Neural Query Synthesis and Domain-Specific Ranking Templates for Multi-Stage Clinical Trial Matching

Ronak Pradeep
Yilin Li
Yuetong Wang
Jimmy Lin
David R. Cheriton School of Computer Science,
University of Waterloo
Waterloo, Ontario, Canada

ABSTRACT

In this work, we propose an effective multi-stage neural ranking system for the clinical trial matching problem. First, we introduce NQS, a neural query synthesis method that leverages a zero-shot document expansion model to generate multiple sentence-long queries from lengthy patient descriptions. These queries are independently issued to a search engine and the results are fused. We find that on the TREC 2021 Clinical Trials Track, this method outperforms strong traditional baselines like BM25 and BM25 + RM3 by about 12 points in nDCG@10, a relative improvement of 34%. This simple method is so effective that even a state-of-the-art neural *relevance* ranking method trained on the medical subset of MS MARCO passage, when reranking the results of NQS, fails to improve on the ranked list. Second, we introduce a two-stage neural reranking pipeline trained on clinical trial matching data using tailored ranking templates. In this setting, we can train a pointwise reranker using just 1.1k positive examples and obtain effectiveness improvements over NQS by 24 points. This end-to-end multi-stage system demonstrates a 20% relative effectiveness gain compared to the second-best submission at TREC 2021, making it an important step towards better automated clinical trial matching.

CCS CONCEPTS

• **Information systems → Retrieval models and ranking**.

KEYWORDS

sequence-to-sequence models, TREC

ACM Reference Format:
Ronak Pradeep, Yilin Li, Yuetong Wang, and Jimmy Lin. 2022. Neural Query Synthesis and Domain-Specific Ranking Templates for Multi-Stage Clinical Trial Matching. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '22), July 11–15, 2022, Madrid, Spain.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3477495.3531853

SIGIR '22, July 11–15, 2022, Madrid, Spain

ACM ISBN 978-1-4503-8732-3/22/07…$15.00
https://doi.org/10.1145/3477495.3531853

1 INTRODUCTION

Clinical trials are vital to the validity of scientific research and form the foundation on which we measure progress in medicine. However, recruiting at least the minimum number of patients for these trials within time constraints is a challenging task and a key reason why most fail to kick off. This has substantial negative consequences both for the advancement of new treatments and for the patients who miss out on the potential health benefits from such clinical interventions.

The clinical trial matching problem we consider can be described as such: Given a patient and the patient's electronic health record (EHR) as the "query" and a collection of actively recruiting clinical trials, return those that the patient is eligible for. This is where the information retrieval community can play a role. In the TREC 2021 Clinical Trials Track [14], participants build systems that match patient case descriptions to eligible clinical trials. Such systems can be of significant assistance to both parties.

The contribution of our work is an effective multi-stage ranking system for the clinical trial matching task. More specifically:

(1) We attempt to handle challenges caused by long EHR description topics. We propose NQS, a neural query synthesis method that leverages doc2query–T5 [9] trained on the MS MARCO V2 passage ranking test collection to generate multiple single-sentence long queries. Then, we independently issue these queries using either BM25 or BM25 + RM3 as the ranking function and fuse the results using RRF [1] to produce our first-stage candidate list. This model demonstrates a relative score improvement of over 33% compared to the baseline of simply issuing the entire EHR description as the query.

(2) We evaluate the effectiveness of a powerful neural sequence-to-sequence ranking model, called "Med-Mono-T5" [12], that is fine-tuned on the clinical trial matching dataset created by Koopman and Zuccon [4]. To alleviate issues caused by a scarcity of training data, we use a ranking template that is specifically designed for matching clinical trials. In particular, our template includes segments from the fields "title", "condition", "eligibility", and "description" in the clinical trial. Our model is used to select the best eligibility segment and the best description segment of each candidate trial retrieved by our first-stage method. After this, the same ranker scores the trial based on a new multi-field combination of both these segments. This two-step method helps avoid the quadratic inference costs of enumerating all

Patient Description - #23

A 39-year-old man came to the clinic with cough and shortness of breath that was not relieved by his inhaler. He had these symptoms for 5 days during the past 2 weeks. He doubled his oral corticosteroids in the past week. He is a chef with a history of asthma for 3 years, suffering from frequent cough, wheezing, and shortness of breath and chest tightness. The symptoms become more bothersome within 1-2 hours of starting work every day and worsen throughout the work week. His symptoms improve within 1-2 hours outside the workplace. Spirometry was performed revealing a forced expiratory volume in the first second (FEV1) of 63% of the predicted. His past medical history is significant for seasonal allergic rhinitis in the summer. He doesn't smoke or use illicit drugs. His family history is significant for asthma in his father and sister. He currently uses inhaled corticosteroid (ICS) and fluticasone 500 mcg/salmeterol 50 mcg, one puff twice daily.

Clinical Trial

Title: Salmeterol/Fluticasone Easyhaler in the Treatment of Asthma and COPD

Eligibility:

Main Inclusion Criteria:

- Male or female patients with asthma or COPD who have been using salmeterol/fluticasone propionate combination treatment for at least 3 months before the study
- Age $\geq$ 18 years.
- Written informed consent obtained.

Main Exclusion Criteria:

- Pregnant or lactating female patients.
- Participation in other clinical studies during the study.
- Known hypersensitivity (allergy) to salmeterol, fluticasone propionate or the excipient lactose.

Condition: N/A

Description: A prospective, open-label, non-interventional, multicentre study in adult patients with asthma or COPD who are treated with Salmeterol/fluticasone Easyhaler. During the study the Salmeterol/fluticasone Easyhaler will be used according to the Summary of Product Characteristics. Clinical effectiveness of the treatment will be evaluated with change in asthma or COPD symptoms during 12 weeks treatment.

Table 1: Example of a patient's EHR description and a relevant clinical trial from the TREC 2021 Clinical Trials Track.

eligibility and description segment pairs and feeding them to the computationally expensive neural ranker.

This multi-stage neural ranking architecture that leverages NQS for first-stage retrieval was the best automatic system at the TREC 2021 Clinical Trials Track [14] by at least a 20% relative improvement over submissions from other teams in terms of the primary metrics.

2 TASK DESCRIPTION AND DATA

In the Clinical Trials Track [14] at TREC 2021, participants were required to retrieve eligible clinical trial descriptions given lengthy topics (5–10 sentences) comprising patient case descriptions taken from EHRs. System outputs were then graded on a 3-level scale: *eligible* if the patient is eligible for the trial based on the inclusion/exclusion criteria, *excluded* if the clinical trial is relevant to the patient but is excluded, and *non-relevant* otherwise.

The clinical trial matching task at TREC comprised a corpus of 375,580 clinical trials and 75 patient notes forming the topics. Each clinical trial has a title, condition, summary, description, and eligibility field. Here, the eligibility field holds the inclusion/exclusion criteria, a core aspect of the clinical trial matching task. Table 1 provides an example of a patient's EHR description (top) and excerpts from a relevant clinical trial (bottom). We can see that treating the EHR as a bag-of-words query may be problematic: while it provides vital information, the EHR also contains many unimportant terms.

Most similar to the target task is the work by Koopman and Zuccon [4]. The collection they created has 204,855 clinical trials forming the corpus, 60 patient notes providing the topics, and 3870 relevance judgment labels. The labels represent a three-point scale for relevance judgments: 0 if the patient is not referred to the clinical trial, 1 for possible referral, and 2 for highly possible referral. There are 2764 trials judged a 0, 685 trials a 1, and 421 trials a 2. We use this collection in our experiments as the training set. Since we only have 1106 positives, we work in a data-poor regime.

3 SYSTEM ARCHITECTURE

3.1 Neural Query Synthesis

Since IR researchers develop most retrieval methods with sentence-length (or shorter) queries in mind, the systems are not adept at tackling long patient descriptions that comprise the topics. These descriptions run between 5–10 sentences long. To address this limitation, we hypothesized that using doc2query–T5 [9], a neural document expansion technique, can help generate multiple single-sentence long queries for the longer patient description.

Nogueira and Lin [9] use a T5-base model trained on the MS MARCO V1 passage ranking task for document expansion to generate multiple queries for the entire corpus, comprising approximately 8.8 million passages. In this task, since we only need to generate multiple queries for 75 topics, and since the community has now moved on to the newer and cleaner MS MARCO V2 passage ranking test collection, we instead choose to use a T5-3B model trained on the newer dataset.

We train the model using the same general approach: given a passage, the task is to generate a query for which the passage is relevant. We use the same experimental setup, fine-tuning using a constant learning rate of $1 \cdot 10^{-3}$ and a batch size of 256. The model is fine-tuned for 4k iterations, corresponding to roughly four epochs with the MS MARCO V2 passage ranking training set. We use a maximum of 512 input tokens and 64 output tokens.

During inference, we utilize top-k sampling ($k = 10$) to generate the single-sentence queries for each topic patient description p. We vary the number of single-sentence queries we sample and whether or not we include the topic description to form a query set Q_p for a particular p. The method that includes the patient description is denoted NQS+PD, distinguished from NQS, which does not.

Given this query set Q_p, we issue each query independently to retrieve the top-1000 results based on a choice of scoring functions, either BM25 or BM25 + RM3. We accomplish this using the default parameters of the Pyserini IR Toolkit [5]. Reciprocal rank fusion [1] is then used to fuse the multiple ranked lists for each query in Q_p to give us the final ranked list, which is passed on to the neural ranking modules downstream. We believe that this step is essential to alleviate concerns from the model hallucinating unfaithful information [7] and being detrimental to retrieval effectiveness.

3.2 Zero-Shot T5-Based Relevance Ranking

Inspired by the success of T5 [13], where the researchers formulate *every* natural language processing task as feeding a sequence-to-sequence model some input text and training it to generate some output text, Nogueira et al. [8] proposed an adaptation for relevance ranking. Their approach is based on an input template to capture the pointwise ranking task:

$$\text{Query: } q \quad \text{Document: } d \quad \text{Relevant:} \tag{1}$$

where q and d are replaced with the query and the (candidate) document text, respectively. The target is one of the "true" or "false" tokens. In other words, ranking is formulated as feeding the model the query and a segment of text, and "asking" the model to predict relevance as an output token.

Given a model trained with this sequence input/output behavior, which Nogueira et al. [8] dub monoT5, at inference time, a softmax is applied to only the logits corresponding to the "true" and "false" tokens to extract meaningful probabilities. In other words, the model estimates $\Pr[\text{relevant} = 1|q, d]$ as the probability score assigned to the "true" token normalized in this manner. The candidates from a first-stage retriever are then reranked based on these scores. This model forms a vital backbone of various state-of-the-art systems for knowledge intensive tasks [3, 10–12]

In this work, our base ranker is a monoT5-3B model [8] fine-tuned on the MS MARCO V1 passage ranking test collection with a batch size of 128 and a learning rate of $1 \cdot 10^{-3}$ for 10k iterations and then fine-tuned again on Med-MARCO, which is a subset of the MS MARCO V1 passage ranking collection where only queries containing medical terms are retained [6]. For this second stage fine-tuning process, we train the model for another 1k iterations using a batch size of 128 and a learning rate of $1 \cdot 10^{-3}$. This model, which we denote $\text{monoT5}_{\text{MED}}$, was first described in Pradeep et al. [12] and was shown to be successful in biomedical retrieval tasks like TREC-COVID. This application of $\text{monoT5}_{\text{MED}}$ on the clinical trial matching task can be thought of as zero-shot.

Beginning here, we make careful choices trying to design the best domain-specific ranking template for the task of clinical trial matching. All the neural rankers use the entire patient description p as the query. Since we are optimizing for early precision here in the reranking stage, as opposed to recall in the NQS retrieval stage (i.e., first-stage retrieval), we believe that it is critical for the model to properly attend to every part of the patient description given the clinical trial to be scored.

Each clinical trial document has two lengthy multi-sentence fields, "eligibility" and "description", and two much smaller fields, "title" and "condition". Hence, during inference, we run sliding-window segmentation using $(n_{\text{length}}, n_{\text{stride}}) = (6, 3)$ on the eligibility and description fields independently and always make sure to include the other two smaller fields entirely. More specifically, we run $\text{monoT5}_{\text{MED}}$ across various eligibility segments of a particular trial t using the ranking template:

$$\begin{array}{lll} \text{Query: } p & \text{Document:} & \text{title: } t_{\text{title}} \\ & & \text{condition: } t_{\text{condition}} \\ & & \text{eligibility: } t_{\text{eligibility}} \quad \text{Relevant:} \end{array} \tag{2}$$

We also run $\text{monoT5}_{\text{MED}}$ across various description segments of a particular trial t using the ranking template:

$$\begin{array}{lll} \text{Query: } p & \text{Document:} & \text{title: } t_{\text{title}} \\ & & \text{condition: } t_{\text{condition}} \\ & & \text{description: } t_{\text{description}} \quad \text{Relevant:} \end{array} \tag{3}$$

Then, the standard MaxP approach [2] is used to assign single scores to a patient and clinical trial pair based on either just the eligibility field or just the description field, or both, denoted by $\text{monoT5}^{\text{E}}_{\text{MED}}$, $\text{monoT5}^{\text{D}}_{\text{MED}}$, and $\text{monoT5}^{\text{A}}_{\text{MED}}$, respectively. To clarify, the $\text{monoT5}^{\text{A}}_{\text{MED}}$ setting takes the highest score over *all* segments considered in templates (2) and (3).

We design the ranking template this way because $\text{monoT5}_{\text{MED}}$ is trained using the input template "Query: q Document: d". During inference, we want the fields of the clinical trial document that replace the document "d" to respect the original input template while also providing the model with domain-specific information.

3.3 Multi-Field Ranking Templates

Ideally, we want our input template to capture all the fields of the clinical trial (e.g., description and eligibility), each having a sufficient window length while being considerate of the total input sequence length limits on T5. However, we face two major issues in building an effective ranker given such constraints. First, such a model would need to flourish in a data-scarce setting during training, given that we have a small training set of only 1.1k positive pairs. Second, even if we can train an effective model, inference needs to be performed over all pairs of description and eligibility segments; since both these fields are lengthy, this is a computationally expensive task. We call these the "training problem" and the "inference problem", respectively.

We first attempt to solve the "training problem". Given the domain-specific input template used in Section 3.2, we further explore if we can train a more effective monoT5 model specific to the

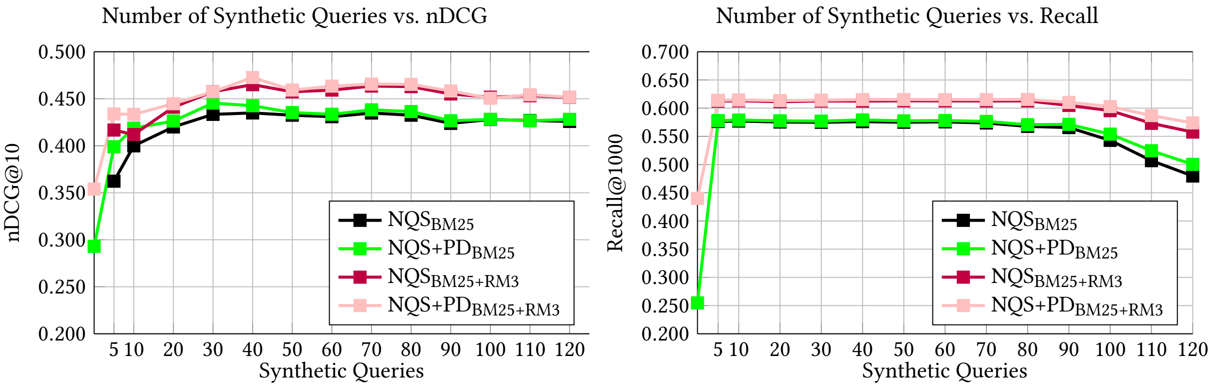

Figure 1: nDCG@10 (left) and Recall@1k (right) for the four NQS settings while varying the number of synthetic queries.

task of clinical trial matching. The caveat here is that we work in a data-scarce regime since we only have 1.1k positive (query, trial) pairs. Our hypothesis is that leveraging a domain-specific ranking template that accounts for the various fields of a trial in conjunction with fine-tuning from a pretrained monoT5$_{\text{MED}}$ model will help us successfully train such a ranker. The latter is motivated by previous success in another task [10].

We fine-tune monoT5$_{\text{MED}}$ on the clinical trial test collection curated by Koopman and Zuccon [4]. We call this model monoT5$_{\text{CT}}$. A clinical trial is labeled positive if it had a graded relevance score of at least one and negative otherwise. We use monoT5$_{\text{MED}}$ to select two segments per positive trial as input for training monoT5$_{\text{CT}}$: one being its highest-scoring description segment and the other its highest-scoring eligibility segment.

Given these two segments, we add to the training set as positive examples, the templates (2), (3), and the following:

$$\begin{aligned}\text{Query: } p \quad \text{Document: } & \text{title: } t_{\text{title}} \\ & \text{condition: } t_{\text{condition}} \\ & \text{eligibility: } t_{\text{eligibility}} \\ & \text{description: } t_{\text{description}} \quad \text{Relevant:}\end{aligned} \tag{4}$$

We sample negative segments from negative trials for training. More specifically, we construct a "hard" negatives set including the three templates that would come with the highest-scoring monoT5$_{\text{MED}}$ description and eligibility segment for each of the negative trials. We also have a "weak" negative set that includes any of the segments from all the negative trials. Then, for each positive template realized with the segments, we sample with a probability $\frac{1}{4}$ a "weak" negative and with a probability $\frac{3}{4}$ a "strong" negative. We include a smaller fraction of "weak" negatives with the hopes of pushing our model in the right direction during training, especially if the "strong" negatives are hard to distinguish from the positives given the scarcity of training data. We train our model for 1k steps with a batch size of 128 and a learning rate of $1 \cdot 10^{-3}$.

Given this trained model, we run inference in a manner similar to monoT5$_{\text{MED}}$, using the same sliding-window segmentation approach and MaxP settings to give us monoT5$^{\text{E}}$$_{\text{CT}}$, monoT5$^{\text{D}}$$_{\text{CT}}$, and monoT5$^{\text{A}}$$_{\text{CT}}$. We also introduce a new setting, which we dub monoT5'$_{\text{CT}}$, that takes the highest-scoring monoT5$^{\text{E}}$$_{\text{CT}}$ segment and the highest-scoring monoT5$^{\text{D}}$$_{\text{CT}}$ segment, rephrases them using template (4) and passes this new input through our trained model. To account for the larger input size, we use a larger input token limit of 1024, an expansion allowed in T5 because it uses relative positional embeddings.

We can view this design as an instantiation of multi-stage ranking. Here, instead of enumerating all possible description and eligibility segment pairs (similar to the duoT5 ranking model in Pradeep et al. [12]), which is a computationally expensive task, we instead first independently find the highest-scoring description and eligibility segment. We then combine these to give the model additional context, to better rank the clinical trials. This ends up as our solution to the "inference problem". To be clear, all of our settings use the *same* trained model.

4 RESULTS

4.1 Neural Query Synthesis

The effectiveness of our neural query synthesis (NQS) technique is shown in Figure 1, where we plot the nDCG@10 (left) and Recall@1k scores (right) for topics from the TREC 2021 Clinical Trials Track as a function of the number of synthetic queries for the four different settings described in Section 3.1. Recall that the NQS+PD settings additionally include the ranked lists attained by querying with the entire EHR description in the fusion.

In both plots, for the NQS+PD$_{\text{BM25}}$ and NQS+PD$_{\text{BM25+RM3}}$ settings, using zero synthetic queries (i.e., the left edge of the plots) corresponds to using *only* the EHR description as the query. We also call these settings Base$_{\text{BM25}}$ and Base$_{\text{BM25+RM3}}$, respectively, as they represent baseline effectiveness without NQS.

From the left plot, we see that for all four settings, nDCG@10 increases as we increase the number of synthetic queries to around 40. After this point, the effectiveness plateaus and remains roughly in the same region. Using RM3 yields better scores than BM25, which is consistent with the literature. We also see that including the patient description p in Q_p results in an effectiveness boost at smaller values of $|Q_p|$. However, for larger values, we do not see any value in including p. Considering the most effective setting, NQS+PD$_{\text{BM25+RM3}}$, we find that the model goes from an nDCG@10 score of 0.3539, when the number of synthetic queries is zero, i.e., Base$_{\text{BM25+RM3}}$, to an nDCG@10 score of 0.4726, when the number

Patient Description - #23: A 39-year-old man came to the clinic with cough and shortness of breath that was not relieved by his inhaler. · · ·

Query 1: causes for wheezing and shortness of breath
Query 2: what could be wrong when a chef has a cough and is short of breath all of a sudden
Query 3: how often should fluticasone be used for asthma
Query 4: what causes shortness of breath even with inhaler
⋮

Table 2: Examples of synthetic queries for the patient description in Table 1.

of generated queries is 40. This represents a 33.5% relative bump in terms of nDCG@10.

Moving on to the right plot, we see a slightly different picture. For all settings, Recall@1k scores increase until the number of synthetic queries reaches 5, after which scores remain mostly flat until we add around 80 queries. After this point, Recall@1k drops gradually until we reach 120 queries, the limit in our experiments. Here again, BM25 + RM3 performs consistently better. Based on these two plots, it appears that 40 synthetic queries represents a good setting to maximize both Recall@1k and nDCG@10.

Table 2 provides a qualitative example of a patient description topic from the TREC 2021 Clinical Trials Track and four corresponding synthetic queries that are generated by our model. As we can see, NQS generates diverse queries that capture critical aspects of the patient description. For this topic, we note that the first single-sentence query is concise and includes terms like "wheezing" and "shortness of breath". The second captures the occupation of the patient and includes the term "cough" that could be relevant. The third query is more scientific and looks for the ideal dosage of "fluticasone". The fourth query is a reformulation of the first query and introduces a new term, "inhaler". Given that different synthetic queries capture diverse aspects of the patient description, we can independently issue bag-of-words queries using these single-sentence long segments and then fuse the results to create a strong first-stage retriever.

4.2 Multi-Stage Clinical Trial Matching

Table 3 shows the results from the TREC 2021 Clinical Trials Track. We include scores for the primary metrics nDCG@10, P@10, and RR. It is worth noting that for measures based on binary judgments, only trials labeled "eligible" are treated as relevant.

For reference, row (1) provides the median score across all runs submitted. Rows (2a) and (2b) present the second and third-ranked submissions (from unique groups) in the track. Rows (3a) and (3b) present our bag-of-words baselines that use the EHR description as the query. Row (3c) presents our $\text{NQS+PD}_{\text{BM25+RM3}}$ run with 40 synthetic queries. The zero-shot pointwise *relevance* ranking models are shown in rows (4a)–(4c). Rows (4d)–(4f) refer to our domain-specific ranking model. Row (4g) shows the result of the multi-stage method discussed at the end of Section 3.3. This row corresponds to the top-scoring run in the evaluation.

As already discussed, moving the first-stage retrieval method from $\text{Base}_{\text{BM25+RM3}}$, row (3b), to $\text{NQS+PD}_{\text{BM25+RM3}}$, row (3c), brings large improvements in effectiveness across the board. This observation highlights the benefits of the NQS method. All neural rerankers we describe here use this run as the base run (i.e., first-stage retriever). Among the zero-shot neural rankers, rows (4a)–(4c), we see that only $\text{monoT5}^{\text{E}}_{\text{MED}}$, row (4c), shows effectiveness comparable to the base run in row (3c). The fact that this simple NQS method *alone* performs on par with the state-of-the-art $\text{monoT5}_{\text{MED}}$ model (albeit applied in a zero-shot manner) further demonstrates its effectiveness.

	Run	nDCG@10	P@10	RR
(1)	Median	0.3040	0.1613	0.2942
(2a)*	`damoebrtog`	0.5953	0.4093	0.6083
(2b)*	`CSIROmed_inc`	0.5320	0.3173	-
(3a)	$\text{Base}_{\text{BM25}}$	0.2923	0.1680	0.3015
(3b)*	$\text{Base}_{\text{BM25+RM3}}$	0.3539	0.2040	0.3659
(3c)*	$\text{NQS+PD}_{\text{BM25+RM3}}$	0.4726	0.2760	0.4304
(4a)	+ $\text{monoT5}^{\text{A}}_{\text{MED}}$	0.2994	0.1973	0.3560
(4b)	+ $\text{monoT5}^{\text{D}}_{\text{MED}}$	0.2311	0.1507	0.3223
(4c)*	+ $\text{monoT5}^{\text{E}}_{\text{MED}}$	0.4715	0.2987	0.4830
(4d)	+ $\text{monoT5}^{\text{A}}_{\text{CT}}$	0.6763	0.5480	0.7253
(4e)	+ $\text{monoT5}^{\text{D}}_{\text{CT}}$	0.4493	0.3267	0.6260
(4f)*	+ $\text{monoT5}^{\text{E}}_{\text{CT}}$	0.6792	0.5493	0.7161
(4g)*	+ $\text{monoT5'}_{\text{CT}}$	0.7118	0.5933	0.8162

Table 3: Results in the TREC 2021 Clinical Trials Track. Rows corresponding to officially submitted runs are denoted as $(\cdot)^*$.

Training the base model on the clinical trial matching task using templates tailored to the domain brings improvements over their zero-shot counterparts, rows (4d)–(4f) vs. (4a)–(4c). It is also clear that scoring only the description segments, row (4e), leads to worse effectiveness than evaluating all segments or only the eligibility segments, rows (4d) and (4e), both of which process the inclusion/exclusion criteria, which are critical parts of the clinical trial under consideration.

Finally, we see that multi-stage ranking using the clever selection of the eligibility and description segments and then combining them using template (4) results in the most effective system across the board, row (4g). Comparing this run to the second top-scoring submission, row (2a), we note a 20% relative effectiveness gain based on the nDCG@10 score, demonstrating the effectiveness of our architecture.

5 CONCLUSION

In this paper, we design an effective multi-stage clinical trial matching system. We introduced NQS, a neural query synthesis method that leverages doc2query–T5 to generate synthetic single-sentence queries to replace lengthy patient descriptions. When these queries are independently issued using BM25 + RM3 and the results fused using RRF, the final ranked list is an effective first-stage retrieval method, performing on par with zero-shot state-of-the-art neural ranking methods. Additionally, we train a pointwise reranking model leveraging domain-specific ranking templates to efficiently learn the matching task in a data-poor regime. We believe these techniques represent an important step towards better automated clinical trial matching and can provide a foundation for future work.

ACKNOWLEDGEMENTS

This research was supported in part by the Canada First Research Excellence Fund, the Natural Sciences and Engineering Research Council (NSERC) of Canada, and the Waterloo–Huawei Joint Innovation Laboratory. Computational resources were provided in part by Compute Ontario, Compute Canada, and Google in the form of Google Cloud credits.

REFERENCES

[1] Gordon V. Cormack, Charles L. A. Clarke, and Stefan Büttcher. 2009. 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 2009)*. Boston, Massachusetts, 758–759.

[2] Zhuyun Dai and Jamie Callan. 2019. Deeper Text Understanding for IR with Contextual Neural Language Modeling. In *Proceedings of the 42nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2019)*. Paris, France, 985–988.

[3] Kelvin Jiang, Ronak Pradeep, and Jimmy Lin. 2021. Exploring Listwise Evidence Reasoning with T5 for Fact Verification. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*. 402–410.

[4] Bevan Koopman and Guido Zuccon. 2016. 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 2016)*. Pisa, Italy, 669–672.

[5] Jimmy Lin, Xueguang Ma, Sheng-Chieh Lin, Jheng-Hong Yang, Ronak Pradeep, and Rodrigo Nogueira. 2021. Pyserini: A Python Toolkit for Reproducible Information Retrieval Research with Sparse and Dense Representations. In *Proceedings of the 44th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2021)*. 2356–2362.

[6] Sean MacAvaney, Arman Cohan, and Nazli Goharian. 2020. SLEDGE: A Simple Yet Effective Zero-Shot Baseline for Coronavirus Scientific Knowledge Search. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 4171–4179.

[7] Yuning Mao, Xiang Ren, Heng Ji, and Jiawei Han. 2021. Constrained Abstractive Summarization: Preserving Factual Consistency with Constrained Generation. *arXiv:2010.12723* (2021).

[8] Rodrigo Nogueira, Zhiying Jiang, Ronak Pradeep, and Jimmy Lin. 2020. Document Ranking with a Pretrained Sequence-to-Sequence Model. In *Findings of the Association for Computational Linguistics: EMNLP 2020*. 708–718.

[9] Rodrigo Nogueira and Jimmy Lin. 2019. From doc2query to docTTTTTquery.

[10] Ronak Pradeep, Xueguang Ma, Rodrigo Nogueira, and Jimmy Lin. 2021. Vera: Prediction Techniques for Reducing Harmful Misinformation in Consumer Health Search. In *Proceedings of the 44th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2021)*. 2066–2070.

[11] Ronak Pradeep, Xueguang Ma, Xinyu Zhang, Hang Cui, Ruizhou Xu, Rodrigo Nogueira, and Jimmy Lin. 2020. H_2oloo at TREC 2020: When all you got is a hammer... Deep Learning, Health Misinformation, and Precision Medicine. In *Proceedings of the Twenty-Ninth Text REtrieval Conference (TREC 2020)*.

[12] Ronak Pradeep, Rodrigo Nogueira, and Jimmy Lin. 2021. The Expando-Mono-Duo Design Pattern for Text Ranking with Pretrained Sequence-to-Sequence Models. *arXiv:2101.05667* (2021).

[13] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *Journal of Machine Learning Research* 21, 140 (2020), 1–67.

[14] Kirk Roberts, Dina Demner-Fushman, Ellen M. Voorhees, Steven Bedrick, and Willian R. Hersh. 2021. Overview of the TREC 2021 Clinical Trials Track. In *Proceedings of the Thirtieth Text REtrieval Conference (TREC 2021)*.