Multistage BiCross Encoder: Team GATE Entry for MLIA Multilingual Semantic Search Task 2

Iknoor Singh, Carolina Scarton, and Kalina Bontcheva

The University of Sheffield, United Kingdom
{i.singh, c.scarton, k.bontcheva}@sheffield.ac.uk

Abstract. The Coronavirus (COVID-19) pandemic has led to a rapidly growing 'infodemic' online. Thus, the accurate retrieval of reliable relevant data from millions of documents about COVID-19 has become urgently needed for the general public as well as for other stakeholders. The COVID-19 Multilingual Information Access (MLIA) initiative is a joint effort to ameliorate exchange of COVID-19 related information by developing applications and services through research and community participation. In this work, we present a search system called Multistage BiCross Encoder, developed by team GATE for the MLIA task 2 Multilingual Semantic Search. Multistage BiCross-Encoder is a sequential three stage pipeline which uses the Okapi BM25 algorithm and a transformer based bi-encoder and cross-encoder to effectively rank the documents with respect to the query. The results of round 1 show that our models achieve state-of-the-art performance for all ranking metrics for both monolingual and bilingual runs.

1 Introduction

The COVID-19 disease outbreak, which is declared as a global pandemic by World Health Organization (WHO), has infected more than 84M people worldwide to date. During this time, researchers and journalists published a lot of content related to health, COVID-19 prevention methods, new laws and policies among others. The information on online media comes from different parts of the world in multiple languages and it becomes really challenging for people to rapidly navigate through this information as the volume of data can be overwhelming. Thus, fast and accurate retrieval from the growing amount of information has become paramount during this crisis. In response to this need, the COVID-19 Multilingual Information Access (MLIA) initiative [2] is a joint effort to improve exchange of COVID-19 related information, across all EU languages and beyond. MLIA proposes three tasks which include Information Extraction (Task 1), Multilingual Semantic Search (Task 2) and Machine Translation (Task 3). In Task 2 [6], the goal is to create systems capable of searching

Covid-19 MLIA @ Eval Initiative, http://eval.covid19-mlia.eu/.

the growing information related to the novel coronavirus, in different languages and with different levels of knowledge about a specific topic. The task follows CLEF-style [17] evaluation process where participants are given a collection of documents and a set of topics which are used as a query to produce various runs. The dataset provided by the organisers contains a corpus of documents and a set of 30 topics, both available in multiple languages which include English, French, German, Greek, Italian, Spanish, Swedish and Ukranian. For our experiments, we considered English, French, German and Spanish documents. There are two subtasks: subtask 1 is focused on high precision whilst subtask 2 is oriented towards high-recall systems. Each participant team could submit both monolingual and bilingual runs where both topic and document are of different language.

In this paper, we analyse the participation of our team GateNLP in MLIA task 2. We propose a novel Multistage BiCross Encoder method and show that it achieves strong results in multilingual semantic search and document retrieval. In particular, this approach tops the leader board in round 1 of MLIA task 2. Our method follows a sequential three-stage ranking pipeline which include an initial lexical retrieval stage followed by two neural-based semantic retrieval stages. For lexical retrieval, we use Okapi BM25 to reduce the search space from a large number of documents (e.g. 1.4M in case of English documents) to a small set of possibly relevant documents. In the second stage, which we call the neural refinement phase, we leverage a transformer-based bi-encoder model to encode both query and document individually into contextualised representations [18] and use them to efficiently re-rank the documents using a sentence-pair scoring function such as cosine similarity. As the representations are separate, bi-encoder can store and reuse the encoded representation of inputs for faster predictions during inference. In the final neural re-ranking phase, we use a transformer-based cross-encoder [15] which performs full self-attention over query and document pair to get the relevance score which is used to rank the final list documents with respect to the query. In the final phase, we only re-rank a subset of top ranked documents from the neural refinement phase. This is because cross-encoder lacks the ability to make use of cached representations as it recomputes the encoding every time during inference which makes them slow and less useful for practical use when the number documents to be ranked is large. On the other side of the spectrum, cross-encoder tend to attain significantly higher accuracy when compared to bi-encoder [10], due to the rich interactions and cross self-attention over query and document pair. Since the performance gains come at a steep computational cost and hence, we use both bi-encoder and cross-encoder with former having more documents to re-rank as compared to the latter. We also explore various rank fusion techniques to combine the output from previous stages to get a single relevance score which is used to sort the final list of documents. Although our method is conceptually simple, the scores of round 1 show that our system is capable of twofold benefit of high precision at top ranked documents as well as high recall for all the retrieved documents. Following this section, we discuss the work related to our approach. Section 3 gives a more detailed description of Multistage BiCross Encoder and Section 6 contains the details of our runs. In Section 5, we present the evaluation results of our system and conclusion in Section 6.

2 Related Work

Neural models in information retrieval tasks are generally used in a two-stage pipeline architecture where the re-ranking is done only for top results retrieved by traditional ranking methods such TF-IDF of BM25 [14, 15, 23], as the cost of using neural models for the entire dataset is high [9]. Nogueira et al. [15] are the first to demonstrate that the BERT model can also be used for fine-tuning on passage re-ranking tasks and it has shown to be effective for ad-hoc document ranking. They use sequence of tokens by concatenating the query tokens and the passage tokens, separated by [SEP] token as an input to the BERT [5] model and then the output embedding of the [CLS] token is passed to a single layer neural network to obtain the probability of the passage being relevant to the query. In addition, people tried various methods to improve the effectiveness of neural models in document re-ranking tasks. Nogueira1 et al. [16] used a query generator model to expand the document before indexing to get additional gains on the retrieval performance. Yilmaz et al. [23] computed relevance scores using sentence-level evidence to rank large documents for ad hoc retrieval. Other work also uses rank fusion methods [4, 7] to combine various runs in order to improve the performance of retrieval system. Clipa et al. [3] analyse and compare various state-of-the-art information retrieval methods and ranking fusion approaches in case of medical publication retrieval.

Reimers et al. [18] show that transformer-based models such as vanilla BERT do not produce good sentence embedding and they present SBERT where they trained BERT-based models using Siamese network architecture to get semantically meaningful sentence representations. These can be leveraged for other tasks such as semantic search where the sentence representations can be compared using cosine-similarity. We explore a similar approach in the bi-encoder used in our neural refinement phase. Recently, the TREC-COVID challenge [19] invited participants to develop information retrieval systems for scientific literature containing tens of thousands of scholarly articles related to COVID-19. The participants of TREC-COVID used various neural models and methods for retrieving scientific documents [11, 24], some of which have been tried in this paper as described in Subsection 3.1 and 3.3.

3 Multistage BiCross Encoder

This section outlines the details of our multistage BiCross encoder method for document retrieval and semantic search. For retrieving and ranking the documents, we use a sequential three stage ranking pipeline which includes initial BM25 retrieval followed by neural refinement (bi-encoder) and neural re-ranking (cross-encoder) phase respectively. As far as sentence-pair scoring tasks are concerned, cross-encoder are more accurate than bi-encoder but they are compute-intensive and time-consuming when compared to bi-encoder which can make use of cached representations for faster inference. In our method, we use both bi-encoder and cross-encoder to effectively retrieve semantically similar documents from the corpus with respect to the query. The overview of our method is illustrated in Figure 1. The details of each phase can be found in the subsequent subsections.

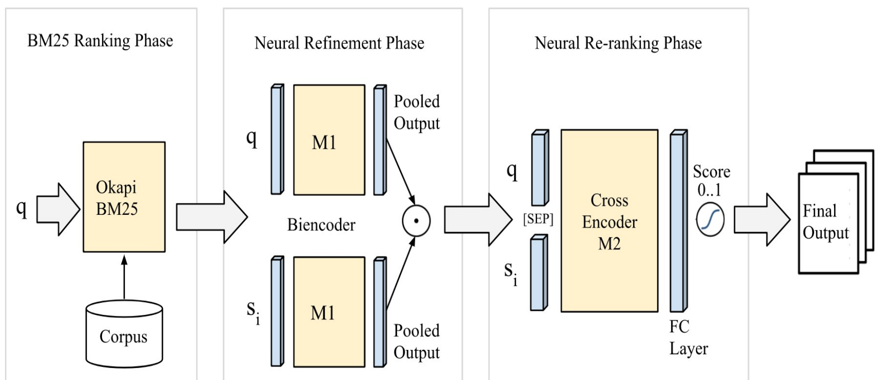

Fig. 1. Overview of Multistage BiCross Encoder, q is the query, s_i is the ith sentence of the document, Model $M1$ is used as bi-encoder and model $M2$ is used as cross-encoder.

3.1 BM25 Ranking Phase

First, we preprocess all the documents in the corpus and index them using Elasticsearch[1]. All the documents belonging to different language are indexed separately and in our case, we consider English, Spanish, French and German documents. The corpus provided by organisers contains documents in the form of XML files and we have only used the text inside the <p> tags and have excluded all the boilerplate tags. Text pre-processing methods such as stopwords removal and lemmatisation have been used before indexing the documents. Each topic in the topic file contains a description of the information need, in this case, it is composed of three fields: a keyword, i.e., a set of relevant keywords; a conversational, i.e., question asked in a verbal way; and a explanation, i.e., a more detailed description to the assessors when performing relevance assessments. For our runs, we use a concatenation of the keyword and conversational

[1] https://www.elastic.co/elasticsearch/

(*key_conv*) field as a query for our experiments. We also used this keyword and conversational field to generate three more queries using the T5-base doc2query model [16] and concatenated all to form a single query, called hereafter *t5_query*. Other than this, we also tried the Udels Query [24] from TREC-COVID task [19] as it gives a slight increase in performance and is used by many participants. The initial retrieval is done by using BM25 Okapi algorithm with default parameters ($k1 = 1.2$ and $b = 0.75$) as shown in Figure 1. This stage will filter out all the lexically dissimilar documents with respect to the query.

3.2 Neural Refinement Phase

In the second phase, the top 1,000 documents retrieved by BM25 are re-ranked using a bi-encoder which is based on Siamese networks. We use an approach similar to [18], where a pre-trained transformer-based model is used to encode both document and query separately into fixed length embeddings by using mean pooling on the output layer. In the same vector space, similar sentences lie in proximity of each other and can be efficiently retrieved using cosine similarity as shown in the neural refinement phase of Figure 1. Also, the encoded representations are cached and can be reused for faster predictions in future. Following [22, 23], each document in the corpus is split into sentences, and we apply inference on each sentence separately. As documents in the MLIA dataset are large, we only considered first 30 sentences for inference to reduce overall computations and also research [8, 12] shows that any relevant document is likely to contain relevant sentences at the beginning of the document. The score of each document is determined by combining the top k scoring sentences in the document as follows: $S_{doc} = \sum_{i=1}^{k} w_i \cdot S_{Bi_i}$, where $k = 3$ and S_{Bi_i} is the i-th top sentence score according to the model. The parameters w_i can be tuned via exhaustive grid search but due to lack of relevance labels for round 1, we have set $w_1 = 1, w_2 = 0.9$ and $w_3 = 0.8$. We choose these as initial parameters such that $w_1 > w_2 > w_3$, because we want to give more weight to the sentence which is more relevant as compared to the less relevant sentences. In other words, high scoring sentences contribute more to the final relevance score of the document.

Since there are no relevance labels provided by the organisers, external datasets were used to train our model. We prepared TC+IFCN data which is a combined version of the TREC-COVID (TC) [19] dataset and the IFCN dataset [20]. TREC-COVID dataset has 69,318 relevance judgement scores of 50 topics. The IFCN data consists of around 7k claims debunked by IFCN (International Fact-Checking Network). Here we use each claim as the query and the debunked page text as the document. We prepare a sentence-level dataset in which for each query, we used top two or three sentences similar to the query, from the document. We also developed a Cross_TC+IFCN, i.e. a multilingual version of the TC+IFCN data where we translated it to Spanish, French and German using OPUS-MT [21]. For training the bi-encoder, we utilised the models provided by the sentence-transformers[2] library, which includes BERT-based

[2] `https://github.com/UKPLab/sentence-transformers`

models [18] fine-tuned using siamese and triplet networks to get semantically meaningful sentence representations. We further fine-tuned the SBERT models on our domain specific dataset. The details of the models used in our experiments is as follows

- For monolingual English runs, we tried two models which include msmarco-distilroberta-base-v2, RoBERTa [13] base model trained on MSMARCO passage ranking dataset and stsb-roberta-large, a RoBERTa large model trained on natural language inference and semantic textual similarity dataset. We used these as base models to fine-tune on the TC+IFCN dataset with regression objective function [18]. The final models are referred as TCIN-msmarco-distilroberta-base and TCIN-stsb-roberta-large.
- For bilingual runs, we used a pretrained xlm-r-distilroberta-base-paraphrase-v1 [18], a xlm-roberta-base model trained on a large scale paraphrase dataset of more than 50 languages. Transfer learning was used to fine-tune this model on Cross_TC+IFCN dataset using the same objective function as mentioned above and the final multilingual model is named as CrossTCIN-xlm-r-paraphrase.

We call this stage as neural refinement phase as it helps to filter out all the semantically unrelated documents and it also works much faster as compared to a cross-encoder-based approach where a pair of sentences are passed together to the model every time during inference.

3.3 Neural Re-ranking Phase

In the third phase, top 400 documents retrieved by neural refinement phase are re-ranked using a cross-encoder architecture. In this, both query tokens and the document tokens separated by [SEP] token are passed to the transformer-based model to perform full self-attention over the given input and the output of [CLS] token is passed to the linear layer with sigmoid activation to get relevance scores from 0 to 1 [15] as illustrated in neural re-ranking phase of Figure 1. Similar to the neural refinement phase, here we also split the document into sentences and apply sentence-level inference with the query. The final score of each document is determined by combining the top k scoring sentences of the document i.e. $S_{doc} = \sum_{i=1}^{k} w_i \cdot S_{Cross_i}$, where S_{Cross_i} is the i-th top sentence score according to the cross-encoder model and all other parameters are kept the same as before. The description of models used as cross-encoder is as follows

- For monolingual English runs, we used ELECTRA model fine-tuned on the MSMARCO dataset from [11] as it performed well and was among the top positions in the second round of TREC-COVID task. The model was further fine-tuned on the TC+IFCN data using binary cross-entropy loss. We call this model as TCIN-electra-msmarco.
- For bilingual runs, as of now, there does not exist any multilingual model trained on MSMARCO passage ranking dataset. Hence, we used state-of-the-art multilingual transformer-based models such as xlm-roberta-base [13]

and distilbert-base-multilingual [5] as the base model for fine-tuning on the MSMARCO passage dataset for a epoch with a learning rate of 1e-5, batch size of 16 and a maximum of 512 input sequence length. The models are further fine-tuned on Cross_TC+IFCN dataset using sigmoid cross-entropy loss and the final models are referred to as CrossTCIN-xlm-roberta-msmarco and CrossTCIN-distilbert-multilingual-msmarco.

In case of bilingual runs, where query and the documents are in different language, we simply use Google Translate to translate the query into the target language of the document, and apply exactly the same methods as described above in a monolingual setting. For some runs, we also combined scores from previous stages using various rank fusion algorithms. These can be broadly classified into score-based and rank-based fusion algorithms. For score-based method, we used weighted CombSUM which is a slight modification of CombSUM [7] algorithm where we add the weighted scores from different ranking models. The equation is as follows

$$wCombSUM(d) = \alpha \cdot \sum_{i=1}^{k} w_i \cdot S_{Cross_i} + \beta \cdot \sum_{i=1}^{k} w_i \cdot S_{Bi_i} + (1 - \alpha - \beta) \cdot S_{BM25} \quad (1)$$

S_{BM25}, S_{Bi_i} and S_{Cross_i} are the min-max normalised scores of document d from first , second and third phase respectively. The parameters α and β are such that $\alpha > \beta$ and for the round 1 we have fixed $\alpha = 0.5$ and $\beta = 0.4$, giving more weight to cross-encoder ranking followed by bi-encoder and BM25 ranking respectively. $wCombSUM(d)$ is the weighted CombSUM score of the document. For rank-based fusion methods, we tried Reciprocal Rank Fusion (RRF) [4] and Borda Fusion [1]. In this, the fused score of the document simply rely on the rank of document from different ranking models and in our work, we only used the output of second and third neural phase. The equation of RRF and Borda fusion is as follows

$$RRFScore(d) = \frac{1}{k + R_{Cross}(d)} + \frac{1}{k + R_{Bi}(d)} \quad (2)$$

$$BordaScore(d) = \frac{N - R_{Cross}(d) + 1}{N} + \frac{N - R_{Bi}(d) + 1}{N} \quad (3)$$

where $R_{Bi}(d)$ and $R_{Cross}(d)$ are the ranks of the document d from neural refinement and neural re-ranking phase. For the RRF method, we set the constant $k = 60$ default as mentioned in their respective paper [4]. N is the total number of documents during fusion.

4 Implementation Details

We implemented our approach to check its effectiveness in both monolingual and bilingual semantic search setting. For the round 1, we submitted a total of 22 monolingual runs and 15 bilingual runs. In these runs, we tried different models, type of query and rank fusion methods. The description of all the runs is given below. All our experiments were conducted on NVIDIA Titan RTX GPU.

- gatenlp_run1 / gatenlp_run2 / gatenlp_run3 : In run 1, Udels method was used to generate the query, *t5_query* was used in run 2 and a concatenation of the keyword and conversational (*key_conv*) field in run 3. The bi-encoder is TCIN-msmarco-distilroberta-base and cross-encoder is TCIN-electra-msmarco. The encoder models are kept same in all the runs to see which type of query gives the best results.
- gatenlp_run5 / gatenlp_run7 : In run 5 and run 7, we used a different bi-encoder i.e. TCIN-stsb-roberta-large and the cross-encoder model is kept same as for the previous runs. The only difference between both the runs is that for run 7 we used Udels query and for run 5 we used *key_conv* as query.
- gatenlp_run8 / gatenlp_run10 : In these runs, we retrieved 1000 documents for each query. Run 8 just retrieves the output of the bi-encoder and run 10 fuses the scores from both bi-encoder and cross-encoder using Reciprocal Rank Fusion. Here, bi-encoder is TCIN-msmarco-distilroberta-base and cross-encoder is TCIN-electra-msmarco.
- gatenlp_es_run25 / gatenlp_fr_run26 / gatenlp_de_run27 / gatenlp_es_run28 / gatenlp_fr_run29 / gatenlp_de_run30 : These are monolingual Spanish, French and German runs. Here, we use multilingual model where bi-encoder is CrossTCIN-xlm-r-paraphrase and cross-encoder is CrossTCIN-distilbert-multilingual-msmarco. In case of run 25, 26 and 27, CombSum_wtd fusion was used whereas for run 28, 29 and 30, RRF fusion was used to get the final ranked documents. In all these runs and the following ones, we used *key_conv* as query to retrieve the documents.
- gatenlp_es_run31 / gatenlp_fr_run32 / gatenlp_de_run33 / gatenlp_es_run34 / gatenlp_fr_run35 / gatenlp_de_run36 / gatenlp_es_run37 / gatenlp_fr_run38 / gatenlp_de_run39 : In the above runs, we used XLM-based model i.e. CrossTCIN-xlm-roberta-msmarco as a cross-encoder and CrossTCIN-xlm-r-paraphrase as bi-encoder. Here, we used rank-based fusion where run 31, 32 and 33 uses RRF and run 34, 35 and 36 used Borda fusion. For run 37, 38 and 39, we directly use the output of cross-encoder to rank the final list of documents.
- gatenlp_en2es_run40 / gatenlp_en2fr_run41 / gatenlp_en2de_run42 / gatenlp_en2es_run43 / gatenlp_en2fr_run44 / gatenlp_en2de_run45 / gatenlp_en2es_run46 / gatenlp_en2fr_run47 / gatenlp_en2de_run48 : These are all bilingual runs where both query and the retrieved documents are in different language. Here again we use multilingual models where the bi-encoder is CrossTCIN-xlm-r-paraphrase and cross-encoder is CrossTCIN-xlm-roberta-msmarco. For run 40, 41 and 42, RRF is used to calculate the final relevance score. Apart from these, all other runs just use the output of cross-encoder but we use *key_conv* in run 43, 44 and 45 and Udels query in run 46, 47 and 48.
- gatenlp_en2es_run49 / gatenlp_en2fr_run50 / gatenlp_en2de_run51 / gatenlp_en2es_run52 / gatenlp_en2fr_run53 / gatenlp_en2de_run54 : For these

runs, we use bi-encoder as CrossTCIN-distilbert-multilingual-msmarco and cross-encoder as CrossTCIN-xlm-roberta-msmarco. Run 49, 50 and 51 uses Udels query and run 52, 53 and 54 uses *key_conv* as query for retrieving the documents.

5 Evaluation and Results

In this section, we explore the effectiveness of our approach using various ranking metrics. We evaluate the performance of Multistage BiCross Encoder using the relevance judgements provided by the MLIA organisers for the round 1. All the submitted runs are evaluated using Recall, Precision at 5 (P@5), P@10, R-Precision (RPrec), Average Precision (AP), Normalized Discounted Cumulative Gain (nDCG) and NDCG@10. In the following subsections, we discuss our results for both monolingual and bilingual runs.

5.1 Monolingual Runs

Table 1 shows the results of the monolingual English runs. From the table of results, all gatenlp runs outperform all other participant runs in all metrics by a significant margin. In our runs, gatenlp_run5 performs best followed by gatenlp_run3 and other runs shown in the table. The gatenlp_run5 uses *key_conv* as a query and TCIN-stsb-roberta-large as a bi-encoder model. This suggests that the use of model trained on semantic text similarity data proved to be beneficial when compared to the ones trained on MSMARCO dataset. Regarding the type of query, the results show that the use of *key_conv* query achieves large gains as compared to Udels query and *t5_query* method. Furthermore, the gatenlp_run1 performs best in case of recall, demonstrating the effectiveness Udels query for recall oriented task. The second half of Table 1 contains the runs which retrieve 1000 documents per query as these were a part of subtask 2. In this case also, our gatenlp_run10 achieves the best results for most of the metrics. It uses RRF to fuse the scores from bi-encoder and cross-encoder. Except this, run ims_nlex achieves the highest scores for P@10 and NDCG@10.

In monolingual Spanish runs (Table 2), again, our runs outperformed all other submissions, in most cases to a statistically significant degree. gatenlp_run37 scores highest in Precision whereas gatenlp_run25 gave the best results for NDCG and recall. We also find that MAP and Rprec are highest for gatenlp_run31. Similarly, Table 3 and Table 4 shows the results for the monolingual French runs and monolingual German runs. Overall, we find that the runs which use weighted CombSUM on model CrossTCIN-xlm-r-paraphrase (bi-encoder) and CrossTCIN-distilbert-multilingual-msmarco (cross-encoder) give comparatively high Recall and NDCG scores. Also, the runs that use CrossTCIN-xlm-roberta-msmarco as cross-encoder give highly competent scores. Additionally, we also compared the performance of multilingual models on different languages and we found that for most of the metrics, the scores of German runs are comparatively higher, followed by French and Spanish runs respectively.

Run ID	P@5	P@10	MAP	NDCG@10	NDCG	Rprec	Recall
gatenlp_run5	**0.9333**	**0.9000**	**0.2944**	**0.8331**	**0.5187**	**0.3486**	0.4382
gatenlp_run3	0.9200	0.8900	0.2912	0.8223	0.5155	0.3484	0.4375
gatenlp_run2	0.9000	0.7967	0.2560	0.7775	0.4925	0.3215	0.4278
gatenlp_run1	0.8867	0.8633	0.2776	0.8139	0.5067	0.3310	**0.4411**
gatenlp_run7	0.8667	0.8800	0.2719	0.8212	0.5014	0.3305	0.4292
CUNIMTIR_Run1	0.5933	0.4800	0.1145	0.4254	0.2802	0.1976	0.2613
CUNIMTIR_Run3	0.3600	0.3233	0.0609	0.2712	0.1444	0.1046	0.1278
CUNIMTIR_Run4	0.3533	0.3267	0.0530	0.2688	0.1422	0.0940	0.1239
ims_bm25_1k	0.3067	0.2433	0.0688	0.2391	0.2418	0.1579	0.2595
ims_bm25_2k	0.2400	0.1833	0.0478	0.1744	0.1789	0.1277	0.2028
ims_bm25_3k	0.2067	0.1633	0.0396	0.1413	0.1582	0.1075	0.1930
ims_bm25_4k	0.1933	0.1533	0.0367	0.1546	0.1483	0.1037	0.1677
gatenlp_run10	**0.9200**	0.8767	**0.3427**	0.8108	**0.6255**	**0.3711**	0.6334
ims_nlex	0.8933	**0.9000**	0.3055	**0.8365**	0.5740	0.3408	0.5593
gatenlp_run8	0.8867	0.8533	0.2999	0.8126	0.6092	0.3295	0.6334
ims_c-bm25	0.8600	0.8267	0.2771	0.7592	0.5945	0.3089	0.6482
ims_v-csum	0.8533	0.8233	0.2999	0.7693	0.6092	0.3450	**0.6516**
ims_bm25	0.7200	0.6900	0.2269	0.6202	0.5264	0.2673	0.6079
CUNIMTIR_Run5.	0.6867	0.6900	0.1908	0.5780	0.4574	0.2364	0.5160
CUNIMTIR_Run1	0.6800	0.5033	0.1659	0.4928	0.4450	0.2223	0.5148
ims_nsle	0.5067	0.5133	0.1595	0.4084	0.4145	0.2205	0.4837
CUNIMTIR_Run3	0.4800	0.3367	0.0882	0.2944	0.2500	0.1221	0.2986
CUNIMTIR_Run2	0.4667	0.3033	0.0646	0.3005	0.2379	0.1163	0.2683
CUNIMTIR_Run4	0.4267	0.3400	0.0658	0.2809	0.2200	0.1051	0.2662

Table 1. Performance of different monolingual English runs. Best results are bolded.

5.2 Bilingual Runs

Table 5 compares the performance of our different bilingual runs. These include English to Spanish (en2es), English to French (en2fr) and English to German (en2de) runs. In all the runs, we find that runs that which use RRF where bi-encoder is CrossTCIN-xlm-r-paraphrase and cross-encoder is CrossTCIN-xlm-roberta-msmarco give comparatively higher scores for all the metrics. In addition, if we compare run 43, 44 and 45 with run 46, 47 and 48, we see that the former runs which use *key_conv* as query give better results than the latter ones which use Udels query. We see similar results for run 49, 50 and 51 which use Udels query and run 52, 53 and 54 which use *key_conv* as query for retrieving the documents. For English to German runs, gatenlp-en2de-run42 has the highest scores for all the metrics. Apart from this, we couldn't find any single model which perform good for all the languages as different models and methods give distinct results for different metrics as shown in Table 5.

Run ID	P@5	P@10	MAP	NDCG@10	NDCG	Rprec	Recall
gatenlp_run37	**0.8333**	**0.7933**	0.2043	0.7263	0.3705	0.2806	0.3086
gatenlp_run25	0.8133	0.7767	0.2154	0.7455	**0.3808**	0.2795	**0.3111**
gatenlp_run28	0.8067	0.7767	0.2113	**0.7478**	0.3768	0.2758	0.3086
gatenlp_run34	0.7933	0.7833	0.2173	0.7383	0.3769	0.2858	0.3086
gatenlp_run31	0.7933	0.7867	**0.2246**	0.7362	0.3790	**0.2873**	0.3086
sinai_sinai1	0.5200	0.4867	0.0900	0.4629	0.2177	0.1557	0.1767
sinai_sinai2	0.4400	0.4067	0.0631	0.3868	0.1835	0.1284	0.1537
sinai_sinai4	0.3600	0.3067	0.0535	0.2904	0.1738	0.1243	0.1594
sinai_sinai3	0.2267	0.1733	0.0284	0.1820	0.1121	0.0786	0.1011
sinai_sinai5	0.2267	0.1733	0.0155	0.1832	0.0634	0.0407	0.0444
ims_bm25_1k	0.2067	0.1867	0.0577	0.1812	0.1944	0.1366	0.2142
ims_bm25_2k	0.2000	0.1800	0.0591	0.1745	0.2003	0.1402	0.2275
ims_bm25_3k	0.1733	0.1433	0.0444	0.1359	0.1744	0.1196	0.2072
ims_bm25_4k	0.0867	0.0800	0.0309	0.0793	0.1535	0.1046	0.1900
ims_c-bm25	**0.7000**	0.6933	0.1654	0.6346	**0.3993**	0.2224	**0.4084**
ims_v-csum	0.6867	**0.7133**	0.1697	**0.6604**	0.3797	0.2171	0.3612
ims_csum	0.6800	0.6200	**0.1720**	0.5822	0.3769	**0.2259**	0.3779
ims_bm25	0.6133	0.5800	0.1458	0.5263	0.3540	0.2020	0.3740
sinai_sinai1	0.5200	0.4867	0.1000	0.4629	0.2839	0.1560	0.2928
sinai_sinai2	0.4400	0.4067	0.0715	0.3868	0.2436	0.1285	0.2618
sinai_sinai4	0.3600	0.3067	0.0626	0.2904	0.2368	0.1247	0.2689
sinai_sinai5	0.2267	0.1733	0.0157	0.1832	0.0693	0.0408	0.0550
sinai_sinai3	0.2267	0.1733	0.0342	0.1820	0.1644	0.0788	0.1906

Table 2. Performance of different monolingual Spanish runs. Best results are bolded.

Run ID	P@5	P@10	MAP	NDCG@10	NDCG	Rprec	Recall
gatenlp_run26	**0.8800**	**0.7533**	**0.3505**	**0.7490**	**0.5672**	**0.3773**	**0.5267**
gatenlp_run29	0.8600	0.7400	0.3302	0.7324	0.5406	0.3651	0.4926
gatenlp_run32	0.8133	0.7367	0.3161	0.7180	0.5297	0.3593	0.4926
gatenlp_run35	0.8133	0.7267	0.3125	0.7116	0.5268	0.3541	0.4926
gatenlp_run38	0.7867	0.6400	0.2752	0.6436	0.5030	0.3269	0.4926

Table 3. Performance of different monolingual French runs. Best results are bolded.

Run ID	P@5	P@10	MAP	NDCG@10	NDCG	Rprec	Recall
gatenlp_run30	**0.9067**	**0.8767**	0.4537	**0.8234**	0.6403	0.4794	0.6253
gatenlp_run27	0.9000	0.8667	**0.4629**	0.8211	**0.6488**	0.4858	**0.6339**
gatenlp_run36	0.8733	0.8267	0.4442	0.7772	0.6377	0.4843	0.6253
gatenlp_run33	0.8733	0.8300	0.4531	0.7793	0.6399	**0.4972**	0.6253
gatenlp_run39	0.7733	0.7700	0.4227	0.7078	0.6200	0.4601	0.6253
ims_bm25_1k	0.1667	0.1633	0.0700	0.1475	0.2288	0.1413	0.3063
ims_bm25_2k	0.1667	0.1600	0.0793	0.1515	0.2176	0.1388	0.2769
ims_bm25_4k	0.1467	0.1433	0.0629	0.1396	0.1967	0.1120	0.2589
ims_bm25_3k	0.1400	0.1367	0.0650	0.1276	0.1924	0.1163	0.2488
ims_v-csum	**0.7267**	**0.6733**	**0.3447**	**0.6341**	**0.6174**	**0.3737**	0.7080
ims_csum	0.6267	0.5700	0.3072	0.5315	0.5731	0.3507	0.6940
ims_c-bm25	0.6133	0.5633	0.2890	0.5150	0.5667	0.3131	**0.7114**
ims_bm25	0.5933	0.5333	0.2869	0.4912	0.5572	0.3173	0.6924

Table 4. Performance of different monolingual German runs. Best results are bolded.

Run ID	P@5	P@10	MAP	NDCG@10	NDCG	Rprec	Recall
gatenlp_en2es_run49	**0.8533**	0.7367	0.1579	0.7042	0.3214	0.2273	**0.2565**
gatenlp_en2es_run52	0.8200	0.7700	0.1666	**0.7368**	**0.3287**	0.2286	**0.2565**
gatenlp_en2es_run40	0.8067	**0.7733**	**0.1740**	0.7211	0.3277	**0.2380**	**0.2565**
gatenlp_en2es_run43	0.8000	0.6867	0.1538	0.6555	0.3155	0.2287	**0.2565**
gatenlp_en2es_run46	0.7733	0.6367	0.1439	0.6330	0.3120	0.2231	**0.2565**
gatenlp_en2fr_run53	**0.8400**	**0.7467**	**0.2870**	**0.7245**	**0.4993**	0.3220	**0.4452**
gatenlp_en2fr_run41	0.8267	0.7033	0.2811	0.6980	0.4966	**0.3294**	**0.4452**
gatenlp_en2fr_run44	0.7667	0.6667	0.2527	0.6622	0.4801	0.3107	**0.4452**
gatenlp_en2fr_run47	0.7400	0.6300	0.2378	0.6234	0.4633	0.2980	0.4360
gatenlp_en2fr_run50	0.7133	0.6700	0.2506	0.6521	0.4712	0.3054	0.4360
gatenlp_en2de_run42	**0.8000**	**0.7600**	**0.2776**	**0.7300**	**0.4546**	**0.3307**	**0.3950**
gatenlp_en2de_run54	0.7733	0.7267	0.2680	0.7007	0.4484	0.3221	**0.3950**
gatenlp_en2de_run51	0.7200	0.6867	0.2475	0.6568	0.4334	0.3029	0.3907
gatenlp_en2de_run45	0.7133	0.6700	0.2474	0.6444	0.4349	0.3076	**0.3950**
gatenlp_en2de_run48	0.6867	0.6433	0.2292	0.6099	0.4187	0.2978	0.3907

Table 5. Performance of different bilingual Spanish (en2es), French (en2fr) and German (en2de) runs. Best results are bolded.

6 Conclusion

In this work we present Multistage BiCross Encoder for MLIA Multilingual Semantic Search Task 2. Multistage BiCross encoder is a three stage approach consisting of an initial retrieval using Okapi BM25 algorithm followed by a transformer-based bi-encoder and cross-encoder to effectively rank the documents with respect to the query. While our approach is simple, we found it to be highly effective at achieving state-of-the art performance on a wide range of metrics, including precision (P@10 & P@10) and NDCG at top ranks, R-precision, mean average precision and high recall for all the retrieved documents. For future rounds, we plan to make further improvements to our approach, as well as extensively explore BiCross encoder for document retrieval for future research.

References

[1] Aslam, J.A., Montague, M.: Models for metasearch. In: Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval, pp. 276–284 (2001)

[2] Casacuberta, F., Ceausu, A., Choukri, K., Declerck, T., Deligiannis, M., Di Nunzio, G.M., Domingo, M., Eskevich, M., Ferro, N., García-Martínez, M., Grouin, C., Herranz, M., Jacquet, G., Papavassiliou, V., Piperidis, S., Prokopidis, P., Zweigenbaum, P.: The Covid-19 MLIA @ Eval initiative: Developing multilingual information access systems and resources for Covid-19. `https://bitbucket.org/covid19-mlia/organizers-overall/src/master/report/` (2021)

[3] Clipa, T., Di Nunzio, G.M.: A study on ranking fusion approaches for the retrieval of medical publications. Information **11**(2), 103 (2020)

[4] Cormack, G.V., Clarke, C.L., Buettcher, S.: 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, pp. 758–759 (2009)

[5] Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)

[6] Di Nunzio, G.M., Eskevich, M., Ferro, N.: The Covid-19 MLIA @ Eval initiative: Overview of the multilingual semantic search task. `https://bitbucket.org/covid19-mlia/organizers-task2/src/master/report/` (2021)

[7] Fox, E.A., Shaw, J.A.: Combination of multiple searches. NIST special publication SP **243** (1994)

[8] Hammache, A., Boughanem, M.: Term position-based language model for information retrieval. Journal of the Association for Information Science and Technology (2020)

[9] Hofstätter, S., Hanbury, A.: Let's measure run time! extending the ir replicability infrastructure to include performance aspects. arXiv preprint arXiv:1907.04614 (2019)

[10] Humeau, S., Shuster, K., Lachaux, M.A., Weston, J.: Poly-encoders: Transformer architectures and pre-training strategies for fast and accurate multi-sentence scoring. arXiv preprint arXiv:1905.01969 (2019)

[11] Li, C., Yates, A., MacAvaney, S., He, B., Sun, Y.: Parade: Passage representation aggregation for document reranking. arXiv preprint arXiv:2008.09093 (2020)

[12] Li, X., Liu, Y., Mao, J., He, Z., Zhang, M., Ma, S.: Understanding reading attention distribution during relevance judgement. In: Proceedings of the 27th ACM International Conference on Information and Knowledge Management, pp. 733–742 (2018)

[13] Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V.: Roberta: A robustly optimized bert pre-training approach. arXiv preprint arXiv:1907.11692 (2019)

[14] McDonald, R., Brokos, G.I., Androutsopoulos, I.: Deep relevance ranking using enhanced document-query interactions. arXiv preprint arXiv:1809.01682 (2018)

[15] Nogueira, R., Cho, K.: Passage re-ranking with bert. arXiv preprint arXiv:1901.04085 (2019)

[16] Nogueira, R., Lin, J., Epistemic, A.: From doc2query to docttttquery. Online preprint (2019)

[17] Peters, C.: Cross-language information retrieval and evaluation workshop of the cross-language evaluation forum, clef 2000 lisbon, portugal, september 21–22, 2000 revised papers. In: Conference proceedings CLEF, p. 132, Springer (2000)

[18] Reimers, N., Gurevych, I.: Sentence-bert: Sentence embeddings using siamese bert-networks. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics (11 2019), URL https://arxiv.org/abs/1908.10084

[19] Roberts, K., Alam, T., Bedrick, S., Demner-Fushman, D., Lo, K., Soboroff, I., Voorhees, E., Wang, L.L., Hersh, W.R.: Trec-covid: Rationale and structure of an information retrieval shared task for covid-19. Journal of the American Medical Informatics Association (2020)

[20] Song, X., Petrak, J., Jiang, Y., Singh, I., Maynard, D., Bontcheva, K.: Classification aware neural topic model and its application on a new covid-19 disinformation corpus. arXiv preprint arXiv:2006.03354 (2020)

[21] Tiedemann, J., Thottingal, S.: OPUS-MT — Building open translation services for the World. In: Proceedings of the 22nd Annual Conferenec of the European Association for Machine Translation (EAMT), Lisbon, Portugal (2020)

[22] Yang, W., Zhang, H., Lin, J.: Simple applications of bert for ad hoc document retrieval. arXiv preprint arXiv:1903.10972 (2019)

[23] Yilmaz, Z.A., Yang, W., Zhang, H., Lin, J.: Cross-domain modeling of sentence-level evidence for document retrieval. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pp. 3481–3487 (2019)

[24] Zhang, E., Gupta, N., Tang, R., Han, X., Pradeep, R., Lu, K., Zhang, Y., Nogueira, R., Cho, K., Fang, H., et al.: Covidex: Neural ranking models and keyword search infrastructure for the covid-19 open research dataset. arXiv preprint arXiv:2007.07846 (2020)