DEPARTMENT OF
COMPUTER SCIENCE

BRUNO DANTAS CARDOSO

BSc in Computer Science

TRIALMATCH

A TRANSFORMER ARCHITECTURE TO MATCH PATIENTS TO CLINICAL TRIALS

MASTER IN COMPUTER SCIENCE

NOVA University Lisbon
December, 2022

DEPARTMENT OF
COMPUTER SCIENCE

TRIALMATCH

A TRANSFORMER ARCHITECTURE TO MATCH PATIENTS TO CLINICAL TRIALS

BRUNO DANTAS CARDOSO

BSc in Computer Science

Adviser: João Miguel da Costa Magalhães
Associate Professor, NOVA University Lisbon

Co-adviser: Bruno Emanuel da Graça Martins
Associate Professor, University of Lisbon

Examination Committee:

Chair: Nuno Manuel Ribeiro Preguiça
Associate Professor, NOVA University Lisbon

Rapporteur: Francisco José Moreira Couto
Associate Professor, University of Lisbon

Adviser: João Miguel da Costa Magalhães
Associate Professor, NOVA University Lisbon

MASTER IN COMPUTER SCIENCE

NOVA University Lisbon
December, 2022

TrialMatch

This document was created using the (pdf/Xe/Lua)LaTeX processor, based on the NOVAthesis template, developed at the Dep. Informática of FCT-NOVA by João M. Lourenço. [50]

Acknowledgements

I would like to thank my adviser, Dr. João Magalhães, for his guidance, availability, and patience through the development of this thesis. I would like to thank my co-advisor, Dr. Bruno Martins, for all his help and valuable knowledge.

Furthermore, I would like to thank NOVA School of Science and Technology | FCT NOVA and the NOVA Search Group for giving me the necessary tools to complete this work.

Finally, to my friends and family, a special thanks, because your encouragement and help throughout these years, made it possible.

"Live as if you were to die tomorrow. Learn as if you were to live forever." (Mahatma Gandhi)

Abstract

Around 80% of clinical trials fail to meet the patient recruitment requirements, which not only hinders the market growth but also delays patients' access to new and effective treatments. A possible approach is to use Electronic Health Records (EHRs) to help match patients to clinical trials. Past attempts at achieving this exact goal took place, but due to a lack of data, they were unsuccessful. In 2021 Text REtrieval Conference (TREC) introduced the Clinical Trials Track, where participants were challenged with retrieving relevant clinical trials given the patient's descriptions simulating admission notes. Utilizing the track results as a baseline, we tackled the challenge, for this, we resort to Information Retrieval (IR), implementing a pipeline for document ranking where we explore the different retrieval methods, how to filter the clinical trials based on the criteria, and reranking with Transformer based models. To tackle the problem, we explored models pre-trained on the biomedical domain, how to deal with long queries and documents through query expansion and passage selection, and how to distinguish an eligible clinical trial from an excluded clinical trial, using techniques such as Named Entity Recognition (NER) and Clinical Assertion. Our results let to the finding that the current state-of-the-art Bidirectional Encoder Representations from Transformers (BERT) bi-encoders outperform the cross-encoders in the mentioned task, whilst proving that sparse retrieval methods are capable of obtaining competitive outcomes, and to finalize we showed that the use of the demographic information available can be used to improve the final result.

Keywords: TREC, Clinical Trial, Electronic Health Record, Transformer, BERT, T5, Information Retrieval, Document Ranking, Learning to Rank, Biomedical Domain.

Resumo

Cerca de 80% dos ensaios clínicos não satisfazem os requisitos de recrutamento de pacientes, o que não só dificulta o crescimento do mercado como também impede o acesso dos pacientes a novos e eficazes tratamentos. Uma abordagem possível é utilizar os Prontuários Eletrônicos para ajudar a combinar doentes a ensaios clínicos. Tentativas passadas para alcançar este exato objetivo tiveram lugar, mas devido à falta de dados, não foram bem sucedidos. Em 2021, a TREC introduziu a *Clinical Trials Track*, onde os participantes foram desafiados com a recuperação ensaios clínicos relevantes, dadas as descrições dos pacientes simulando notas de admissão. Utilizando os resultados da track como base, enfrentámos o desafio, para isso recorremos à Recuperação de Informação, implementando uma *pipeline* para a classificação de documentos onde exploramos os diferentes métodos de recuperação, como filtrar os ensaios clínicos com base nos critérios, e reclassificação com modelos baseados no *Transformer*. Para enfrentar o problema, explorámos modelos pré-treinados no domínio biomédico, como lidar com longas descrições e documentos, e como distinguir um ensaio clínico elegível de um ensaio clínico excluído, utilizando técnicas como Reconhecimento de Entidade Mencionada e Asserção Clínica. Os nossos resultados permitem concluir que os actuais bi-encoders de última geração BERT superam os cross-encoders BERT na tarefa mencionada, provamos que os métodos de recuperação esparsos são capazes de obter resultados competitivos, e para finalizar mostramos que a utilização da informação demográfica disponível pode ser utilizada para melhorar o resultado final.

Palavras-chave: TREC, Ensaios Clínicos, Prontuário Eletrônico, Transformer, BERT, T5, Recuperação de Informação, Classificação de Documentos, Aprender a Classificar, Domínio Biomédico.

Contents

List of Figures

List of Tables

Acronyms

BERT	Bidirectional Encoder Representations from Transformers vi, vii, xi, 9, 11, 12, 13, 14, 15, 16, 21, 22, 23, 37, 38, 40
C4	Colossal Clean Crawled Corpus 12
CT	Clinical Trials x, 2, 6, 17, 24, 33, 43
DFR	Divergence from Randomness 6
EHR	Electronic Health Record vi, 1, 2, 3, 23
IE	Information Extraction 21
IR	Information Retrieval vi, 2, 6, 7, 9, 11, 19, 32, 48, 61
LM	Language Model 6, 40
LTR	Learning to Rank 7, 8, 9
ML	Machine Learning 7, 8
MLM	Masked Language Model 12
NER	Named Entity Recognition vi, 19, 21, 23, 45, 46, 47, 59, 61
NIH	National Institutes of Health 1
NLI	Natural Language Inference 14
NLP	Natural Language Processing 10, 11, 13, 19, 21, 22
NSP	Next Sentence Prediction 12, 14
PM	Precision Medicine x, 1, 17
PRF	Probabilistic Relevance Framework 6, 7

QA	Question Answering 14
RE	Relation Extraction 21, 23
RRF	Reciprocal Rank Fusion 17, 18, 34, 36
SA	Sentiment Analysis 47, 59, 61
SBERT	Sentence-BERT 10, 15, 16
T5	Text-to-Text Transfer Transformer 12, 16
TCP	Target Corpus Pretraining 21
TREC	Text REtrieval Conference vi, vii, 1, 2, 17, 33

1

Introduction

1.1 Context and Motivation

Clinical trials, research studies focused on testing new treatments (e.g. a new drug) and evaluating their effects on human health, for this to be possible, every study needs to meet a certain number of participants just to be able to begin. The recruitment of patients for a clinical trial requires a lot of manual and laborious work that, in most cases, ends up not paying off, given that the National Institutes of Health (NIH) estimated that about 80% of clinical trials fail to recruit enough patients [79], which not only poses a waste of time and money, but more critically it poses as a real issue for patients, that are not able to obtain new effective treatments and are in turn exposed to outdated ones and their possible side effects. Due to how ineffective the patient recruitment is, a not soo common approach is for the patient himself to do his own search and volunteering for clinical trials, this has become a trend with patients that have a particular condition for which there is no known or effective treatment available.

A possible solution to overcome the problem of matching patients to the clinical trials is to utilize EHRs, an EHR contains a patient's medical history, such as diagnoses. These documents are used in various tools in the medical care field with success. The use of EHR in clinical trial recruitment is a well-known use case, as stated in [98]. The patient's EHR can be composed in several ways, being the critical difference in this work, the use of unstructured free-text patient descriptions. The use of semi-structured data is well studied in the TREC PM Track [75, 76, 77], to match oncologist patients to relevant trials, and so this thesis will be limited to the study of unstructured EHRs.

Past attempts at exploring the use of unstructured EHRs, started with the TREC Medical Records Track 2011-2012 [95], which was discontinued due to the lack of a large enough dataset, this track sought to explore the patient-trial match task. Afterward, the Clinical Decision Support Track [73, 74, 84] took place annually from 2014 to 2016, where the participants had the task of retrieving biomedical articles according to patient's EHRs, in the 2014 run, experts built a synthetic dataset with plain text describing the patient, and based on this dataset in 2016 a collection for matching patients to clinical trials was introduced (SIGIR-CT [32]), which aimed to help future development in the field.

1.2 Problem Statement

Following the problem of matching patients with clinical trials, we formalize it as follows, given a patient p and a collection of clinical trials C, the goal is to return an ordered list of (p, c) pairs, where $c \in C$, such that c is a relevant clinical trial for that patient, and the higher the clinical trial is on the list, the more relevant it is for that patient. To solve this, we resort to Ranking in IR, and using the solutions provided by this discipline, we tackle the task. Still, first, we need to overcome the two core problems, the biomedical domain that makes it harder to use well-researched general domain tools, and the length of the documents (EHR and clinical trials).

The primary objective is to build a retrieval system, that could possibly be integrated in an autonomous system capable of helping clinical trials meet their recruitment requirements, which means that only matching patients with relevant clinical trials is insufficient. Therefore, the relevant trials group can be divided into two subgroups, taking the criteria requirements into account, excluded clinical trials, and eligible ones. For this, we will have to find solutions that "pick apart" the criteria and can take the medical and demographic terms into account in the decision-making (Figure 1.1), mostly approaches based on the Transformer [94] were explored. To encourage the research of this, TREC introduced in 2021 the CT Track[1].

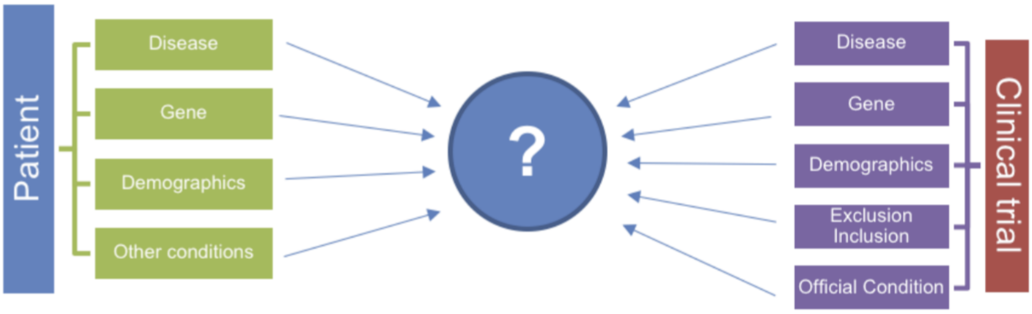


Figure 1.1: Input data of a retrieval system [6]

1.2.1 TREC Clinical Trials Track

In 2021, TREC introduced the CT Track, with the objective of utilizing EHRs to assist patient recruitment of clinical trials. This track flips the trial-to-patients paradigm to a patient-to-trials paradigm, enabling the evaluation of systems that match the patient to a

[1] http://www.trec-cds.org/2021.html

trial and building a test collection for clinical trial search, meaning the topics are the patient descriptions and the corpus are the clinical trials. The participants of the track were tasked with retrieving clinical trials from ClinicalTrials.gov, a required registry for clinical trials in the United States. The trials are composed of various fields, creating a lengthy document, but the core aspect of this work is the inclusion/exclusion criteria (Figure 1.2). The topics consist of synthetic patient case descriptions that simulate an admission statement in an EHR (Figure 1.3), the admission notes were created by individuals with a medical background and are between 5 to 10 sentences long.

With the task of retrieving the top 1000 relevant clinical trials for 75 synthetic EHR, the test collection was published where the clinical trials are judged in the following manner:

- **Not Relevant**. The clinical trial is not in any way relevant to the patient.
- **Excluded**. The patient has the condition targeted by the trial but does not meet the eligibility criteria.
- **Eligible**. The clinical trial is relevant and eligible for the patient.

1.3 Contributions

As a result of the work performed in this thesis, we highlight the following contributions:

- Exploration of a sparse retrieval method, where query expansion techniques are explored and rank fusion.
- Systematic testing of cross-encoders, fine-tuning of in-domain pretrained models, and exploration of different approaches to ranking with such models.
- The addition of a stage to the ranking pipeline, which we call post-rank filter.

1.4 Document Structure

The remainder of this work is structured as follows:

- **Chapter 2, Background and Related Work**. Introduces the core concepts and state-of-the-art around document ranking, the Transformer, and the biomedical domain.
- **Chapter 3, Experimental Setup**. Describes the proposed setup to solve this task and the datasets involved.
- **Chapter 4, Approaching the Matching Problem** . Details the developed work done, from the retrieval stage to the use of neural ranking models, while also exploring how to make a distinction between Eligible and Excluded clinical trials.

```
<eligibility>
                                   ...
<criteria>
<textblock>
Inclusion Criteria:
- Patients must have a diagnosis of cancer of any histologic type.
- Patients must have a Karnofsky performance status great or equal to 70%.
- Patients must have an expected survival for at least four months.
- Normal healthy volunteers to serve as control for this study.
- All patients must sign informed consent approved by the Committee on the Use of Human
Subjects at the University of Minnesota
Exclusion Criteria:
- Pregnant or lactating women. Females of child-bearing potential will be asked to take a preg-
nancy test before receiving vaccines.
- Serious intercurrent medical illnesses which would interfere with the ability of the patient to
carry out the follow-up monitoring program.
- Immunization should not be administered during the course of any febrile illness or acute
infection.
- Hypersensitivity to any component of the vaccine, including Thimerosal, a mercury derivative.
- The occurrence of any type of neurologic symptoms to tetanus vaccine in th past.
- Patients with a history of seafood allergy are excluded from receiving KLH.
- Subjects who have had tetanus toxoid within the last 7 years are not eligible for tetanus vaccine
component of this protocol.
</textblock>
</criteria>
                                   ...
<gender>All</gender>
<minimum_age>18 Years</minimum_age>
<maximum_age>N/A</maximum_age>
                                   ...
```

Figure 1.2: Example from a Clinical Trial (NCT00000105)

> A 2-year-old boy is brought to the emergency department by his parents for 5 days of high fever and irritability. The physical exam reveals conjunctivitis, strawberry tongue, inflammation of the hands and feet, desquamation of the skin of the fingers and toes, and cervical lymphadenopathy with the smallest node at 1.5 cm. The abdominal exam demonstrates tenderness and enlarged liver. Laboratory tests report elevated alanine aminotransferase, white blood cell count of 17,580/mm, albumin 2.1 g/dL, C-reactive protein 4.5 mg, erythrocyte sedimentation rate 60 mm/h, mild normochromic, normocytic anemia, and leukocytes in urine of 20/mL with no bacteria identified. The echocardiogram shows moderate dilation of the coronary arteries with possible coronary artery aneurysm.

Figure 1.3: Example Topic (0)

- **Chapter 5, Evaluation and Result Analysis**. Provides and discusses the obtained results using the methods described throughout the dissertation.
- **Chapter 6, Conclusions and Future Work**. Presents the conclusions of this thesis and avenues for future work.

2

Background and Related Work

This chapter presents the required background and concepts related to ranking, transformers, and the CT Track. First, we introduce IR and its classic ranking models, and then we follow up on how our words and sentences are represented with embeddings. Thirdly, we introduce the Transformer and more modern variations, and in the fourth section, we present the various techniques used to utilize the Transformer in document ranking. The fifth section introduces what was done by participating teams in the 2021 CT Track and a critical analysis of their work. In the subsequent sections, we mention how to tackle some of the problems of this task, the sequence length, how to extract some key elements, and how the biomedical domain affects the problem.

2.1 Classic Text Ranking Models

In classic IR literature various ranking models have been proposed [7], a in-depth comprehension of the models that we explored is done in the subsequent subsections.

2.1.1 Retrieval Models

With the use case of the initial step of fetching information from the dataset, these IR models are used as the starting point in retrieving relevant documents given a topic/query, although they don't provide the best rank, these processes are computationally efficient and can fastly retrieve from a large corpus, and the ranking can be subsequently refined. This retrieval process is generally used as a first-stage retrieval thus, high recall is looked after, as this process retrieves the top-k documents from an extensive data collection.

The retrieval models can be viewed as three types [17]: Boolean Models, Vector Space Models, and Probabilistic Models. In the probabilistic models, the information retrieval process is captured in a probabilistic framework, following the Probabilistic Ranking Principle [80], where the relevance of a pair query-document is the probability of the document being relevant and the ranking list is ordered in that way.

There are a number of probabilistic IR models, notably: Language Model (LM) [35, 36], Divergence from Randomness (DFR) [5], and Probabilistic Relevance Framework

(PRF) [81]. Emphasizing the PRF family, where we find BM25, a popular well established and successful ranking function used in many IR systems and research.

2.1.1.1 Okapi BM25

Okapi Best Matching 25 [81], the name of the actual ranking function is BM25, it includes Okapi due to the Okapi IR system being the first to use it. The usual BM25 scoring function is as follows:

$$score(D,Q) = \sum_{i=1}^{n} IDF(q_i) \cdot \frac{f(q_i,D) \cdot (k_1 + 1)}{f(q_i,D) + k_1 \cdot \left(1 - b + b \cdot \frac{|D|}{avg\_document\_len}\right)} \tag{2.1}$$

where:

- D: Document;
- Q: Query comprised of terms such that $Q = \{q_1, \ldots, q_n\}$;
- q_i: Query term such that $q_i \in Q$;
- $IDF(q_i)$: The inverse document frequency of the query term q_i, usually computed as:

  $$IDF(q_i) = \ln\left(1 + \frac{N - n(q_i) + 0.5}{n(q_i) + 0.5}\right) \tag{2.2}$$

  Where N is the total number of documents in the collection, and $n(q_i)$ is the number of documents containing q_i;
- $f(q_i,D)$: The term frequency of query term q_i in document D;
- $|D|$: Length of document D;
- $avg\_document\_len$: Average document length in the collection;
- k_1 and b: Parameters for fine-tuning.

Other versions built on BM25, include BM25F [81] a modification in which the document several fields (title, main text, etc.) are given different degrees of importance, and BM25+ [52] where one deficiency of the original BM25 is addressed, where long documents are often unfairly scored when compared to shorter documents that do not contain the query terms at all.

2.1.2 Learning to Rank

Many ranking models have been introduced in IR literature, and most contain parameters for fine-tuning, just like k_1 and b in BM25 (2.1.1.1), these parameters need to be optimized in a validation set, whilst not overfitting. Therefore Learning to Rank (LTR) [45] is adopted, being a class of algorithmic techniques that apply supervised Machine Learning

(ML) to solve ranking problems. Traditional ML solves classification or regression, aiming to obtain a class or a numerical value for a single item, the LTR framework differs, the aim is to solve a ranking problem on a list of items, coming up with an optimal order. In Figure 2.1, we find the typical flow of LTR, and how it resembles supervised learning. Given a list of items with judgments, the ground truth is derived for the selected approach, after the split into training and test set is done.

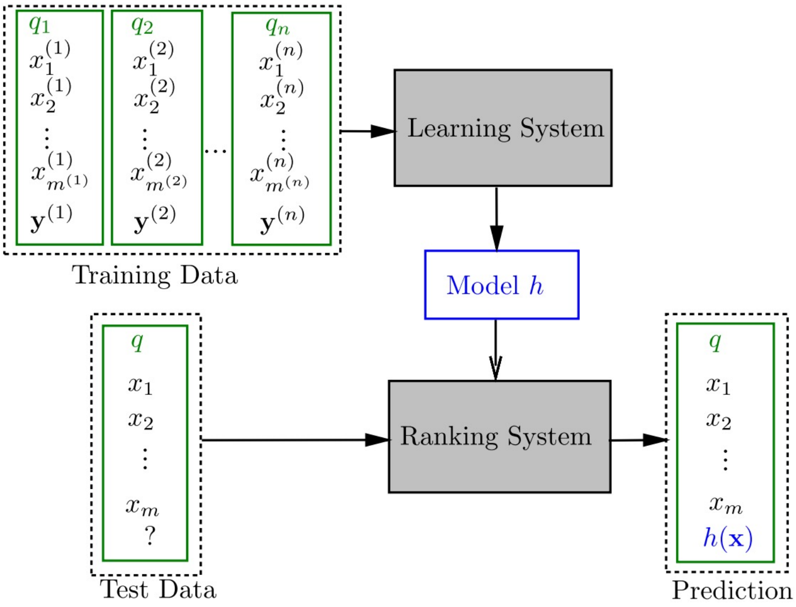

q_1 $x_1^{(1)}$ $x_2^{(1)}$ $x_{m^{(1)}}^{(1)}$ $\mathbf{y}^{(1)}$
q_2 $x_1^{(2)}$ $x_2^{(2)}$ $x_{m^{(2)}}^{(2)}$ $\mathbf{y}^{(2)}$
q_n $x_1^{(n)}$ $x_2^{(n)}$ $x_{m^{(n)}}^{(n)}$ $\mathbf{y}^{(n)}$
Training Data
Learning System
Model h
Ranking System
q x_1 x_2 x_m ?
Test Data
q x_1 x_2 x_m $h(\mathbf{x})$
Prediction

Figure 2.1: LTR Framework [45]

LTR approaches differ in their grounding hypotheses, input and output spaces, and loss functions. LTR can be divided into three approaches:

- **Pointwise Approach.** The pointwise approach assumes a query-document pair and a score (relevance) is attributed, in this way LTR can be approximated as a regression or classification problem. The objective is to learn a function that predicts the relevance of a document to a query. The loss function examines the accurate prediction of the ground truth label for every single document. This approach does not consider the interdependency among documents, the position of the documents in the ranked list is not taken into account in the loss function. Most ML supervised learning algorithms can be applied.

- **Pairwise Approach.** In the pairwise approach the idea is given a pair of documents, to learn which is more relevant. This is approximated as classification problem, by learning a binary classifier, the loss function measures the inconsistency of the prediction to the ground truth. Some examples are, PRM [46], and DirectRanker [33].

- **Listwise Approach.** The listwise approach tries to directly optimize ranking metrics (Section 3.3), averaged over all queries, given a set of documents. Most ranking metrics cannot be used, so approximations or bounds of this measures are used. A few examples include, PiRank [88], SetRank [65], and GSF [1].

2.1.3 Multi-Stage Architectures

Using LTR (Subsection 2.1.2) algorithms at inference time brings high latency due to the computational resources needed (computational complexity), although obtaining higher ranking effectiveness, it is not feasible using such algorithms on a large document collection. The idea behind multi-stage architectures is to split the ranking process into a series of pipeline stages, which typically starts with a retrieval model (Subsection 2.1.1), and each subsequent stage of the pipeline reranking a smaller number of retrieved documents passed from the previous stage. This approach has been used through the years in the IR field [54, 55].

Multi-stage ranking architectures strive to find the balance between complexity and latency, a good way to control this is to simply change the number of documents that are passed to a set stage of the pipeline. In Figure 2.2 we see a simple retrieve-and-rerank architecture that is the most adopted approach in research, a first-stage retrieval is used to generate the list of documents that will next be reranked by the chosen model.

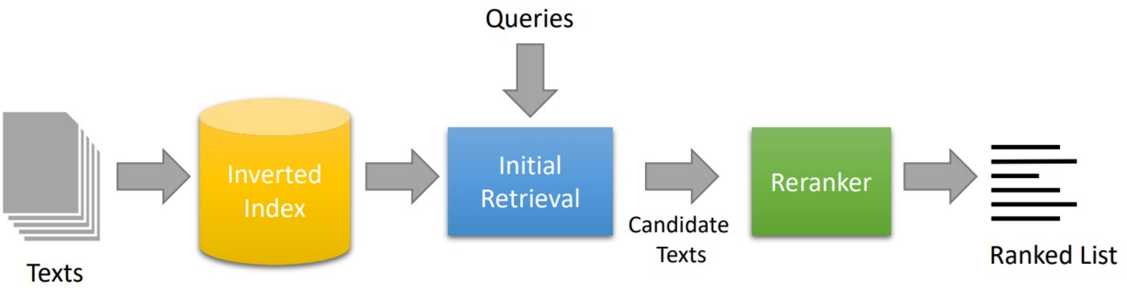


Figure 2.2: Retrieve-and-Rerank Architecture [42]

In [61] a multi-stage ranking architecture (Figure 2.3) is proposed, using BERT based cross-encoder models (Subsection 2.4.1), in this example, BM25 is used as first-stage retrieval, followed with a reranking stage using monoBERT, and then the output is passed to the last stage where duoBERT reranks the output.

2.2 Embeddings

For the computer to process any kind of text, first, we need to represent it in a way the computer can understand, so we represent the text as numbers, which we call embedding. Here we have two options word embedding and sentence embedding. These techniques bring two critical properties, dimensionality reduction, and contextual similarity.

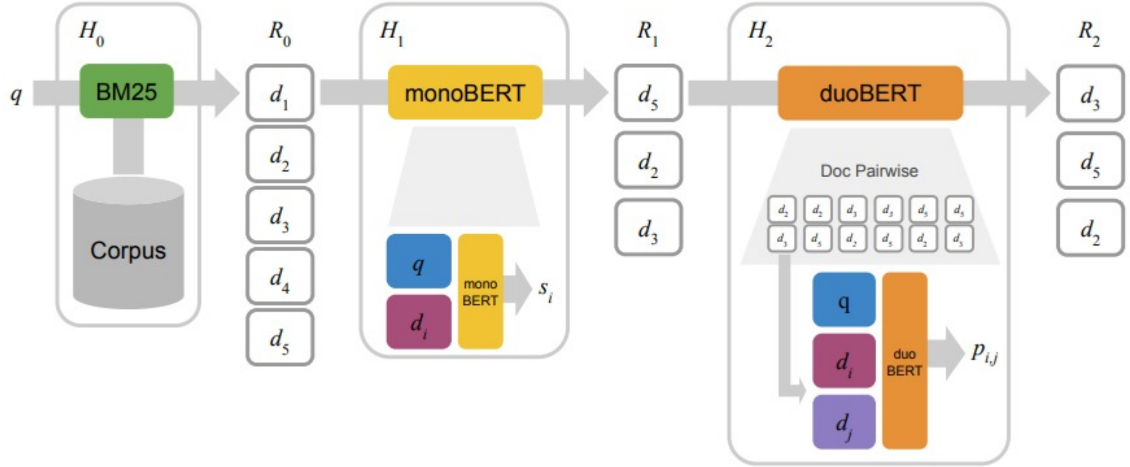


Figure 2.3: BERT based Multi-Stage Architecture [61]

Word embedding is a technique in NLP* used to map individual words into a vector of real numbers in a predefined vector space. Each word is represented by a vector, often with hundreds of dimensions, contrasted to the millions of dimensions required for a sparse word representation technique. [97, 23] To produce the word embeddings, a historical approach is to use Word2Vec [56] and GloVe [67]. Two different models were introduced, CBOW and Skip-Gram, both models are focused on learning about words in their context. The significant benefit of this method is the efficiency in producing high-quality word embeddings, allowing for larger dimensions to be used.

With sentence embedding, we create an embedding for the sentence instead of dealing with each word. This helps the computer understand context, intention, and other nuances as the vector retain semantic information. To produce sentence embeddings, one of the most popular techniques is Doc2Vec [37], an extension of Word2Vec, another option and the most dominant at the moment is Sentence-BERT (SBERT) [72], which makes use of the powerful Transformer.

2.3 The Transformer

Recurrent neural networks are firmly established as state-of-the-art approaches in various Natural Language Processing (NLP) fields, in [11, 87], we see some of the models that pioneered this trend. Taking an in-depth look at such models, we easily find an encoder-decoder architecture in which the computations typically depend on the previous computations, meaning each step generates a state that is needed on the next step, ruling out any possibility of parallelization, making the use of such models impractical.

Attention mechanisms, introduced and refined in [51, 8], quickly became an integral part of various models, significantly increasing the quality of such models. However, using these mechanisms did not resolve the parallelization problem, keeping the same bottleneck.

The Transformer [94] was then introduced in 2017 relying entirely on a self-attention mechanism, allowing for significantly more parallelization and setting itself as state-of-the-art. The Transformer's architecture kept an encoder-decoder structure as is usual in sequence transduction models. The encoder maps an input sequence of symbol representations to a sequence of continuous representations, given that sequence the decoder generates an output sequence of symbols one element at a time, at each time-step the model uses the previously generated symbols as additional input. Overall it uses a stacked self-attention and point-wise, fully connected layers for both the encoder and decoder as shown in figure 2.4. On the original paper, the encoder and decoder are composed of a stack of $N = 6$ identical layers. A more in-depth and illustrated explanation can be found in [90] and [89].

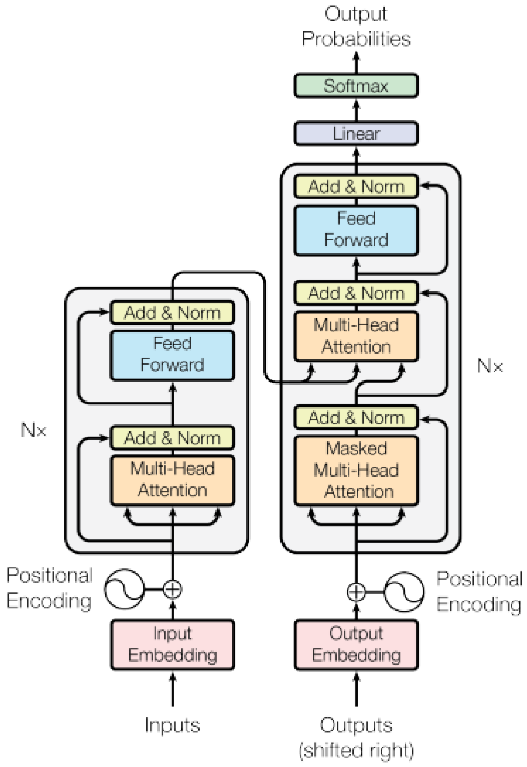


Figure 2.4: The Transformer - model architecture [94]

2.3.1 Bidirectional Encoder Representations from Transformers (BERT)

The Transformer was conceived for sequence transduction tasks (e.g. Language translation), as mentioned its architecture is separated into an encoder-decoder. Focusing on the encoder (Figure 2.5) as this component is the one that gains a notion about the language. By simply stacking the Transformer's encoders we obtain BERT [18]. With BERT we can now approach other problems in NLP and IR, that weren't possible using the encoder-decoder architecture, such as classification tasks.

BERT's architecture is a variable number of stacked encoder layers and self-attention heads. BERT can handle various tasks as it receives as input a single sentence or a pair of sequences (separated with a unique token). To tokenize such sentences WordPiece [99] is used, with a 30000 token vocabulary.

To pre-train BERT two unsupervised tasks are used, Masked Language Model (MLM) and Next Sentence Prediction (NSP). In MLM we randomly mask some percentage of the input tokens and then predict those masked tokens. In NSP we pick two sentences followed by each other or not and the model needs to indicate if the sentence is the next one or not. Pre-training is computationally expensive, but using a pre-trained model we can then further fine-tune and optimize the model for specific tasks, by swapping out the appropriate inputs and outputs expending less resources, and using smaller datasets.

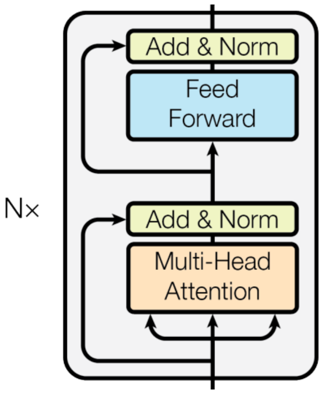


Figure 2.5: The Transformer - encoder [94]

2.3.2 Text-To-Text Transfer Transformer (T5)

The common approach is to pre-train a model on a data-rich task to then further fine-tune it on downstream tasks, which gave rise to various pre-training objectives (token masking, token deletion, etc.), unlabeled data sets, benchmarks, and fine-tuning methods. In [70] propose a unified approach, with Text-to-Text Transfer Transformer (T5). The idea is to treat every text-processing problem as a text-to-text problem, this allows us to apply the same procedure to every task.

T5 model is essentially the same as the Transformer mentioned in section 2.3, except for some minor changes to its inner architecture.

To make the unified approach text-to-text work for all downstream tasks, a task-specific text prefix is needed before the original input to specify which task the model should perform, as shown in figure 2.6.

The model was pre-trained over the Colossal Clean Crawled Corpus (C4) dataset (introduced by the authors), and the unsupervised tasks techniques used were MLM and word dropout. To tokenize the used vocabulary SentencePiece [34] was chosen, with 32000 wordpieces.

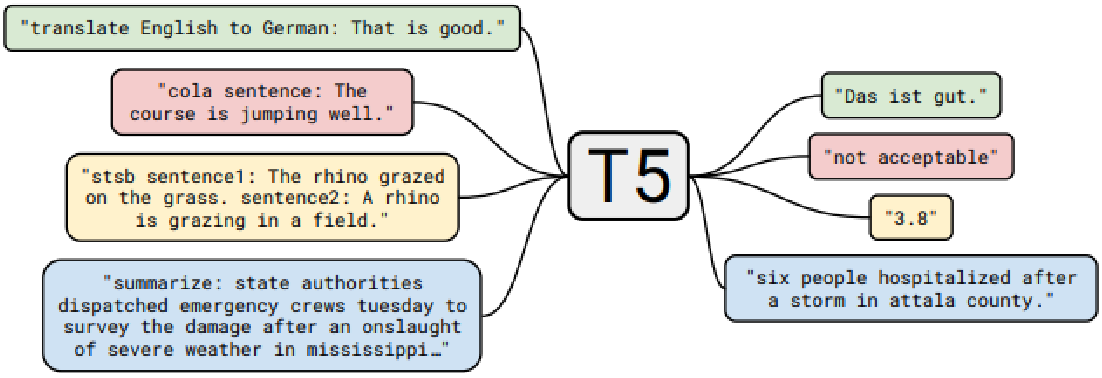

"translate English to German: That is good."
"cola sentence: The course is jumping well."
"stsb sentence1: The rhino grazed on the grass. sentence2: A rhino is grazing in a field."
"summarize: state authorities dispatched emergency crews tuesday to survey the damage after an onslaught of severe weather in mississippi…"
T5
"Das ist gut."
"not acceptable"
"3.8"
"six people hospitalized after a storm in attala county."

Figure 2.6: T5 - diagram of the text-to-text framework [70]

2.4 Ranking with Transformer

Currently, the state-of-the-art architecture models for ranking and biomedical domain-specific NLP tasks, are based on BERT that by fine-tuning on the task, substantial improvements are obtained, respectively the tasks present in the MS MARCO Passage Ranking Leaderboard [16], and the BLURB Leaderboard [24].

The proposed approaches to using Transformer based models are:

- **Cross-Encoder** - performs full self-attention over a given input that is then passed to a classifier, obtaining the final result.
- **Bi-Encoder** - performs self-attention over the query and the document separately, producing sentence embeddings that are then compared, obtaining a similarity score.
- **Encoderd-Decoders** - sequence-to-sequence models that perform full self-attention on both the encoder and the decoder, outputting a sequence that is used for classification.

2.4.1 Cross-Encoders

As said above, cross-encoders perform full cross self-attention over the input, these models usually get higher ranking measures, when compared to bi-encoders, however, the computational complexity of this method is substantial, and as a result, their use as a reranking method comes with high latency.

To use BERT, the first token of every sequence is always a special classification token ([CLS]). The logits corresponding to this token are used as the aggregate sequence representation for the classification tasks (Figure 2.7). The sentences are packed together into a single sequence, these are differentiated in two ways, separated with a special token ([SEP]), and the segment embedding, where a learned embedding is added to every token indicating to which sentence belongs. One of the reasons for the success of BERT in ranking, is the learned knowledge in the pretraining phase, more precisely, the NSP task has been shown to be closely related to representation learning objectives [27, 49], and in BERT original paper [18] it is shown that the NSP task is very beneficial to both Question Answering (QA) and Natural Language Inference (NLI), downstream tasks based on understanding the relationship between two sentences.

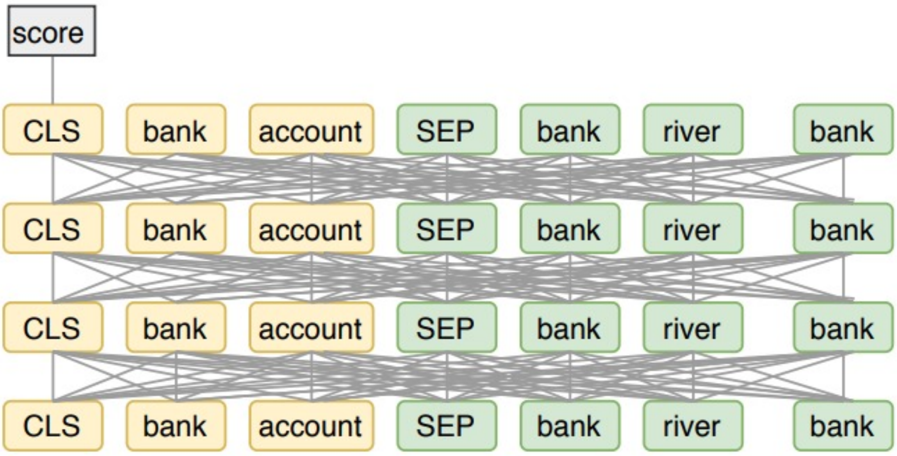


Figure 2.7: Example of BERT Cross-Encoder [22]

In [61], two approaches to using BERT Cross-Encoder for ranking are introduced, monoBERT a pointwise approach, and duoBERT a pairwise approach, both models fine-tuned on the document ranking task.

2.4.1.1 monoBERT

monoBERT is a BERT model used as a binary relevance classifier, and can be denoted as:

$$P(Relevant = 1|q,d) \tag{2.3}$$

where q and d are the query and the document, respectively, the inputs to the model are feed with the following template:

[CLS] q [SEP] d [SEP]

Due to BERT having a maximum length of 512 tokens, in [61] the query is truncated to have at most 64 tokens and the document is truncated in such a way that the input has a maximum length of 512 tokens. The contextual representation of the [CLS] token, is used as input to a single layer fully-connected neural network to obtain the probability p of document d being relevant to query q.

The model was trained for re-ranking using cross-entropy loss:

$$L = -\sum_{j \in J_{pos}} \log(p_j) - \sum_{j \in J_{neg}} \log(1 - p_j), \tag{2.4}$$

where J_{pos} s is the set of indexes of the relevant candidates, and J_{neg} is the set of indexes of the non-relevant candidates. To note that optimizing cross-entropy for classification may not improve ranking metrics.

2.4.1.2 duoBERT

The duoBERT, a pairwise approach, compares pairs of documents in the following fashion:

$$P(d_i \succ d_j | q, d_i, d_j) \tag{2.5}$$

where $d_i \succ d_j$ denotes that d_i is more relevant than d_j, the following input template is used:

$$\text{[CLS] } q \text{ [SEP] } d_i \text{ [SEP] } d_j \text{ [SEP]}$$

In [61], a similar truncation strategy is employed, where the maximum length of q, d_i, and d_j is 62, 223, and 223 tokens, respectively, so that the input has a maximum length of 512 tokens. Similar to monoBERT, the final representation of the [CLS] token is used as input to a fully-connected layer to obtain the probability $p_{i,j}$. The following loss was used to train the model:

$$L = -\sum_{i \in J_{pos}, j \in J_{neg}} \log(p_{i,j}) - \sum_{i \in J_{neg}, j \in J_{pos}} \log(1 - p_{i,j})$$

At inference time the pairwise scores $p_{i,j}$ are aggregated so that each document receives a single score, five aggregation methods were investigated (SUM, BINARY, MIN, MAX, and SAMPLE), SUM and BINARY methods seem to be the better performing. duoBERT shows better performance than monoBERT, but with monoBERT the latency increases linearly as more documents are considered, and the input length is usually smaller, duoBERT brings huge latency and complexity.

2.4.2 Bi-Encoders

Bi-encoders, first introduced in [26], perform self-attention over the input separately, producing meaningful sentence embeddings, that can then be compared using similarity measures such as cosine-similarity. In [72] SBERT is introduced, to show how reduced the complexity of bi-encoders is when compared to cross-encoders, the authors show that the task of sentence similarity with 10000 sentences, on the original BERT takes arround 65 hours, while with SBERT takes 5 seconds. SBERT modifies the original architecture to be used as a siamese network (Figure 2.8), that outputs fixed sized vectors for the input of each sentence that are then similarity measured, this reduces the computation needed exponentially.

SBERT adds a pooling layer to the output of BERT to derive a fixed-size sentence embedding. Given two sentences, we can now compute the similarity between the produced embeddings, and rank using this score. With bi-encoders, we can also index the

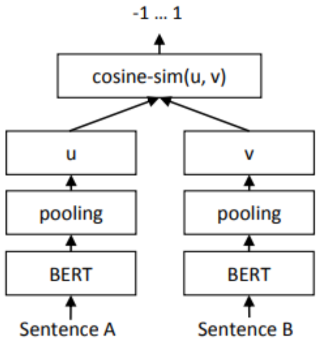


Figure 2.8: SBERT architecture to compute cosine similarity scores. [72]

collection, producing the embeddings and at inference time, only the embedding of the queried sentence needs to be computed (dense passage retrieval). But SBERT focus is on textual similarity and not ranking, and the use of bi-Encoders for ranking brings some negative aspects, mostly in the fine-tuning phase.

2.4.3 Encoders-Decoders

The most commonly adopted document ranking approach is the use of encoder-only transformer architectures such as BERT. In [59] a novel adaptation of T5 for document ranking, is introduced. Exploiting the model's latent knowledge learned through pre-training just like BERT ranking models. The authors adapted the sequence-to-sequence model, using the following input template:

$$\text{Query: } q \text{ Document: } d \text{ Relevant:}$$

where q and d are the query and the document, respectively. The model is fine-tuned to produce the words "true"or "false"depending on whether the document is relevant or not to the query. To produce a relevance probability for the query-document pair, at inference time a softmax is applied on the logits of "target words"tokens ("true"and "false"). The retrieved list is then reranked according to these probabilities assigned to the "true"token.

The model was fine-tuned on the MS MARCO passage ranking dataset [58] and tested on it, and on the Robust04 test collection [96] and the results show that the approach shines in data-poor regimes and significantly outperforms BERT with limited training examples. This model will be referred to as monoT5, due to the similarity to monoBERT.

Another use of T5 in this field is as a passage expansion technique, docTTTTTquery [60], this model when given a passage as input, generates passages related to the input. This model can be used as a document expansion technique or query expansion technique, and further fine-tuned on a domain-specific dataset.

```
<topic number="1">
<disease>melanoma</disease>
<gene>BRAF (V600E)</gene>
<treatment>Dabrafenib</treatment>
</topic>
```

Figure 2.9: Example Topic from PM Track

```
<topic number="1">
A 2-year-old boy is brought to the emergency department by his parents for 5 days of high fever and irritability. The physical exam reveals conjunctivitis, strawberry tongue, inflammation of the hands and feet, desquamation of the skin of the fingers and toes, and cervical lymphadenopathy with the smallest node at 1.5 cm. The abdominal exam demonstrates tenderness and enlarged liver. Laboratory tests report elevated alanine aminotransferase, white blood cell count of 17,580/mm, albumin 2.1 g/dL, C-reactive protein 4.5 mg, erythrocyte sedimentation rate 60 mm/h, mild normochromic, normocytic anemia, and leukocytes in urine of 20/mL with no bacteria identified. The echocardiogram shows moderate dilation of the coronary arteries with possible coronary artery aneurysm.
</topic>
```

Figure 2.10: Example Topic from CT Track

2.5 TREC 2021 Clinical Trials Track

TREC has organized various biomedical track series, and the most relevant to this work is the 2021 CT Track [79], where given a patient case description we are tasked to retrieve relevant clinical trials, similar to one of the tasks of the PM Track (Figure 2.9 shows the structured topic used), but the tasks differentiate in several key differences, one is that the CT Track is not specific to cancer, another difference is that the topics are long unstructured text as shown in Figure 2.10, and for last is that the focus now is on retrieving eligible clinical trials and not just relevant ones.

In the 2020 PM Track [78] all of the best scoring runs used a multi-stage architecture, with the first stage using a retrieval such as BM25 and DRF models, and a Transformer based models (BioBERT [38] and T5 [70]) to re-rank the top retrieved documents [28, 69, 82].

Following the success of the multi-stage architectures used before, in the 2021 CT Track, most teams followed that, but still, we find some interesting methods that achieved better results. In [43] the best scoring and a well implemented method, that is used as reference in this work. It is a multi-stage reranking architecture, with the first-stage retrieval based on BM25 + RM3 and Reciprocal Rank Fusion (RRF), and the rerank stage using a T5 based model. The submissions delivered represent the various "phases"of the pipeline, for the first-stage retrieval, due to the sentence-length of the topics, and most retrieval methods being built having shorter queries in mind, the authors generated multiple single sentence long queries for a given topic, using docTTTTTquery [60], a neural passage expansion technique, the model is the T5-3B fine-tuned on the MS MARCO V1 passage ranking training set, 40 queries were sampled for each topic. In the table 2.1, we observe how impactful this query expansion technique is (desc_rm3 vs. f_d2q_rm3), the run desc_rm3 is the BM25 + RM3 (with Pyserini's default parameters) retrieval technique on an indexed corpus using the patient descriptions as queries, and f_d2q_rm3 is the RRF of 41 runs (desc_rm3 run plus the runs on the 40 generated queries).

The following runs (f_0_mt5, f_t_mt5, and f_t_mt5_2) are somewhat extensions of each other, using the monoT5-3B [59] as the base model for reranking, fine-tuned on the MS MARCO V1 passage ranking test collection, this model was then further fine-tuned on the Med-MARCO [53], which is a subset of the MS MARCO V1 passage ranking, that only contains queries with medical terms in the test collection, giving origin to a model called monoT5$_{MED}$, that was first described in [68]. The results of this zero shot approach are in the table under the run name f_0_mt5 (Table 2.1), we see a very small change in some metrics and an improvement in others, revealing that the zero shot T5 based rerank is relatively successful.

The monoT5$_{MED}$ was then fine-tuned on an in-domain task, the CSIRO clinical trial test collection [32], creating what the authors call the monoT5$_{CT}$. The input template of the reranker is the following:

Query: q Document: d Relevant:

Two multi-sentence long fields are used "eligibility" and "description", as well as two smaller fields "title" and "condition", due to the length of the multi-sentence long fields during inference a sliding window segmentation was used with a length of 6 and a stride of 3. The eligibility and description fields are used separately, as shown bellow on the two input templates for the document:

Document: title: d$_{title}$ condition: d$_{condition}$ eligibility: d$_{eligibility}$

Document: title: d$_{title}$ condition: d$_{condition}$ description: d$_{description}$

After a MaxP approach is used to select one score for the topic and clinical trial pair. The f_t_mt5, and f_t_mt5_2 runs use the model monoT5$_{CT}$, the first uses only the eligibility field, while the second uses both the eligibility field and the description field, to reveal the two highest scoring segments per trial, these are combined together to give a single larger segment for that trial, with these composed segments, the monoT5$_{CT}$ is used again for rerank.

With the use of monoT5$_{CT}$ a model trained on an in-domain task, we see a big improvements in all metrics (Table 2.1), and with the f_t_mt5_2 run, we identify how using different fields of the clinical trial improves the model.

In [62], the RM3Filtered run, BM25 + RM3 retrieval model shows better results than any simple retrieval model on the table 2.1, reveling that optimizing the retrieval step hyperparameters improves all metrics and also that filtering the clinical trials prior to retrieval is a viable approach. The results can be compared between the runs, RM3Filtered vs. desc_rm3 (does not filter the clinical trials). Other submitted runs explored summarizing the queries with transformer models (BART [39] and T5), but showed worst results.

An original approach is found in [31], that instead of using IR as the core engine, they used a simple approach, based on document similarity. The pipeline can be divided into three stages, first, summarization of the patient's cases and clinical trials, secondly, producing semantically meaningful sentence embeddings, and finally, similarity computation. For the summarization NER extraction was used, specifically, the authors extracted the following medical entity types: Problem, Treatment, and Test. The model used was the Bio+ClinicalBERT [4], which was further fine-tuned on the i2b2/VA 2010 corpus [92], that the authors converted to the CoNLL format [83]. The model was then applied to the topics and clinical trials obtaining a distilled corpus. The models used to produce sentence embeddings were chosen with the idea to evaluate multilingual and scientific domain models, LaBSE which supports 109 languages [21], and allenai-specter [12]. The runs LaBSE and specter, use the embeddings of their respective model (LaBSE and allenai-specter), text similarity between the clinical trials and the patients cases is computed using cosine similarity between the embeddings.

The LaBSE rerank and spect_rerank runs, built on the previously mentioned runs, is a rerank of the runs with a binary bag-of-words representation method. The spec_rrk_fqv reranks the run specter, using Term Frequency. The reranks are done by using the weighted average calculated between embedding cosine scores and the select method.

Looking into the table 2.1 under the team uni_pais_vasco we see all their runs. Clearly using only the embedding method alone underperforms when compared to the other methods seen. We can also see that the pipeline using a scientific domain model slightly outperforms the multilingual model. These results may be a result of how the NER system trained in the scientific domain extracts mostly medical entities.

2.6 Self-Attention and Sequence Length

Transformer based models have shown to be very successful for many NLP tasks, such as document ranking as we have seen, achieving state-of-the-art results. This is partly due to the self-attention mechanism, but this mechanism brings a limitation, having a quadratic time and memory complexity, which scales with input sequence length. To address this limitation, two alternative strategies have been proposed, discussed below.

2.6.1 Alternative Attention Mechanism

One approach researchers have proposed is of reducing attention mechanism computational complexity. To address it, Longformer [10] was introduced, proposing sliding window attention pattern that scales linearly with the input sequence, and BigBird [104] with a sparse attention mechanism, that also scales linearly with sequence length.

Table 2.1: Runs Scores of Each Team Mentioned on the TREC CT 2021 [79]

Run	**P@10**	**NDCG@10**	**MRR**	**RPrec**
h2oloo				
desc_rm3	0.2040	0.3539	0.3659	0.1270
f_d2q_rm3	0.2760	0.4726	0.4304	0.1740
f_0_mt5	0.2987	0.4715	0.4830	0.1742
f_t_mt5	0.5493	0.6792	0.7161	**0.2639**
f_t_mt5_2 (Best Submission)	**0.5933**	**0.7118**	**0.8162**	0.2628
ims_unipd				
RM3Filtered	**0.3360**	**0.5149**	0.4936	**0.2078**
uni_pais_vasco				
LaBSE	0.1347	0.2551	0.2687	0.0474
specter	0.1480	0.2555	0.2731	0.0681
LaBSE rerank	0.1413	0.2900	0.3353	0.0537
spect_rerank	**0.2093**	**0.3614**	**0.3680**	**0.0838**
spec_rrk_fqv	0.1547	0.2694	0.2760	0.0620
All TREC Submissions				
Median	0.1613	0.3040	0.2942	-
Best	0.7480	0.8491	1.0000	-

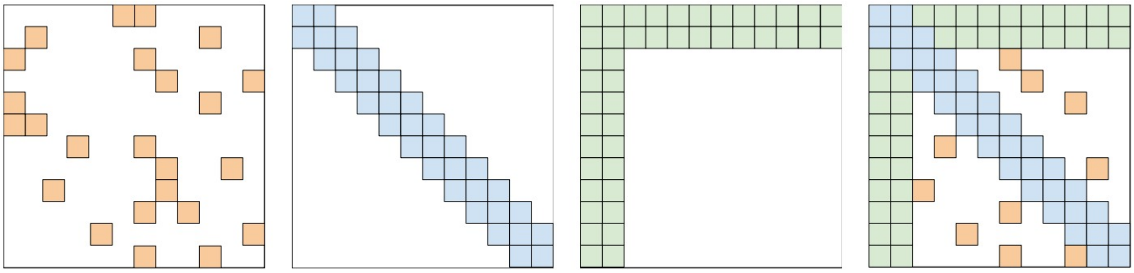


Figure 2.11: Comparing Alternative Attention Patterns, Random Attention, Window Attention, Global Attention, and BigBird's Attention, respectively. [104]

In Figure 2.11 we can visualize some of the proposed attention patterns. For random attention, each token is randomly attended to other tokens. In a sliding window a fixed-size window attention is employed for each token, giving importance to local context, given a fixed window size w, each token attends to $\frac{1}{2}w$ tokens on each side, obtaining a $O(n \times w)$, scaling linearly with the input sequence n. For classification tasks where the model aggregates the representation into logits of the [CLS] token, such as ranking, the model needs to compare the query with the document, global attention is done on a few selected tokens, and those selected tokens attend to all other tokens, all those tokens attend to those tokens, on the example of classification tasks, we use global attention on the [CLS] token, this method also scales linearly to the input length. BigBird's attention mechanism combines all of the above, the authors call it, generalized attention mechanism.

2.6.2 Passage-Level Relevance

To solve the input sequence length problem, we can resolve to split the document into passages, various approaches are available, like fixed-size spans or dividing into natural sentences, but how wide should these passages be? And should they overlap? There are lots of alternatives. [42]

But for training, it is unclear what to do. The issue raises from the relevance judgments being provided at document level, although the document is relevant, only some passages might be relevant to a query. And at inference time we still need to somehow aggregate the scores produced.

One solution, Birch [3], where they avoid the training problem by using data that does not have length issues, and at inference time aggregating only the top-3 scoring passages.

2.7 Information Extraction

In this small section we explore Information Extraction (IE) work in the biomedical domain, namely NER and Relation Extraction (RE) tasks. NER is the recognition of known entity names (Age, Gender, Treatment, Measurement, etc), and RE is the identification of relations between these entities.

In [64], the introduced model, aims at extracting key components of the reports of the clinical trials, namely: Intervention, Comparator, and Outcome. Referred as ICO triplets, to identify these entities a BiLSTM-CRF sequence tagging model, trained on the EBM-NLP data from [63], was used (Extract phase). The tokens used by the NER model are encoded, using SciBERT [9], the predicted mentions then are assigned to the provided entity that has the highest cosine similarity, building the ICO triplet (Link phase).

Although deep learning-based methods, such as attention-based BiLSTMs, like the model described above, have been successful the introduction of BERT demonstrated state-of-the-art performance in various NLP tasks, CT-BERT, introduced in [47], looks to explore the use of BERT in IE, by fine-tuning a set of pre-trained models (BERT, BioBERT, ClinicalBERT, and BlueBERT), on annotated ClinicalTrials.gov clinical trials [91]. The NER identifies 15 types of clinical entities, covering common types as well as speciality types and value ranges. The RE module is used to associate attribute entities (e.g. measurements) with their base entities. The BioBERT based model achieved new state-of-the-art performance.

2.8 Biomedical Domain-Specific Language Models

Although BERT is very successful in NLP tasks, its focus is on general domain corpora, in specialized domains such as biomedicine, further domain-specific pre-training can be benefit, such as Target Corpus Pretraining (TCP), but in [24] the authors show that pretraining language models from scratch is preferable.

Biomedical Term	Category	BERT	SciBERT	PubMedBERT (Ours)
diabetes	disease	✓	✓	✓
leukemia	disease	✓	✓	✓
lithium	drug	✓	✓	✓
insulin	drug	✓	✓	✓
DNA	gene	✓	✓	✓
promoter	gene	✓	✓	✓
hypertension	disease	hyper-tension	✓	✓
nephropathy	disease	ne-ph-rop-athy	✓	✓
lymphoma	disease	l-ym-ph-oma	✓	✓
lidocaine	drug	lid-oca-ine]	✓	✓
oropharyngeal	organ	oro-pha-ryn-ge-al	or-opharyngeal	✓
cardiomyocyte	cell	card-iom-yo-cy-te	cardiomy-ocyte	✓
chloramphenicol	drug	ch-lor-amp-hen-ico-l	chlor-amp-hen-icol	✓
RecA	gene	Rec-A	Rec-A	✓
acetyltransferase	gene	ace-ty-lt-ran-sf-eras-e	acetyl-transferase	✓
clonidine	drug	cl-oni-dine	clon-idine	✓
naloxone	drug	na-lo-xon-e	nal-oxo-ne	✓

Figure 2.12: Comparison of how the models, with different pre-training, break biomedical terms into work pieces, the check mark indicates that the word is not broken down. [24]

In Figure 2.12, we have the models: BERT, pre-trained on general domain corpora; SciBERT, starting from BERT further scientific domain pre-train is made; PubMedBERT, pre-trained on the biomedical domain. Here we see how the differences in pre-training affects how the model breaks or not biomedical terms into word pieces, which may affect the model performance in downstream biomedical tasks. Which is supported by the performance of these models in the BLURB Leaderboard[1] [24], where PubMedBERT outperforms both models, and SciBERT outperforms BERT.

Some interesting biomedical domain models are described next:

- BioLinkBERT [102]: Based on LinkBERT, a model that outperforms BERT on various downstream tasks due to its innovative pre-training task (linking documents). BioLinkBERT is pre-trained on PubMed documents, and achieves new state-of-the-art results in various downstream tasks in the NLP biomedical domain;

- PubMedBERT [24]: BERT inspired model wiht the same architecture, but pre-trained from scratch on PubMed abstracts;

- BioBERT [38]: using BERT as a starting point, further pretraining is done on PubMed abstracts (PubMed) and PubMed Central (PMC) full-text articles;

- ClinicalBERT [25]: starting with BERT, the model is further pretrained on clinical notes, the Medical Information Mart for Intensive Care III (MIMIC-III) [29] dataset;

[1] https://microsoft.github.io/BLURB/leaderboard.html

- BEHRT [41]: BERT based model, fine-tuned for disease prediction, on the Clinical Practice Research Datalink (CPRD) [20], which contains longitudinal EHR patients, inputting the past patient information extracted from the respective EHRs, plus the new age embedding;

- CT-BERT [47]: its a model fine-tunned for NER and RE tasks on [91], dataset for information extraction of clinical trials eligibility criteria. A set of pre-trained models were tested, BERT, BioBERT, ClinicalBERT, and BlueBERT [66]. The BioBERT based model achieved the best results.

3

Experimental Setup

In this chapter, we describe and analyze the data used in this work and its characteristics, the protocol that each step in the pipeline will take to ensure cohesion between distinct runs, and finally how each run will be evaluated and the meaning of the chosen metrics.

3.1 Datasets

3.1.1 TREC 2021 Clinical Trials Track

The TREC 2021 Clinical Trials Track test collection, which will be mentioned as TREC-CT from now on, was made available to evaluate the runs of participants of the track. This dataset is nowadays used to test retrieval methods in the CT domain.

3.1.1.1 Documents

The clinical trials available at ClinicalTrials.gov were used as the documents in the creation of this dataset. An April 27, 2021 snapshot of the collection was taken and was made available for download[1] by the 2021 Clinical Trials Track. Each Clinical Trial is formatted using the ClinicalTrials.gov XML schema.

Out of the 375580 clinical trials, 841 are redacted, due to not being approved or cleared by the U.S. FDA. Leaving 374739 clinical trials to be used as the corpus.

Taking a look into the clinical trial structure, it includes various fields describing the trial but only some fields were found relevant, most of which are unstructured text. We can divide the fields into "describing"fields and "filtering"fields.

The "describing"fields are the following (Figure 3.1):

- **brief_title**: A small version of the title containing only essential keywords;
- **official_title**: The title with all keywords and expanded biomedical terms;
- **brief_summary**: A summary of the trial, and a brief description of the subject matter;

[1] http://www.trec-cds.org/2021.html#documents

Table 3.1: Whitespace Tokenization for each Field (Mean Length, and Standard Deviation) and Presence of the Fields in the Clinical Trials.

Field	Mean	Standard Deviation	Presence
brief_title	11.7	5.1	100%
official_title	18.3	8.4	97%
brief_summary	92.2	88.5	100%
detailed_description	286.5	336.6	67%
eligibility study_pop	23.9	23.9	21%
eligibility criteria	207.2	238.6	100%

- **detailed_description**: Detailed description of the study.

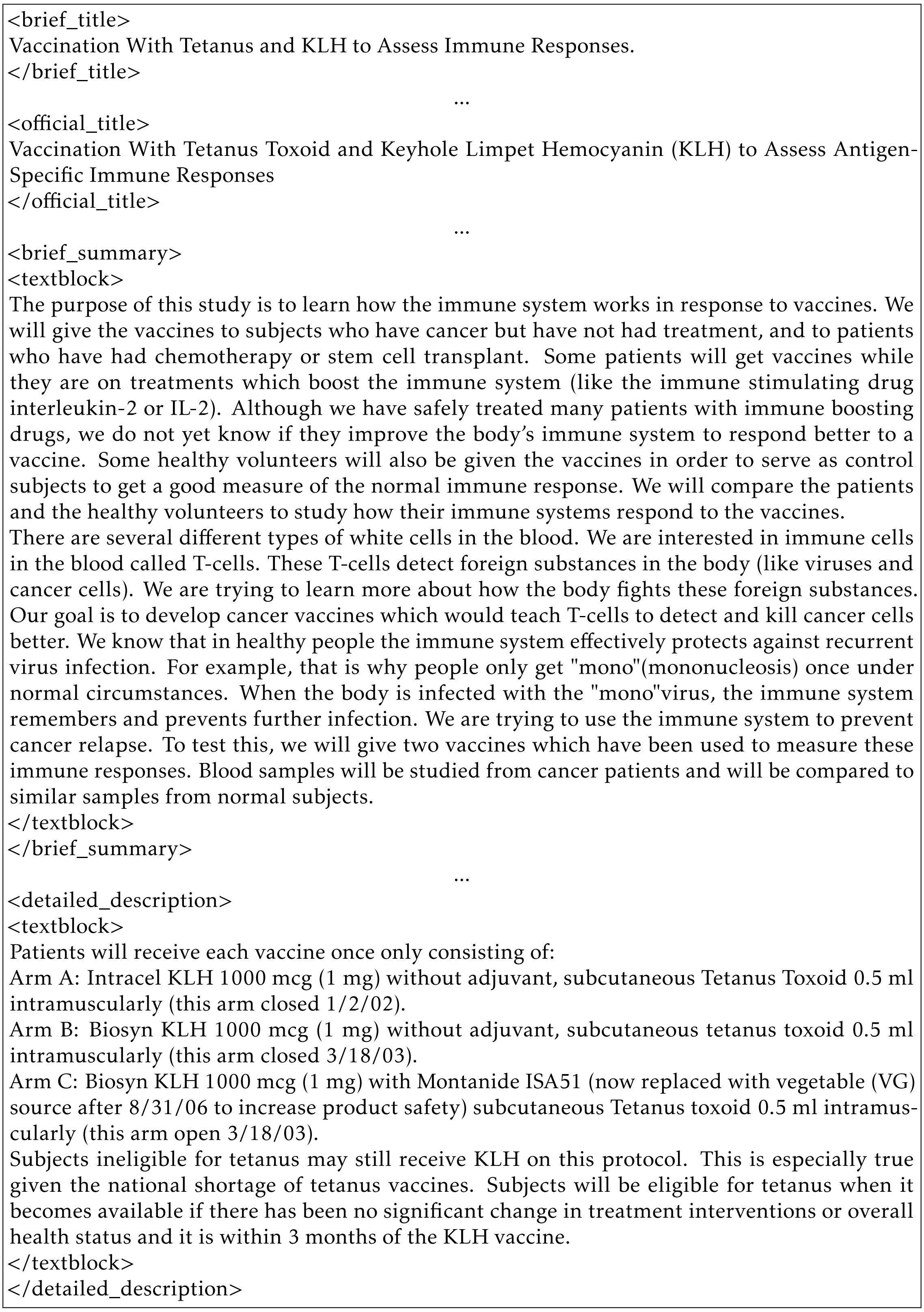
The "filtering"fields (Figure 3.2 and 3.3):

- **study_pop**: This field indicates specific characteristics the patients must have, and presents some redundancy with the next field;
- **criteria**: Here we find both inclusion and exclusion criteria of a patient for eligibility of the clinical trial, sometimes there is this semi-structure text, as shown in the figure, with the headers "Inclusion Criteria:"and "Exclusion Criteria:", and sometimes it is unstructured text;
- **condition**: One or multiple keywords of the conditions the patient needs to have;
- **gender**: In the case the trial targets a gender, its indicated here;
- **minimum_age** and **maximum_age**: The age gap in which the patient needs to fall;
- **healthy_volunteers**: Indicates if the clinical trial accepts healthy patients or not.

For the free-text fields, statistics for the length of each field were calculated. Using whitespace tokenization the mean length and the standard deviation were computed. In table 3.1 we can see that for the smaller fields there is some consistency in their length and that the use of some fields might be impractical due to not being present in most trials.

3.1.1.2 Topics

The collection has 75 topics, designed to represent a patient admission note, all topics are lengthy (5 to 10 sentences) as shown in Figure 3.4. The topics are synthetic patient cases created by individuals with medical training, with a further breakdown of the full data we notice that the topics can be grouped into two clusters: The first one, and the biggest group, has the information as unstructured text resembling traditional medical case descriptions (as shown in Figure 3.4); The second, has the information in the form of a list, semi-structured text.



```
<brief_title>
Vaccination With Tetanus and KLH to Assess Immune Responses.
</brief_title>
...
<official_title>
Vaccination With Tetanus Toxoid and Keyhole Limpet Hemocyanin (KLH) to Assess Antigen-Specific Immune Responses
</official_title>
...
<brief_summary>
<textblock>
The purpose of this study is to learn how the immune system works in response to vaccines. We will give the vaccines to subjects who have cancer but have not had treatment, and to patients who have had chemotherapy or stem cell transplant. Some patients will get vaccines while they are on treatments which boost the immune system (like the immune stimulating drug interleukin-2 or IL-2). Although we have safely treated many patients with immune boosting drugs, we do not yet know if they improve the body’s immune system to respond better to a vaccine. Some healthy volunteers will also be given the vaccines in order to serve as control subjects to get a good measure of the normal immune response. We will compare the patients and the healthy volunteers to study how their immune systems respond to the vaccines.
There are several different types of white cells in the blood. We are interested in immune cells in the blood called T-cells. These T-cells detect foreign substances in the body (like viruses and cancer cells). We are trying to learn more about how the body fights these foreign substances. Our goal is to develop cancer vaccines which would teach T-cells to detect and kill cancer cells better. We know that in healthy people the immune system effectively protects against recurrent virus infection. For example, that is why people only get "mono"(mononucleosis) once under normal circumstances. When the body is infected with the "mono"virus, the immune system remembers and prevents further infection. We are trying to use the immune system to prevent cancer relapse. To test this, we will give two vaccines which have been used to measure these immune responses. Blood samples will be studied from cancer patients and will be compared to similar samples from normal subjects.
</textblock>
</brief_summary>
...
<detailed_description>
<textblock>
Patients will receive each vaccine once only consisting of:
Arm A: Intracel KLH 1000 mcg (1 mg) without adjuvant, subcutaneous Tetanus Toxoid 0.5 ml intramuscularly (this arm closed 1/2/02).
Arm B: Biosyn KLH 1000 mcg (1 mg) without adjuvant, subcutaneous tetanus toxoid 0.5 ml intramuscularly (this arm closed 3/18/03).
Arm C: Biosyn KLH 1000 mcg (1 mg) with Montanide ISA51 (now replaced with vegetable (VG) source after 8/31/06 to increase product safety) subcutaneous Tetanus toxoid 0.5 ml intramuscularly (this arm open 3/18/03).
Subjects ineligible for tetanus may still receive KLH on this protocol. This is especially true given the national shortage of tetanus vaccines. Subjects will be eligible for tetanus when it becomes available if there has been no significant change in treatment interventions or overall health status and it is within 3 months of the KLH vaccine.
</textblock>
</detailed_description>
```

Figure 3.1: Example from a Clinical Trial (NCT00000105)

```
<eligibility>
<study_pop>
<textblock>
- Normal volunteers
- Patients with Cancer (breast, melanoma, hematologic)
- Transplant patients (umbilical cord blood transplant, autologous transplant)
- Patients receiving other cancer vaccines
</textblock>
</study_pop>
                                    ...
<criteria>
<textblock>
 Inclusion Criteria:
- Patients must have a diagnosis of cancer of any histologic type.
- Patients must have a Karnofsky performance status great or equal to 70%.
- Patients must have an expected survival for at least four months.
- Normal healthy volunteers to serve as control for this study.
- All patients must sign informed consent approved by the Committee on the Use of Human Subjects at the University of Minnesota
 Exclusion Criteria:
- Pregnant or lactating women. Females of child-bearing potential will be asked to take a pregnancy test before receiving vaccines.
- Serious intercurrent medical illnesses which would interfere with the ability of the patient to carry out the follow-up monitoring program.
- Immunization should not be administered during the course of any febrile illness or acute infection.
- Hypersensitivity to any component of the vaccine, including Thimerosal, a mercury derivative.
- The occurrence of any type of neurologic symptoms to tetanus vaccine in th past.
- Patients with a history of seafood allergy are excluded from receiving KLH.
- Subjects who have had tetanus toxoid within the last 7 years are not eligible for tetanus vaccine component of this protocol.
</textblock>
</criteria>
```

Figure 3.2: Example from a Clinical Trial (NCT00000105)

```
<condition>Cancer</condition>
<gender>All</gender>
<minimum_age>18 Years</minimum_age>
<maximum_age>N/A</maximum_age>
<healthy_volunteers>Accepts Healthy Volunteers</healthy_volunteers>
```

Figure 3.3: Example from a Clinical Trial (NCT00000105)

```
<topic number=-1">
A 2-year-old boy is brought to the emergency department by his parents for 5 days of high fever and irritability. The physical exam reveals conjunctivitis, strawberry tongue, inflammation of the hands and feet, desquamation of the skin of the fingers and toes, and cervical lymphadenopathy with the smallest node at 1.5 cm. The abdominal exam demonstrates tenderness and enlarged liver. Laboratory tests report elevated alanine aminotransferase, white blood cell count of 17,580/mm, albumin 2.1 g/dL, C-reactive protein 4.5 mg, erythrocyte sedimentation rate 60 mm/h, mild normochromic, normocytic anemia, and leukocytes in urine of 20/mL with no bacteria identified. The echocardiogram shows moderate dilation of the coronary arteries with possible coronary artery aneurysm.
</topic>
```

Figure 3.4: Example Topic from TREC-CT

3.1.1.3 Relevance Judgments

All 75 available topics were matched with some clinical trials, ending up with a total of 35832 query relevance. The relevance judgments are assessed on a 3-point scale:

- **Not Relevant**. The clinical trial is not in any way relevant to the patient;
- **Excluded**. The patient has the condition targeted by the trial but does not meet the eligibility criteria;
- **Eligible**. The clinical trial is relevant for the patient and the eligibility criteria are met.

This dataset is relatively balanced in terms of class frequency for each topic.

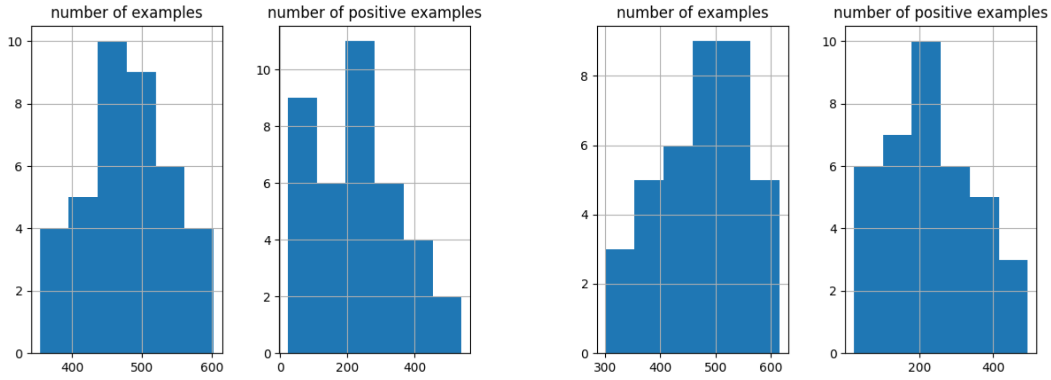
3.1.1.4 Splitting the Data

We decided to split this dataset into a training and evaluation split, although runs done on these splits won't be able to be compared to runs done on the full data, we find that there would be some interest in exploring the use of the dataset in this manner, and fine-tune models on the training split.

To make the split, as each topic had a different number of relevances (with different class frequencies), a detailed look into the data was made, so that the number of examples in total was about the same for each split, whilst maintaining a similar class frequency. In Figure 3.5, we see the resulting class distribution between the training and evaluation splits, which will be mentioned as TREC-CT-Train and TREC-CT-Eval.

3.1.2 SIGIR 2016 Clinical Trials

The SIGIR 2016 Clinical Trials dataset [32], which will be mentioned as SIGIR-CT, is the most similar collection found in the literature.



(a) TREC-CT-Train

(b) TREC-CT-Eval

Figure 3.5: In this Figure we have two sub-figures (3.5(a) and 3.5(b)), each for a different dataset, the left histogram shows the distribution of the number of examples each topic has, and the right histogram shows the distribution of positive examples (Excluded + Eligible) per topic.

64-year-old obese female with diagnosis of diabetes mellitus and persistently elevated HbA1c. She is reluctant to see a nutritionist and is not compliant with her diabetes medication or exercise. She complains of a painful skin lesion on the left lower leg. She has tried using topical lotions and creams but the lesion has increased in size and is now oozing.

Figure 3.6: Example Topic Description

3.1.2.1 Documents

Like the TREC-CT collection, the clinical trials are from ClinicalTrials.gov, and a December 16, 2015 snapshot of the collection is used, with a total of 204855 clinical trials.

The same analysis of the TREC-CT documents applies to this.

3.1.2.2 Topics

With a total of 60 topics, they were adopted from past editions of the Clinical Decision Support Track, 30 topics from 2014 [84] and 30 topics from 2015 [73]. The topics are in three formats (ad-hoc queries, summaries, and descriptions), but we will only examine the "descriptions"format as it is the one that most resembles the TREC-CT topics format.

An example of a topic is found in Figure 3.6, and we easily see that its length is about half of the example topic from TREC-CT. The topics averages about 78 words.

3.1.2.3 Relevance Judgments

With a small count of 3870 query relevance, they are assessed on a 3-point scale:

- **Not Relevant**. The clinical trial is not in any way relevant to the patient;

- **Would Consider**. "Would consider referring this patient to this clinical trial upon further investigation.";
- **Relevant**. "Highly likely to refer this patient for this clinical trial".

There is a contrast in these labels when compared to the TREC-CT labels, but we believe the within the classes is acceptable. This dataset is relatively imbalanced having mostly Not Relevant labels for each topic.

3.2 Protocol

The protocol for the multiple steps of the pipeline is as follows:

1. First, we index the document collection and retrieve the necessary information (e.g. [105]), this way the selected retrieval algorithm can efficiently return a few thousand relevant documents from the large document collection.
2. For the retrieval stage, this needs to be efficient, the stage will retrieve the top-k (at least 1000 documents) clinical trials for each query.
3. In the case of a reranking pipeline, the rank list will be passed on to a transformer based model, which will rerank always the top 1000 documents.
4. As we saw in section 2.5, the ims_unipd team explored the use of filters pre-retrieval, based on demographic information, here we explore the use of filters post-ranking. So any filtering done will be the last step of the pipeline.
5. To finish off, we evaluate our runs following the approach in [79], where the 3-point scale is used explicitly only for the NDCG, and for all other metrics, the excluded trials are treated as not relevant. The evaluation metrics used, are mentioned in the next Subsection 3.3.

3.3 Evaluation Metrics

In this section, we take a quick overview of the evaluation measures mentioned in this document.

Precision

Precision (P) is the fraction of retrieved documents that are relevant:

$$precision = \frac{\#(\text{relevant items retrieved})}{\#(\text{retrieved items})} \tag{3.1}$$

Precision at k (P@k) corresponds to the fraction of relevant results among the top K retrieved documents.

R-Precision (RPrec) is the precision after R documents have been retrieved, where R is the number of relevant documents for the query.

Recall

Recall (R) is the fraction of the relevant documents that are successfully retrieved:

$$recall = \frac{\#(\text{relevant items retrieved})}{\#(\text{relevant items})} \tag{3.2}$$

Normalized Discounted Cumulative Gain

Discounted Cumulative Gain (DCG) penalizes relevant documents appearing lower in the retrieved list, the traditional formula:

$$DCG_p = \sum_{i=1}^{p} \frac{rel_i}{\log_2(i+1)} \tag{3.3}$$

where p is a given rank cut-off position.

Normalized Discounted Cumulative Gain (NDCG), by dividing DCG by the best possible score, normalizes the metric to obtain a score between 0 and 1:

$$nDCG_p = \frac{DCG_p}{IDCG_p} \tag{3.4}$$

Where $IDCG_p$ is ideal discounted cumulative gain:

$$IDCG_p = \sum_{i=1}^{|REL_p|} \frac{rel_i}{\log_2(i+1)} \tag{3.5}$$

Mean Reciprocal Rank

Mean Reciprocal Rank (MRR) is the average of the reciprocal ranks of results, the reciprocal rank of a query response is the multiplicative inverse of the rank of the first correct answer:

$$MRR = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \frac{1}{rank_i} \tag{3.6}$$

4

Approaching the Matching Problem

In this chapter, we present our approaches to tackling the problem, and following the protocol mentioned in the previous Section, we first worked on the first-stage retrieval, where we experiment with different indexes and retrieval techniques, next we mention our approaches to rerank using Transformer based models, and to terminate we describe how we perform the post-rerank filtering. All of the developed code, necessary to replicate the work done, is available in the TrialMatch git repository.

4.1 First-Stage Retrieval

As is typical in all IR systems, the initial step is the retrieval stage, here we fetch information from the clinical trial corpus given information about the patient (the topic). Due to the size of the document collection, this operation needs to be efficient and fast while giving relevant results, at least a high recall, in order to not compromise the next steps of the pipeline.

In this chapter, we decided to utterly depend on Pyserini[1] [44] for our sparse retrieval methods. Pyserini is a Python toolkit for reproducible information retrieval research, in particular, it supports various sparse retrieval algorithms. Pyserini is a python interface to Anserini [100, 101] and its sparse retrieval support comes from Lucene[2], which provides diverse manners of controlling aspects of indexing and retrieval.

4.1.1 Sparse Retrieval

4.1.1.1 Indexing

The first step of this stage is indexing the documents. As mentioned in 3.1.1.1, in the TREC-CT dataset we find an XML file to represent each clinical trial in which we see various fields of meaningful text. To build the sparse Lucene inverted index on this document collection first we need to convert from the XML format to a JSON format, this file will be

[1] https://github.com/castorini/pyserini

[2] https://lucene.apache.org/

comprised of two fields, “id” and “contents” that represent the document’s unique identifier and the contents of the document in one string, respectively. For this, a small Python script was written, where we took care of the preprocessing of the text needed, using just a white space tokenizer. We tested multiple indexes of the TREC-CT corpus, creating an inverted index for the following fields: brief_title, official_title, brief_summary, detailed_description, study_pop, and criteria. To finish off we also created an index for the concatenation of all the previously mentioned fields, joining them with a white space. All these indexes were created using Pyserini which is called the DefaultLuceneIndexer class that created the desired index.

As for the option of exploring multiple preprocessing techniques, like, a punctuation tokenizer or changing the capitalization of words, we found that there was no sense in its use due to in the Biomedical domain we easily find terms where punctuation and capitalization matter for its definition. While the use of other document processing tools such as stemming and stopword removal were chosen not to be explored as it wasn’t the focus of this work. The results of this retrieval stage also presented themselves as good enough, proving that there was no need to explore all these variables.

4.1.1.2 Retrieval

The choice of a good retrieval method is imperative, basing our choice on modern literature and the team’s submissions on the 2021 TREC CT Track, BM25 was chosen as it has proven to obtain good results consistently on various tasks. The Pyserini’s BM25 implementation with the default parameters was used to retrieve the top 1000 documents (clinical trials) for each query (topic), on the indexes mentioned before: brief_title, official_title, brief_summary, detailed_description, study_pop, criteria, and concatenation. In this way, we understand how each field carries more relevant information, so we can make later decisions based on the results. In Figure 4.1 we see the pipeline used for the retrieval.

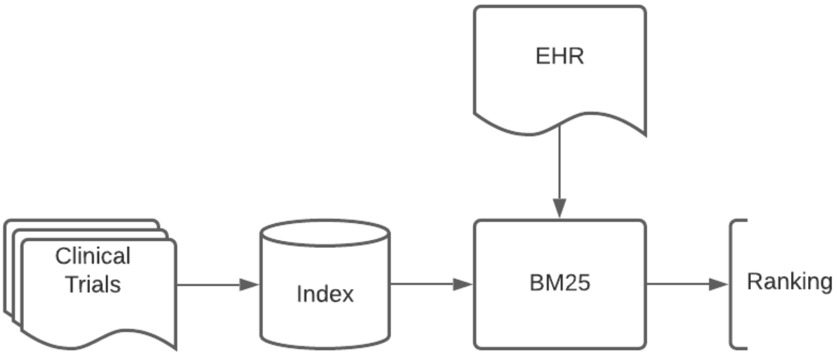


Figure 4.1: First-stage retrieval pipeline with BM25

To mitigate some drawbacks of BM25, we decided to jointly use RM3 [36], a query expansion strategy that was shown to effectively boost the effectiveness of the retrieval step, in turn, return a better ranking list. Further exploring the use of BM25 + RM3, we tested this retrieval method on the TREC-CT-Train and TREC-CT-Eval splits, optimizing the parameters to obtain a higher recall (on the TREC-CT-Train), by testing thousands of possible combinations for the parameters.

4.1.2 Rank Fusion

The queries that serve as input for our system are 5 to 10 sentences long and BM25 is known to not deal with long queries in a proficient way, so inspiring ourselves in the work done in [43], we decided to proceed in a similar manner, using doc2query-T5 [60], a neural passage expansion technique.

With doc2query-T5, we input a document and the model infers questions that the document might answer. Now giving our patients descriptions as input, generating multiple queries for each description (topic), these queries can now be used as input for our retrieval system. With this method, we have decomposed each topic into multiple smaller sentences. Now for each patient, we have multiple topics, and we use our sparse retrieval method to create multiple ranking lists for just a given patient, to deal with joining all lists into a single one we resort to rank fusion.

RRF [14], a simple method that has shown to consistently improve evaluation metrics when combining results of multiple retrieval methods, when compared to the best of those methods, also the RRF consistently outperforms other fusion algorithms, such as CombMNZ [57].

RRF works by favoring documents at the "top"of the rank and penalizing the ones below the "top". This "top"can be controlled via a constant, mitigating its impact. The function is as follows:

$$RRFscore(d \in D) = \sum_{r \in R} \frac{1}{k + r(d)} \tag{4.1}$$

where:

- D: Document Collection to be ranked;
- R: Set of rankings, each a permutation on $1..|D|$;
- k: Constant, defaulted to $k = 60$;
- $r(d)$: The rank position of document d in the rank r.

4.1.2.1 Query Expansion

As mentioned above, the topics were broken down into smaller sentences, using the neural passage expansion technique doc2query-T5 [60]. At first, the objective was to use a T5-3B fine-tuned on the MS MARCO V1 passage ranking training set, to mimic [43], Patient is a 45-year-old man with a history of anaplastic astrocytoma of the spine complicated by severe lower extremity weakness and urinary retention s/p Foley catheter, high-dose steroids, hypertension, and chronic pain. The tumor is located in the T-L spine, unresectable anaplastic astrocytoma s/p radiation. Complicated by progressive lower extremity weakness and urinary retention. Patient initially presented with RLE weakness where his right knee gave out with difficulty walking and right anterior thigh numbness. MRI showed a spinal cord conus mass which was biopsied and found to be anaplastic astrocytoma. Therapy included field radiation t10-l1 followed by 11 cycles of temozolomide 7 days on and 7 days off. This was followed by CPT-11 Weekly x4 with Avastin Q2 weeks/ 2 weeks rest and repeat cycle.

1: "anaplastic astrocytoma of the spine treatment"
2: "anaplastic astrocytoma of the spine"
...
19: "t-l tumor location treatment"
20: "where is t-l spine"
...
39:"what is temozolomide for resectable astrocytoma"
40:"anaplastic astrocytoma"

Figure 4.2: TREC Clinical Trials - Topic 1 Query Expansion

but we ran into GPU memory issues, that were solved by downsizing the model to the T5-Large, as it is available on Hugging Face.

The fine-tuning process of the doc2query-T5 (both T5-3B and T5-Large) was made with a constant learning rate of $1 \cdot 10^{-4}$ and a batch size of 256 for 4K iterations, on the adapted MS MARCO V1 passage ranking training set, which corresponds to 2 epochs. At inferring time we used a maximum of 512 input tokens and 64 output tokens and a top-k sampling ($k = 10$), to sample 40 queries for each of the 75 topics.

In Figure 4.2 we see an example of the final result on topic number 1, here we find that the model generated with success shorter queries, a maximum of 8 words long, that maintain important keywords such as diseases and treatment names. With these smaller topics in hand, we can actually explore the use of them in future steps of the ranking pipeline, as most methods have smaller queries in mind when they were created.

4.1.2.2 DeepSpeed

As we ran into GPU memory issues when fine-tuning a T5-3B model, on our GPU NVIDIA A100 with 40GB, the first approach was just to reduce the batch size but the problem persisted for future work, and with this in mind, we were forced to find a solution and turned into DeepSpeed[3]. DeepSpeed enables us to use Zero Redundancy Optimizer (ZeRO) [71] which allows us to fine-tune large models across multiple GPUs in an efficient and optimized way, or on a single GPU using CPU of load making use of the large amounts of RAM present in the server. The main problem that we tackled here was to enable the use of DeepSpeed on the NOVA Search Hub.

[3] https://www.deepspeed.ai/

4.1.3 Dense Retrieval

Work done by our colleague João Pereira in [15], where the use of dense retrieval using bi-encoders on this exact task is explored.

Using the FAISS[4] [30] library, to index the sentence embeddings representing the trials, the indexes are stored as a flat index. A similar approach to ours was taken, where various fields were indexed individually and different combinations of them too, including a concatenation of all fields.

For retrieval FAISS efficiently accesses the chosen index and retrieves the top 1000 documents, by computing the dot product between the trial embedding and the topic embedding.

The models tested using the above approach are the following:

- msmarco-bert-base-v5[5]: BERT-Base model fine-tuned for semantic search on the MSMARCO Passage Ranking dataset.

- all-distilroberta-v1[6]: A distilled version of RoBERTa [48], it is an all-round model fine-tuned on large and diverse datasets for different use cases.

Model fine-tuning of both of the above models was explored, using a Multiple Negative Ranking Loss and Negative Sampling and Batching, using the SIGIR-CT dataset and the TREC-PM dataset.

His work is mentioned here, due to us testing his models in our work. We refer to his thesis [15] for an in-depth understanding.

4.1.4 Summary

Resuming our developed work in this Section 4.1, composing all that was mentioned into a single pipeline, we first start by expanding our 75 patient descriptions with the doc2query-T5-Large model producing a total of 40 short sentences. With a total of 41 topics per patient (40 expanded queries + original topic) and using a chosen index created on the TREC-CT corpus, we compute a total of 41 ranking lists that were retrieved using BM25 + RM3, these lists are then fused using RRF, producing a single final rank for our first-stage retrieval.

Notice that the query expansion method adds some latency to the online phase of the pipeline, if used in a real case scenario, the only offline step here is the creation of the indexes. The final pipeline is represented in Figure 4.3.

Further fine-tuning of the parameters of BM25, RM3, and RRF was looked into, on the TREC-CT-Train and its effects tested on the TREC-CT-Eval, focusing on improving the recall as it is the most important metric in the retrieval stage.

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

[5] https://huggingface.co/sentence-transformers/msmarco-bert-base-dot-v5

[6] https://huggingface.co/sentence-transformers/all-distilroberta-v1

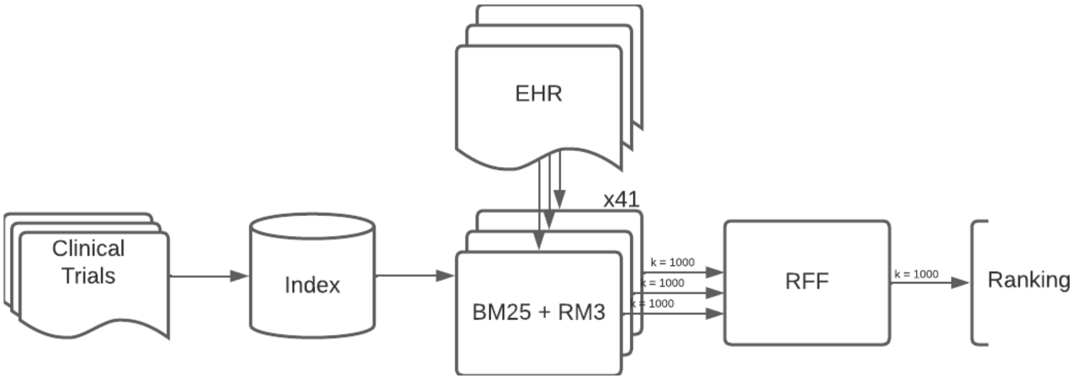


Figure 4.3: First-stage retrieval pipeline with BM25 + RM3 + RFF

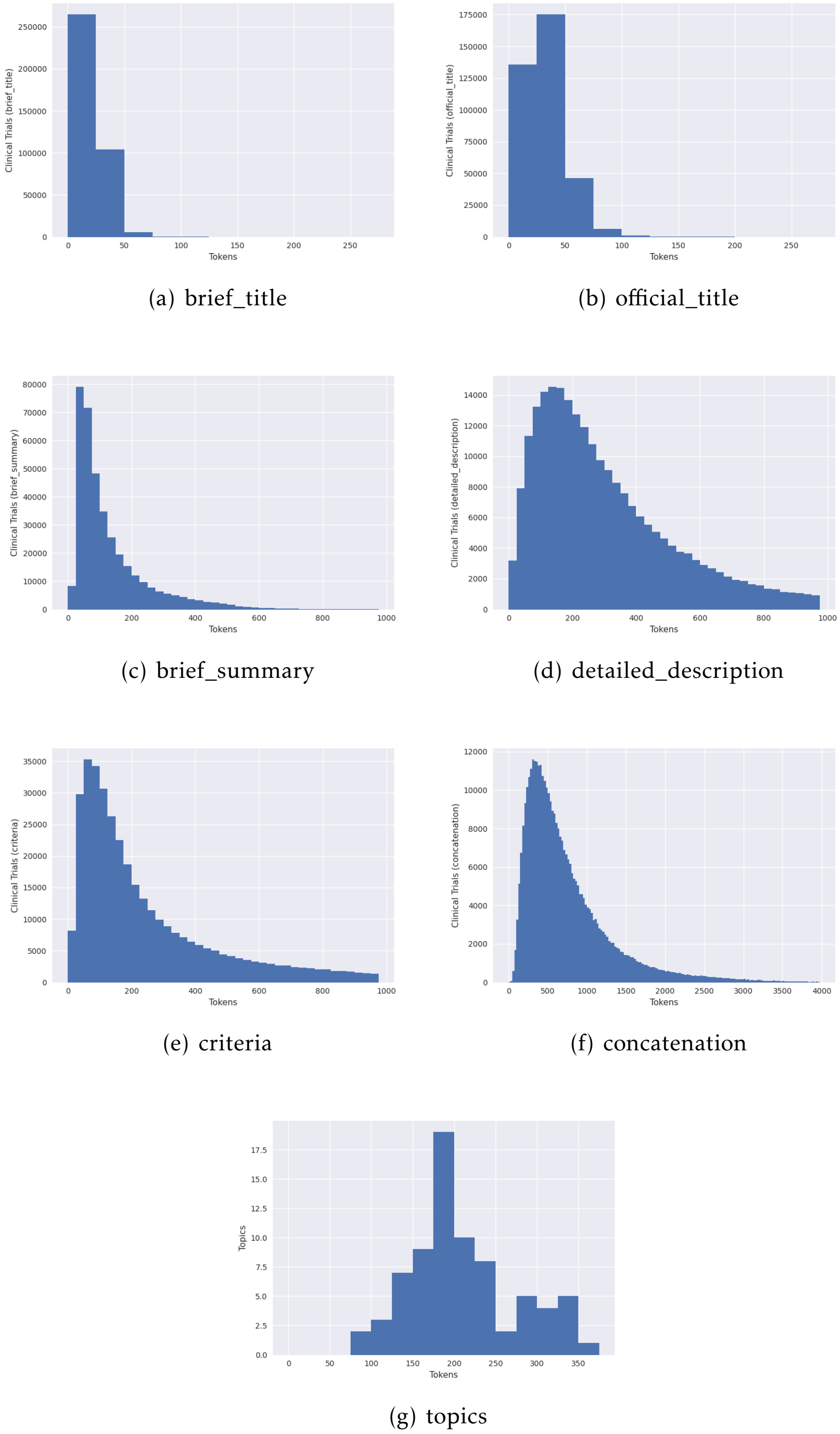
4.2 Reranking

In this section, we delve into the use of neural ranking models described in Section 2.4 and describe our approach to make use of various model architectures for the task of reranking our retrieved list. Here we find ourselves with various options to make use of BERT [18] for the task, basing our decision making on state-of-the-art literature, we decided to test both cross-encoders and bi-encoders and see how both deal with the task at hand. In this phase we aim to push the Eligible trials to the top of the ranking list, improving metrics such as P@10 and NDGG@10.

In this task, we operate with both long queries and documents, to add to the already hard task at hand we also need to deal with the clinical domain-specific language. Firstly to see how each of the selected fields would impact the input length of a selected model, we used a BERT tokenizer and individually tokenize the topics, and each field separately (and the concatenation of these fields), building histograms based on this information, visible in Figure 4.4. Most models have a maximum input length of 512 tokens, here we observe that in most cases the topic will occupy half of the input sequence, not leaving much space left for the document, in the use case of cross-encoders.

We are posed with various variables and to explore them it was decided to explore fine-tuned models on other similar tasks and test them as a zero-shot approach to the problem, we also explored models with different maximum input lengths to see how this would affect the results, and to finish we also tested models pre-trained on the biomedical domain. We proceeded to fine-tune selected models on the task at hand, analyzing various architectures. For working on the problem, we used both Hugging Face[7] and SentenceTransformers[8].

[7] https://huggingface.co/
[8] https://www.sbert.net/



(a) brief_title

(b) official_title

(c) brief_summary

(d) detailed_description

(e) criteria

(f) concatenation

(g) topics

Figure 4.4: Histograms of the token length (using a BERT tokenizer) of multiple fields, the concatenation of those fields. and the topics.

4.2.1 Bi-Encoders

Starting with bi-encoders, as a reminder, these types of models perform self-attention over the input separately producing sentence embeddings for both the query and the document, these embeddings are then compared using similarity measures. In our case, we based our work on [15], and explored the use of the same models, both the zero-shot approach models and the fine-tuned model (best performing one), and used them by choosing our original topics as the query and the concatenation of the selected fields as the documents, after producing the sentence embeddings the dot product between them is performed, obtaining the final score that is used for reranking the list.

4.2.2 Cross-Encoders

Cross-encoders are computationally heavy due to them performing full cross self-attention over the input, but hopefully, the high latency returns good results. Based on the token length both the topic and trials shown in Figure 4.4, the duoBERT architecture was not explored, as there was no possibility to fit the needed information in the input template given the maximum length of 512 tokens. So we explored the monoBERT, using the following input template:

$$[\text{CLS}]\ q\ [\text{SEP}]\ d\ [\text{SEP}]$$

where q and d are the patient description and the clinical trial, respectively.

In this Subsection we seek to explore three defined objectives:

1. Approaching the task with models fine-tuned on similar tasks.
2. Exploring different input lengths of our models, more explicitly, 512 tokens, 1024 tokens, and 4096 tokens.
3. Fine-tuning models pre-trained on the biomedical domain, and general domain models.

4.2.2.1 Zero-Shot Approach

To rerank using models that are not fine-tuned on the task, we look into the output format of a given model and chose the most similar class to the Eligible class, the output logits of this chosen class are used as the score for reranking the pairs. We follow the asked input format for each model, using as query the original topic, and as the document the criteria of each clinical trial. Below we have a brief description of the models that were tested, notice that the first two models have a maximum input length of 512 tokens, the following two an input length of 1024 tokens, and the last two an input length of 4096 tokens:

- **ms-marco-MiniLM-L-12-v2**[9]: A model trained on the MS Marco Passage Ranking

[9] https://huggingface.co/cross-encoder/ms-marco-MiniLM-L-12-v2

task. This task is similar to ours, with the exception that is on general-domain corpura.

- **mmarco-mMiniLMv2-L12-H384-v1**[10]: A model trained on the MMARCO dataset, it is a machine translated version of MS MARCO using Google Translate, being a multilingual model.
- **quora-roberta-large**[11]: This model was trained on the Quora Duplicate Questions dataset. Inferring similarity between two texts might be a good approach.
- **stsb-roberta-large**[12]: This model was trained on the STS benchmark dataset. It is a model that infers semantic similarity between two sentences.
- **longformer-base-4096-finetuned-squadv1**[13]: This is a Longformer [10] model fine-tuned on SQuAD v1 dataset for question answering task.
- **longformer-base-plagiarism-detection**[14]: Longformer trained on the Machine-Paraphrased Plagiarism Dataset.

4.2.2.2 Fine-Tuning Models

Starting on pre-trained LMs, mostly based on BERT [18], but also the Longformer [10]. We explored adaptations of these models for the rerank task. Focusing on the defined objectives mentioned before we fine-tuned, general domain models of different sizes (BERT), a model with an input size of 4096 tokens (Longformer), and biomedical domain models (some of the models mentioned in 2.8).

To maintain most variables in common when testing different models, as the focus is not on optimizing and getting the most out of a single model but to see how a model behaves in the given task, we maintained our hyper-parameters static, a learning rate of 5e−5, zero warmup steps, a linear scheduler, and insuring that the backpropagation step would only be computed after a batch size of 32 (when the batch size did not fit into the GPU memory we changed the gradient accumulation steps to match the preferable batch size), we are of course needed to change the loss function according to the output format, CrossEntropyLoss for single label classification, BCEWithLogitsLoss for multi label classification, and MSELoss for regression.

Given that the relevance judgments are assessed on a 3-point scale: Not Relevant (0), Excluded (1), and Eligible (2). We hypothesized four different output formats for our models, the first three, where we maintain the classification problem and use the logits of our target class as the rank score, and the last one, where we turn the task into a regression problem:

[10] https://huggingface.co/cross-encoder/mmarco-mMiniLMv2-L12-H384-v1

[11] https://huggingface.co/cross-encoder/quora-roberta-large

[12] https://huggingface.co/cross-encoder/stsb-roberta-large

[13] https://huggingface.co/valhalla/longformer-base-4096-finetuned-squadv1

[14] https://huggingface.co/jpwahle/longformer-base-plagiarism-detection

Eligible vs. All (2A) Here we turn to our early objective of making a distinction between Excluded and Eligible trials, and we unit the Not Relevant and Excluded trials into a single class (0), leaving all the Eligible trials in a separate class (1);

Not Relevant vs. All (2B) In this case, we simply turned to the previous hypothesis and did the "opposite" by leaving the Not Relevant trials in class (0) and joining the Excluded and Eligible trials into class (1);

Multiclass (3) In this output format, we decided to deviate from the monoBERT binary classification and left the original classes: Not Relevant (0), Excluded (1), and Eligible (2);

Regression (R) As the previous format left questions on how to rerank with a model that outputs more than two classes, we decided to turn the problem into a regression task, giving the following values to each class: Not Relevant (-1), Excluded (0), and Eligible (1). This decision was made on the basis that in a rank list, we want the Eligible trials above the Excluded and the Excluded above the Not Relevant ones.

Founded on the input template mentioned above, the multiple output formats, and using the same training parameters, we fine-tuned numerous models, on the following training datasets.

SIGIR-CT

Starting with the SIGIR-CT dataset, which contains 3870 examples, we use the patient descriptions offered because they most resemble the topics in TREC-CT. For each of the output formats mentioned before, a respective training dataset was built where q is the patient description and d is the criteria field from the clinical trial, as follows:

$$[\text{CLS}]\ \textit{description}\ [\text{SEP}]\ \textit{criteria}\ [\text{SEP}]$$

the input is then padded and truncated to the maximum length of the model (512 tokens).

We end up with the following statistics for the datasets (a total of 3870 examples):

(SIGIR-CT-2A) With 3449 negative examples (Not Relevant + Excluded) and 421 positive examples (Eligible);

(SIGIR-CT-2B) With 2764 negative examples (Not Relevant) and 1106 positive examples (Excluded + Eligible);

(SIGIR-CT-3 and SIGIR-CT-R) Having 2764 Not Relevant examples, 685 Excluded examples, and 421 Eligible examples.

SIGIR-CT$_{Balanced}$

As we easily visualize, the SIGIR-CT dataset is composed mainly of negative examples. To address this problem, we repeated the positive examples to balance the datasets. We built these new datasets using the previous template and method.

Below we show how this affected the datasets:

(SIGIR-CT-2A$_B$) With 10135 examples, of which 5504 are negative examples (Not Relevant + Excluded), and 4631 are positive examples (Eligible);

(SIGIR-CT-2B$_B$) With 8030 examples, of which 2764 are negative examples (Not Relevant), and 5266 are positive examples (Excluded + Eligible);

(SIGIR-CT-3$_B$ and SIGIR-CT-R$_B$) Having 2764 Not Relevant examples, 2740 Excluded examples, and 2526 Eligible examples, ending up with a total of 8030 examples.

TREC-CT-Train

Then we tested it out with the TREC-CT-Train split we built. There are some differences between the SIGIR-CT and TREC-CT datasets, as the latter has lengthier topics describing the patient, and the distribution of the classes is more balanced. Using the input template shown below:

$$\text{[CLS] } admission\ note \text{ [SEP] } criteria \text{ [SEP]}$$

now using as the query the admission notes, we proceed with the same method as before, building these new four datasets.

The statistics are the following (with 35825 examples each):

(TREC-CT-Train-2A) With 30255 negative examples (Not Relevant + Excluded) and 5570 positive examples (Eligible);

(TREC-CT-Train-2B) With 24236 negative examples (Not Relevant) and 11589 positive examples (Excluded + Eligible);

(TREC-CT-Train-3 and TREC-CT-Train-R) Having 24236 Not Relevant examples, 6019 Excluded examples, and 5570 Eligible examples.

TREC-CT-Train$_{Expanded}$

The admission notes are long, and most models are fine-tuned with smaller queries in mind. We used the query expansion technique used previously, doc2query-T5, using the 40 inferred queries for each patient admission note. We expanded our dataset using the template below:

$$\text{[CLS] } inferred\ query \text{ [SEP] } criteria \text{ [SEP]}$$

this made our dataset 40 times bigger and made it possible for more tokens of the criteria field to be present in the input without being padded. To infer using 40 queries for a single pair topic-trial, a MaxP approach is used, meaning that out of the 40 logits the model inferred the maximum value was chosen as the rank score for the pair.

Ending up with the following statistics (totaling in 1433000 examples):

(TREC-CT-Train-2A$_E$) With 1210200 negative examples (Not Relevant + Excluded) and 222800 positive examples (Eligible);

(TREC-CT-Train-2B$_E$) With 969440 negative examples (Not Relevant) and 463560 positive examples (Excluded + Eligible);

(TREC-CT-Train-3$_E$ and TREC-CT-Train-R$_E$) Having 969440 Not Relevant examples, 240760 Excluded examples, and 222800 Eligible examples.

4.2.3 Summary

In this Section, we presented two different approaches to the reranking task where the aim is to push better matching trials of a patient to the top of the list. When using bi-encoders, sentence embeddings for both the topics and the trials are produced, then the dot product between them is computed and we use that value as the rank score. For the use of cross-encoders, full cross self-attention over the input is performed and then we pass the logits of the [CLS] token to an output head that produces the rank score.

We explored the use of adaptations for these models, testing both the original topic and the expanded topics as the query, and testing as the document multiple fields of the trial (including the concatenation of them), but giving a big emphasis to the criteria.

To enable the fine-tuning of our cross-encoders, four training sets were created based on existing ones, being them the following: SIGIR-CT; SIGIR-CT$_{Balanced}$; TREC-CT-Train; TREC-CT-Train$_{Expanded}$. Where we explore various options on how to label the data.

4.3 Filtering

In this section, we tackle what we mention as post-ranking filtering. As the name indicates, what we do is pick the final ranking list that our pipeline produced and we filter all documents present, by trying to understand if the patient is eligible for a given clinical trial. Reminding that the trials are judged as Not Relevant, Excluded, and Eligible, in this Section we want to make a distinction between the Excluded trials and the Eligible ones, assuming that no Not Relevant trial is in the final ranking list. This task is one of the core aspects of the 2021 CT track, as mentioned by its creators where they hard emphasize the distinction between Excluded and Eligible trials using the inclusion/exclusion criteria.

In the clinical trial XML structure, we have the attribute "eligibility"(Figure 4.5) where we have all the needed information, excluding the "criteria", the other mentioned fields represent demographic rules and are structured, then back to the "criteria"field, we find

```
<eligibility>
                                        ...
<criteria>
<textblock>
 Inclusion Criteria:
- Patients must have a diagnosis of cancer of any histologic type.
- Patients must have a Karnofsky performance status great or equal to 70%.
- Patients must have an expected survival for at least four months.
- Normal healthy volunteers to serve as control for this study.
- All patients must sign informed consent approved by the Committee on the Use of Human
 Subjects at the University of Minnesota
 Exclusion Criteria:
- Pregnant or lactating women. Females of child-bearing potential will be asked to take a preg-
 nancy test before receiving vaccines.
- Serious intercurrent medical illnesses which would interfere with the ability of the patient to
 carry out the follow-up monitoring program.
- Immunization should not be administered during the course of any febrile illness or acute
 infection.
- Hypersensitivity to any component of the vaccine, including Thimerosal, a mercury derivative.
- The occurrence of any type of neurologic symptoms to tetanus vaccine in th past.
- Patients with a history of seafood allergy are excluded from receiving KLH.
- Subjects who have had tetanus toxoid within the last 7 years are not eligible for tetanus vaccine
 component of this protocol.
</textblock>
</criteria>
                                        ...
<gender>All</gender>
<minimum_age>18 Years</minimum_age>
<maximum_age>N/A</maximum_age>
                                        ...
```

Figure 4.5: Example from a Clinical Trial (NCT00000105)

a semi-structured text (in most cases) in which we have inclusion criteria and exclusion criteria about what the patient medical situation needs to be.

We approach the problem in two different ways, one by using the demographic restrictions and the other by using the criteria field, these solutions were then tested by adding them as the last step of the pipeline.

4.3.1 Demographic Rules

The demographic fields in clinical trials (Figure 4.5) easily inform us of the demographics that a patient needs to have, as these fields are structured, we effortlessly retrieve the information, so the main problem is to retrieve these characteristics from a given patient's description (topic).

When looking into the original topic in Figure 4.2, we observe that there is natural language pointing to the gender and age of the patient, for example, a male patient can be referred to as a man or a boy, and a similar language is used to refer a patient age. Given this information, the use of computational heavy models was not attempted, as with the

```
r"\d+ *(yo|year old|-year-old|year-old|y/o|-year old|year|-day-old|months old)?"
r"(( *(\w*|African-American) *),6(man|woman|female|gentleman|male|boy|girl|F|M))?"
```

Figure 4.6: Regular Expression patterns used to extract demographic information from the topics

use of Regular Expressions (RegEx), the creation of two regular expressions (Figure 4.6) was enough to solve the problem of retrieving the demographic information of the topic, creating metadata that indicates the age and gender of a given patient.

With these rules in mind, we wrote an algorithm that compares each pair topic-trial present in the ranking list, by extracting the metadata from the topic and then comparing it to the demographics eligibility of the clinical trial (age gap, and gender). As each pair topic-trial is associated with a score that indicates how relevant the clinical trial is to the patient, we proceed to penalize such score by a value p in the case the patient does not satisfy the eligibility age or gender of the trial.

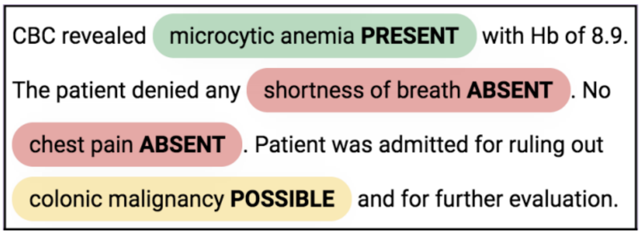
4.3.2 Named Entity Recognition

Intending to explore the use of the criteria field (Figure 4.5) to make a distinction between the Excluded and Eligible trials, but how? Our hypothesized approach is to make use of NER models and retrieve clinical entities from the topics and the trial's criteria, and with the due context for those entities, we could maybe imply if a rule from the inclusion criteria or exclusion criteria, is satisfied or broken.

To retrieve clinical entities, the first approach was to use models in the literature, that show good results on the task at hand. The following services were found, Lexigram[15], and BERN2 [86], both have public APIs, although they work very well at extracting the right entities, both APIs would stop responding due to the quantity requests becoming unusable at inferring time. This brings as to the next step, the use of local models, exploring some of the popular transformers present in Hugging Face[16], the resulting extractions of clinical entities was not successful, as we were presented with non-medical entities, as these models were fine-tuned on general domain datasets. The solution is then to use transformers trained in the biomedical domain, but even these had problems, for example, a disease name would in most cases be broken into multiple tokens, due to pretraining, supporting what was said in Section 2.8.

We took on the task of fine-tuning a NER model, by using the biomedical domain model and the fine-tuning script made available by [103] in their git repository. Inspired by the BLURB [24] leaderboard where BioLinkBERT-Large [103] has the highest score on the NER tasks. We decided to use this exact model, by sequentially fine-tuning it on the following tasks (made available by BLURB): BC2GM [85], BC5-Chem [40], BC5-Disease

[15] https://www.lexigram.io/
[16] https://huggingface.co/



CBC revealed microcytic anemia **PRESENT** with Hb of 8.9. The patient denied any shortness of breath **ABSENT**. No chest pain **ABSENT**. Patient was admitted for ruling out colonic malignancy **POSSIBLE** and for further evaluation.

Figure 4.7: Example of how the Clinical Assertion and Negation Classification BERT classifies entities. [2]

[40], JNLPBA [13], and NCBI Disease [19]. Ending up with a powerful model able to extract the important clinical entities, from both the topics and the criteria.

So for each pair of topic-criteria, we extracted the entities and created metadata about the pair, informing us, of the entities in common, and for each entity if it corresponds to the topic or the clinical trial, and if it belongs to the trial it indicates if it is in the inclusion criteria or the exclusion criteria, plus we save the entity position in the original text. We have multiple options from this step, which will be detailed next.

4.3.2.1 Clinical Assertion

Starting from the previous step of NER, one of our hypotheses to have context awareness for an entity and understand its significance in the rules present in the inclusion and exclusion criteria is to make use of models capable of classifying if an entity has a positive weight or negative in the topic and the trial, and thus establish a score to the pair based on this information.

We select the following model, Clinical Assertion and Negation Classification BERT [2], that enables us to assert if a clinical entity (symptoms, treatments, and other relevant entities) is present, a possibility, or absent in the patient diagnosis, as Figure 4.7 indicates.

To achieve this the model needs the input sequence in the form of a sentence with one marked entity to classify, the entity in question is identified with the special token [entity] surrounding it, as follows:

[CLS] Hypersensitivity ... including [entity] Thimerosal [entity] ... [SEP]

To make use of the model, starting from the NER phase, for each topic-trial in the ranking list, we take into account their clinical entities in common, and for each entity, we create a span of text (on the topic and the clinical trial) that has its center on the entity position on the original text and goes 30 characters back and forward, this span is then converted to the mentioned input format and passed to the model, which infers the probability of the three output classes. To notice that an entity can appear multiple times in the original text, in this case, the mean distribution of the probabilities is calculated. In the case the entity belongs to the exclusion criteria, we switch the probabilities of the PRESENT and ABSENT classes.

After which, still working on the "for each entity", we selected the PRESENT class and subtract the probability the entity got on the topic from the probability the entity got on the trial, obtaining the absolute value of the subtraction, for example, if the topic got PRESENT=0.2 and the trial got PRESENT=0.7, we get a score of PRESENT=0.5 for the pair. We treat this score as to how similar the context of the entity in both texts is, the lower the better. Expecting the higher the score an entity gets the more likely it is for the criteria rule the entity is part of is being broken, making the patient not eligible.

Each pair in the ranking list has now a list of common entities with an associated score, between 0 and 1, to enable this post-filtering we actually chose the highest score present in the list of common entities, calling it p, and penalized the pair's score on the rank by subtracting p.

4.3.2.2 Sentiment Analysis

Continuing from the above idea, we looked into different models to classify an entity context and considered Sentiment Analysis (SA). SA models classify sentences as positive, negative, or neutral based on the emotional tone present in the text. Although at first glance it makes no sense to explore this, due to clinical writing normally being neutral and formal, we thought to try as maybe there could be some keywords left into spans of text that could transfer some positive or negative sentiment, after all the text is written by Humans.

So following step by step the above algorithm, we just switched our model with a DistilBERT model fine-tuned on the SA task[17], taking into account the new classes, we selected the POSITIVE class, and again calculated the context similarity scores.

4.3.3 Summary

To summarize the work done in this section, to be able to differentiate Excluded from Eligible clinical trials we started by analyzing the contents of the eligibility attribute where we made a distinction between the semi-structured criteria and the structured demographics criteria.

For the demographic criteria, we successfully created rules based on RegEx that extracted the needed information from the patient description, and based on this information, we penalized the score of the trials which broke these rules.

On another hand, focusing on the text field “criteria”, to achieve a system that understood the inclusion and exclusion rules we used a NER model to extract clinical entities, after which we used the context of these entities in both the topic and the clinical trials, to infer a score that we would hope to indicate if a criteria rule is being broken or not.

[17] https://huggingface.co/distilbert-base-uncased-finetuned-sst-2-english

5

Evaluation and Result Analysis

In this chapter, we present and discuss the results obtained using the approaches mentioned in the previous chapter. We discuss the results obtained in the first-stage retrieval, the reranking phase, and lastly the post-ranking filtering, using the different techniques mentioned for each method. To close the chapter, the last section addresses the questions this research investigated and its key findings.

5.1 Experimental Results

To evaluate the runs, trec_eval[1] is the standard tool. While trec_eval has brought many benefits, it has a downside that it is available only as a standalone executable. As our IR system is implemented using Python, two python packages were used for evaluating, pytrec_eval[2] [93] and ranx[3], both expose a similar interface of trec_eval. The chosen metrics are the following: P@10; NDCG@10; MRR; RPrec; R@10; R@100; R@500; R@1000. Remember that the 3-point scale is used explicitly only for the NDCG, and for all other metrics the Excluded trials are treated as Not Relevant.

5.1.1 First-Stage Retrieval

Our first experiment focused on investigating how each field affected the results, in this phase our focus in on the recall value, although, how each method affects the rest of the metrics is also in our scope. In Table 5.1, we see the results of our approach to indexing the documents and using Pyserini's BM25 implementation with the default parameters, here we notice that most fields have meaningful information as most of the obtained scores are similar, with the exception of the detailed_description, study_pop, and criteria. We notice that the most critical information for the matching task is in the fields detailed_description and criteria, given that the results implied that the criteria field has the most impact. For the study_pop the low evaluation scores can be explained

[1] https://github.com/usnistgov/trec_eval

[2] https://github.com/cvangysel/pytrec_eval

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

Table 5.1: First-Stage Retrieval of the top 1000 clinical trials using BM25 on different indexes.

Index	P@10	NDCG@10	MRR	RPrec	R@1000
brief_title	0.0813	0.1264	0.1698	0.0471	0.2197
official_title	0.0800	0.1391	0.1838	0.0504	0.2157
brief_summary	0.0827	0.1648	0.2359	0.0550	0.2142
detailed_description	0.0987	0.1859	0.2923	0.0554	0.1702
eligibility study_pop	0.0707	0.1069	0.2037	0.0301	0.1169
eligibility criteria	0.1173	0.2471	0.2795	0.0594	0.1606
concatenation	**0.1640**	**0.2901**	**0.3125**	**0.0927**	**0.2610**

by the lack of trials having this field, only around 21% have it present. To finalize the analysis of this table, we notice that the concatenation of all fields leads to better results as no information is lost, this is an expected result when using sparse retrieval methods.

In Table 5.2, we have the three described stages of our sparse retrieval method, as the baseline, we chose the BM25 method applied on the concatenation index. The BM25 + RM3 method shows us that the retrieval can be improved by a fair amount without adding much complexity, RM3 is relatively light when compared to other query expansion strategies. To finalize we have our BM25 + RM3 + RRF method, which significantly adds latency in a real use case scenario as the doc2query-T5 needs to create the shorter queries, this stage improves the metrics greatly. We believe that the trade-off in computational complexity is worth it when comparing the baseline with our final retrieval method, as the recall value increases from 26% to 60%, which is reasonably good for a retrieval stage of 1000 documents.

Table 5.2 also shows the expected results when approaching the task with dense retrieval, the first two models used were fine-tuned on different tasks, these runs show that a zero-shot approach to the problem using bi-encoders leads to similar recall values to the sparse methods BM25 and BM25 + RM3, which are less computational complex. The last model, all-distilroberta-v1 fine-tuned (best model in [15]), which was fine-tuned on the TREC-PM dataset, presents a recall of 60% that is the same as our BM25 + RM3 + RRF method. Still, the dense retrieval technique is significantly more computationally heavy, turning out to not be worth its use. To notice that when comparing all-distilroberta-v1 fine-tuned with BM25 + RM3 + RRF, the rest of the metrics (with the exception of NDCG@10) improved, begging to question the use of bi-encoders on the rerank phase.

We observe that the method BM25 + RM3 + RRF with the default parameters (zero-shot approach) is on par with a fine-tuned bi-encoder, which begs the question of optimizing our sparse method to the task at hand. In Table 5.3, we explored the fine-tuning of the BM25 + RM3 + RRF parameters and chose the parameter combination that gave the biggest improvement in the recall on the TREC-CT-Train collection, we notice that there is a big jump in the recall value for the TREC-CT-Train that is translated in a smaller improvement in the TREC-CT-Eval. Showing that there is value in fine-tuning sparse methods.

Table 5.2: First-Stage Retrieval of the top 1000 clinical trials using the different stages of our sparse methods on the concatenation index, and different bi-encoders.

Model	P@10	NDCG@10	MRR	RPrec	R@1000
		Sparse Retrieval			
BM25	0.1640	0.2901	0.3125	0.0927	0.2610
BM25 + RM3	0.2053	0.3498	0.3537	0.1338	0.4494
BM25 + RM3 + RRF	0.2360	**0.3817**	0.3606	0.1654	**0.6012**
		Dense Retrieval			
msmarco-bert-base-dot-v5	0.1347	0.2343	0.3117	0.0740	0.2701
all-distilroberta-v1	0.1973	0.3220	0.4501	0.1280	0.4221
all-distilroberta-v1 fine-tuned	**0.2720**	0.3672	**0.4459**	**0.1832**	0.5980

Table 5.3: First-Stage Retrieval of the top 1000 clinical trials on the concatenation index, using the BM25 + RM3 + RRF method, where its parameters where fine-tuned to improve the recall on the TREC-CT-Train dataset.

Finetuned	Dataset	P@10	NDCG@10	MRR	RPrec	R@1000
No	TREC-CT-Train	0.2289	0.3910	0.3420	0.1559	0.6195
No	TREC-CT-Eval	0.2432	0.3722	0.3797	0.1752	0.5825
Yes	TREC-CT-Train	0.2000	0.3471	0.3674	0.1533	0.6608
Yes	TREC-CT-Eval	**0.2486**	**0.3675**	**0.4247**	**0.1916**	**0.5962**

5.1.2 Reranking

In this stage, we are tasked with reranking our retrieved list of 1000 trials for each topic, produced by our selected retrieval method (BM25 + RM3 + RRF). We start by using bi-encoders, which produce sentence embeddings for both the topic and the document (we used the concatenation of the fields), in Table 5.9 we have the results of the approach by reranking (1), the models (2) and (3) before described, are a zero-shot approach, the model (3) slightly improved the P@10, the NDCG@10, and the RPrec, whit a big improvement in the MRR. Not to our surprise, model (4) optimized for retrieval when applied for reranking on a retrieved list with high recall, the results are highly improved.

In Table 5.9 under the Cross-encoders-Zero-shot row, we have the results of applying models fine-tuned on other tasks, to this matching task. Model (5) has a maximum input length of 512 tokens, the following two (6) and (7) have an input length of 1024 tokens, and the last two (8) and (9) have an input length of 4096 tokens, all of these models are pre-trained on general domain corpora. Looking at the results, we can not infer much about the input length, as the results are bad across the board, but we notice that the best performing model is (5), which is fine-tuned on the most similar task to ours, the MS Marco Passage Ranking task.

Next, we explored fine-tuning a biomedical domain model, BioLinkBERT-Large, on our task. To enable this, we used our four output formats: Eligible vs. All (2A); Not Relevant vs. All (2B); Multiclass (3); Regression (R). Also, we explored the use of our four training datasets: SIGIR-CT; SIGIR-CT$_{Balanced}$; TREC-CT-Train; TREC-CT-Train$_{Expanded}$.

For each case, we fine-tuned a model for five epochs and tested it to see how each epoch affected the final result.

In Table 5.4, we have the results of fine-tuning BioLinkBERT-Large on the SIGIR-CT. We notice that after the second epoch in most cases the results would be around their peak and that the best results were obtained using the Not Relevant vs. All (2B) output format, the results are in fact interesting and lead to the assumption that they are due to the class imbalances the SIGIR-CT has. These models all had worse results than the original run, although the model BioLinkBERT-large_SIGIR-CT-2B was able to push most of the Eligible trials to the top 500 (R@500 similar to R@1000).

Table 5.5 shows the results of fine-tuning on SIGIR-CT$_{Balanced}$. Here we rapidly analyze that the Not Relevant vs. All (2B) output format has the worse results, which is to be expected as we are grouping Excluded and Eligible trials in the same class, this supports the above statement about the imbalances in the SIGIR-CT dataset. The best output formats are the Multiclass (3) and the Regression (R), these models stabilize their results around the fourth and fifth training epochs, although the evaluation metrics are very similar to the original run, the NDCG@10 dropped significantly, which means that our models do not distinguish very well Excluded trials from Eligible trials.

Table 5.4: Rerank results of the "BM25 + RM3 + RRF"run with the BioLinkBERT-Large fine-tuned on SIGIR-CT.

Run	P@10	NDCG@10	MRR	RPrec	R@10	R@100	R@500	R@1000
BioLinkBERT-large_SIGIR-CT-2A								
Epoch 1	0.0605	0.0731	0.1153	0.0592	0.0102	0.0918	0.3863	0.6608
Epoch 2	0.0684	0.0931	0.1400	0.0667	0.0134	0.1168	0.4020	0.6608
Epoch 3	0.0553	0.0786	0.1777	0.0612	0.0134	0.0871	0.3511	0.6608
Epoch 4	0.0316	0.0506	0.1046	0.0326	0.0056	0.0475	0.2747	0.6608
Epoch 5	0.0763	0.1076	0.1894	0.0636	0.0140	0.1119	0.3806	0.6608
BioLinkBERT-large_SIGIR-CT-2B								
Epoch 1	0.1474	0.1814	0.3120	0.1135	0.0286	0.1668	0.4543	0.6608
Epoch 2	0.1921	**0.2327**	0.3548	0.1627	0.0404	0.2723	0.5835	0.6608
Epoch 3	0.1737	0.2250	**0.3929**	**0.1704**	0.0420	0.2940	0.5985	0.6608
Epoch 4	0.1684	0.2121	0.3111	0.1571	0.0352	0.2842	0.6006	0.6608
Epoch 5	**0.1974**	0.2305	0.3093	0.1684	**0.0466**	**0.3015**	**0.6058**	0.6608
BioLinkBERT-large_SIGIR-CT-3								
Epoch 1	0.1579	0.1845	0.3310	0.1254	0.0255	0.1944	0.5249	0.6608
Epoch 2	0.1816	0.1980	0.2696	0.1290	0.0343	0.2190	0.5490	0.6608
Epoch 3	0.1763	0.2038	0.2903	0.1419	0.0383	0.2493	0.5441	0.6608
Epoch 4	0.1474	0.1940	0.2940	0.1520	0.0347	0.2574	0.5526	0.6608
Epoch 5	0.1474	0.1948	0.2528	0.1538	0.0403	0.2597	0.5440	0.6608
BioLinkBERT-large_SIGIR-CT-R								
Epoch 1	0.0868	0.1293	0.2133	0.0551	0.0145	0.0715	0.3501	0.6608
Epoch 2	0.0395	0.0605	0.1048	0.0414	0.0063	0.0601	0.3083	0.6608
Epoch 3	0.0526	0.0715	0.1864	0.0510	0.0099	0.0746	0.3355	0.6608
Epoch 4	0.0684	0.0949	0.1624	0.0522	0.0111	0.0706	0.3393	0.6608
Epoch 5	0.0316	0.0590	0.1608	0.0368	0.0061	0.0597	0.3180	0.6608

Table 5.5: Rerank results of the "BM25 + RM3 + RRF"run with the BioLinkBERT-Large fine-tuned on SIGIR-CT$_{Balanced}$.

Run	P@10	NDCG@10	MRR	RPrec	R@10	R@100	R@500	R@1000
BioLinkBERT-large_SIGIR-CT-2A_B								
Epoch 1	0.1079	0.1434	0.3182	0.0844	0.0157	0.1487	0.4742	0.6608
Epoch 2	0.1211	0.1417	0.2173	0.0917	0.0164	0.1541	0.4762	0.6608
Epoch 3	0.1500	0.1802	0.3192	0.1126	0.0264	0.1476	0.4403	0.6608
Epoch 4	0.1158	0.1558	0.2784	0.1040	0.0233	0.1528	0.4350	0.6608
Epoch 5	0.1237	0.1582	0.2944	0.0932	0.0222	0.1479	0.4510	0.6608
BioLinkBERT-large_SIGIR-CT-2B_B								
Epoch 1	0.0500	0.0589	0.1643	0.0650	0.0156	0.1176	0.4060	0.6608
Epoch 2	0.0605	0.0812	0.1594	0.0492	0.0083	0.0739	0.3393	0.6608
Epoch 3	0.0447	0.0737	0.1511	0.0420	0.0055	0.0548	0.3171	0.6608
Epoch 4	0.0342	0.0641	0.1158	0.0376	0.0037	0.0601	0.3286	0.6608
Epoch 5	0.0368	0.0659	0.1427	0.0357	0.0040	0.0591	0.3397	0.6608
BioLinkBERT-large_SIGIR-CT-3_B								
Epoch 1	0.1895	0.2241	0.3297	0.1590	0.0415	0.2781	**0.5848**	0.6608
Epoch 2	0.1947	0.2303	0.4253	0.1479	0.0479	0.2220	0.4828	0.6608
Epoch 3	0.2105	0.2593	0.4361	0.1512	0.0457	0.2030	0.4025	0.6608
Epoch 4	0.2132	0.2731	**0.5315**	0.1505	0.0496	0.2153	0.4016	0.6608
Epoch 5	0.2105	0.2645	0.4895	0.1369	0.0399	0.2055	0.3728	0.6608
BioLinkBERT-large_SIGIR-CT-R_B								
Epoch 1	0.1816	0.2166	0.3273	0.1731	0.0444	0.2987	0.5802	0.6608
Epoch 2	0.2026	0.2568	0.4248	0.1585	0.0495	0.2901	0.5837	0.6608
Epoch 3	0.2079	0.2611	0.4150	0.1804	0.0558	0.2891	0.5651	0.6608
Epoch 4	0.2289	0.2801	0.4474	0.1855	0.0589	0.2973	0.5467	0.6608
Epoch 5	**0.2368**	**0.2915**	0.4739	**0.1899**	**0.0613**	**0.3169**	0.5767	0.6608

Table 5.6: Rerank results of the "BM25 + RM3 + RRF"run with the BioLinkBERT-Large fine-tuned on TREC-CT-Train.

Run	P@10	NDCG@10	MRR	RPrec	R@10	R@100	R@500	R@1000
BioLinkBERT-large_TREC-CT-Train-2A								
Epoch 1	0.0605	0.0785	0.1742	0.0437	0.0082	0.0731	0.3367	0.6608
Epoch 2	0.0474	0.0772	0.1853	0.0458	0.0078	0.0726	0.3252	0.6608
Epoch 3	0.0421	0.0698	0.0870	0.0455	0.0048	0.0654	0.3317	0.6608
Epoch 4	0.0605	0.0848	0.1562	0.0520	**0.0119**	0.0723	0.3416	0.6608
Epoch 5	0.0447	0.0637	0.1412	**0.0526**	0.0059	0.0794	0.3360	0.6608
BioLinkBERT-large_TREC-CT-Train-2B								
Epoch 1	0.0474	0.0790	0.1727	0.0433	0.0048	0.0700	0.3305	0.6608
Epoch 2	0.0474	0.0595	0.1488	0.0407	0.0055	0.0614	0.3401	0.6608
Epoch 3	0.0447	0.0611	0.1080	0.0420	0.0109	0.0773	0.3348	0.6608
Epoch 4	0.0421	0.0622	0.1503	0.0513	0.0070	**0.0821**	0.3427	0.6608
Epoch 5	0.0474	0.0702	0.1168	0.0380	0.0071	0.0628	0.3374	0.6608
BioLinkBERT-large_TREC-CT-Train-3								
Epoch 1	0.0526	0.0636	0.1203	0.0508	0.0085	0.0741	0.3380	0.6608
Epoch 2	0.0316	0.0540	0.1224	0.0423	0.0044	0.0684	0.3258	0.6608
Epoch 3	**0.0658**	**0.0978**	0.1958	0.0495	0.0106	0.0618	**0.3473**	0.6608
Epoch 4	0.0553	0.0741	0.1221	0.0440	0.0097	0.0498	0.3124	0.6608
Epoch 5	0.0368	0.0633	0.1103	0.0467	0.0067	0.0806	0.3466	0.6608
BioLinkBERT-large_TREC-CT-Train-R								
Epoch 1	0.0395	0.0744	0.1121	0.0430	0.0080	0.0779	0.3405	0.6608
Epoch 2	0.0421	0.0665	0.1325	0.0437	0.0053	0.0621	0.3241	0.6608
Epoch 3	0.0500	0.0718	0.1399	0.0450	0.0077	0.0754	0.3432	0.6608
Epoch 4	0.0395	0.0615	0.1026	0.0427	0.0071	0.0570	0.3368	0.6608
Epoch 5	0.0579	0.0964	**0.2079**	0.0440	0.0106	0.0615	0.3201	0.6608

Table 5.7: Rerank results of the "BM25 + RM3 + RRF"run with the BioLinkBERT-Large fine-tuned on TREC-CT-Train$_{Expanded}$.

Run	**P@10**	**NDCG@10**	**MRR**	**RPrec**	**R@10**	**R@100**	**R@500**	**R@1000**
BioLinkBERT-large_TREC-CT-Train-2A_E								
Epoch 1	0.0447	0.0749	0.1957	0.0519	0.0094	0.0715	0.3456	0.6608
Epoch 2	0.0316	0.0679	0.1105	0.0450	0.0062	0.0672	0.3317	0.6608
Epoch 3	0.0316	0.0568	0.1205	0.0455	0.0041	0.0692	0.3249	0.6608
Epoch 4	0.0395	0.0515	0.1199	**0.0748**	0.0038	0.1408	**0.4626**	0.6608
Epoch 5	0.0395	0.0515	0.1199	**0.0748**	0.0038	0.1408	**0.4626**	0.6608
BioLinkBERT-large_TREC-CT-Train-2B_E								
Epoch 1	0.0500	0.0729	**0.1964**	0.04700	**0.0118**	0.0641	0.3139	0.6608
Epoch 2	0.0421	0.0721	0.1447	0.0401	0.0068	0.0723	0.3236	0.6608
Epoch 3	0.0395	0.0676	0.1514	0.0441	0.0113	0.0610	0.3273	0.6608
Epoch 4	0.0421	0.0653	0.1007	0.0413	0.0053	0.0655	0.3433	0.6608
Epoch 5	0.0395	0.0634	0.0991	0.0438	0.0042	**0.0778**	0.3199	0.6608
BioLinkBERT-large_TREC-CT-Train-3_E								
Epoch 1	0.0368	0.0575	0.1004	0.0488	0.0040	0.0693	0.3361	0.6608
Epoch 2	0.0316	0.0615	0.1180	0.0423	0.0040	0.0641	0.3369	0.6608
Epoch 3	0.0474	0.0701	0.1283	0.0474	0.0057	0.0726	0.3414	0.6608
Epoch 4	0.0526	0.0865	0.1937	0.0491	0.0063	0.0616	0.3426	0.6608
Epoch 5	0.0368	0.0636	0.0840	0.0391	0.0037	0.0603	0.3483	0.6608
BioLinkBERT-large_TREC-CT-Train-R_E								
Epoch 1	0.0553	0.0819	0.1722	0.0403	0.0062	0.0746	0.3362	0.6608
Epoch 2	0.0158	0.0348	0.0889	0.0427	0.0027	0.0731	0.3235	0.6608
Epoch 3	0.0316	0.0655	0.1544	0.0416	0.0042	0.0653	0.3439	0.6608
Epoch 4	**0.0632**	**0.0922**	0.1694	0.0458	0.0071	0.0676	0.3449	0.6608
Epoch 5	0.0263	0.0498	0.1034	0.0395	0.0037	0.0583	0.3445	0.6608

Table 5.8: Rerank using fine-tuned cross-encoders on the SIGIR-CT$_{Balanced}$, with different fields as part of the input.

Input	**P@10**	**NDCG@10**	**MRR**	**RPrec**
BERT-Base				
brief_title	0.0507	0.0739	0.1534	0.0484
official_title	0.0387	0.0652	0.1469	0.0452
brief_summary	0.0947	0.1272	0.2472	0.0628
detailed_description	0.0400	0.0661	0.1285	0.0391
eligibility criteria	0.0427	0.0584	0.1384	0.0468
concatenation	0.0747	0.1078	0.1972	0.0625
Longformer-Base				
brief_title	0.0427	0.0685	0.1412	0.0423
official_title	0.1467	0.1774	0.2938	0.1082
brief_summary	0.0373	0.0571	0.1011	0.0429
detailed_description	0.1478	0.1846	0.3123	0.1079
eligibility criteria	0.1493	0.1930	0.3174	0.1089
concatenation	0.0907	0.1154	0.2177	0.0731
BioLinkBERT-Base				
brief_title	0.2160	0.2682	0.3760	0.1675
official_title	0.1627	0.1953	0.3367	0.1239
brief_summary	0.1627	0.1913	0.2904	0.1418
detailed_description	0.2040	0.2736	**0.4140**	**0.1844**
eligibility criteria	**0.2400**	**0.2987**	0.4027	0.1803
concatenation	0.1387	0.2070	0.3161	0.1370

Table 5.9: Rerank results of the "BM25 + RM3 + RRF"run. Using both bi-encoders and cross-encoders, at the end of the table, we also have the results of the post-rank filtering on the "BM25 + RM3 + RRF"run. The input column indicates which field or fields of the clinical trials were used as part of the input.

Model	**Input**	**P@10**	**NDCG@10**	**MRR**	**RPrec**
(1) BM25 + RM3 + RRF	-	0.2360	0.3817	0.3606	0.1654
Bi-encoders-Zero-shot					
(2) msmarco-bert-base-dot-v5	all	0.1893	0.3099	0.3578	0.1250
(3) all-distilroberta-v1	all	0.2493	0.3932	0.4956	0.1760
Bi-encoders-Fine-tuned					
(4) all-distilroberta-v1	all	0.2693	0.3945	0.4763	0.2066
Cross-encoders-Zero-shot					
(5) ms-marco-MiniLM-L-12-v2	all	0.0737	0.1327	0.2644	0.0593
(6) quora-roberta-large	all	0.0307	0.0627	0.1043	0.0400
(7) stsb-roberta-large	all	0.0493	0.0925	0.1546	0.0547
(8) longformer-base-4096-finetuned-squadv1	all	0.0347	0.0545	0.1080	0.0348
(9) longformer-base-plagiarism-detection	all	0.0587	0.0868	0.1913	0.0503
Cross-encoders-Fine-tuned					
(10) BERT-Base	criteria	0.0427	0.0584	0.1384	0.0468
(11) BERT-Large	criteria	0.0360	0.0637	0.1268	0.0405
(12) Longformer-Base	criteria	0.1493	0.1930	0.3174	0.1089
(13) BioLinkBERT-Base	criteria	0.2400	0.2987	0.4027	0.1803
(14) BioLinkBERT-Large	criteria	0.1627	0.2074	0.3022	0.1401
(15) PubMedBERT-Base-Abstract	criteria	0.1840	0.2356	0.3606	0.1451
(16) PubMedBERT-Base-Fulltext	criteria	0.0613	0.0903	0.1670	0.0563
BM25 + RM3 + RRF + Post-rank filtering					
(17) RegEx	-	0.2853	0.4172	0.4626	0.1988
(18) Clinical Assertion	-	0.2079	0.3276	0.3308	0.1483
(19) Sentiment Analysis	-	0.1789	0.3111	0.3257	0.1328

In Tables 5.6 and 5.7, we have the results of fine-tuning on TREC-CT-Train and TREC-CT-Train$_{Expanded}$. Sadly all of these results are really low and don't follow a pattern, we can not take any conclusion from this. Leading us to have more questions than answers, and not being able to explore the TREC-CT-Train split and make use of our shorted queries.

Following the above results, all models were now trained for five epochs on the SIGIR-CT$_{Balanced}$ dataset, using the Regression (R) output format. Using these models for reranking is simpler as we can use the output logit directly as a rank score.

In Table 5.8, we explore how each field impacts the final list when used as the model input. Here we have two general domain models (BERT-Large and Longformer-Base), a model which receives 4096 tokens as input (Longformer-Base), and a model pre-trained on the biomedical domain (BioLinkBERT-Base). For BERT-Large, the brief_summary field is actually the better performing one, the analysis of this result requires further investigation, as when using Longformer-Base it is the worse performing field, ruling out the impact of being pre-trained on general domain corpora. With Longformer-Base we have some fields that are good when using a model where they fully fit without being truncated, official_title, detailed_description, and criteria. The concatenation of all fields underperformed, which is an interesting result as in most cases this is not truncated and all information is fed to the model, with this we end up with some questions, maybe fully using the model input length is not an optimized way of using the model? Or maybe the window attention mechanism is not the best for this use case? To finalize the use of BioLinkBERT-Base was tested, a model pre-trained on the biomedical domain, here we notice again that the criteria field has most of the relevant information for the matching task. Sadly we were not able to find biomedical domain models with larger input lengths.

In Table 5.9 under the Cross-encoders-Fine-tuned, we have tested multiple models. The two models (11) and (14), are the Large version of models tested (10) and (13), here we look to see how our fine-tuning process affects larger models, the worse results may be to under-training, due to larger models needing more steps to converge. The last two models (15) and (16) are biomedical domain models, with slightly different pre-trained approaches, PubMedBERT-Base-Abstract is pretrained from scratch using abstracts from PubMed, while PubMedBERT-Base-Fulltext is pretrained from scratch using abstracts from PubMed and full-text articles from PubMedCentral, the difference in the scores is really drastic just by this change, leaving on more point for future research. One answer that may explain the result between the runs (15) and (16), is that being a model pre-trained in the biomedical domain is not enough to achieve good results, the language present in the documents used for pre-trained needs to be similar to the language used in the documents for fine-tuned, in our case, the model trained using abstracts achieves the best results, as the abstracts have a very similar language to of our documents and queries, which can not be said about the language present in full-text articles.

When comparing the fine-tuned models with the zero-shot, we notice that both the biomedical domain models and the Longformer outperform them.

To end the evaluation of the rerank stage, we take a look at the runs (1), (4), and (13). Here we notice that both the bi-encoder and the cross-encoder have reranked the original list to a better one, having obtained higher evaluation metrics. Notice that the cross-encoder obtained worse NDCG@10, leading to the conclusion that the model can not distinguish Excluded from Eligible clinical trials. The bi-encoder obtained the best results overall, which is a surprise because in the literature the cross-encoder constantly outperforms the bi-encoder.

5.1.3 Filtering

Here we tackled the task of distinguishing Excluded and Eligible trials, where we had a very simple approach using RegEx, and a really complex algorithm using NER, Clinical Assertion, or SA. In Table 5.9, we have the results of these methods when applied to the ranking list produced by BM25 + RM3 + RRF, its clear that both the methods (18) and (19) were very unsuccessful, although when comparing the use of Clinical Assertion versus the use of SA, it is clearly better to use Clinical Assertion, this NER method has so many variables that can be tested, becoming really hard to make any conclusion without further research. On the other hand, the use of our simple approach (17) where we create metadata about the demographics of the patient, which we then compare to the eligibility of the clinical trial, proved really successful, improving all metrics significantly even when compared to the tested dense models (4) and (13).

5.2 Discussion

Which clinical trial fields carries more information for the matching task? Both when using sparse methods or using dense methods, the *criteria* field carries the most needed information, which is supported by models using the *criteria* as input constantly obtaining better evaluation metrics.

What is the effectiveness of sparse retrieval versus dense retrieval? We have shown that both methods seem to be equally as effective, obtaining 60% recall when retrieving the top 1000 documents, but due to the difference in computational complexity between the two, we say that sparse retrieval is the better alternative.

What is the best output format to rerank with a 3-point scale? Our approach to the task where we need to rank documents that are judged on a 3-point scale, was of transforming it into the multiple four different output formats, the most successful models used the Multiclass (3) and Regression (R) output formats, we opted to use the Regression (R) method as slightly better results were achieved and its easy to use for rerank is facilitate as the output value is used as the rank score.

Is fine-tuning sparse retrieval methods effective? We have shown that with little effort, we can optimize the parameters of a method and obtain higher recall, which may become significant in later stages of the ranking pipeline.

Reranking with bi-encoders versus cross-encoders? Actually a very surprising result, as in most cases cross-encoders perform better, in our work bi-encoders obtained the best result. We might be able to attribute the result to the input length limitation of cross-encoders.

How does the 512 token input length impacts the use of cross-encoders? As we showed the topic occupies more than half of the input sequence, which does not leave enough space in the majority of cases, for the desired trial information to be put in the input sequence without being truncated.

Reranking with general domain models versus biomedical domain models? As expected, biomedical domain models outperformed general domain models, both in with zero-shot approach and with fine-tuning.

Is the post-rank filter worth it? We believe that adding this stage to the pipeline is beneficial when we can find a method capable of doing something a dense ranking method can not, like our RegEx method does with the demographic metadata.

6 Conclusions and Future Work

6.1 Conclusions

In this thesis, we studied and implemented the various components of an Information Retrieval system for document ranking, with the goal of being capable of matching patients to clinical trials.

Specifically, we tackled the ranking of large documents, both on the query side (patients' descriptions) and the document side (clinical trials XML files). A three-stage pipeline was proposed, which is composed of a retrieval stage, a reranking stage, and a filtering stage:

- For the retrieval stage, the clinical trials documents were indexed, following a small study of which fields had more impact on the evaluation metrics, using BM25 as the ranking method. Here we also explored the use of doc2query-T5 to expand our topics, and with newly created queries we used BM25 and RM3 to build various ranking lists, that were merged using RRF, building our retrieval method. This stage was then fine-tuned to optimize the recall.

- For the reranking phase, we explored models based and BERT (pre-trained in the biomedical domain), whilst exploring the Longformer to circumnavigate the lengthy queries and documents. We fine-tuned these models on the task, where we looked at how different output formats and training epochs affected the final results.

- In our filtering stage, we explored the use of the "eligibility" attribute, where we find demographic criteria and the field criteria. For the demographic criteria, we successfully built rules based on RegEx that filtered our lists, by penalizing the clinical trials that broke these rules. For the criteria field, we hypothesized an algorithm, where we explored the use of NER, SA, and Clinical Assertion.

In the last years, the Transformer revolutionized IR, bringing a surge to apply it to document ranking and to the biomedical domain. Having the opportunity to contribute to both of these fields was the main motivation.

6.2 Future Work

Our results have shown to be higher than the median results the TREC CT 2021 participants obtained, which we call a success. Throughout the work we found some answers to our questions and raised other questions, giving space for future work:

- We found, that for this task we need a model pretrained on the biomedical domain, with a higher input length than 512 tokens. These models usually use alternative attention mechanisms, we would call for the use of a global attention mechanism on the [CLS] token, as is indicated in 2.6.
- Another way to resolve the sequence length limitation, is to have smaller queries and documents. An interesting approach would be to explore the use of summarized topics and trials, by applying doc2query to both. As we failed to use our shorter queries.
- A more in-depth study to be able to understand why the bi-encoders seem to perform better in this task, when compared to cross-encoders.
- To finalize, we believe that there is some interest in exploring the use of the criteria field to make a distinction between Excluded and Eligible clinical trials. The exploration of the inclusion and exclusion rules might be beneficial.

Bibliography

[1] Q. Ai et al. “Learning Groupwise Multivariate Scoring Functions Using Deep Neural Networks”. In: *Proceedings of the 2019 ACM SIGIR International Conference on Theory of Information Retrieval*. ICTIR ’19. Santa Clara, CA, USA: Association for Computing Machinery, 2019, pp. 85–92. ISBN: 9781450368810. DOI: 10.1145/3341981.3344218. URL: https://doi.org/10.1145/3341981.3344218 (cit. on p. 9).

[2] B. van Aken et al. “Assertion Detection in Clinical Notes: Medical Language Models to the Rescue?” In: *Proceedings of the Second Workshop on Natural Language Processing for Medical Conversations*. Online: Association for Computational Linguistics, June 2021, pp. 35–40. DOI: 10.18653/v1/2021.nlpmc-1.5. URL: https://aclanthology.org/2021.nlpmc-1.5 (cit. on p. 46).

[3] Z. Akkalyoncu Yilmaz et al. “Applying BERT to Document Retrieval with Birch”. 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): System Demonstrations*. Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 19–24. DOI: 10.18653/v1/D19-3004. URL: https://aclanthology.org/D19-3004 (cit. on p. 21).

[4] E. Alsentzer et al. “Publicly Available Clinical BERT Embeddings”. In: *Proceedings of the 2nd Clinical Natural Language Processing Workshop*. Minneapolis, Minnesota, USA: Association for Computational Linguistics, June 2019, pp. 72–78. DOI: 10.18653/v1/W19-1909. URL: https://aclanthology.org/W19-1909 (cit. on p. 19).

[5] G. Amati and C. J. Van Rijsbergen. “Probabilistic Models of Information Retrieval Based on Measuring the Divergence from Randomness”. In: *ACM Trans. Inf. Syst.* 20.4 (Oct. 2002), pp. 357–389. ISSN: 1046-8188. DOI: 10.1145/582415.582416. URL: https://doi.org/10.1145/582415.582416 (cit. on p. 6).

[6] G. C. de Araújo. *Biomedical information extraction for matching patients to clinical trials*. 2018. URL: http://hdl.handle.net/10362/61552 (cit. on p. 2).

[7] R. A. Baeza-Yates and B. A. Ribeiro-Neto. *Modern Information Retrieval*. ACM Press / Addison-Wesley, 1999. ISBN: 0-201-39829-X. URL: http://www.ischool.berkeley.edu/~hearst/irbook/glossary.html (cit. on p. 6).

[8] D. Bahdanau, K. Cho, and Y. Bengio. *Neural Machine Translation by Jointly Learning to Align and Translate*. 2016. arXiv: 1409.0473 [cs.CL] (cit. on p. 10).

[9] I. Beltagy, A. Cohan, and K. Lo. "SciBERT: Pretrained Contextualized Embeddings for Scientific Text". In: *CoRR* abs/1903.10676 (2019). arXiv: 1903.10676. URL: http://arxiv.org/abs/1903.10676 (cit. on p. 21).

[10] I. Beltagy, M. E. Peters, and A. Cohan. "Longformer: The Long-Document Transformer". In: *CoRR* abs/2004.05150 (2020). arXiv: 2004.05150. URL: https://arxiv.org/abs/2004.05150 (cit. on pp. 19, 40).

[11] K. Cho et al. *Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation*. 2014. arXiv: 1406.1078 [cs.CL] (cit. on p. 10).

[12] A. Cohan et al. *SPECTER: Document-level Representation Learning using Citation-informed Transformers*. 2020. arXiv: 2004.07180 [cs.CL] (cit. on p. 19).

[13] N. Collier and J.-D. Kim. "Introduction to the Bio-entity Recognition Task at JNLPBA". In: *Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and its Applications (NLPBA/BioNLP)*. Geneva, Switzerland: COLING, Aug. 2004, pp. 73–78. URL: https://aclanthology.org/W04-1213 (cit. on p. 46).

[14] G. V. Cormack, C. L. A. Clarke, and S. Buettcher. "Reciprocal Rank Fusion Outperforms Condorcet and Individual Rank Learning Methods". In: *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR '09. Boston, MA, USA: Association for Computing Machinery, 2009, pp. 758–759. ISBN: 9781605584836. DOI: 10.1145/1571941.1572114. URL: https://doi.org/10.1145/1571941.1572114 (cit. on p. 34).

[15] J. da Costa Pereira. *Neural Retrieval Models for Matching Patients to Clinical Trials*. 2022 (cit. on pp. 36, 39, 49).

[16] N. Craswell et al. "MS MARCO: Benchmarking Ranking Models in the Large-Data Regime". In: *CoRR* abs/2105.04021 (2021). arXiv: 2105.04021. URL: https://arxiv.org/abs/2105.04021 (cit. on p. 13).

[17] J. Datta and P. Bhattacharyya. "Ranking in Information Retrieval". In: (2010). URL: https://www.cse.iitb.ac.in/archive/internal/techreports/reports/TR-CSE-2010-31.pdf (cit. on p. 6).

[18] J. Devlin et al. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. 2019. arXiv: 1810.04805 [cs.CL] (cit. on pp. 11, 14, 37, 40).

[19] R. I. Doğan, R. Leaman, and Z. Lu. "NCBI disease corpus: a resource for disease name recognition and concept normalization". en. In: *J. Biomed. Inform.* 47 (Feb. 2014), pp. 1–10 (cit. on p. 46).

[20] H. E et al. "Data Resource Profile: Clinical Practice Research Datalink (CPRD)". In: *International journal of epidemiology* 44.3 (June 2015), pp. 827–836. ISSN: 1464-3685. DOI: 10.1093/IJE/DYV098. URL: https://pubmed.ncbi.nlm.nih.gov/26050254/ (cit. on p. 23).

[21] F. Feng et al. *Language-agnostic BERT Sentence Embedding*. 2020. arXiv: 2007.01852 [cs.CL] (cit. on p. 19).

[22] L. Gao, Z. Dai, and J. Callan. "COIL: Revisit Exact Lexical Match in Information Retrieval with Contextualized Inverted List". In: *CoRR* abs/2104.07186 (2021). arXiv: 2104.07186. URL: https://arxiv.org/abs/2104.07186 (cit. on p. 14).

[23] *Glossary of Deep Learning: Word Embedding*. URL: https://medium.com/deeper-learning/glossary-of-deep-learning-word-embedding-f90c3cec34ca (visited on 01/17/2022) (cit. on p. 10).

[24] Y. Gu et al. "Domain-Specific Language Model Pretraining for Biomedical Natural Language Processing". In: *CoRR* abs/2007.15779 (2020). arXiv: 2007.15779. URL: https://arxiv.org/abs/2007.15779 (cit. on pp. 13, 21, 22, 45).

[25] K. Huang, J. Altosaar, and R. Ranganath. "ClinicalBERT: Modeling Clinical Notes and Predicting Hospital Readmission". In: *CoRR* abs/1904.05342 (2019). arXiv: 1904.05342. URL: http://arxiv.org/abs/1904.05342 (cit. on p. 22).

[26] S. Humeau et al. "Real-time Inference in Multi-sentence Tasks with Deep Pretrained Transformers". In: *CoRR* abs/1905.01969 (2019). arXiv: 1905.01969. URL: http://arxiv.org/abs/1905.01969 (cit. on p. 15).

[27] Y. Jernite, S. R. Bowman, and D. A. Sontag. "Discourse-Based Objectives for Fast Unsupervised Sentence Representation Learning". In: *CoRR* abs/1705.00557 (2017). arXiv: 1705.00557. URL: http://arxiv.org/abs/1705.00557 (cit. on p. 14).

[28] Q. Jin et al. "Aliababa DAMO Academy at TREC Precision Medicine 2020: State-of-the-art Evidence Retriever for Precision Medicine with Expert-in-the-loop Active Learning". In: (2020) (cit. on p. 17).

[29] A. E. Johnson et al. "MIMIC-III, a freely accessible critical care database". In: *Scientific Data* 3.1 (May 2016), p. 160035. ISSN: 2052-4463. DOI: 10.1038/sdata.2016.35. URL: https://doi.org/10.1038/sdata.2016.35 (cit. on p. 22).

[30] J. Johnson, M. Douze, and H. Jégou. *Billion-scale similarity search with GPUs*. 2017. DOI: 10.48550/ARXIV.1702.08734. URL: https://arxiv.org/abs/1702.08734 (cit. on p. 36).

[31] J. Koontz, M. Oronoz, and A. Pérez. "TREC 2021 Clinical Trials Submission for Universidad del País Vasco". In: (2021) (cit. on p. 19).

[32] 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*. SIGIR '16. Pisa, Italy: Association for Computing Machinery, 2016, pp. 669–672. ISBN: 9781450340694. DOI: 10.1145/2911451.2914672. URL: https://doi.org/10.1145/2911451.2914672 (cit. on pp. 2, 18, 28).

[33] M. Köppel et al. "Pairwise Learning to Rank by Neural Networks Revisited: Reconstruction, Theoretical Analysis and Practical Performance". In: *CoRR* abs/1909.02768 (2019). arXiv: 1909.02768. URL: http://arxiv.org/abs/1909.02768 (cit. on p. 8).

[34] T. Kudo and J. Richardson. *SentencePiece: A simple and language independent subword tokenizer and detokenizer for Neural Text Processing*. 2018. arXiv: 1808.06226 [cs.CL] (cit. on p. 12).

[35] J. Lafferty and C. Zhai. "Document Language Models, Query Models, and Risk Minimization for Information Retrieval". In: *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR '01. New Orleans, Louisiana, USA: Association for Computing Machinery, 2001, pp. 111–119. ISBN: 1581133316. DOI: 10.1145/383952.383970. URL: https://doi.org/10.1145/383952.383970 (cit. on p. 6).

[36] V. Lavrenko and W. B. Croft. "Relevance Based Language Models". In: *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR '01. New Orleans, Louisiana, USA: Association for Computing Machinery, 2001, pp. 120–127. ISBN: 1581133316. DOI: 10.1145/383952.383972. URL: https://doi.org/10.1145/383952.383972 (cit. on pp. 6, 33).

[37] Q. V. Le and T. Mikolov. *Distributed Representations of Sentences and Documents*. 2014. DOI: 10.48550/ARXIV.1405.4053. URL: https://arxiv.org/abs/1405.4053 (cit. on p. 10).

[38] J. Lee et al. "BioBERT: a pre-trained biomedical language representation model for biomedical text mining". In: *CoRR* abs/1901.08746 (2019). arXiv: 1901.08746. URL: http://arxiv.org/abs/1901.08746 (cit. on pp. 17, 22).

[39] M. Lewis et al. *BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension*. 2019. arXiv: 1910.13461 [cs.CL] (cit. on p. 19).

[40] J. Li et al. "BioCreative V CDR task corpus: a resource for chemical disease relation extraction". en. In: *Database (Oxford)* 2016 (May 2016), baw068 (cit. on pp. 45, 46).

[41] Y. Li et al. “BEHRT: Transformer for Electronic Health Records”. In: *Scientific Reports* 10.1 (Apr. 2020), p. 7155. ISSN: 2045-2322. DOI: 10.1038/s41598-020-62922-y. URL: https://doi.org/10.1038/s41598-020-62922-y (cit. on p. 23).

[42] J. Lin, R. Nogueira, and A. Yates. *Pretrained Transformers for Text Ranking: BERT and Beyond*. 2021. arXiv: 2010.06467 [cs.IR] (cit. on pp. 9, 21).

[43] J. Lin et al. “New Nails for Old Hammers: Anserini and Pyserini at TREC 2021”. In: (2021) (cit. on pp. 17, 34).

[44] J. Lin et al. *Pyserini: An Easy-to-Use Python Toolkit to Support Replicable IR Research with Sparse and Dense Representations*. 2021. arXiv: 2102.10073 [cs.IR] (cit. on p. 32).

[45] T.-Y. Liu. “Learning to Rank for Information Retrieval”. In: *Foundations and Trends® in Information Retrieval* 3.3 (2009), pp. 225–331. ISSN: 1554-0669. DOI: 10.1561/1500000016. URL: http://dx.doi.org/10.1561/1500000016 (cit. on pp. 7, 8).

[46] W. Liu et al. “Personalized Re-Ranking with Item Relationships for E-Commerce”. In: *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. CIKM ’20. Virtual Event, Ireland: Association for Computing Machinery, 2020, pp. 925–934. ISBN: 9781450368599. DOI: 10.1145/3340531.3412332. URL: https://doi.org/10.1145/3340531.3412332 (cit. on p. 8).

[47] X. Liu et al. *Clinical Trial Information Extraction with BERT*. 2021. arXiv: 2110.10027 [q-bio.QM] (cit. on pp. 21, 23).

[48] Y. Liu et al. *RoBERTa: A Robustly Optimized BERT Pretraining Approach*. 2019. DOI: 10.48550/ARXIV.1907.11692. URL: https://arxiv.org/abs/1907.11692 (cit. on p. 36).

[49] L. Logeswaran and H. Lee. “An efficient framework for learning sentence representations”. In: *International Conference on Learning Representations*. 2018. URL: https://openreview.net/forum?id=rJvJXZb0W (cit. on p. 14).

[50] J. M. Lourenço. *The NOVAthesis LaTeX Template User’s Manual*. NOVA University Lisbon. 2021. URL: https://github.com/joaomlourenco/novathesis/raw/master/template.pdf (cit. on p. iii).

[51] M.-T. Luong, H. Pham, and C. D. Manning. *Effective Approaches to Attention-based Neural Machine Translation*. 2015. arXiv: 1508.04025 [cs.CL] (cit. on p. 10).

[52] Y. Lv and C. Zhai. “Lower-Bounding Term Frequency Normalization”. In: *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*. CIKM ’11. Glasgow, Scotland, UK: Association for Computing Machinery, 2011, pp. 7–16. ISBN: 9781450307178. DOI: 10.1145/2063576.2063584. URL: https://doi.org/10.1145/2063576.2063584 (cit. on p. 7).

[53] S. MacAvaney, A. Cohan, and N. Goharian. "SLEDGE: A Simple Yet Effective Baseline for Coronavirus Scientific Knowledge Search". In: *CoRR* abs/2005.02365 (2020). arXiv: 2005.02365. URL: https://arxiv.org/abs/2005.02365 (cit. on p. 18).

[54] J. Mackenzie et al. "Query Driven Algorithm Selection in Early Stage Retrieval". In: *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*. WSDM '18. Marina Del Rey, CA, USA: Association for Computing Machinery, 2018, pp. 396–404. ISBN: 9781450355810. DOI: 10.1145/3159652.3159676. URL: https://doi.org/10.1145/3159652.3159676 (cit. on p. 9).

[55] I. Matveeva et al. "High Accuracy Retrieval with Multiple Nested Ranker". In: *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR '06. Seattle, Washington, USA: Association for Computing Machinery, 2006, pp. 437–444. ISBN: 1595933697. DOI: 10.1145/1148170.1148246. URL: https://doi.org/10.1145/1148170.1148246 (cit. on p. 9).

[56] T. Mikolov et al. *Efficient Estimation of Word Representations in Vector Space*. 2013. arXiv: 1301.3781 [cs.CL] (cit. on p. 10).

[57] M. Montague and J. A. Aslam. "Condorcet Fusion for Improved Retrieval". In: *Proceedings of the Eleventh International Conference on Information and Knowledge Management*. CIKM '02. McLean, Virginia, USA: Association for Computing Machinery, 2002, pp. 538–548. ISBN: 1581134924. DOI: 10.1145/584792.584881. URL: https://doi.org/10.1145/584792.584881 (cit. on p. 34).

[58] T. Nguyen et al. "MS MARCO: A Human Generated MAchine Reading COmprehension Dataset". In: *CoRR* abs/1611.09268 (2016). arXiv: 1611.09268. URL: http://arxiv.org/abs/1611.09268 (cit. on p. 16).

[59] R. Nogueira, Z. Jiang, and J. Lin. *Document Ranking with a Pretrained Sequence-to-Sequence Model*. 2020. arXiv: 2003.06713 [cs.IR] (cit. on pp. 16, 18).

[60] R. Nogueira and J. Lin. "From doc2query to docTTTTTquery". In: (2019). URL: https://cs.uwaterloo.ca/~jimmylin/publications/Nogueira_Lin_2019_docTTTTTquery-v2.pdf (cit. on pp. 16, 17, 34).

[61] R. Nogueira et al. "Multi-Stage Document Ranking with BERT". In: *CoRR* abs/1910.14424 (2019). arXiv: 1910.14424. URL: http://arxiv.org/abs/1910.14424 (cit. on pp. 9, 10, 14, 15).

[62] G. M. D. Nunzio, G. Faggioli, and S. Marchesin. "Filter, Transform, Expand, and Fuse The IMS Unipd at TREC 2021 Clinical Trials". In: (2021) (cit. on p. 18).

[63] B. E. Nye et al. “A Corpus with Multi-Level Annotations of Patients, Interventions and Outcomes to Support Language Processing for Medical Literature”. In: *CoRR* abs/1806.04185 (2018). arXiv: 1806.04185. URL: http://arxiv.org/abs/1806.04185 (cit. on p. 21).

[64] B. E. Nye et al. *Understanding Clinical Trial Reports: Extracting Medical Entities and Their Relations*. 2020. arXiv: 2010.03550 [cs.CL] (cit. on p. 21).

[65] L. Pang et al. “SetRank: Learning a Permutation-Invariant Ranking Model for Information Retrieval”. In: *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. New York, NY, USA: Association for Computing Machinery, 2020, pp. 499–508. ISBN: 9781450380164. URL: https://doi.org/10.1145/3397271.3401104 (cit. on p. 9).

[66] Y. Peng, S. Yan, and Z. Lu. “Transfer Learning in Biomedical Natural Language Processing: An Evaluation of BERT and ELMo on Ten Benchmarking Datasets”. In: *CoRR* abs/1906.05474 (2019). arXiv: 1906.05474. URL: http://arxiv.org/abs/1906.05474 (cit. on p. 23).

[67] J. Pennington, R. Socher, and C. Manning. “GloVe: Global Vectors for Word Representation”. In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar: Association for Computational Linguistics, Oct. 2014, pp. 1532–1543. DOI: 10.3115/v1/D14-1162. URL: https://aclanthology.org/D14-1162 (cit. on p. 10).

[68] R. Pradeep, R. Nogueira, and J. Lin. “The Expando-Mono-Duo Design Pattern for Text Ranking with Pretrained Sequence-to-Sequence Models”. In: *CoRR* abs/2101.05667 (2021). arXiv: 2101.05667. URL: https://arxiv.org/abs/2101.05667 (cit. on p. 18).

[69] R. Pradeep et al. “H2oloo at TREC 2020: When all you got is a hammer... Deep Learning, Health Misinformation, and Precision Medicine”. In: (2020) (cit. on p. 17).

[70] C. Raffel et al. *Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer*. 2020. arXiv: 1910.10683 [cs.LG] (cit. on pp. 12, 13, 17).

[71] S. Rajbhandari et al. “ZeRO: Memory Optimization Towards Training A Trillion Parameter Models”. In: *CoRR* abs/1910.02054 (2019). arXiv: 1910.02054. URL: http://arxiv.org/abs/1910.02054 (cit. on p. 35).

[72] N. Reimers and I. Gurevych. *Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks*. 2019. arXiv: 1908.10084 [cs.CL] (cit. on pp. 10, 15, 16).

[73] K. Roberts et al. *Overview of the TREC 2015 Clinical Decision Support Track*. URL: https://trec.nist.gov/pubs/trec24/papers/Overview-CL.pdf (cit. on pp. 1, 29).

[74] K. Roberts et al. *Overview of the TREC 2016 Clinical Decision Support Track*. URL: https://trec.nist.gov/pubs/trec25/papers/Overview-CL.pdf (cit. on p. 1).

[75] K. Roberts et al. *Overview of the TREC 2017 Precision Medicine Track*. URL: https://trec.nist.gov/pubs/trec26/papers/Overview-PM.pdf (cit. on p. 1).

[76] K. Roberts et al. *Overview of the TREC 2018 Precision Medicine Track*. URL: https://trec.nist.gov/pubs/trec27/papers/Overview-PM.pdf (cit. on p. 1).

[77] K. Roberts et al. *Overview of the TREC 2019 Precision Medicine Track*. URL: https://trec.nist.gov/pubs/trec28/papers/OVERVIEW.PM.pdf (cit. on p. 1).

[78] K. Roberts et al. *Overview of the TREC 2020 Precision Medicine Track*. URL: https://trec.nist.gov/pubs/trec29/papers/OVERVIEW.PM.pdf (cit. on p. 17).

[79] K. Roberts et al. *Overview of the TREC 2021 Clinical Trials Track*. URL: https://trec.nist.gov/act_part/conference/papers/Overview-CT.pdf (cit. on pp. 1, 17, 20, 30).

[80] S. E. Robertson. "The Probability Ranking Principle in IR". In: *Journal of Documentation* 33.4 (1977), pp. 294–304. ISSN: 00220418. DOI: 10.1108/EB026647/FULL/XML. URL: https://www.emerald.com/insight/content/doi/10.1108/eb026647/full/html (cit. on p. 6).

[81] S. Robertson and H. Zaragoza. "The Probabilistic Relevance Framework: BM25 and Beyond". In: *Found. Trends Inf. Retr.* 3.4 (Apr. 2009), pp. 333–389. ISSN: 1554-0669. DOI: 10.1561/1500000019. URL: https://doi.org/10.1561/1500000019 (cit. on p. 7).

[82] M. Rybinski and S. Karimi. "CSIROmed at TREC Precision Medicine 2020". In: (2020) (cit. on p. 17).

[83] E. F. T. K. Sang and F. D. Meulder. *Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition*. 2003. arXiv: cs/0306050 [cs.CL] (cit. on p. 19).

[84] M. S. Simpson, E. M. Voorhees, and W. Hersh. *Overview of the TREC 2014 Clinical Decision Support Track*. URL: https://trec.nist.gov/pubs/trec23/papers/overview-clinical.pdf (cit. on pp. 1, 29).

[85] L. Smith et al. "Overview of BioCreative II gene mention recognition". In: *Genome Biology* 9.2 (Sept. 2008), S2. ISSN: 1474-760X. DOI: 10.1186/gb-2008-9-s2-s2. URL: https://doi.org/10.1186/gb-2008-9-s2-s2 (cit. on p. 45).

[86] M. Sung et al. "BERN2: an advanced neural biomedical namedentity recognition and normalization tool". In: (2022). arXiv: 2201.02080 [cs.CL] (cit. on p. 45).

[87] I. Sutskever, O. Vinyals, and Q. V. Le. *Sequence to Sequence Learning with Neural Networks*. 2014. arXiv: 1409.3215 [cs.CL] (cit. on p. 10).

[88] R. M. E. Swezey et al. “PiRank: Learning To Rank via Differentiable Sorting”. In: *CoRR* abs/2012.06731 (2020). arXiv: 2012.06731. URL: https://arxiv.org/abs/2012.06731 (cit. on p. 9).

[89] *The Annotated Transformer*. URL: http://nlp.seas.harvard.edu/2018/04/03/attention.html (visited on 01/19/2022) (cit. on p. 11).

[90] *The Illustrated Transformer – Jay Alammar – Visualizing machine learning one concept at a time*. URL: https://jalammar.github.io/illustrated-transformer/ (visited on 10/24/2021) (cit. on p. 11).

[91] Y. Tseo et al. “Information Extraction of Clinical Trial Eligibility Criteria”. In: *CoRR* abs/2006.07296 (2020). arXiv: 2006.07296. URL: https://arxiv.org/abs/2006.07296 (cit. on pp. 21, 23).

[92] Ö. Uzuner et al. “2010 i2b2/VA challenge on concepts, assertions, and relations in clinical text”. In: *J Am Med Inform Assoc* 18.5 (2011). [PubMed Central:PMC3168320] [DOI:10.1136/amiajnl-2011-000203] [PubMed:20819854], pp. 552–556 (cit. on p. 19).

[93] C. Van Gysel and M. de Rijke. “Pytrec_eval”. In: *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval* (June 2018). DOI: 10.1145/3209978.3210065. URL: http://dx.doi.org/10.1145/3209978.3210065 (cit. on p. 48).

[94] A. Vaswani et al. *Attention Is All You Need*. 2017. arXiv: 1706.03762 [cs.CL] (cit. on pp. 2, 11, 12).

[95] E. M. Voorhees and W. Hersh. “Overview of the TREC 2012 Medical Records Track”. In: (). URL: https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=913781 (cit. on p. 1).

[96] E. M. Voorhees. “Overview of the TREC 2004 Robust Retrieval Track”. In: (2004). URL: https://trec.nist.gov/pubs/trec13/papers/ROBUST.OVERVIEW.pdf (cit. on p. 16).

[97] *What Are Word Embeddings for Text?* URL: https://machinelearningmastery.com/what-are-word-embeddings/ (visited on 01/17/2022) (cit. on p. 10).

[98] H. WR. “Adding value to the electronic health record through secondary use of data for quality assurance, research, and surveillance”. In: *The American journal of managed care* 13.6 Part 1 (June 2007), pp. 277–278. ISSN: 1936-2692. URL: https://pubmed.ncbi.nlm.nih.gov/17567224/ (cit. on p. 1).

[99] Y. Wu et al. *Google’s Neural Machine Translation System: Bridging the Gap between Human and Machine Translation*. 2016. arXiv: 1609.08144 [cs.CL] (cit. on p. 12).

[100] P. Yang, H. Fang, and J. Lin. "Anserini: Enabling the Use of Lucene for Information Retrieval Research". In: *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR '17. Shinjuku, Tokyo, Japan: Association for Computing Machinery, 2017, pp. 1253–1256. ISBN: 9781450350228. DOI: 10.1145/3077136.3080721. URL: https://doi.org/10.1145/3077136.3080721 (cit. on p. 32).

[101] P. Yang, H. Fang, and J. Lin. "Anserini: Reproducible Ranking Baselines Using Lucene". In: *J. Data and Information Quality* 10.4 (Oct. 2018). ISSN: 1936-1955. DOI: 10.1145/3239571. URL: https://doi.org/10.1145/3239571 (cit. on p. 32).

[102] M. Yasunaga, J. Leskovec, and P. Liang. *LinkBERT: Pretraining Language Models with Document Links*. 2022. DOI: 10.48550/ARXIV.2203.15827. URL: https://arxiv.org/abs/2203.15827 (cit. on p. 22).

[103] M. Yasunaga, J. Leskovec, and P. Liang. *LinkBERT: Pretraining Language Models with Document Links*. 2022. DOI: 10.48550/ARXIV.2203.15827. URL: https://arxiv.org/abs/2203.15827 (cit. on p. 45).

[104] M. Zaheer et al. "Big Bird: Transformers for Longer Sequences". In: *CoRR* abs/2007.14062 (2020). arXiv: 2007.14062. URL: https://arxiv.org/abs/2007.14062 (cit. on pp. 19, 20).

[105] J. Zobel, A. Moffat, and K. Ramamohanarao. "Inverted Files versus Signature Files for Text Indexing". In: *ACM Trans. Database Syst.* 23.4 (Dec. 1998), pp. 453–490. ISSN: 0362-5915. DOI: 10.1145/296854.277632. URL: https://doi.org/10.1145/296854.277632 (cit. on p. 30).