.0925	0.3128	0.2576	0.2917
	Pre-trained							
(4)	msmarco-bert-base-dot-v5	MSMARCO	Bi-Encoder	0.1347	0.0740	0.3117	0.2701	0.2343
(5)	all-distilroberta-v1	Multiple	Bi-Encoder	0.1973	0.1280	0.4501	0.4221	0.3220
	Fine-tuned							
(6)	all-distilroberta-v1	SIGIR-CT	Bi-Encoder	0.2293	0.1630	0.3971	0.5937	0.3389
(7)	all-distilroberta-v1[(1)]	SIGIR-CT	Bi-Encoder	0.1200	0.0854	0.2896	0.4322	0.1613
(8)	all-distilroberta-v1	TREC-PM	Bi-Encoder	0.2360	0.1777	0.4572	0.5995	0.3501
(9)	all-distilroberta-v1(6)	TREC-PM	Bi-Encoder	0.2720	0.1832	0.4459	0.5980	0.3672
	Regex Filter							
(10)	all-distilroberta-v1(6) + Regex Filter	TREC-PM	Bi-Encoder	**0.2973**	**0.2036**	**0.4573**	**0.5980**	**0.3858**

ples.

Comparing its results with the previous run (5), the NDCG@10 improved very little. However, only recall had a significant improvement, with an increase of 17%.

(7) The model was fine-tuned using a simple random negative sampling technique together with in-batch negatives. The results were the worst for dense retrieval, with a significant decrease in all the evaluated metrics, but mainly the NDCG@10.

The authors in [27] show that random negative sampling methods, such as random negatives and in-batch negatives, minimize the total pairwise errors but cannot effectively minimize the top-K pairwise errors. Moreover, these techniques allow difficult queries to dominate the training easily, resulting in a serious loss of top-ranking performance [27], which supports the decreased value for NDCG@10.

(8) The applied strategy was very similar to the fifth run (6). These two runs differ on the dataset used to fine-tuned the model, where in this run the data used comes from the TREC-PM test collection. The idea of these experiment was to investigate how could a very different dataset affect training and consequently the ranking task.

The results obtained for this run outperform all the other previous experiments. However, these are very similar to the results obtained in the sixth run (6), where the all-distilroberta-v1 model was fine-tuned on the SIGIR-CT dataset.

(9) This experiment used the TREC-PM dataset for training as in the previous run, but instead of training the pre-trained version of the all-distilroberta-v1, it trained the fine-tuned model obtained in the sixth run (6). The ranking result from using this fine-tuned model yields the best performance overall, with an improvement of almost 4% for P@10, and close to 2% for the NDCG@10 when compared to the previous run.

(10) As in the third run (3), this run experimented regular expressions to extract metadata from the topics and create a method that penalizes relevance scores between topic-trial pairs in a ranking list whenever the extracted information from the topic does not satisfies the trial's eligibility demographics. For this run, the ranking list used was the one that led to the best results, the run (9).

Similar to the run (3), the obtained results had gains between 1 and 2% depending on the evaluated metric. Overall, the results obtained in this run are the best ones for the sparse and dense retrieval experiments.

5.3. General Discussion

What is the effectiveness of sparse retrieval when ranking clinical trials using the entire collection? The results presented in Table 2 show a decrease in performance when analysing the evaluated metrics and compare them to the second run with a small combination of fields. Moreover, the quality of the ranking list got worst as the NDCG@10 almost 10% lower. However, first-stage retrieval aims to recall potentially relevant documents as many as possible [5]. Thus, both options would have a very similar behaviour for a first-stage or recall ranking.

Compared to sparse models, how well do BERT-based pre-trained models perform when applied in a zero-shot scenario for retrieval? The obtained results prove that depending on the pre-trained model used, the performance could be worst (e.g., third run) or could outperform

significantly more classical approaches with sparse models such as the one tested in this study, a default implementation of BM25.

It is important to consider both options and select according to some factors. For instance, using BM25 is less computationally expensive when compared to BERT-based models. In addition, BM25 would be a much better option than training BERT from scratch with a small dataset. However, many tools, e.g., HuggingFace, enable users to build, train and deploy ML models based on open-source code and technologies. Hence, one can easily find a pre-trained BERT-based model for any domain without worrying about training the model and the associated costs.

Another factor to be considered is the index size, as the performance of dense representations decreases quicker than sparse representations for increasing index sizes [22]. In addition, lets not forget that term-based models may suffer from the vocabulary mismatch problem and ignores term ordering information [5].

Does training a model on small in-domain data affect the effectiveness of the downstream task and how it performs compared to a zero-shot scenario with a pre-trained model? The experiments with fine-tuning show an improve in effectiveness compared to ranking with pre-trained models on data not related to the domain in study. In fact, the best run comes from one of these experiments with a model fine-tuned on two very different in-domain datasets.

6. Conclusions and Future Work

The focus of this study was to address transformer-based models and investigate their ability to integrate first-stage retrieval and replace traditional approaches with term-based models. These models have a very important role in the whole effectiveness of a multi-stage pipeline, right from the beginning, since they are responsible for retrieving a ranking list of relevant documents to be re-ranked in the following stages of the pipeline. Thus, when first-stage retrieval lacks performance, it affects the entire system.

A fine-tuning setup for bi-encoders, for the task of clinical trials document retrieval (i.e., retrieve relevant trial to patient descriptions) was proposed to improve transformer-based models, and, consequently, improve first-stage retrieval with dense representations. The model fine-tuning setup focuses on batch construction through random negative sampling procedure, more specifically, in-batch negative sampling, and in one experiment another random sampling technique was used based on easy negatives from the relevance judgments. In addition, two very different datasets were used to investigate the effect of a model fine-tuned with small in-domain data and its effect in the retrieval task.

Experiments showed that fine-tuning a model improved the results obtained in other experiments with a sparse model and with pre-trained dense models that were not related to the domain in study. From all the evaluated metrics, recall was the one that showed the best improvements. It would be interesting to compare recall with other experiments submitted to the TREC 2021 Clinical Trials track. Unfortunately, all the submissions found were based on re-ranking.

An extension of this study could be using extraction techniques using transformer models to summarize the clinical trials and patient descriptions producing semantically meaningful sentence embeddings. This could help mitigating the vocabulary mismatch problem, but also resolve the problem of the maximum sequence limitation in the case transformer-based models. This way relevant information could fit the transformer models without losing information.

It would be relevant to test pre-trained models on more related domains or in the same domain if possible to compare their performance with the already experiments. In addition, these models could be fine-tuned with the datasets used in this study and once again compare their performance with previous experiments.

Finally, it would be interesting to experiment with hard negative sampling techniques and see much improvements they can bring, given that the experience with random negatives lead to the worst results for dense retrieval.

Acknowledgements

The author would like to acknowledge the project supervisors, Prof. Bruno Martins and Prof. João Magalhães, for their frequent guidance and dedicated involvement in every step throughout the process. He also gratefully acknowledges the support of INESC-ID for providing the hardware that made possible the concretization of this work.

References

[1] Clinical trial delays: America's patient recruitment dilemma - Clinical Trials Arena, 2022.

[2] L. Biester, V. Joopudi, and B. Dandala. IBM @ TREC Clinical Trials Track 2021. Text REtrieval Conference, 2021.

[3] B. Cardoso. TRIALMATCH: A Transformer Architecture to Match Patients to Clinical Trials. Master's thesis, NOVA University Lisbon, 2022.

[4] G. W. Furnas, T. K. Landauer, L. M. Gomez, and S. T. Dumais. The vocabulary problem

in human-system communication. *Communications of the ACM*, 30(11), 1987.

[5] J. Guo, Y. Cai, Y. Fan, F. Sun, R. Zhang, and X. Cheng. Semantic Models for the First-stage Retrieval: A Comprehensive Review. *ACM Transactions on Information Systems*, 40(4), 2022.

[6] J. Guo, Y. Fan, Q. Ai, and W. B. Croft. A Deep Relevance Matching Model for Ad-hoc Retrieval. *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, 2016.

[7] J. Guo, Y. Fan, L. Pang, L. Yang, Q. Ai, H. Zamani, C. Wu, W. B. Croft, and X. Cheng. A Deep Look into Neural Ranking Models for Information Retrieval. *CoRR*, abs/1903.06902, 2019.

[8] S. Humeau, K. Shuster, M.-A. Lachaux, and J. Weston. Poly-encoders: Transformer Architectures and Pre-training Strategies for Fast and Accurate Multi-sentence Scoring. *CoRR*, abs/1905.01969, 2019.

[9] J. Johnson, M. Douze, and H. Jégou. Billion-scale similarity search with GPUs. *CoRR*, abs/1702.08734, 2017.

[10] V. Karpukhin, B. Oğuz, S. Min, P. Lewis, L. Y. Wu, S. Edunov, D. Chen, and W. tau Yih. Dense Passage Retrieval for Open-Domain Question Answering. *CoRR*, abs/2004.04906, 2020.

[11] B. Koopman and G. Zuccon. A test collection for matching patients to clinical trials. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, 2016.

[12] W. Kusa and Y. Ghafourian. DOSSIER at TREC 2021 Clinical Trials Track. Text REtrieval Conference, 2021.

[13] J. Lei, X. Chen, N. Zhang, M. Wang, M. Bansal, T. L. Berg, and L. Yu. Loop-ITR: Combining Dual and Cross Encoder Architectures for Image-Text Retrieval. *ArXiv*, abs/2203.05465, 2022.

[14] Y. Li, Z. Liu, C. Xiong, and Z. Liu. More Robust Dense Retrieval with Contrastive Dual Learning, July 2021. arXiv:2107.07773 [cs].

[15] J. Lin, X. Ma, S.-C. Lin, J.-H. Yang, R. Pradeep, and R. Nogueira. Pyserini: A Python Toolkit for Reproducible Information Retrieval Research with Sparse and Dense Representations. *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2021.

[16] Y. Luan, J. Eisenstein, K. Toutanova, and M. Collins. Sparse, Dense, and Attentional Representations for Text Retrieval. *CoRR*, abs/2005.00181, 2020.

[17] C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[18] A. Menon, S. Jayasumana, A. S. Rawat, S. Kim, S. Reddi, and S. Kumar. In defense of dual-encoders for neural ranking. In *Proceedings of the 39th International Conference on Machine Learning*. PMLR, 2022. ISSN: 2640-3498.

[19] B. Mitra and N. Craswell. Neural Models for Information Retrieval. *CoRR*, abs/1705.01509, 2017.

[20] H. Nguyen, H. Chen, B. Prasad, H. Zhao, J. Ding, J. Chen, and A. Cleveland. Untiia lab at trec 2021-clinical trial. Text REtrieval Conference, 2021.

[21] R. Nogueira and K. Cho. Passage Re-ranking with BERT. *CoRR*, abs/1901.04085, 2019.

[22] N. Reimers and I. Gurevych. The Curse of Dense Low-Dimensional Information Retrieval for Large Index Sizes. *CoRR*, abs/2012.14210, 2020.

[23] K. Roberts, D. Demner-Fushman, E. M. Voorhees, W. R. Hersh, S. Bedrick, A. J. Lazar, and S. Pant. Overview of the trec 2017 precision medicine track. In *The... text REtrieval conference: TREC. Text REtrieval Conference*, volume 26. NIH Public Access, 2017.

[24] S. Ruder, M. E. Peters, S. Swayamdipta, and T. Wolf. Transfer Learning in Natural Language Processing. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Tutorials*, Minneapolis, Minnesota, 2019. Association for Computational Linguistics.

[25] H. Schäfer, A. Idrissi-Yaghir, W. Galetzka, M. Bexte, and C. Friedrich. Wispermed text at trec clinical trials track 2021. Text REtrieval Conference, 2021.

[26] 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. *CoRR*, abs/2009.12756, 2020.

[27] J. Zhan, J. Mao, Y. Liu, J. Guo, M. Zhang, and S. Ma. Optimizing Dense Retrieval Model Training with Hard Negatives. *CoRR*, abs/2104.08051, 2021.

[28] J. Zhan, J. Mao, Y. Liu, M. Zhang, and S. Ma. RepBERT: Contextualized Text Embeddings for First-Stage Retrieval. *CoRR*, abs/2006.15498, 2020.

[29] L. Zhao and J. Callan. Term necessity prediction. *Proceedings of the 19th ACM international conference on Information and knowledge management*, 2010.