EasyChair Preprint

№ 3826

Latent Retrieval for Large-Scale Fact-Checking and Question Answering with NLI training

Chris Samarinas, Wynne Hsu and Mong Li Lee

EasyChair preprints are intended for rapid dissemination of research results and are integrated with the rest of EasyChair.

September 8, 2020

Latent Retrieval for Large-Scale Fact-Checking and Question Answering with NLI training

Chris Samarinas[1] Wynne Hsu[1,2,3] Mong Li Lee[1,2,3]
[1]Institute of Data Science, National University of Singapore
[2]NUS Centre for Trusted Internet and Community
[3]School of Computing, National University of Singapore
chris.samarinas@gmail.com, {whsu, leeml}@comp.nus.edu.sg

***Abstract*—Passage retrieval is a part of fact-checking and question answering systems that is critical yet often neglected. Most systems usually rely only on traditional sparse retrieval. This can have a significant impact on the recall, especially when the relevant passages have few overlapping words with the query sentence. Recent approaches have attempted to learn dense representations of queries and passages to better capture the latent semantic content of text. While dense retrieval models have been proven effective in question answering, there is no relevant work for improving evidence retrieval in fact-checking. In this work, we show that simple training of a dense retriever is sufficient to outperform traditional sparse representations in both question answering and fact-checking. We constructed a new artificial dataset called Factual-NLI, comprised of factual claims and relevant evidence, and demonstrate that using it to train a dense retriever can improve evidence retrieval significantly. Experimental results on the MSMARCO dataset indicate that pre-training with Factual-NLI, and other NLI datasets, is also effective for large-scale passage retrieval in question answering. Our model is incorporated in a real world semantic search engine that returns snippets containing evidence related to questions and claims about the COVID-19 pandemic.**

***Index Terms*—passage retrieval, fact-checking, question answering, natural language inference**

I. Introduction

Misinformation can appear in various forms for different reasons, and in many cases its detection is challenging. When it is not intentional, it may be due to mistakes done by journalists or the inability to verify information. However, it can also be intentional, like hoaxes, rumours, click-bait, satire or fraud. Many online sources are strongly motivated by reader engagement and profits, and intentionally spread false or unverifiable information to capture the internet users' interest. Aided by the high speed of information diffusion, the spread of fake news has become a serious issue. One measure to counter this spread is to perform fact-checking. However, manual fact-checking at scale is intractable and there is a need for a system to assist in this process.

Many approaches have been proposed for automated fact-checking. They can be categorized based on their *input representations*, their *sources of evidence* and the *methodology* they use [1]. A common input representation that has been used in many systems is *subject-predicate-object triples* [2]–[6]. The problem with this representation of claims is that the types of predicates can be limited. Some claims are more complex, and their semantics cannot be fully expressed using such representation.

Sources of evidence for fact-checking include *source identity*, metadata [7] and their *estimated veracity* [8]. Some systems depend on trustworthy sources or fact-checking organizations [9], [10] to match claims with already verified claims in a repository of unstructured text articles. Structured data sources like *databases* and *knowledge graphs* have also been used [11]–[13]. The main issue with approaches depending on structured data is that the data sources may not be comprehensive and they require maintenance. In terms of methodology, models for stance classification have been trained to check the stance of existing headlines on a given claim [14]–[17]. Natural Language Inference models have also been utilized for the entailment classification of evidence-claim pairs [18].

Fact-checking and question answering systems require the retrieval of relevant passages from a text corpus in their first stage. Until recently, most approaches have relied on the traditional TF-IDF or BM25 retrieval [19], [20]. TF-IDF and BM25 represent text as sparse high-dimensional vectors that can be searched efficiently using an inverted index data structure. These sparse representations can be effective in reducing the search space based on keywords. For example, when we want to answer a question like *"Who directed the movie Inception?"*, we obviously want to focus on passages containing the words *movie* and *Inception*. However, sparse representations can be restrictive because they require word overlaps between the query and the passage, and they fail to capture latent semantic relationships. For instance, if we want to validate the claim *"Young person died from COVID-19"*, relevant passages like *"baby died from COVID-19"* or *"boy died from COVID-19"* will be missed.

In this paper, we present a semantic matching model for evidence retrieval called QR-BERT, that has the potential to scale to a corpus with millions of passages. We constructed a new artificial factual natural language inference dataset (Factual-NLI), and demonstrated that QR-BERT trained on this dataset outperforms sparse evidence retrieval. When using additional samples from traditional NLI datasets for pre-training, we observed further improvement. The same pre-training scheme improves the model on passage retrieval for question answering (MSMARCO dataset) and leads to improvement over other state-of-the-art approaches when used in a hybrid ranking

architecture. Finally, we describe the architecture of a real-world semantic search engine called Quin, that utilizes QR-BERT to return snippets related questions or claims about the COVID-19 pandemic.

II. Related work

Neural models based on transformers [21] and pre-trained on language modeling tasks, like BERT [22], GPT [23], and T5 [24], have lead to significant improvements in many natural language processing tasks, including passage retrieval. Noguerira et al. [25] used BERT as a re-ranking model in a multi-stage document ranking architecture that relies on sparse retrieval in the first stage. While achieving state-of-the-art results, the performance of their architecture is bounded by the recall of a sparse retriever. Seo et al. [26] tackled the open-domain question answering problem by using a BERT-based model to generate query-agnostic dense representations of phrases. Their approach however, fails to outperform a basic two-step open domain question answering system than relies on sparse retrieval and a question answering model [20], [27]. Lee et al. [28] presented an open-domain question answering system with a BERT-based dense retriever and a BERT-based reader, that was trained jointly with question-answer pairs without any traditional information retrieval system. They also introduced the Inverse Close Task, that attempted to solve the cold start problem of a dense retriever. Chang et al. [29] carried out further experiments with dense retrievers, introduced pre-training tasks and showed that a dual-encoder dot product retriever based on BERT, with proper pre-training and fine-tuning on a passage retrieval dataset, can outperform sparse retrieval. Guu et al. [30] proposed a retrieval-augmented language model pre-training technique, that trains a knowledge retriever with masked language modeling without any supervision. Their work showed additional improvement in open-domain question answering over the previous approaches. In concurrent work related to this paper, Luan et al. [31] and Karpukhin et al. [32], also investigated the effectiveness of dense dot product retrieval language models.

While there is a lot of work towards improving passage retrieval for question answering, there is not much relevant work on improving evidence retrieval for fact-checking. One relevant work is by Nie et al. [18] who proposed a semantic matching model to identify the most relevant documents to factual claims. They observed an improvement in their fact-checking system on the FEVER dataset. However, their semantic matching model cannot scale, and its performance is also bounded by a sparse retriever.

Pre-training on natural language inference data, has been proven quite effective for learning dense latent representations of text. The idea was first introduced by Conneau et al. [33], who proposed training a text encoder on a textual entailment classification task. The same idea was used by Reimers et al. [34], who used BERT as the encoder.

III. The Factual-NLI Dataset

In order to train and evaluate models for evidence retrieval in fact-checking, we constructed a new synthetic dataset called Factual-NLI. Factual-NLI is comprised of claim-evidence pairs from the FEVER dataset [35] as well as additional synthetic examples generated from the Natural Questions dataset [36] and the MSMARCO dataset [37] which are in the form of question-passage-answer triples.

The additional examples are derived by converting the question-answer pairs to factual statements. This conversion is performed by fine-tuning T5-base, a pre-trained sequence-to-sequence language model [24], using the well-formed answers (182,887 examples in total), with their respective questions and short answers from the MSMARCO question answering dataset. The training is done on a TPU pod for 2 epochs with batch size 512. The input sequence is a pair of question and short answer, separated by "?". The output sequence is the factual statement. For example the pair: *'Which is the tallest building in the world ? Burj Khalifa'* is converted to the statement: *'Burj Khalifa is the tallest building in the world.'*

Besides the synthetic entailed factual statements from question-answer pairs, we also generate additional contradictory statements using the following rules:

R1. *Entity replacement*. We replace named entities (person, organization, location) and numerical entities with a random entity of the same type. For example:
- Alexander Graham Bell invented the first telephone in 1976. → Thomas Edison invented the first telephone in 1976. (named entity replacement)
- Alexander Graham Bell invented the first telephone in 1976. → Alexander Graham Bell invented the first telephone in 2004. (numerical entity replacement)

R2. *Antonym replacement*. We replace the first mentioned adjective with its antonym using Wordnet[1]. For example:
- Burj Khalifa is the tallest building in the world. → Burj Khalifa is the shortest building in the world.

R3. *Verb negation*. We convert the first verb to its negative form. For example:
- Alexander Graham Bell invented the first telephone in 1976. → Alexander Graham Bell did not invent the first telephone in 1976.

Table I shows the characteristics of the Factual-NLI training and testing datasets.

IV. Methodology

A. Problem Definition

The retrieval problem we attempt to solve is defined as follows: Given a corpus of documents D and a query q, we want to return the top-k most relevant passages in D using

[1] https://wordnet.princeton.edu/

Type	Source	# Training	# Testing
True statements	FEVER	115,569	13,329
Conversion to factual statements	Natural Questions + MSMARCO	330,829	28,361
	Subtotal	446,398	41,690
False statements	FEVER	29,734	6,660
Contradictions obtained with rule R1	Natural Questions + MSMARCO	233,014	19,773
Contradictions obtained with rule R2	Natural Questions + MSMARCO	101,497	9,032
Contradictions obtained with rule R3	Natural Questions + MSMARCO	100,503	9,388
	Subtotal	464,748	44,853
	Total	911,146	86,543

TABLE I: Statistics of the Factual-NLI dataset.

a scoring function r_θ with two arguments q and $d \in D$ that computes a relevance score $r_\theta(q, d) \in \mathbb{R}$. In question answering, the query is a question and the relevant passages are expected to answer the question, while in fact-checking, the query is a statement and the relevant passages support or contradict the given statement.

B. Dense Retrieval Model

The corpus D may consist of millions of documents, thus the time complexity of the scoring function is important for a responsive real-time system. In this work, we use the dot product of $\phi(q)$ and $\phi(d)$ as our scoring function:

$$r_\theta(q, d) = \phi(q)^T \phi(d) \tag{1}$$

where $\phi(.)$ is an embedding function that maps a passage or query to a dense vector. The choice of this function r_θ allows us to use efficient maximum inner product search [38], and easily scale our system to millions of documents. As for the embedding function $\phi(.)$, we use the average token embedding of the BERT-base language model [22] which has been fine-tuned on a number of tasks:

$$\phi(d) = \frac{1}{|d|} \sum_{i}^{|d|} \text{BERT}_i(d) \tag{2}$$

where $\text{BERT}_i(d)$ is the embedding of the i-th token in document d, and $|d|$ is the number of tokens in d. Figure 1 shows the semantic matching model, referred to as QR-BERT, that determines whether a passage is relevant to the query. Our dense retrieval model is comprised of one encoder that embeds the query and the passage to the same k-dimensional space. The similarity between the query and the passage is given by the cosine similarity of their embedding representations.

C. Training

QR-BERT is trained on a set of query-passage examples. Let D be the set of all the passages in our training dataset and D^+ be the set of positive query-passage pairs. We estimate the model parameters θ of the scoring function by maximizing the log likelihood as follows:

$$max_\theta \sum_{(q,d) \in D+} log(p_\theta(d|q)) \tag{3}$$

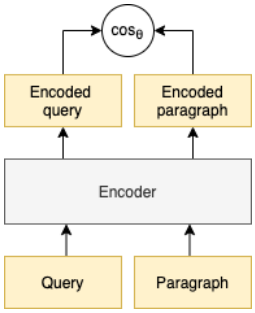


Fig. 1: Siamese semantic matching model

where conditional probability is approximated by the softmax:

$$p_\theta(d|q) = \frac{e^{r_\theta(q,d)}}{\sum_{d_i \in D} e^{r_\theta(q,d_i)}} \tag{4}$$

Note that obtaining the denominator over all the passages in Equation 4 is computationally expensive. Hence, we limit the computation to only passages in the current training batch as is widely used in [28]–[30], [39]. The final loss function is given by:

$$max_\theta \sum_{(q,d) \in D_B^+} r_\theta(q, d) - log\Big(\sum_{d_i \in D_B} e^{r_\theta(q,d_i)} \Big) \tag{5}$$

where D_B is the set of passages in a training batch B, and D_B^+ is the set of positive query-passage pairs in B.

For evidence retrieval, the model is trained and evaluated on the Factual-NLI dataset, and for answer retrieval on MS-MARCO. We train it with Adam optimizer, initial learning rate 2×10^{-5}, batch size 256 and 10,000 warmup steps.

D. Pre-training Tasks

We experiment with two pre-training tasks in an attempt to further improve the performance of QR-BERT:

Inverse Cloze Task. A semantic matching model may suffer from the cold start problem as observed in the dense retrieval models. The Inverse Cloze Task (ICT) has recently been used as a pre-training task to solve this issue [28]–[30].

Given a passage d with n sentences $d = \{s_1, s_2, ..., s_n\}$, the query q is a sentence s_i drawn randomly from the passage d, and the relevant passage is the remaining sentences $\{s_1, ..., s_{i-1}, s_{i+1}, ..., s_n\}$. Here, we adopt the same pre-training process for QR-BERT. We source 50 million training samples from Wikipedia, and pre-train our siamese encoder model for 400,000 steps with Adam optimizer, weight decay 0.01, initial learning rate 2×10^{-5}, batch size 256 and 10,000 warmup steps.

Natural Language Inference. It is recognized that pre-training embedding models on natural language inference (NLI) data is effective for semantic text similarity tasks [33], [34]. As such, we merge two popular NLI datasets, namely SNLI [40] and MultiNLI [41], into one dataset (referred to as NLI) and perform pre-training with the classification objective function (as used in [33], [34]):

$$o = softmax(W[u; v; |u - v|]) \tag{6}$$

where u is the embedding of the premise sentence and v is the embedding of the hypothesis sentence, $W_{3\times 3k}$ a linear transformation matrix, where $k = 768$ is the dimensionality of the hidden representation of QR-BERT (based on BERT-base) and $[u; v; |u - v|]$ is the concatenated vector of u, v and their absolute difference $|u-v|$. We pre-train QR-BERT using cross-entropy loss, Adam optimizer, initial learning rate 2×10^{-5} and batch size 64, until convergence.

V. Performance Study

In this section, we evaluate the performance of QR-BERT and compare the different pre-training methods. We compute the recall@k metrics (whether the evicence passage is returned in the top k results) and the mean reciprocal rank (MRR) for the top 10 results on the Factual-NLI and the MSMARCO dataset, which is defined as follows:

$$\text{MRR} = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{rank_i}$$

where N is the total number of queries and $rank_i$ the position of the correct passage to the i-th query in the returned results. To compute the MRR@10, we ignore all the ranks > 10 in the calculation of the metric.

A. Experiments on Evidence Retrieval

We first demonstrate the effectiveness of QR-BERT for evidence retrieval, where the query is a statement or a claim. We evaluate the performance of QR-BERT with different pre-training methods on the Factual-NLI testing set. For comparison, we implement a baseline BM25 retriever using lemmatization, unigrams and bigrams. We have trained QR-BERT in one or more stages using:

- FEVER (with sampled softmax - eq. 5)
- NLI (with classification objective - eq. 6)
- Factual-NLI (with sampled softmax)
- ICT (with sampled softmax) and Factual-NLI
- NLI and Factual-NLI without contradiction examples
- NLI and Factual-NLI using all the examples

Table II shows the results. We observe that training using the FEVER dataset, the currently popular dataset for fact-checking [42], is insufficient and leads to worse performance than sparse retrieval, while training using the extended Factual-NLI dataset is very effective. The best performing model is the one that has been pre-trained on NLI and trained on Factual-NLI. When we compare training QR-BERT on NLI with the classification objective versus Factual-NLI with sampled softmax, we see a vast improvement in the recall as well as the MRR. Pre-training on NLI and training on Factual-NLI leads to slight improvement compared to pre-training on ICT and training on Factual-NLI. Looking at the last two rows in Table II, we find that including contradiction examples in the training dataset is important, as it improves MRR by 0.021 and R@100 by 3.22%. The advantage of QR-BERT over sparse retrieval is more obvious when we perform evaluation on a subset of the test examples, excluding those for which the sparse retriever returns the relevant passage in the top 5 results. In this subset, as seen in Table III, QR-BERT is still able to achieve high recall in the top 20 results. This makes clear that when keyword matching is insufficient, dense representations can help in retrieving relevant passages.

B. Experiments on Answer Retrieval

Besides evidence retrieval, QR-BERT is also effective for answer retrieval. In this set of experiments, we evaluate QR-BERT on a large scale MAchine Reading COmprehension dataset (MSMARCO) [37]. MSMARCO contains 8,841,823 passages extracted from 3,563,535 web documents retrieved by the Bing search engine. It also contains 1,010,916 queries with 1,026,758 unique answers, 182,887 of which are also written as well-formatted sentences. This dataset is widely used for evaluating reading comprehension and passage retrieval models. We examine the performance of the following QR-BERT models on MSMARCO:

- fine-tuned using only MSMARCO
- pre-trained on ICT and fine-tuned using MSMARCO
- pre-trained on NLI and fine-tuned using MSMARCO
- pre-trained on NLI and Factual-NLI without fine-tuning
- pre-trained on NLI and Factual-NLI with fine-tuning using MSMARCO

Table IV shows the evaluation results using only the passages in the development set as a small scale retrieval benchmark. We observe that using the model we trained for evidence retrieval without fine-tuning on MSMARCO, leads to a very poor performance. Pre-training on NLI improves MRR by 0.017 and R@1 by 2.26% (fifth row in Table IV). If we include Factual-NLI dataset in the pre-training, we obtain an additional 0.031 improvement in MRR and 3.71% in R@1 (see the sixth row). Overall, pre-training using natural language inference examples seems more effective than pre-training on

the inverse cloze task for both evidence and answer retrieval. However, it is worth mentioning that we use a relatively small batch size of 256 due to limited resources, while much larger batches were used in previous works [29]. All the models outperform the BM25 retrieval baseline.

C. Multi-task Retrieval Model

Additionally, we trained a multi-task retrieval model for passage retrieval in fact-checking *and* question answering. We used the QR-BERT model initially pre-trained on NLI, and trained it using examples from the *union* of Factual-NLI and MSMARCO. The resulting model performs a little worse than the single task models (as seen in the last rows of Table II and Table IV). However, the multi-task model still outperforms sparse retrieval, and also seems to rank a little better the top 10 results in the reduced evidence retrieval evaluation set (as seen in table Table III).

D. Experiments with Re-Ranking

The learned representation of a dense retrieval model like QR-BERT is query-agnostic. Nevertheless, learning a representation conditioned on both the query and the passage should help in identifying the relevant passages more accurately. To validate this, we experiment with a two-stage retrieval process, where a list of candidate passages is first obtained from a retrieval model followed by a re-ranking of the list of candidates [25] with a binary relevance classifier. The relevance classifier is based on BERT-large, fine-tuned on 20M query-passage pairs from the MSMARCO dataset. More specifically, we take the embedded representation of the [CLS] token and apply a linear transformation:

$$r(q, d) = softmax(W \cdot \text{BERT}_{[\text{CLS}]}([q\ ;\ d]) + b) \qquad (7)$$

where $W_{2\times d}$ a linear transformation matrix, d the dimensionality of the BERT embeddings ($d = 1024$ for BERT-large), and $b_{2\times 1}$ the added bias in the transformation. The final normalized relevance scores are given by the softmax. We trained the ranking model by minimizing the cross-entropy loss with Adam optimizer, initial learning rate $2 \times 10^{-}5$ and batch size 128, on a validation set of 20,000 samples until convergence. We evaluated the following configurations:

- Use BM25 to generate a list of candidates without re-ranking.
- Use QR-BERT to generate a list of candidates without re-ranking,
- Use BM25 to generate a list of candidates, followed by re-ranking
- Use QR-BERT to generate a list of candidates, followed by re-ranking,
- Use the union of the top results from BM25 and QR-BERT as the list of candidates, followed by re-ranking.

Table V shows the results. We observe that using a two-stage approach with retrieval and re-ranking, leads to higher recall and MRR, confirming our hypothesis that query-dependent representations are better in identifying relevant passages. The two-stage ranking approach with a BM25 sparse retriever seems to perform a little better in terms of the MRR metric than the one with a QR-BERT dense retriever. This is possibly because in many query-passage pairs in MSMARCO, there is a word overlap that constitutes a strong relevance signal, and dense representations sometimes cannot capture it. It is worth noting however, that QR-BERT with re-ranking has higher recall@100 compared to BM25 with re-ranking. The best performance is obtained when we combine the top retrieved results of BM25 and QR-BERT during the retrieval stage. Our two-step approach with the hybrid retrieval and re-ranking even outperforms the state-of-the-art solutions by Nogueira et al. [25], namely, BM25 retrieval and re-ranking with a relevance classifier referred to as monoBERT, as well as their three-stage approach with BM25 retrieval, re-ranking using monoBERT followed by an additional pairwise ranking model called duoBERT.

E. Qualitative Analysis

Table VI shows the top snippets retrieved for some questions and claims by QR-BERT and BM25. For the first two queries, only QR-BERT returns a relevant passage. This is because keyword based retrieval is not sufficient to surface the most relevant passages. For the third query, both models succeed in retrieving a relevant snippet. An interesting observation for this query is that the snippet retrieved by QR-BERT has only the word *viruses* in common with the query. This shows that the model has the ability to capture the latent meaning of the text and identify synonym terms like *called off* (for *canceled*) and semantically close terms like *conferences* (close to *events*).

In conclusion, sparse retrieval works well in cases when the query is specific enough to allow easy discovery of the relevant passage through keyword matching. However, this is insufficient in practice. A dense retrieval model can give more accurate results, and when used in an architecture with sparse retrieval combined with re-ranking, we can achieve significantly higher recall.

VI. The Quin System

Using the dense retrieval model, we developed Quin[2], a scalable semantic search engine that returns snippets of up to five sentences containing the answer to a question or claim related to the COVID-19 pandemic. Figure 3 shows a screenshot of the system.

A. Indexing and search

Quin has a module for crawling RSS feeds, and storing the html source of the news articles. We remove the boilerplate and isolate the main content of the news articles. From the clean text, we extract snippets of 5 sentences each, by using a sliding window on the sequence of sentences of every article. We utilize the nltk library[3] to split the documents into sentences.

[2]https://quin.algoprog.com
[3]https://www.nltk.org

Model	R@1	R@5	R@10	R@20	R@100	MRR@10
BM25 with lemmatization, unigrams and bigrams	78.18	89.15	92.61	95.16	98.03	0.8272
QR-BERT on FEVER	62.80	73.47	78.07	81.87	87.21	0.6735
QR-BERT on NLI	23.31	28.98	33.01	37.42	48.78	0.2667
QR-BERT on Factual-NLI	79.82	89.30	92.06	93.86	95.50	0.8369
QR-BERT on ICT + Factual-NLI	82.27	92.37	95.24	97.15	98.76	0.8643
QR-BERT on NLI + Factual-NLI (no contradictions)	81.39	89.97	92.53	94.13	95.77	0.8492
QR-BERT on NLI + Factual-NLI	**82.91**	**93.06**	**95.79**	**97.61**	**98.99**	**0.8707**
QR-BERT on NLI + Factual-NLI $\bigcup$ MSMARCO (multi-task)	80.19	89.59	92.31	93.92	95.23	0.8402

TABLE II: Evidence retrieval evaluation on Factual-NLI.

Model	R@1	R@5	R@10	R@20	R@100	MRR@10
BM25 with lemmatization, unigrams and bigrams	0	0	32.57	55.93	82.32	0.0706
QR-BERT on NLI + Factual-NLI	45.55	65.75	76.78	**84.30**	**90.82**	0.5457
QR-BERT on NLI + Factual-NLI $\bigcup$ MSMARCO (multi-task)	**47.36**	**66.87**	**76.94**	83.72	89.28	**0.5582**

TABLE III: Evidence retrieval evaluation on Factual-NLI excluding examples with BM25 rank ≤ 5.

Model	R@1	R@5	R@10	R@20	R@100	MRR@10
BM25 with lemmatization, unigrams and bigrams	57.03	78.72	84.45	88.71	94.69	0.6684
QR-BERT on NLI + Factual-NLI	39.21	60.65	67.48	73.13	83.15	0.4916
QR-BERT on MSMARCO	66.90	87.87	92.03	94.63	97.66	0.7621
QR-BERT on ICT + MSMARCO	65.28	86.79	91.20	94.03	97.40	0.7482
QR-BERT on NLI + MSMARCO	69.16	89.05	92.80	95.17	97.84	0.7798
QR-BERT on NLI + Factual-NLI + MSMARCO	**72.87**	**91.64**	**94.92**	**96.79**	**98.85**	**0.8114**
QR-BERT on NLI + Factual-NLI $\bigcup$ MSMARCO (multi-task)	67.96	87.06	90.48	92.47	94.69	0.7638

TABLE IV: Answer retrieval evaluation on MSMARCO (passages from development set)

Model	R@1	R@5	R@10	R@20	R@100	MRR@10
BM25 with lemmatization, unigrams and bigrams	9.77	27.08	36.39	44.96	64.62	0.1713
QR-BERT	13.91	35.15	44.85	53.79	70.49	0.2285
BM25 (top 200) + re-rank	24.19	51.35	59.86	66.05	71.64	0.3556
QR-BERT (top 200) + re-rank	22.55	47.92	56.96	64.37	74.36	0.3324
QR-BERT (top 200) $\bigcup$ BM25 (top 200) + re-rank	**25.43**	**55.71**	**66.68**	**75.67**	**87.93**	**0.3817**
BM25 (top 1000) + monoBERT (Nogueira et al.) [25]	-	-	-	-	-	0.3650
BM25 (top 1000) + monoBERT + duoBERT [25]	-	-	-	-	-	0.3790

TABLE V: Answer retrieval evaluation on MSMARCO (all candidate passages)

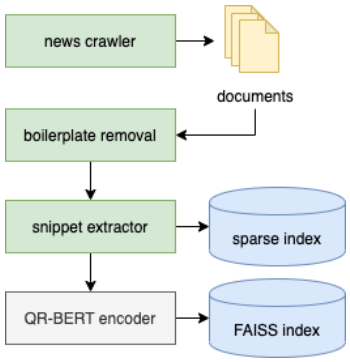
To facilitate efficient large-scale retrieval, we build two indexes on the snippets: (a) an efficient sparse inverted index for BM25 retrieval, and (b) a FAISS dense index [43] that supports maximum inner product search. Building a FAISS index of 1M passages takes about 26 seconds using an NVIDIA V100 GPU. The index is able to process about 1000 top-100 queries per second on a DGX-2 server with a Dual Intel Xeon Platinum CPU. Figure 2a summarizes this process. Figure 2b shows the search process. The query is used to perform a search on a sparse and on a dense (FAISS) index of snippets. We retrieve the top 500 results from each index, and compute a relevance score for each result. The results are ranked by their relevance score, and we output the final ranked list of results R.

B. Time-sensitive ranking

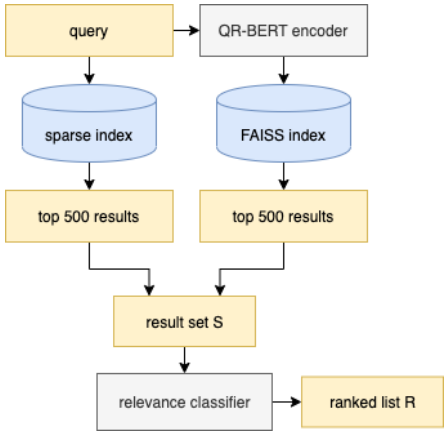
When dealing with news articles, the answers to many questions change over time. When ordering the results by semantic relevance, some of the retuned snippets might contain outdated answers. For example, when we have a question like *'How many are the virus cases in Italy?'*, the top results might contain answers stating that the number of cases is a few hundreds, while the actual number is already more than 200,000. Sorting the passages above a relevance threshold by date is not the best solution, because finding an optimal cuttoff threshold is non-trivial and could lead to lower recall. On the other hand, ignoring completely the relevance score in the final ranking can lead to irrelevant passages ranked high in the results. To mitigate this issue, we pass the relevance scores to an exponential decay function to priotitise the snippets from more recently published articles:

$$r'(q, d, t) = r(q, d) \times 2^{-\frac{t_c - t}{h}} \tag{8}$$

where $r'(q, d, t)$ is the time-sensitive relevance of the snippet d to the query q from a news article with unix timestamp t, $r(q, d)$ the relevance score of the snippet d to the query q (from the ranking model), t_c is the current unix timestamp and h a halving interval in seconds (one year in our system). Table VII shows an example where the time-sensitive relevance surfaces a more up-to-date answer.



(a) Indexing snippets from news articles



(b) Searching for relevant snippets

Question / Claim	Top snippet returned by QR-BERT	Top snippet returned by BM25
Is COVID-19 the same as SARS?	... Coronaviruses are a class of pathogens, seven of which are known to infect humans. **Covid-19** is said to be more genetically **similar** to **Sars** than any other virus of that class. The International Committee on Taxonomy of Viruses is even calling **Covid-19** "severe acute respiratory syndrome coronavirus 2". ...	... The patent numbers listed are indeed real, but they are for **SARS**, caused by **SARS**-CoV (or **SARS**-CoV-1), not for **COVID-19**, caused by **SARS**-CoV-2. Both mention "**SARS**-CoV" multiple times but have no mention of "**SARS**-CoV-2", the new strain causing **COVID-19**. ...
What is COVID-19?	... **Covid-19 is** one of seven strains of the coronavirus class that are known to infect humans. Others range from the mild common cold to severe acute respiratory syndrome (**Sars**), which killed 774 people in 2004. Most of the people who initially became unwell from **Covid-19** worked at, or visited, a seafood and live animal market in the Chinese city of Wuhan. ...	... CMO Brendan Murphy has repeatedly ruled out any link between the technology and the spread of **COVID-19**. T̈here is no link between 5G and **COVID-19**. 5G does not cause **COVID-19**. It does not spread **COVID-19**. Nor does it increase the severity of **COVID-19** or make people more susceptible to **COVID-19**,ẗhe said on Friday in a statement. ...
Events were canceled because of the virus	... **Conferences** are particularly conducive to spreading **viruses** because they bring large crowds together in close proximity from many locations. Several major tech **conferences**, including the Mobile World Congress in Barcelona and Facebook's F8 developers' **conference**, have been **called off** because of the **coronavirus**. ...	... The Google I/O 2020 physical **event** has been **canceled** and will be held digitally. Refunds will be given to those who have purchased tickets for I/O 2020. People who were awarded tickets for the 2020 event will be automatically awarded tickets to the 2021 event. ...

TABLE VI: Snippets retrieved by QR-BERT and BM25 for queries related to COVID-19

Scoring	Top retrieved snippet
Relevance	... More than 100 people have died from the virus in Italy with more than **3,000** confirmed cases. ...
Time-sensitive relevance	... Italy's total known infections stand at **249,756**. Three more deaths since Thursday raised Italy's overall confirmed death toll to 35,190. ...

TABLE VII: Retrieved snippets using relevance and time-aware relevance scores

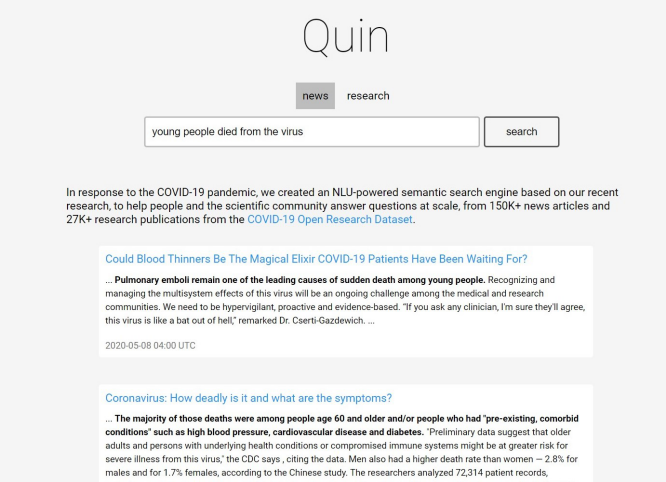


Fig. 3: Screenshot of the Quin system

VII. Conclusion

In this work, we demonstrated that a latent dot product retrieval model based on BERT, trained with sampled softmax loss, can outperform the traditional sparse retrieval as a standalone model, and leads to significant improvements when used in a multi-stage ranking architecture. We constructed a new synthetic dataset for evidence retrieval evaluation in fact-checking called Factual-NLI, and showed that pre-training on existing NLI datasets, and the new dataset, improves significantly the retrieval model. Using the trained retrieval model, we built a semantic search system of news articles to demonstrate its effectiveness in a real-world large-scale dataset. Our model and the used datasets are publicly released as part of our Quin semantic search framework [4].

In future work, it would be interesting to see how dense retrieval benefits a complete fact-checking system on a much larger scale. Augmenting Factual-NLI with noise from web results would give a more difficult benchmark, which could showcase better the efficiency and recall issues of the traditional sparse retrieval in a real-world setting.

Acknowledgement

This work was supported by Tensorflow Research Cloud program that provided free access to TPU pods.

References

[1] J. Thorne and A. Vlachos, "Automated fact checking: Task formulations, methods and future directions," in *Proceedings of the 27th International Conference on Computational Linguistics*, 2018.

[2] N. Nakashole and T. M. Mitchell, "Language-aware truth assessment of fact candidates," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, 2014.

[3] A. Magdy and N. Wanas, "Web-based statistical fact checking of textual documents," in *Proceedings of the 2nd International Workshop on Search and Mining User-Generated Contents*, 2010.

[4] W. Y. Lim, M. L. Lee, and W. Hsu, "Claimfinder: A framework for identifying claims in microblogs," in *WWW Workshop on Making Sense of Microposts*, 2016.

[5] ——, "Ifact: An interactive framework to assess claims from tweets," in *Proceedings of ACM on Conference on Information and Knowledge Management (CIKM)*, 2017.

[6] ——, "End-to-end time-sensitive fact check," in *ACM SIGIR Workshop on Reducing Online Misinformation Exposure (ROME)*, 2019.

[7] W. Y. Wang, ""liar, liar pants on fire": A new benchmark dataset for fake news detection," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, 2017.

[8] L. Derczynski and K. Bontcheva, "Pheme: Veracity in digital social networks," *CEUR Workshop Proceedings*, vol. 1181, pp. 19–22, 01 2014.

[9] N. Hassan, G. Zhang, F. Arslan, J. Caraballo, D. Jimenez, S. Gawsane, S. Hasan, M. Joseph, A. Kulkarni, A. K. Nayak, and et al., "Claimbuster: The first-ever end-to-end fact-checking system," *PVLDB*, vol. 10, no. 12, p. 1945–1948, 2017.

[10] N. Hassan, F. Arslan, C. Li, and M. Tremayne, "Toward automated fact-checking: Detecting check-worthy factual claims by claimbuster," in *Proceedings of the ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, 2017.

[11] P. K. Agarwal, "Finding, monitoring, and checking claims computationally based on structured data," in *Computation+Journalism Symposium*, 2014.

[12] Y. Wu, P. K. Agarwal, C. Li, J. Yang, and C. Yu, "Toward computational fact-checking," *PVLDB*, vol. 7, no. 7, 2014.

[13] Y. Wu, B. Walenz, P. Li, A. Shim, E. Sonmez, P. K. Agarwal, C. Li, J. Yang, and C. Yu, "icheck: computationally combating "lies, d–ned lies, and statistics"," in *Proceedings of the ACM SIGMOD Conference*, 2014.

[14] R. Baly, M. Mohtarami, J. R. Glass, L. Màrquez, A. Moschitti, and P. Nakov, "Integrating stance detection and fact checking in a unified corpus," *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL)*, 2018.

[15] W. Ferreira and A. Vlachos, "Emergent: a novel data-set for stance classification," in *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2016.

[16] M. Mohtarami, R. Baly, J. Glass, P. Nakov, L. Màrquez, and A. Moschitti, "Automatic stance detection using end-to-end memory networks," in *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2018.

[17] L. Poddar, W. Hsu, and M. L. Lee, "Predicting stances in twitter conversations for detecting veracity of rumors: A neural approach," in *Proceedings of the IEEE International Conference on Tools for Artificial Intelligence (ICTAI)*, 2018.

[18] Y. Nie, H. Chen, and M. Bansal, "Combining fact extraction and verification with neural semantic matching networks," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2019.

[19] D. Chen, A. Fisch, J. Weston, and A. Bordes, "Reading wikipedia to answer open-domain questions," in *ACL*, 2017.

[20] W. Yang, Y. Xie, A. Lin, X. Li, L. Tan, K. Xiong, M. Li, and J. Lin, "End-to-end open-domain question answering with bertserini," *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2019.

[21] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems (NIPS)*, 2017.

[22] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL)*, 2019.

[23] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving language understanding by generative pre-training," 2018.

[24] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *ArXiv*, vol. abs/1910.10683, 2019.

[25] R. Nogueira, W. Yang, K. Cho, and J. Lin, "Multi-stage document ranking with bert," *ArXiv*, vol. abs/1910.14424, 2019.

[4] https://github.com/algoprog/Quin

[26] M. J. Seo, J. Lee, T. Kwiatkowski, A. P. Parikh, A. Farhadi, and H. Hajishirzi, "Real-time open-domain question answering with dense-sparse phrase index," *preprint arXiv:1906.05807v2*, 2019.

[27] Z. Wang, P. Ng, X. Ma, R. Nallapati, and B. Xiang, "Multi-passage bert: A globally normalized bert model for open-domain question answering," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2019.

[28] K. Lee, M. Chang, and K. Toutanova, "Latent retrieval for weakly supervised open domain question answering," *CoRR*, 2019.

[29] W.-C. Chang, F. X. Yu, Y.-W. Chang, Y. Yang, and S. Kumar, "Pre-training tasks for embedding-based large-scale retrieval," in *International Conference on Learning Representations (ICLR)*, 2020.

[30] K. Guu, K. Lee, Z. Tung, P. Pasupat, and M.-W. Chang, "Realm: Retrieval-augmented language model pre-training," *https://arxiv.org/abs/2002.08909*, 2020.

[31] Y. Luan, J. Eisenstein, K. Toutanova, and M. Collins, "Sparse, dense, and attentional representations for text retrieval," *preprint arXiv:2005.00181*, 2020.

[32] V. Karpukhin, B. Oğuz, S. Min, P. Lewis, L. Wu, S. Edunov, D. Chen, and W. tau Yih, "Dense passage retrieval for open-domain question answering," *preprint arXiv:2004.04906v2*, 2020.

[33] A. Conneau, D. Kiela, H. Schwenk, L. Barrault, and A. Bordes, "Supervised learning of universal sentence representations from natural language inference data," *ArXiv*, vol. abs/1705.02364, 2017.

[34] N. Reimers and I. Gurevych, "Sentence-bert: Sentence embeddings using siamese bert-networks," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2019.

[35] J. Thorne, A. Vlachos, C. Christodoulopoulos, and A. Mittal, "FEVER: a large-scale dataset for fact extraction and verification," *CoRR*, vol. abs/1803.05355, 2018.

[36] T. Kwiatkowski, J. Palomaki, O. Redfield, M. Collins, A. Parikh, C. Alberti, D. Epstein, I. Polosukhin, M. Kelcey, J. Devlin, K. Lee, K. N. Toutanova, L. Jones, M.-W. Chang, A. Dai, J. Uszkoreit, Q. Le, and S. Petrov, "Natural questions: a benchmark for question answering research," *Transactions of the Association of Computational Linguistics*, 2019.

[37] D. F. Campos, T. Nguyen, M. Rosenberg, X. Song, J. Gao, S. Tiwary, R. Majumder, L. Deng, and B. Mitra, "Ms marco: A human generated machine reading comprehension dataset," *ArXiv*, vol. abs/1611.09268, 2016.

[38] Q. Ding, H.-F. Yu, and C.-J. Hsieh, "A fast sampling algorithm for maximum inner product search," in *Proceedings of Machine Learning Research*, 2019.

[39] M. Henderson, R. Al-Rfou, B. Strope, Y.-H. Sung, L. Lukács, R. Guo, S. Kumar, B. Miklos, and R. Kurzweil, "Efficient natural language response suggestion for smart reply," *ArXiv*, vol. abs/1705.00652, 2017.

[40] S. R. Bowman, G. Angeli, C. Potts, and C. D. Manning, "A large annotated corpus for learning natural language inference," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2015.

[41] A. Williams, N. Nangia, and S. Bowman, "A broad-coverage challenge corpus for sentence understanding through inference," in *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL)*, 2018.

[42] J. Thorne, A. Vlachos, C. Christodoulopoulos, and A. Mittal, "Fever: a large-scale dataset for fact extraction and verification," *arXiv preprint arXiv:1803.05355*, 2018.

[43] J. Johnson, M. Douze, and H. Jégou, "Billion-scale similarity search with gpus," *IEEE Transaction on Big Data*, 2017.