Information retrieval in an infodemic: the case of COVID-19 publications

Sohrab Ferdowsi*[1], Nikolay Borissov[2,3], Elham Kashani[5], David Vicente Alvarez[1], Jenny Copara[1,4], Racha Gouareb[1], Poorya Amini[2,3], and Douglas Teodoro*[1,4]

[1]*HES-SO - University of Applied Sciences and Arts of Western Switzerland*
[2]*Riskclik AG, Bern, Switzerland*
[3]*Clinical Trials Unit, University of Bern, Bern, Switzerland*
[4]*SIB Swiss Institute of Bioinformatics, Lausanne, Switzerland*
[5]*Institute of Pathology, University of Bern, Bern, Switzerland*

***Abstract*—In the context of searching for COVID-19 related scientific literature, we present an information retrieval methodology for effectively finding relevant publications for different information needs. We discuss different components of our architecture consisting of traditional information retrieval models, as well as modern neural natural language processing algorithms. We present recipes to better adapt these components to the case of an infodemic, where, from one hand, the number of publications has an exponential growth and, from the other hand, the topics of interest evolve as the pandemic progresses. The methodology was evaluated in the TREC-COVID challenge, achieving competitive results with top ranking teams participating in the competition. In retrospect to this challenge, we provide additional insights with further useful impacts.**

***Keywords*—information retrieval, neural language models, infodemic, information, retrieval for pandemic, COVID-19, NLP**

I. Introduction

*In parallel to its public health crisis with vast social and economic impacts, the COVID-19 pandemic has resulted in an explosive surge of activities within scientific communities and across many disciplines [1]. The number of publications related to the pandemic has had an exponential growth since early 2020 when the pandemic was officially announced. In addition to the volume and velocity of the generated data, the heterogeneity of the data as as result of the typical variety of concept naming found in the biomedical field (e.g., COVID-19, SARS-COV2, Coronavirus disease 19, etc.), spelling mistakes (e.g., chloroquin, hydoxychloroquine), and the different collection types (scientific reports, grey literature, preprint archives, etc.), among others, make searching and finding relevant literature within these corpora an important challenge.

To help clinical researchers, epidemiologists, medical practitioners and healthcare policy makers, among others, find the most relevant information for their needs, effective information retrieval models for these large and fast changing corpora became then a prominent necessity. The information retrieval community, in turn, has responded actively and quickly to this extraordinary situation and has been aiming at addressing these challenges. To facilitate research for all the communities involved, the COVID-19 Open Research Dataset (CORD-19) [2] collection was built to maintain all the related publications to the family of corona-viruses. This dataset helped research in various directions and several tasks are build around it, including NLP-related tasks like question answering and language model pre-training, and information retrieval challenges in Kaggle[†], as well as the TREC-COVID[‡].

The TREC-COVID [3], [4] challenge ran in 5 rounds, each asking for a particular set of information needs to be retrieved from the publications of the CORD-19 collection. During each round of the TREC-COVID challenge, the participants were asked to rank documents of the CORD-19 corpus in decreasing order of the likelihood of containing answers to corresponding query topics. At the end of each round, experts provided judgments of the relevance of the top-ranking documents submitted by different participants using a pooling strategy [5], [3]. TREC-COVID used a ternary labeling scheme to specify the relevance of each publication-topic (document-query) pair. Although limited to the first several top-submissions of the participating teams, these relevance judgments are valuable training examples to train retrieval models for the subsequent rounds of the challenge.

More than 50 teams participated in the TREC-COVID challenge worldwide. Several of them have published their methodologies used to approach this complex task [6], [7], [8], [9], [10], [11]. These works describe different information retrieval and natural language processing (NLP) techniques, as well as how to further adapt them to the infodemic case of the COVID-19. Having participated in the TREC-COVID challenge, this paper follows the same trend by detailing our retrieval pipeline, which brought us competitive results with the top-ranking teams. The aim is to help practitioners and researchers from the information retrieval community to provide timely solutions in case of potential future infodemics.

*{sohrab.ferdowsi | douglas.teodoro}@hesge.ch

[†]https://www.kaggle.com/allen-institute-for-ai/CORD-19-research-challenge

[‡]https://ir.nist.gov/covidSubmit

II. Related work

A. Two-stage information retrieval

Currently, two main methodologies are used to rank documents in information retrieval systems: i) the classic query-document probabilistic approaches, such as BM25 [12] and language models [13], and ii) the learning to rank approaches, which usually post-process results provided by classic systems to improve the original ranked list [14], [15]. When there are sufficient training data, i.e., queries with relevance judgments in the case of information retrieval, learning to rank models tend to outperform classic one-stage retrieval systems [14], [16]. Nevertheless, empiric results have also shown that the re-ranking step may degrade the performance of the original rank [17]. Progress on learning to rank algorithms have been fostered thanks to the public release of annotated benchmark datasets, such as the LETOR [18] and the Microsoft Machine Reading Comprehension (MS MARCO) [19].

The learning to rank approaches can be categorised in three main classes of algorithms - pointwise, pairwise and listwise - based on whether they consider one document, a pair of documents or the whole ranking list in the learning loss function, respectively [20], [15], [14], [16]. Variations of these learning to rank algorithms are available based on neural networks [20], [15] and other learning algorithms, such as boosting trees [21]. More recently, pointwise methods leveraging the power of deep language models have attracted great attention [22], [23]. These models use the query and document learning representations provided by the language models to classify whether a document in the ranked list is relevant to query. While these two stage neural re-rankers provide interesting features, such as learned word proximity, in practice, the first stage based on classic probabilistic retrieval algorithms is indispensable, as the algorithmic complexity of the re-ranking methods makes them prohibitive to classify the whole collection [16].

B. TREC-COVID retrieval efforts

Recently, the specific case of retrieval of COVID-related scientific publications has been addressed in several efforts [6], [7], [8], [9], [10], [11]. These works follow mostly the above two-stage retrieval process. Among the first efforts is the SLEDGE system [6], where the authors detail their solutions for the first round of TREC-COVID challenge using a BM25-based ranking method followed by a neural re-ranker. An important difficulty for the first round of the challenge is the absence of labelled data. To overcome this limitation, the authors lightly tune the hyper-parameters of the first stage ranking model using minimal human judgements on a subset of the topics. As for the second stage, they use SciBERT [24] pre-trained on biomedical texts and fine tuned on the general MS MARCO set [19] with a simple cross-entropy loss. Their work provides further insights, including the usefulness of the date filtering, i.e., excluding all the publications before 1st of January 2020, as well as the effectiveness of using only specific query fields of the standard topics for both stages of search. CO-Search [8] uses a slightly different approach, wherein they incorporate semantic information, as captured by Siamese-BERT [25], also within the initial retrieval stage. Moreover, they use the citation information of publications in their ranking pipeline.

In the work of COVIDex [7], the authors provide a full-stack search engine including an active web service dedicated to the CORD-19 data. Their system follows a multi-stage ranking pipeline, where their first stage is based on the Anserini information retrieval toolkit [26]. To address the issue of length variability between the articles and to account for the lack of abstract or full-text for some of the publications, the authors study different scenarios on what to consider as the atomic unit of a document, including concatenation of abstract and title, versus a paragraph-level index. They provide different strategies for neural re-ranking. In particular, starting from round 2 of the challenge, they use the official relevance feedbacks provided by domain experts as labels to train a classification algorithm on the data from the previous round.

III. Material and methods

In this section, we describe the TREC-COVID challenge and our methodology for searching COVID-19 related literature. We start by introducing the CORD-19 dataset, which is the corpus used in the competition. We then detail the queries and briefly the challenge organisation. Then, we describe our ranking methodology. Finally, we present the evaluation criteria used to score the participant's submissions. For further details on the TREC-COVID challenge, please see [3], [4].

A. The CORD-19 dataset

A prominent effort to gather publications, preprints and reports related to the corona-viruses (COVID-19, SARS, MERS) is the CORD-19 collection of the Allen Institute for Artificial Intelligence (in collaboration with other partners) [2]. As shown in Figure 1, this is a large and dynamically growing semi-structured dataset from various sources like PubMed, PubMed Central, WHO and preprint servers like bioRxiv, medRxiv, and arXiv. The dataset contains document metadata, including *title*, *abstract*, *authors*, among others. A diverse set of related disciplines, e.g., from virology and immunology to genetics, are represented in the collection. Throughout the challenge, the dataset was updated on a daily basis and snapshot versions representing its status at a certain time were provided to the participants for each round. In the last round of the TREC-COVID challenge, the corpus contained almost 200'000 documents.

B. The TREC-COVID query and challenge organisation

The TREC-COVID challenge provided a query set trying to capture important information needs of researchers during the pandemic from the CORD-19 collection described above [3], [4]. These needs are stated in different "topics", where each topic consists of a 'query", a "question" and a "narrative", as shown in Table I. The challenge comprised 5 rounds, starting

query	school reopening coronavirus
question	what are the benefits and risks of re-opening schools in the midst of the COVID-19 pandemic?
narrative	With the possibility of schools re-opening while the COVID-19 pandemic is still ongoing, this topic is looking for evidence or projections on what the potential implications of this are in terms of COVID-19 cases, hospitalizations, or deaths, as well as other benefits or harms to re-opening schools. This includes both the impact on students, teachers, families, and the wider community.

TABLE I: An example of an information need "topic" (# 48 out of 50) from the TREC-COVID challenge with 3 fields.)

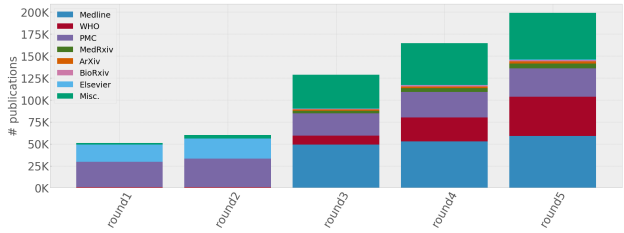


Fig. 1: Growth of the CORD-19 corpus across the TREC-COVID rounds stratified by source.

with 30 topics in the round 1 and adding 5 new topics at each new round, reaching thus 50 topics in round 5.

In each round, the participants provided ranked lists of candidate publications, which best answered the query topics. Each ranked list was generated by a different information retrieval model, so called *run*, with up to 5 runs in the first 4 rounds and 7 runs in the last round. At the end of each round, domain experts examined the top-k candidate publications (where k is defined by the organisers) from the priority runs of each team and judged them as "highly relevant", "somehow relevant", or "irrelevant". Then, based on the consolidated relevance judgments, the participants were evaluated using standard metrics of information retrieval that we describe in section III-D. Judged documents for a specific topic from previous rounds were excluded from the relevance judgement of the subsequent rounds. The participants runs were categorized either as "automatic", i.e., directly coming from a retrieval algorithm and without benefiting from relevance judgments, or "feedback", i.e., benefiting from the relevance judgments of the previous rounds, or "manual", i.e., when a human intervenes the results of a retrieval algorithm in some sort, e.g., by removing irrelevant publications.

C. Proposed methodology

Fig. 2 shows the different components of our proposed retrieval system. These components can be divided into 3 main categories: *i)* first-stage retrieval using classic probabilistic methods; *ii)* (neural) re-ranking models; and *iii)* rank fusion algorithms. In the next sections, we detail these different components.

1) First-stage retrieval: We first apply a set of pre-processing steps to the publication texts. This includes lower-casing, removal of symbols, Porter stemming, as well as enrichment of some COVID-related keywords with a set of synonyms, which is a minimal set of commonly occurring variations of keywords that we manually extract, e.g., "covid-19", "sars-cov-2", ..

Our implementation of the indexing is based on the *Elastic-search* framework, where we use 3 separate indices for each round of the challenge. In particular, we use the classical BM25 [27], DFR [28] and the LMD [29] as our basic indices.

These methods are based on the general idea of bag-of-words-based ranking, where a histogram of tokenized words within a document replaces the text of the document. This idea, while neglecting the relative position of words in documents, brings about the significant advantage of using the inverted-file structure, where the query words can find the frequency of their occurrences within the documents very efficiently since these histogram-like statistics are calculated once and offline.

This efficiency of inverted-files is the core basis of very-large size retrieval systems, making the bag-of-words-based ranking an indispensable part of information retrieval. However, simply scoring the documents by their frequency count of the query words provides poor results, since commonly occurring words (like the article "the" in English) will dominate the score.

A re-weighting of the words is used using the "tf-idf" framework, which is the product of term-frequency and the inverse document frequency statistics. Denote $f(t,d)$ as the number of times a term t appears in a document d within a collection C. The basic tf-idf is calculated as:

$$tfidf(t,d;C) = tf(t,d) \cdot idf(t;C) = \log\big(1+f(t,d)\big) \cdot \log\big(\frac{|C|}{n_t}\big), \tag{1}$$

where n_t is the number of documents containing the term t, and $|C|$ is the size of the collection.

Okapi BM25 [27] is based on tf-idf and is a very popular and effective ranking function. For a query q containing $\{q_1, \cdots, q_i, \cdots, q_n\}$ terms, the score of each document in the collection is calculated as:

$$s(d;q,C) = \sum_{1=1}^{n} idf(q_i;C) \cdot \frac{f(q_i,d) \cdot (k_1+1)}{f(q_i,d) + k_1 \cdot \big(1 - b + b \cdot \frac{|d|}{\mathbb{E}\big[|D|\big]}\big)}, \tag{2}$$

where $|d|$ is the length of the document d, $\mathbb{E}\big[|D|\big]$ is the average length of the documents in the collection C, and b_1 and k are the hyper-parameters of the model to be tuned.

As another probabilistic model for information retrieval, we also use the DFR [28] which measures the divergence from randomness of the frequency of a term in a document. We furthermore use the LMD [29], which is a language model-based retrieval with Dirichlet-prior smoothing. In a retrieval

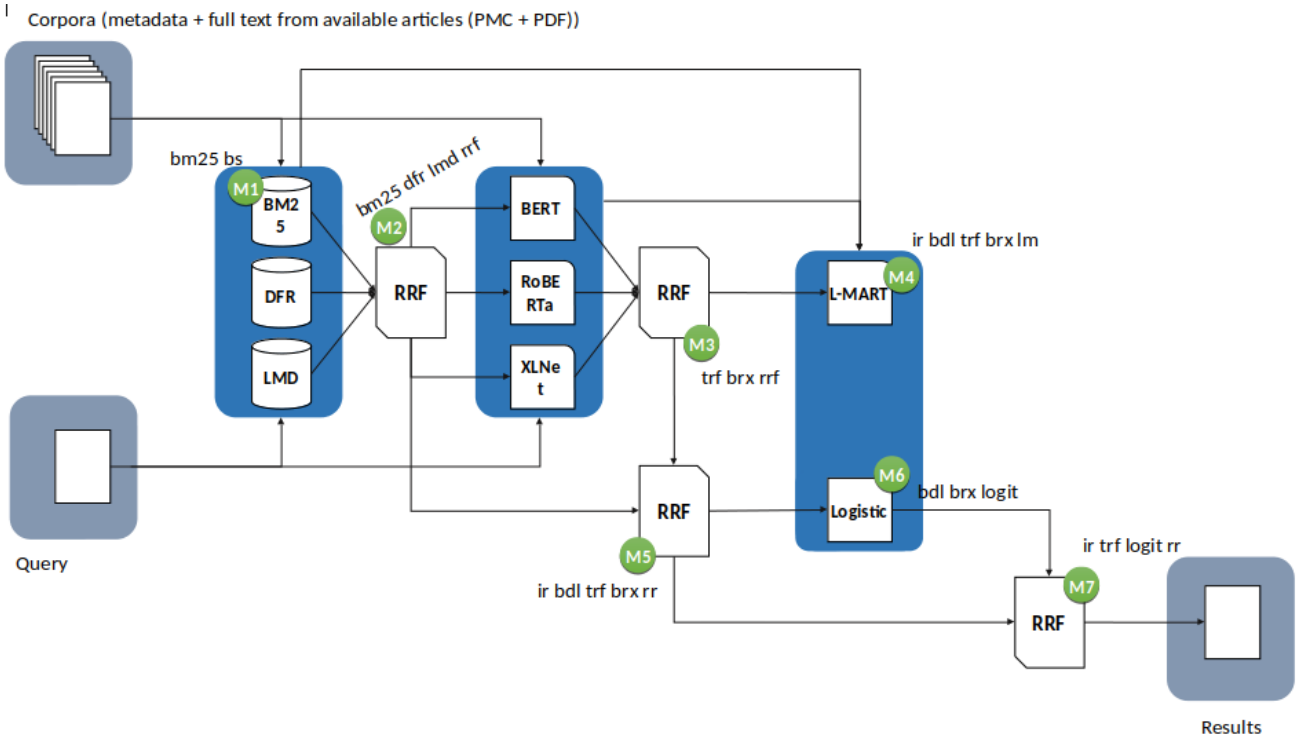


Fig. 2: General schema of our proposed method. M1-M7 denote the different submitted runs. RRF is the Reciprocal Rank Fusion.

context, a language model specifies the probability that a document generated a query, and smoothing is used to avoid overfitting. Each of these models has one hyper-parameter to tune.

At each round of the challenge (except for the first round), we minimally fine-tune the basic hyper-parameters of each index by cross-validation. As an example, to tune the b and k for the BM25 index of round 3, we take the index of round 2 and submit the topics of round 2 and tune the hyper-parameters for the best $P@10$ w.r.t. the relevance judgments of round 2 (excluding round 1). We then preset the index of round 3 with these values.

A particularly effective technique in the case of TREC-COVID (also adopted by some other participating teams) is filtering documents based on their publication dates. As for our submissions, we cut off all publications before December 2019, when the outbreak was first detected. This reduces a bit the recall, but highly improves the precision.

Finally, we use the Reciprocal Rank Fusion [30] to combine the results of different retrieval runs. This is a very simple, yet highly effective technique to re-score the retrieval list based on the scores of multiple retrieval lists. We used the standard hyper-parameter ($k = 60$), and noticed consistent improvement in different experiments.

Querying strategies: Apart from the indexing models which we discussed above, the creation of the indices, as well as the querying strategies can have substantial impacts on the retrieval performance. In particular, the atomic unit of index, i.e., the "document" in the IR language can be defined in different ways. Also, different components of the query can be arranged in different ways.

As for the atomic document, we notice that separately indexing each of the paragraphs of the publications does not provide a noticeable advantage, so we consider each publication as the basic unit. We consider the title and abstract of the publications as separate fields for our index, and use full-text of them, if possible to download from PubMed Central.

We submit queries to the index in a combinatorial way, in terms of these fields, as well as the fields of the topics. This means that, for each of the "query", "question" and "narrative" parts of each of the challenge "topics", we submit a separate query for each of the title and abstract fields of the publications. This provides us with 6 queries for each topic that we submit to each of the 3 indices. We simply sum up the scores of each of these queries together to provide us with an initial list for each index.

2) Neural re-ranking stage: The above models are all based on bag-of-words like statistics, where essentially, we only look at the histogram of words in a document and neglect the sequential nature of text. Significant achievements have been made within the NLP communities to take the sequential nature of language into consideration. Many neural network-based language models have been proposed.

A seminal effort in this direction is BERT [22], showing

significant success in a wide range of NLP tasks. This is based on a bi-directional version of the famous transformer architecture [31], and is pre-trained with the tasks of masked language modeling, as well as next sentence prediction. Ever since BERT was introduced, several works tried to augment its performance. A successful work in this direction is RoBERTa [32], using larger and more diverse corpora for pre-training, as well as removing the next sentence prediction task, but using a novel masked language modeling. While it needs larger compute power, it noticeably improves the performance of BERT across different tasks. Another similar effort is the XLNet [33] with a permutation-based masking, showing consistent improvement over BERT.

In our pipeline, after the first-stage retrieval, we use neural language models to fine-tune our results. Fig. 3 sketches the general idea of how we use a transformer into our pipeline. So for each query topic at round 4, we take a list of top 1000 publications retrieved and use the relevance judgments of round 4 to classify them under $\{1, 2, 3\}$ relevance classes. For this, we use the `[CLS]` token of neural language models and use the cross-entropy loss for the classification. We limit the input to the neural models to 512 tokens.

We learn a BERT, a RoBERTa and an XLNet separately using the above strategy and combine the re-ranked lists for each of these models for round 5 using the RRF technique.

Another element in this stage of our pipeline is to use explicit learning to rank losses. While the BERT-based classification was an instance of a simple point-wise ranking loss, more intricate losses are possible. A successful algorithm is LambdaMART [15] which uses a pair-wise loss and is based on gradient-boosted decision trees.

3) Submitted runs: With the elements discussed thus far, Table II summarizes our submitted runs for round 5 of TREC-COVID.

Our first run *bm25-bs* is the baseline BM25 and our second run *bm25-dfr-lmd-rrf* is fusion of BM25, DFR and LMD with the RRF technique. Run3 *trf-brx-rrf* uses the fusion of BERT, RoBERTa and the XLNet each trained independently from the ranked list of run2 as detailed above. On top of these, run4 *ir-bdl-trf-brx-lm* re-ranks the results based on LambdaMART, while run5 *ir-bdl-trf-brx-rrf* combines them again with run2 using RRF. Run6 *bdl-brx-logit* takes the classic IR features, along with neural language model features and simply uses logistic regression to re-rank the results. Finally, run7 *ir-trf-logit-rrf* merges it with run3 and run5 using RRF.

D. Evaluation criteria

Per each of the queries, the retrieval pipeline provides an ordered list of candidate publications. The quality of the returned list of candidates is evaluated against the relevance judgments provided by the trec-evaluators, and is measured by the following metrics.

Precision at K ($P@K$), simply measures the ratio of relevant publication in the top-K items returned. More precisely:

$$P@K = \frac{1}{Q}\sum_{q=1}^{Q}\frac{\#\text{of relevant documents to query } q \text{ in top-}K}{K}, \tag{3}$$

where $q = \{1, \cdots, Q\}$ is the set of queries (topics) of the challenge. Note that both the "highly relevant" and "somehow relevant" documents are considered simply as relevant documents with the same weight.

Non-Discounted Cumulative Gain at K ($NDCG@K$) takes into account the degree of relevance $\text{rel}_q(d_t)$ of the document d returned at position t for the query q, as well as the position in which the document is presented in the submitted run. This is to give increasingly less importance as the position of a publications increases in the list.

This measure is based on the $DCG_q@K$, defined as:

$$DCG_q@K = \sum_{t=1}^{K}\frac{\text{rel}_q(d_t)}{\log_2(t+1)}, \tag{4}$$

as well as the $IDCG_q@K$ which measures its ideal value, and is normalized and averaged over all queries as:

$$NDCG@K = \frac{1}{Q}\sum_{q=1}^{Q}\frac{DCG_q@K}{IDCG_q@K}. \tag{5}$$

These two measures, however, do not directly take into account the "recall" and are based on "precision".

Mean Average Precision (*MAP*) also takes into account the recall of the retrieval, and is defined as the area under the precision-recall curve averaged over all queries:

$$MAP = \frac{1}{Q}\sum_{q=1}^{Q}\int_0^1 p_q(r)dr, \tag{6}$$

where $p_q(r)dr$ is the precision-recall curve. In practice, this curve is discretized and a finite sum is used instead of the integral.

Binary Preference (*Bpref*) measures how frequently relevant publications are ranked before non-relevant ones. This is more robust than the above measures to the missing values of the relevance judgments and is defined as:

$$Bpref = \frac{1}{R}\sum_{r=1}^{R}\left(1 - \frac{|n \text{ ranked higher than } r|}{R}\right), \tag{7}$$

where R is the number of judged relevant documents that were retrieved, r is one of those, and n is an irrelevant document that was retrieved in a list of size R.

IV. Results

Table III shows the official results of the leading submitted runs of each team for the 5th round of TREC-COVID, where we ranked 4th out of the 22 participating teams.

Table IV sketches the results of each of our 7 submitted runs. In order to see the effect of each of the components of our runs (sketched in Fig. 2 and summarized in Table II), Fig. 4 shows the cumulative effect of each of these runs on the final performance. Note that in our submitted runs, each run is not

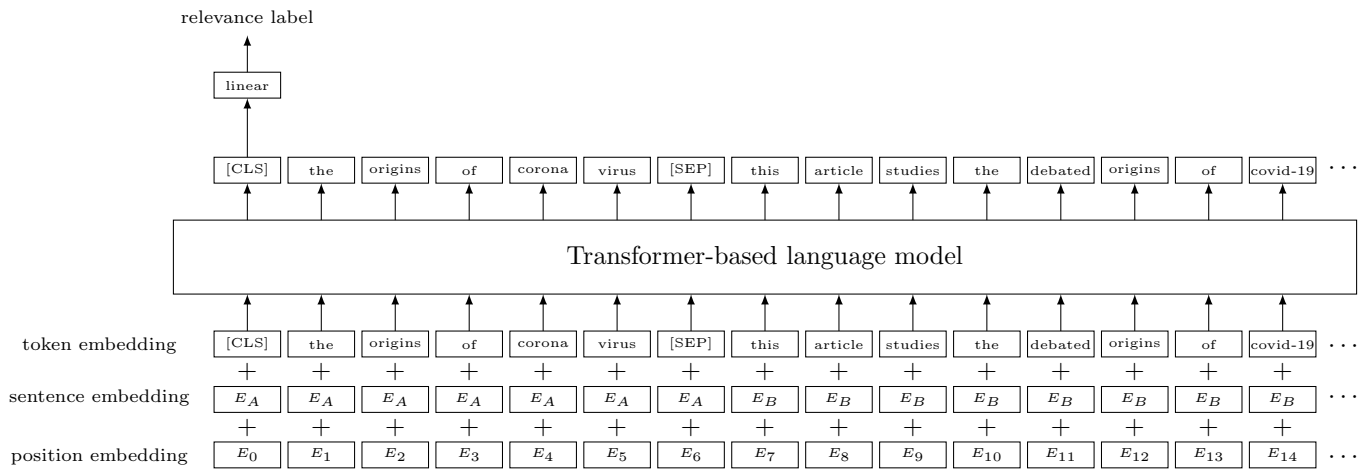


Fig. 3: Transformer-based retrieval used within the second stage of our retrieval pipeline. As inputs to the pre-trained transformer, the topics and candidate publications are separated by the *[CLS]* tag. Moreover, position embedding to enforce the sequential structure of text, as well as sentence embedding are added to the main embedding for each token. The output of the special *[CLS]* token is finally used to determine the relevance of the candidate publication to the queried information topic.

M1: *bm25-bs*	Baseline BM25 model using the metadata index and trained on round 4 queries.
M2: *bm25-dfr-lmd-rrf*	A combination of BM25, DFR and LM Dirichlet runs using RRF ($k = 60$) on metadata and full text indices.
M3: *trf-brx-rrf*	A combination of deep neural language models (BERT, RoBERTa and XLNet) runs, fine tuned on round 4 QREL and combined using RRF ($k = 60$). Classic IR models (BM25, DFR, XLNet) was used as the baseline run.
M4: *ir-bdl-trf-brx-lm*	A learning-to-rank model using the LambdaMART algorithm with features from classic information retrieval models (BM25, DFR and LM Dirichlet) and deep neural language models (BERT, RoBERTa and XLNet) extracted from metadata and full text indices.
M5: *ir-bdl-trf-brx-rrf*	A combination of classic information retrieval models (BM25, DFR and LM Dirichlet) with neural language models. RRF ($k = 60$) was used to combined the runs.
M6: *bdl-brx-logit*	A logistic regression model trained on round 4 QREL using features (query, question, narrative, title, abstract, etc.) of the BM25, DFR and LM Dirichlet information retrieval models and deep neural language models (BERT, RoBERTa and XLNet).
M7: *ir-trf-logit-rrf*	A combination of information retrieval (BM25, DFR, LM Dirichlet), neural language (BERT, RoBERTa, XLNet) and logistic regression models trained on round 4 QREL. RRF ($k = 60$) was used to combined the runs.

TABLE II: Description of each run-tag used in Table IV. Refer to Fig. 2 for a pictorial description.

necessarily based on a previous run, so this figure is only for comparing how each operation would impact the final result.

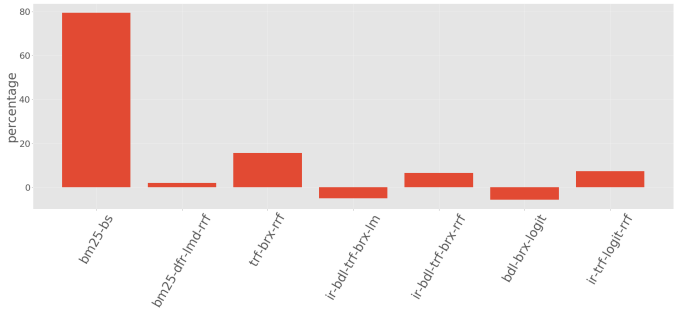


Fig. 4: Cumulative contribution of each run on the overall performance in terms of *NDCG*@20.

Apart from the obvious BM25 baseline, the most significant contribution comes from the inclusion of neural language models. It is important to notice the consistent benefit of Reciprocal Rank Fusion, which always tends to improve the results. Interestingly, the effect of LambdaMART seems to be detrimental for *P*@10, *NDCG*@20 and *MAP*, but beneficial for *Bpref*. Also, logistic regression, when applied alone provides poor results, but helps when combined with other runs.

A. Per-topic performance

To get a per-topic picture of our performance, Fig. 6 shows the performance of our *ir-bdl-trf-brx-lm* run per each of the 50 topics in comparison to the submissions from other participants to the challenge and in terms of *Bpref*.

To further analyze the effect of different runs per each of the topics, Fig. 6 shows the *NDCG*@20 for our run2: *bm25-dfr-lmd-rrf* with classical IR and run7: *ir-trf-logit-rrf* with neural re-ranking.

As is clear from the figure, while the average performance has significantly improved in run7 compared to run2, the topics which we perform the worst (#12, #19 and #50) in run7, are also among the worst performing topics in run2. This shows that in this double-stage retrieval, not surprisingly, the low

TABLE III: Official leader-board of the final round of TREC-COVID

team	*NDCG*@20	*P*@20	*rbp*_p5	*Bpref*	*map*
Unique-ptr	0.8496	0.876	0.9399	0.6378	0.4731
covidex	0.8311	0.846	0.937	0.533	0.3922
Elhuyar-NLP-team	0.8116	0.834	0.9028	0.6284	0.4169
risklick (ours)	0.7956	0.826	0.88	0.5759	0.3789
Udel-fang	0.7929	0.827	0.8988	0.5555	0.3682
CIR	0.7921	0.832	0.8885	0.5735	0.3983
uogTr	0.7913	0.842	0.8683	0.5709	0.3901
UCD-CS	0.7859	0.844	0.8457	0.4488	0.3348
sabir	0.7791	0.821	0.8902	0.6078	0.4061
mpiid5	0.7761	0.811	0.862	0.5873	0.3903
UIowaS	0.7719	0.82	0.8657	0.573	0.3768
anserini	0.7346	0.779	0.8424	0.5103	0.328
BioinformaticsUA	0.6879	0.726	0.8251	0.5205	0.2614
WiscIRLab	0.6877	0.74	0.7987	0.5748	0.3254
UH_UAQ	0.6788	0.701	0.8045	0.3925	0.2448
SFDC	0.6772	0.715	0.7939	0.5253	0.3089
BRPHJ	0.6731	0.711	0.7667	0.4506	0.2836
xj4wang	0.6685	0.703	0.7767	0.5254	0.2647
HKPU	0.6494	0.704	0.7226	0.4883	0.2973
CSIROmed	0.6439	0.672	0.773	0.4013	0.2225

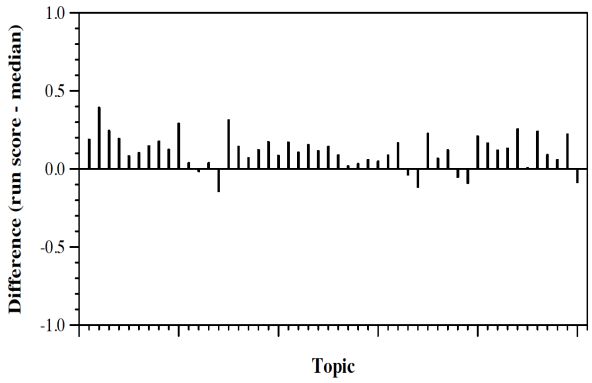


Fig. 5: Difference of the *Bpref* to the median of all submissions per each topic of round 5.

recall associated to the classical IR stage is imposed to the neural re-ranking, no matter how powerful they are.

An interesting case is topic # 19, where apparently most teams performed very poorly. This can be seen in Fig. 7, where the number of relevant publications judged by TREC-COVID domain experts are very low.

V. Discussion

We mentioned in III-C1 about the effect of excluding publications before the detection of the outbreak, which are about other coronaviruses before the SARS-COV-2. The effect

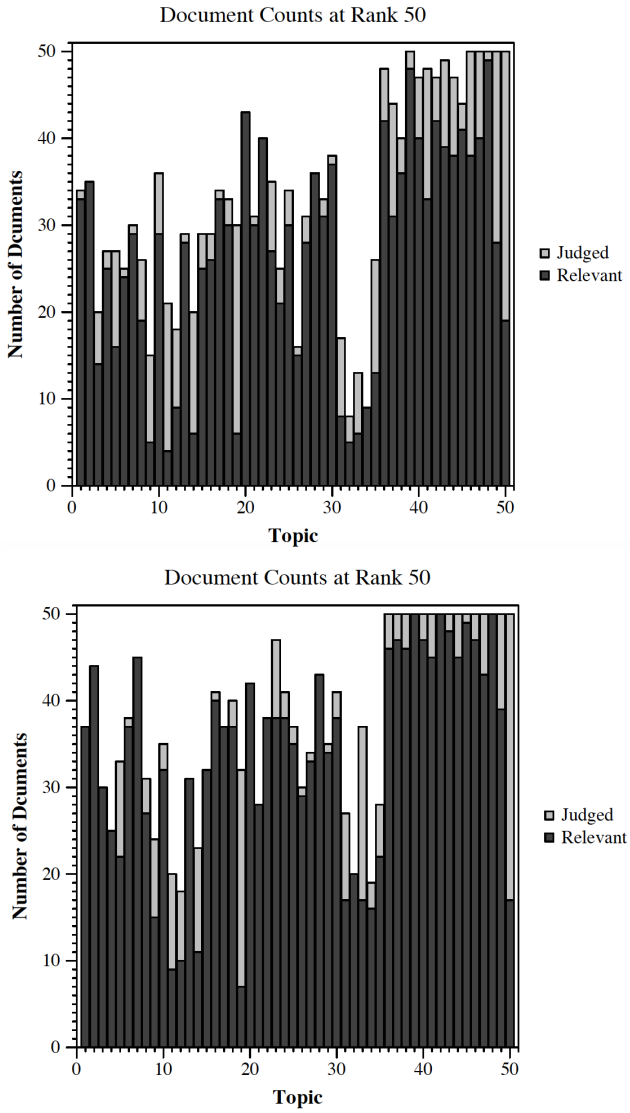


Fig. 6: Per topic proportion of relevant publications retrieved up to rank 50 to the total judged documents of round 5. (top) run2: *bm25-dfr-lmd-rrf* (bottom): run7: *ir-trf-logit-rrf*

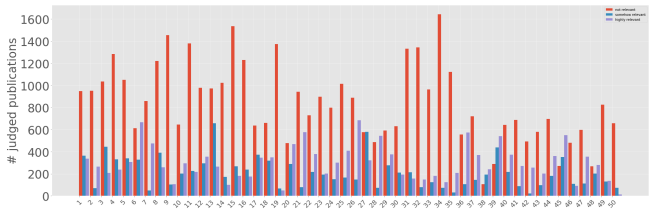


Fig. 7: Distribution of relevance judgments per topics of TREC-COVID. Among the huge amounts of publications, only the top retrievals of the participating teams for the 50 topics of the challenge are judged.

was essentially a small reduction in recall, but with a very

metric	*bm25-bs*	*bm25-dfr-lmd-rrf*	*trf-brx-rrf*	*ir-bdl-trf-brx-lm*	*ir-bdl-trf-brx-rrf*	*bdl-brx-logit*	*ir-trf-logit-rrf*
P@10	0.7200	0.7440	0.8740	0.8400	0.8800	0.8480	**0.8880**
NDCG@20	0.6320	0.6475	0.7716	0.7312	0.7826	0.7374	**0.7956**
MAP	0.2707	0.2778	0.3468	0.3068	0.3719	0.3439	**0.3789**
Bpref	0.5021	0.5174	0.5680	**0.5759**	0.5616	0.5719	0.5659

TABLE IV: Our submitted runs to round 5 of TREC-COVID. Refer to Fig. 2 and Table II for the description of different runs.

significant improvement of the precision.

However, in retrospect, we notice that another useful technique, which is very natural to an infodemic case could be to decay the score of publications by their distance to the present time, rather than a hard cut-off (at December 2019) . To see this, Fig. 8 sketches the distribution of the highly relevant documents judged for each of the 5 rounds of TREC-COVID.

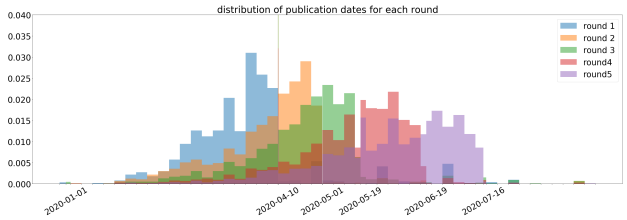


Fig. 8: Distribution of the publication dates of the "highly relevant" articles for each of the TREC-COVID rounds.

As is clear from the figure, more recent publications tend to have a higher probability of being more relevant to the information need. This is somehow expected in the case of a pandemic where new evidence is arriving with an explosive rate, possibly refuting the older knowledge.

VI. Conclusions

For the case of the COVID-19 pandemic, we presented our information retrieval pipeline to find the most relevant publications to different information needs. As the global pandemic continues, the amount of scientific publications grows in an unprecedented rate causing an infodemic within the many different disciplines involved. From the other hand, finding the most relevant answers to different information needs within this huge volume of data becomes of utmost necessity.

To help medical doctors and practitioners, as well as healthcare decision makers and other researchers, our information retrieval pipeline can provide a potential solution to this unique situation. We detailed the different components of this pipeline, including the traditional index-based information retrieval solutions, the modern NLP-based neural network solutions, as well as many insights and practical recipes to increase the quality of information retrieval of scientific publications targeted to the case of a pandemic. We grounded our results to the TREC-COVID challenge, where around 50 different teams participated in 5 rounds of competition. We showed very competitive results as judged by the official leaderboard of the challenge.

Apart from the case of COVID-19, we believe our solutions can also serve useful for other potential infodemics in the future.

References

[1] Milad Haghani, Michiel CJ Bliemer, Floris Goerlandt, and Jie Li, "The scientific literature on coronaviruses, covid-19 and its associated safety-related research dimensions: A scientometric analysis and scoping review," *Safety Science*, p. 104806, 2020.

[2] Lucy Lu Wang, Kyle Lo, Yoganand Chandrasekhar, Russell Reas, J. Yang, Darrin Eide, K. Funk, Rodney Michael Kinney, Ziyang Liu, W. Merrill, P. Mooney, D. Murdick, Devvret Rishi, Jerry Sheehan, Zhihong Shen, B. Stilson, A. Wade, K. Wang, Christopher Wilhelm, Boya Xie, D. Raymond, Daniel S. Weld, Oren Etzioni, and Sebastian Kohlmeier, "Cord-19: The covid-19 open research dataset," *ArXiv*, 2020.

[3] Kirk Roberts, Tasmeer Alam, Steven Bedrick, Dina Demner-Fushman, Kyle Lo, Ian Soboroff, Ellen Voorhees, Lucy Lu Wang, and William R Hersh, "TREC-COVID: rationale and structure of an information retrieval shared task for COVID-19," *Journal of the American Medical Informatics Association*, 07 2020, ocaa091.

[4] Ellen Voorhees, Tasmeer Alam, Steven Bedrick, Dina Demner-Fushman, William R Hersh, Kyle Lo, Kirk Roberts, Ian Soboroff, and Lucy Lu Wang, "TREC-COVID: Constructing a Pandemic Information Retrieval Test Collection," *arXiv e-prints*, p. arXiv:2005.04474, May 2020.

[5] Alberto Tonon, Gianluca Demartini, and Philippe Cudré-Mauroux, "Pooling-based continuous evaluation of information retrieval systems," *Information Retrieval Journal*, vol. 18, no. 5, pp. 445–472, 2015.

[6] Sean MacAvaney, Arman Cohan, and Nazli Goharian, "Sledge: A simple yet effective baseline for coronavirus scientific knowledge search," *arXiv preprint arXiv:2005.02365*, 2020.

[7] Edwin Zhang, Nikhil Gupta, Raphael Tang, Xiao Han, Ronak Pradeep, Kuang Lu, Yue Zhang, Rodrigo Nogueira, Kyunghyun Cho, Hui Fang, et al., "Covidex: Neural ranking models and keyword search infrastructure for the covid-19 open research dataset," *arXiv preprint arXiv:2007.07846*, 2020.

[8] Andre Esteva, Anuprit Kale, Romain Paulus, Kazuma Hashimoto, Wenpeng Yin, Dragomir Radev, and Richard Socher, "Co-search: Covid-19 information retrieval with semantic search, question answering, and abstractive summarization," 06 2020.

[9] Vincent Nguyen, Maciek Rybinski, Sarvnaz Karimi, and Zhenchang Xing, "Searching scientific literature for answers on covid-19 questions," *arXiv preprint arXiv:2007.02492*, 2020.

[10] Sarvesh Soni and Kirk Roberts, "An evaluation of publicly available deep learning based commercial information retrieval systems to search biomedical articles related to covid-19," *arXiv preprint arXiv:2007.03106*, 2020.

[11] Canjia Li, A. Yates, Sean MacAvaney, B. He, and Yingfei Sun, "Parade: Passage representation aggregation for document reranking," *ArXiv*, vol. abs/2008.09093, 2020.

[12] Stephen Robertson and Hugo Zaragoza, *The probabilistic relevance framework: BM25 and beyond*, Now Publishers Inc, 2009.

[13] John Lafferty and Chengxiang 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*, 2001, pp. 111–119.

[14] Thorsten Joachims, Hang Li, Tie-Yan Liu, and ChengXiang Zhai, "Learning to rank for information retrieval (lr4ir 2007)," in *Acm Sigir Forum*. ACM New York, NY, USA, 2007, vol. 41, pp. 58–62.

[15] Christopher JC Burges, "From ranknet to lambdarank to lambdamart: An overview," *Learning*, vol. 11, no. 23-581, pp. 81, 2010.

[16] Nick Craswell, Bhaskar Mitra, Emine Yilmaz, Daniel Campos, and Ellen M Voorhees, "Overview of the trec 2019 deep learning track," *arXiv preprint arXiv:2003.07820*, 2020.

[17] Erik Faessler, Udo Hahn, and Michel Oleynik, "Julie lab & med uni graz@ trec 2019 precision medicine track.," in *TREC*, 2019.

[18] Tao Qin, Tie-Yan Liu, Jun Xu, and Hang Li, "Letor: A benchmark collection for research on learning to rank for information retrieval," *Information Retrieval*, vol. 13, no. 4, pp. 346–374, 2010.

[19] Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng, "Ms marco: A human-generated machine reading comprehension dataset," 2016.

[20] Zhe Cao, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai, and Hang Li, "Learning to rank: from pairwise approach to listwise approach," in *Proceedings of the 24th international conference on Machine learning*, 2007, pp. 129–136.

[21] Ping Li, Qiang Wu, and Christopher J Burges, "Mcrank: Learning to rank using multiple classification and gradient boosting," in *Advances in neural information processing systems*, 2008, pp. 897–904.

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

[23] Zeynep Akkalyoncu Yilmaz, Shengjin Wang, Wei Yang, Haotian Zhang, and Jimmy Lin, "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*, 2019, pp. 19–24.

[24] Iz Beltagy, Kyle Lo, and Arman Cohan, "Scibert: A pretrained language model for scientific text," *arXiv preprint arXiv:1903.10676*, 2019.

[25] Nils Reimers and Iryna Gurevych, "Sentence-bert: Sentence embeddings using siamese bert-networks," *arXiv preprint arXiv:1908.10084*, 2019.

[26] Peilin Yang, Hui Fang, and Jimmy 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*, New York, NY, USA, 2017, SIGIR '17, p. 1253?1256, Association for Computing Machinery.

[27] Stephen E Robertson and K Sparck Jones, "Relevance weighting of search terms," *Journal of the American Society for Information science*, vol. 27, no. 3, pp. 129–146, 1976.

[28] Gianni Amati and Cornelis Joost Van Rijsbergen, "Probabilistic models of information retrieval based on measuring the divergence from randomness," *ACM Transactions on Information Systems (TOIS)*, vol. 20, no. 4, pp. 357–389, 2002.

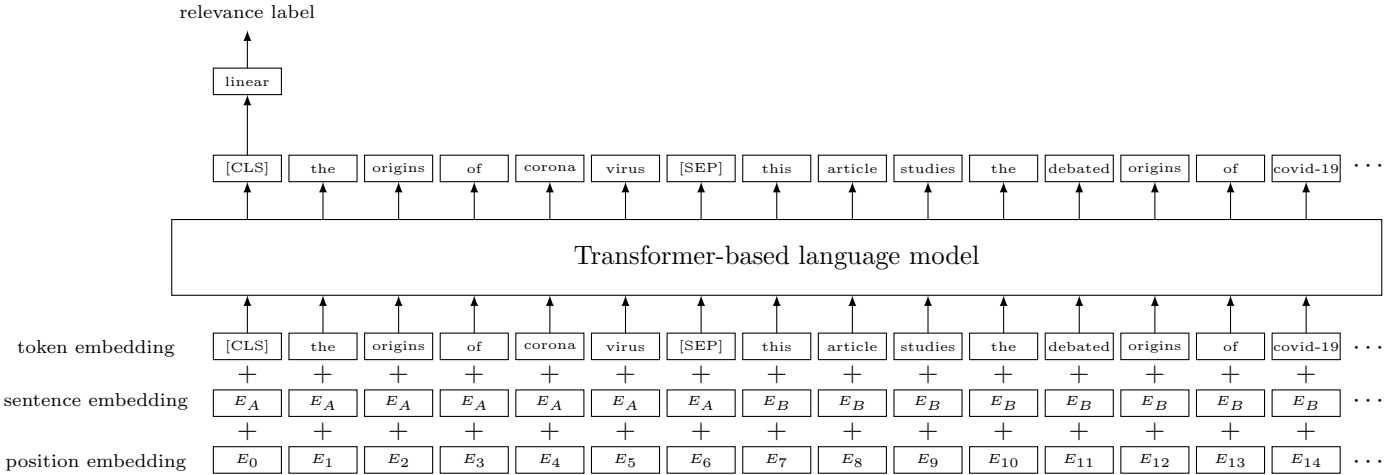
[29] Chengxiang Zhai and John Lafferty, "A study of smoothing methods for language models applied to ad hoc information retrieval," in *ACM SIGIR Forum*. ACM New York, NY, USA, 2017, vol. 51, pp. 268–276.

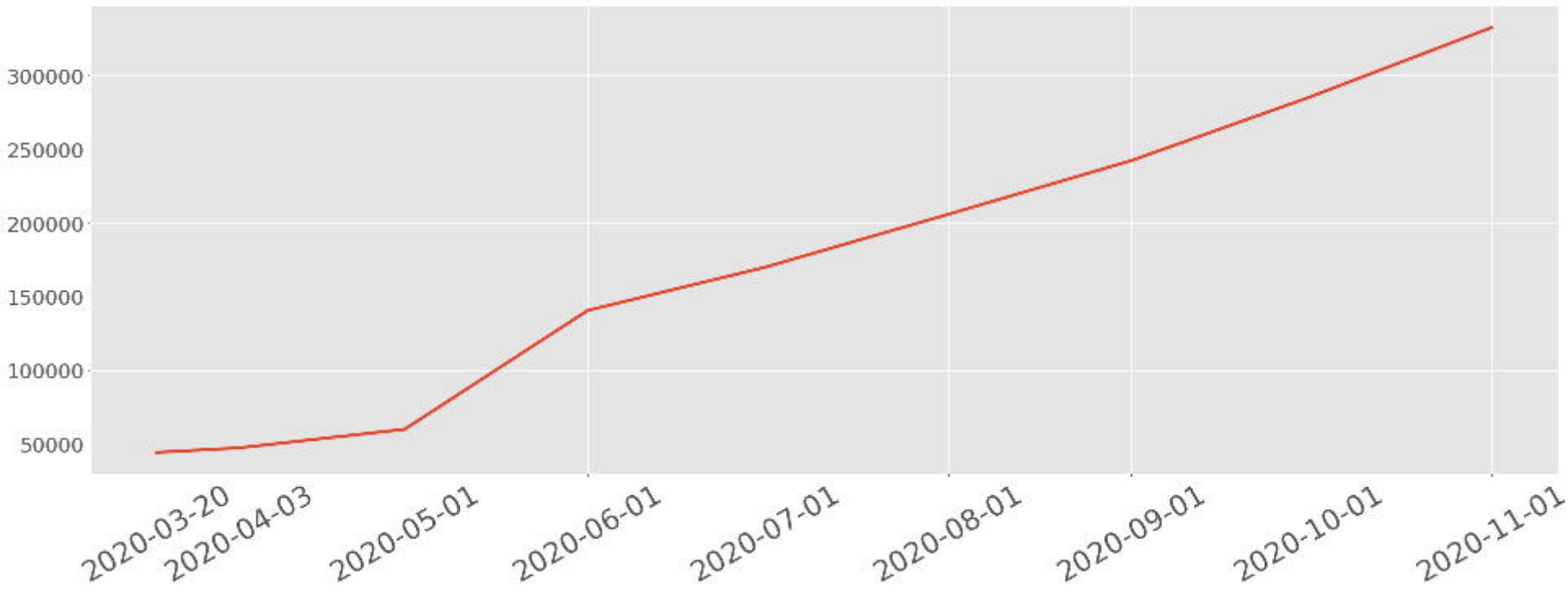
[30] Gordon V Cormack, Charles LA Clarke, and Stefan Buettcher, "Reciprocal rank fusion outperforms condorcet and individual rank learning methods," in *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, 2009, pp. 758–759.

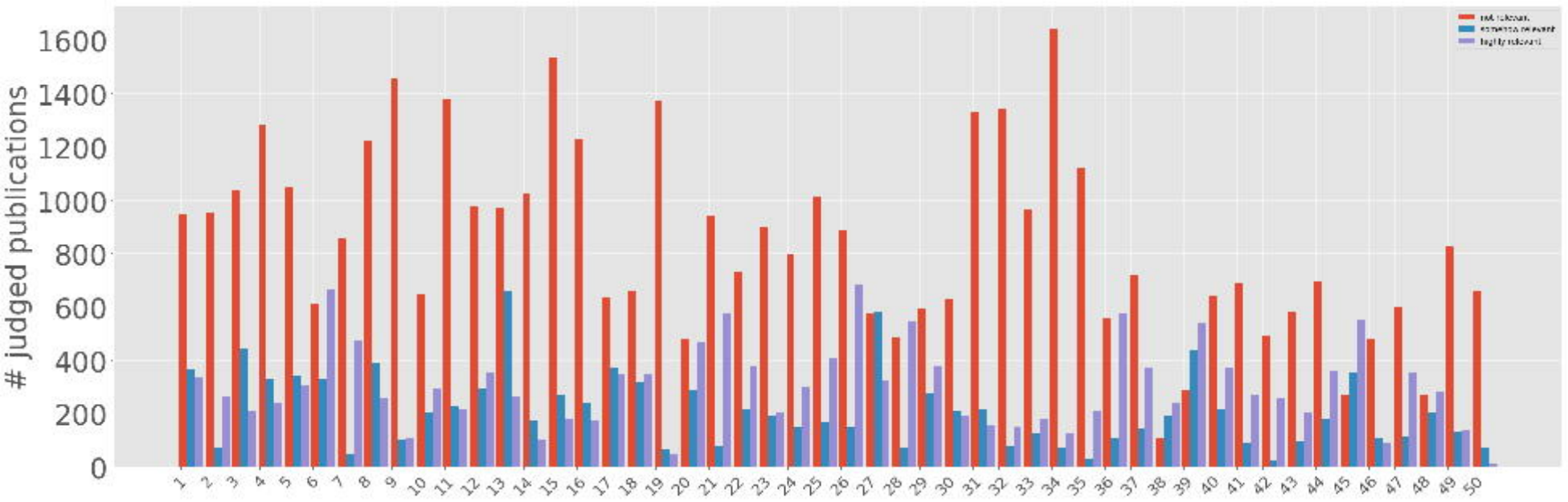
[31] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

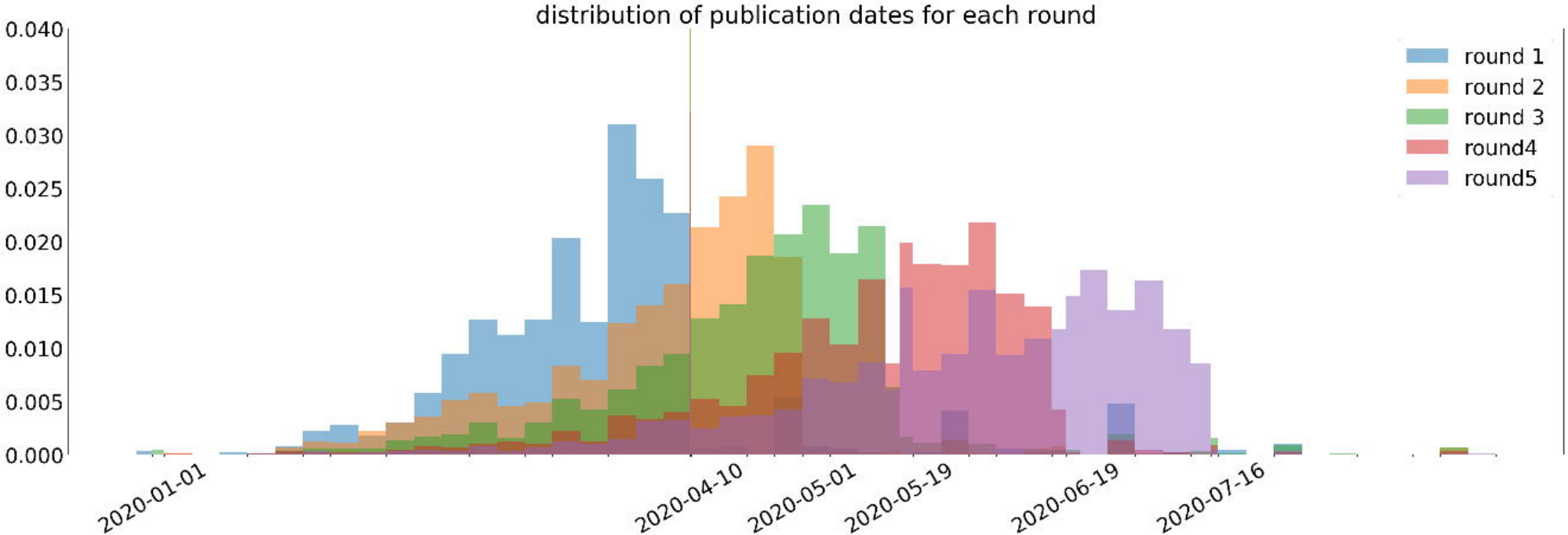
[32] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov, "Roberta: A robustly optimized bert pretraining approach," *arXiv preprint arXiv:1907.11692*, 2019.

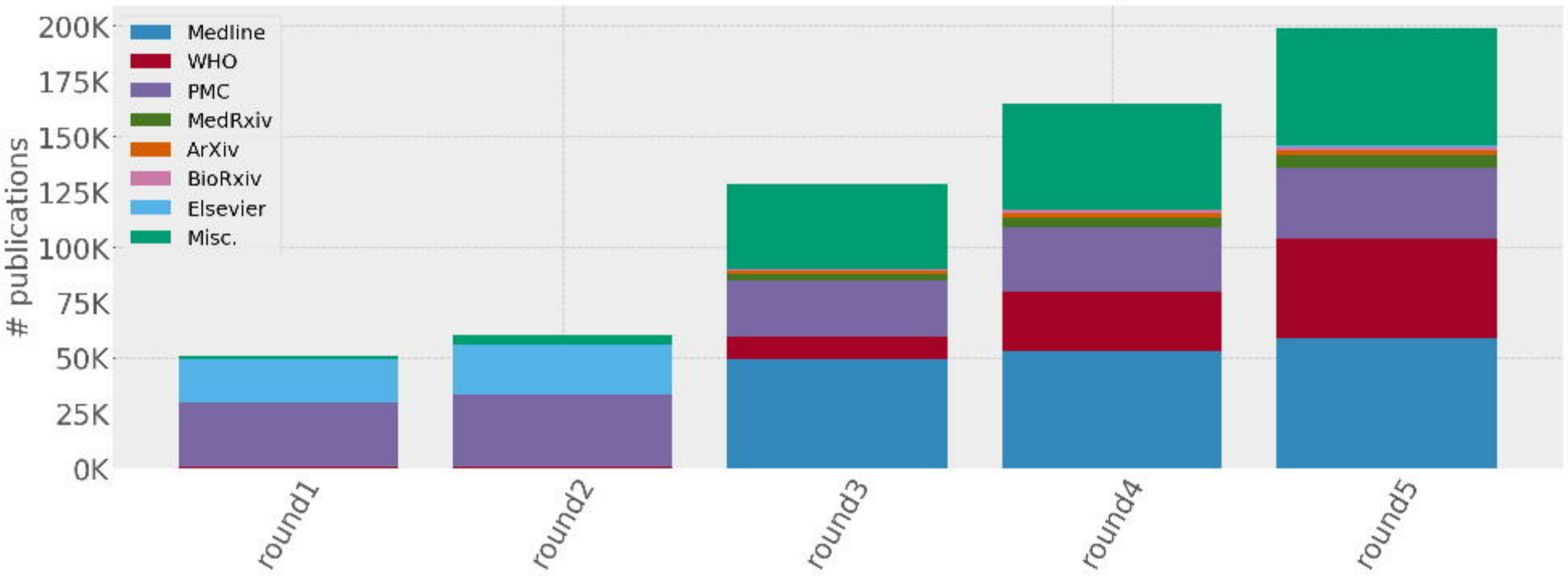
[33] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le, "Xlnet: Generalized autoregressive pretraining for language understanding," in *Advances in neural information processing systems*, 2019, pp. 5753–5763.

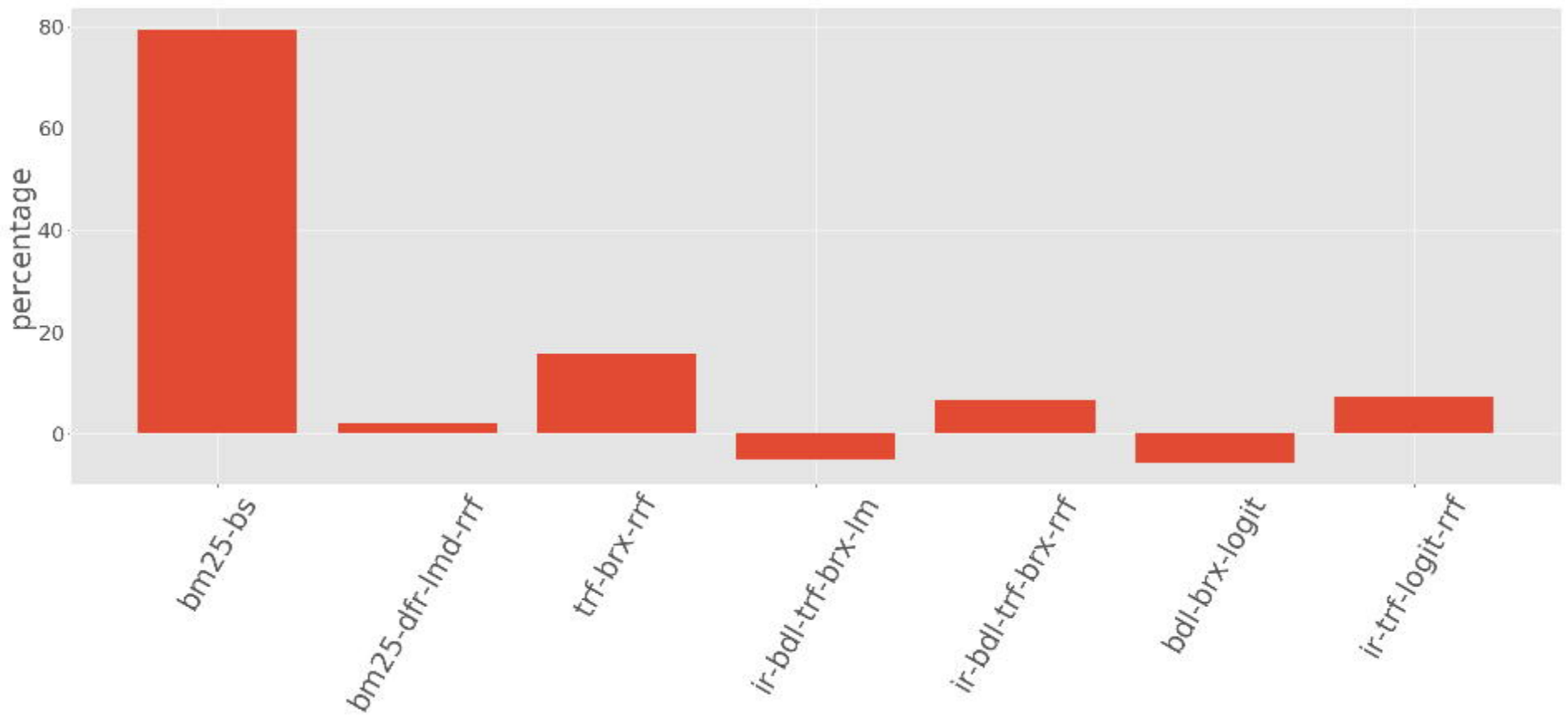

relevance label
linear
[CLS] the origins of corona virus [SEP] this article studies the debated origins of covid-19 ...
Transformer-based language model
token embedding
[CLS] the origins of corona virus [SEP] this article studies the debated origins of covid-19 ...
sentence embedding
E_A E_A E_A E_A E_A E_A E_A E_B E_B E_B E_B E_B E_B E_B E_B ...
position embedding
E_0 E_1 E_2 E_3 E_4 E_5 E_6 E_7 E_8 E_9 E_{10} E_{11} E_{12} E_{13} E_{14} ...

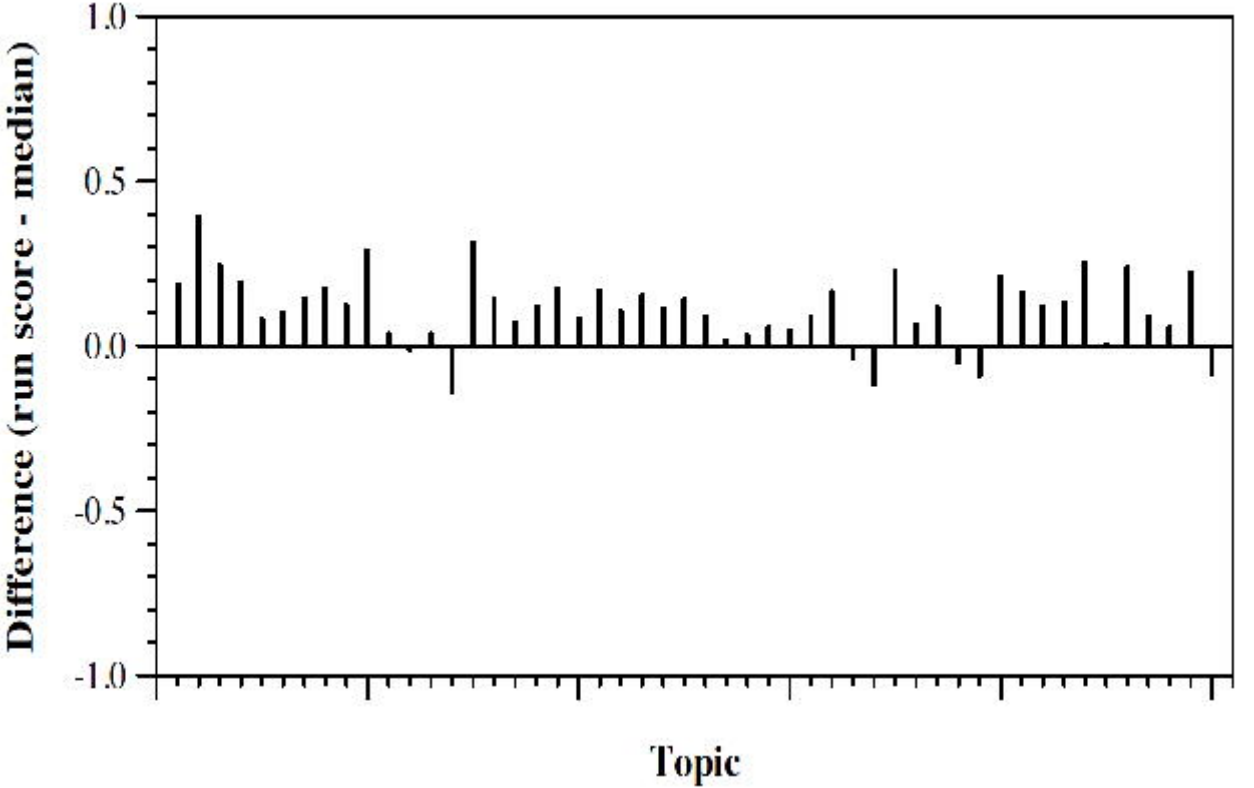

300000
250000
200000
150000
100000
50000
2020-03-20
2020-04-03
2020-05-01
2020-06-01
2020-07-01
2020-08-01
2020-09-01
2020-10-01
2020-11-01

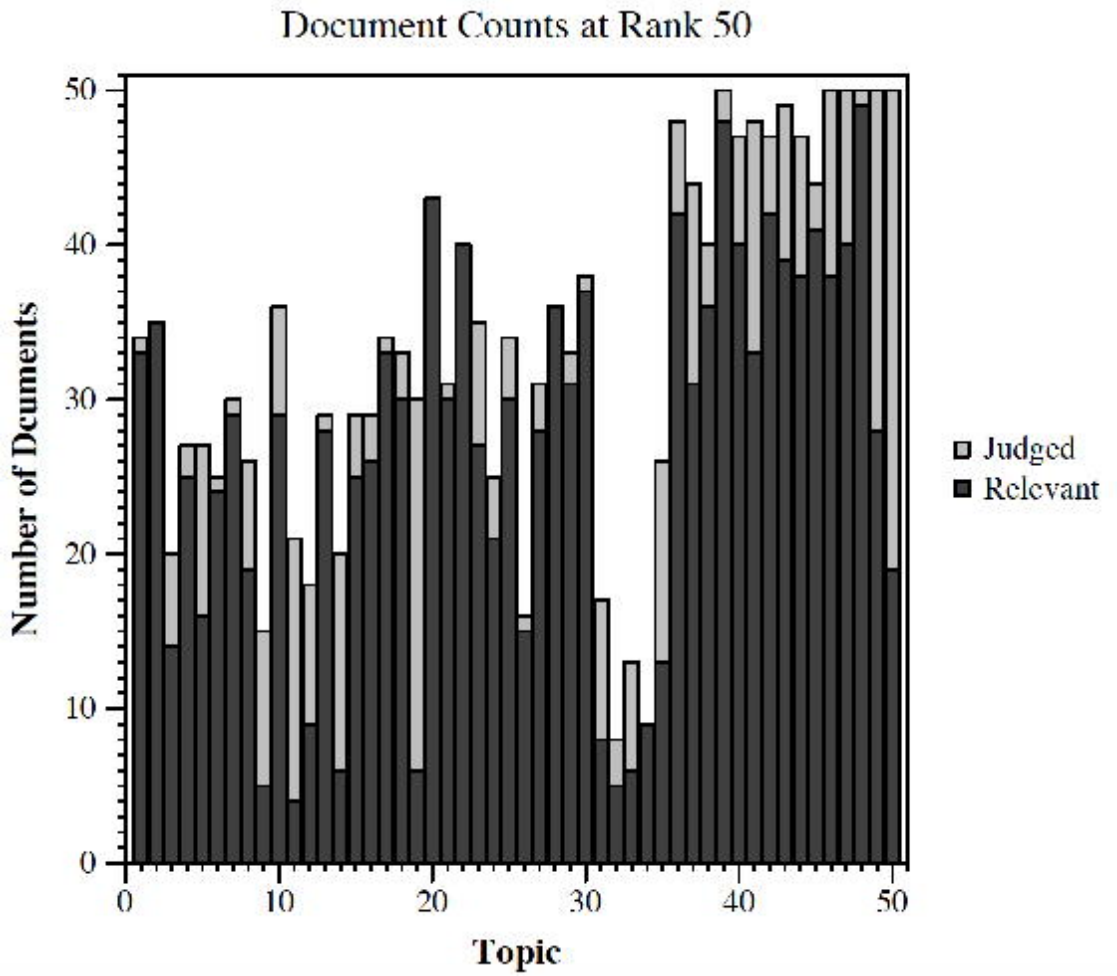

not relevant
somewhat relevant
highly relevant
judged publications
0
200
400
600
800
1000
1200
1400
1600
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50

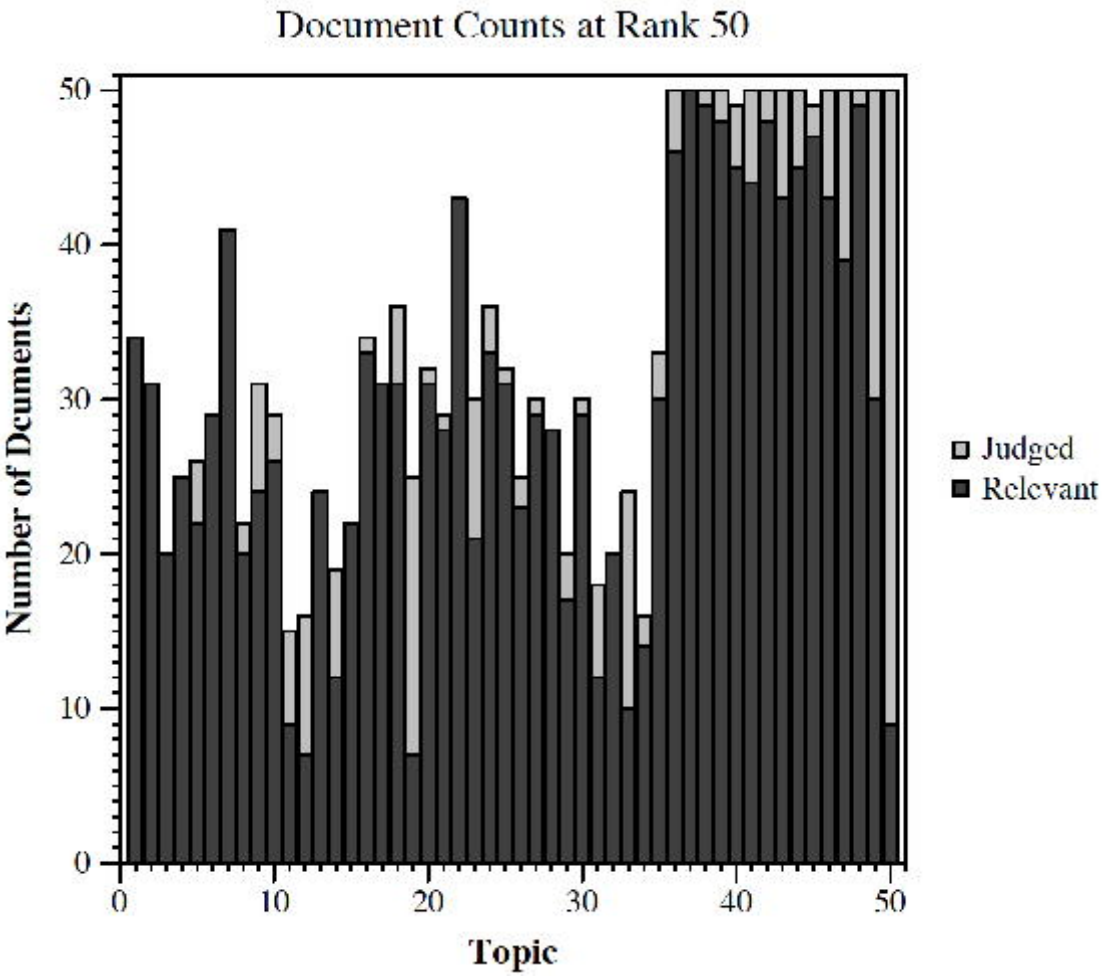

distribution of publication dates for each round
0.040
0.035
0.030
0.025
0.020
0.015
0.010
0.005
0.000
round 1
round 2
round 3
round4
round5
2020-01-01
2020-04-10
2020-05-01
2020-05-19
2020-06-19
2020-07-16

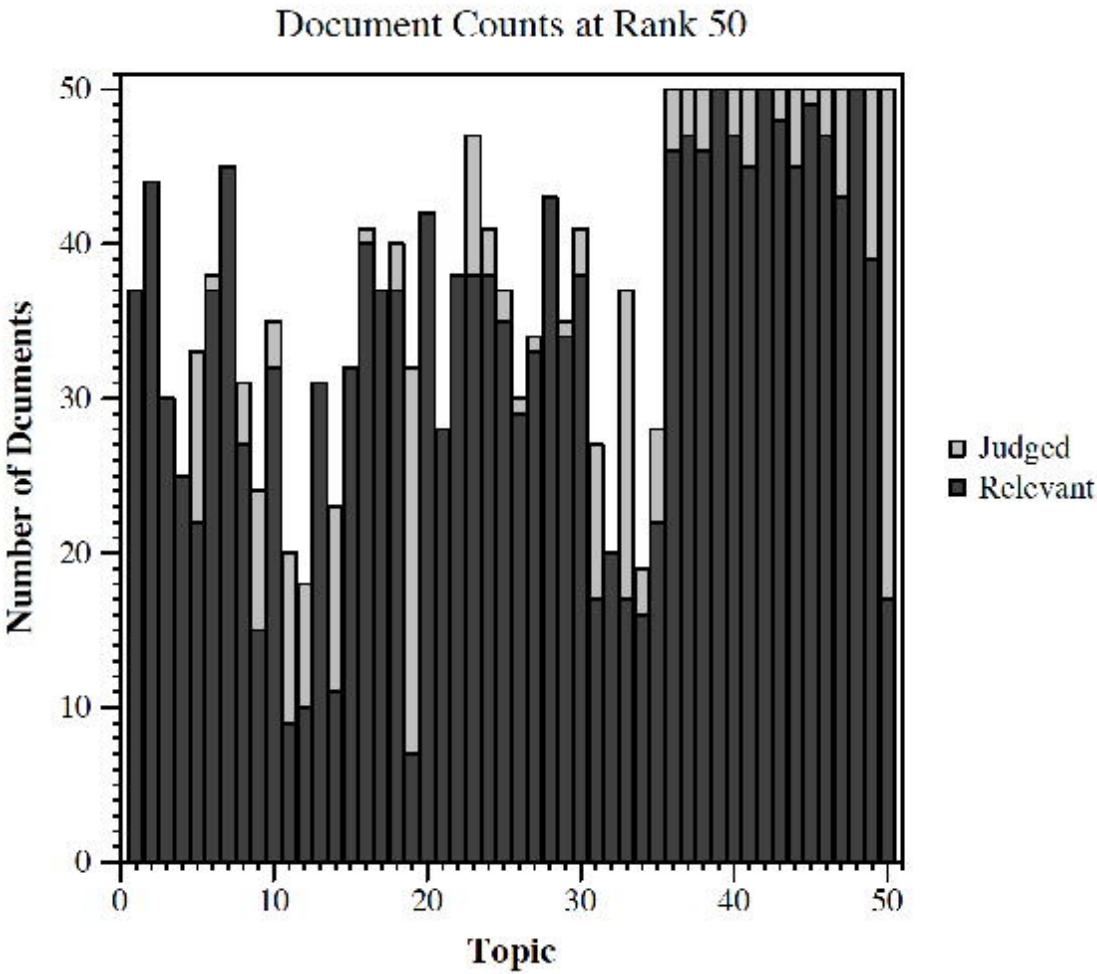

Medline
WHO
PMC
MedRxiv
ArXiv
BioRxiv
Elsevier
Misc.
publications
200K
175K
150K
125K
100K
75K
50K
25K
0K
round1
round2
round3
round4
round5

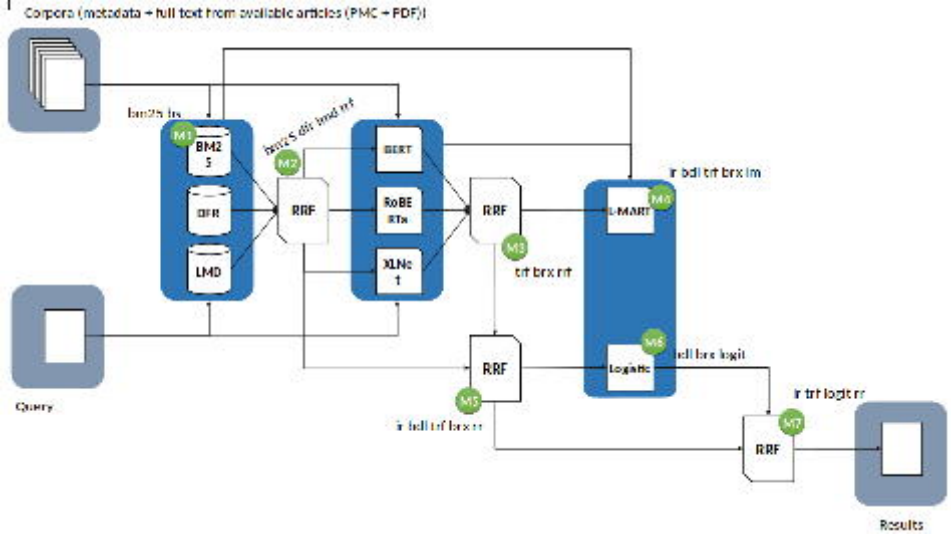

percentage
80
60
40
20
0
bm25-bs
bm25-dfr-lmd-rrf
trf-brx-rrf
ir-bdl-trf-brx-lm
ir-bdl-trf-brx-rrf
bdl-brx-logit
ir-trf-logit-rrf


1.0
0.5
0.0
-0.5
-1.0
Difference (run score - median)
Topic


Document Counts at Rank 50
Number of Documents
Topic
Judged
Relevant
0
10
20
30
40
50


Document Counts at Rank 50
Number of Documents
Topic
0
10
20
30
40
50
Judged
Relevant


Document Counts at Rank 50
Number of Documents
Topic
Judged
Relevant
0
10
20
30
40
50


Corpora (metadata + full text from available articles (PMC + PDF))
M1
BM25
DFR
LMD
M2
RRF
BERT
RoBERTa
XLNet
M3
RRF
L-MART
M4
Logistic
M5
RRF
M6
RRF
Query
Results