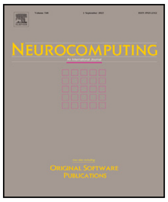
Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom



Information retrieval algorithms and neural ranking models to detect previously fact-checked information

Tanmoy Chakraborty [b], Valerio La Gatta [a], Vincenzo Moscato [a], Giancarlo Sperlì [a,*]

[a] *Department of Electrical Engineering and Information Technology (DIETI), University of Naples "Federico II", Via Claudio 21, Naples, Italy*
[b] *Department of Electrical Engineering, Indian Institute of Technology Delhi, New Delhi, India*

ARTICLE INFO

Communicated by Y. Yan

Keywords:
Fact-checking
Claim retrieval
Learning-to-rank
Semantic matching

ABSTRACT

Although in the last decade several fact-checking organizations have emerged to verify misinformation, fake news has continued to proliferate, especially through social media platforms. Even though adopting improved detection strategies is of utmost importance, the fact-checking process could be optimized by verifying whether a claim has been previously fact-checked. Despite some ad-hoc information retrieval approaches having been recently proposed, the utility of modern (neural) retrieval systems have not been investigated yet. In this paper, we consider the standard two-phases retriever-reranker architecture and benchmark different state-of-the-art techniques from the information retrieval and Q&A literature. We design several experiments on a real-world Twitter dataset to analyze the efficiency and the effectiveness of the benchmark approaches. Our results show that combining standard and neural approaches is the most promising research direction to improve retrievers performance and that complex (neural) rerankers might still be efficient in practice since there is no need to process a high number of documents to improve ranking performance.

1. Introduction

Although fake news is not a new phenomenon, since the last decade it has become one of the major threats to democracy, journalism, and freedom of expression. The rise of social media has been playing a key-role since those platforms enable the creation, the publication and the consumption of news online faster and cheaper. As a result, huge amount of false information which spreads across the population affects our life. For instance, fake news proliferation during the 2016 US Presidential Election undermined public trust in the government [1], and from an economic perspective, the false claim stating that 'Barack Obama was injured in an explosion' wiped out $130 billion in stock value.[1]

The problem of fake news proliferation is being addressed from different perspectives. First, the Duke Reportes' Lab[2] counts more than 400 fact-checking world-wide organizations which try to debunk false information though domain experts' analyses and semi-automatic systems assessing news truthfulness. Second, representatives of online platforms, leading social networks, advertisers and advertising industry agreed on a self-regulatory Code of Practice[3] to converge on a common strategy to deal with the spread of online disinformation. Third, governments are investing in educating the public on this problem and on how to discern false from true information, since it has been proven that, once trained, the probability of people sharing fake news decreases by 400%.[4]

However, the quantity of information is much more than the one people can effectively check and, since fake news spreads comparably or even faster than true ones [2,3], it is essential to speed up and/or ease the verification process. From this perspective, one should consider that the same viral claim is often reposted by thousands of people in a short time-frame and might be shared also after a while in a different context. Moreover, when considering political statements, it is well-known that politicians, even unconsciously, have a tendency to repeat themselves.[5]

Thus, detecting whether a claim has been already fact-checked seems a promising approach on which researchers should focus more for at least three reasons. First, it can ease the manual fact-checkers' effort, increasing their productivity and thus their effectiveness. Second,

* Corresponding author.
E-mail addresses: tanchak@iitd.ac.in (T. Chakraborty), valerio.lagatta@unina.it (V. La Gatta), vincenzo.moscato@unina.it (V. Moscato), giancarlo.sperli@unina.it (G. Sperlì).

[1] https://www.forbes.com/sites/kenrapoza/2017/02/26/can-fake-news-impact-the-stock-market/?sh=1883d50d2fac.

[2] https://reporterslab.org/fact-checking/.

[3] https://ec.europa.eu/digital-single-market/en/code-practice-disinformation.

[4] https://edition.cnn.com/videos/tech/2020/04/17/mark-zuckerberg-facebook-limit-coronavirus-misinformation-cnn-town-hall-vpx.cnn.

[5] https://www.washingtonpost.com/outlook/its-easy-to-fact-check-trumps-lies-he-tells-the-same-ones-all-the-time.

https://doi.org/10.1016/j.neucom.2023.126680
Received 26 April 2022; Received in revised form 28 April 2023; Accepted 11 August 2023
Available online 21 August 2023

automatic fact-checking systems might be improved since the veracity prediction of the input claim could be based on a set of already verified information. Third, journalists, who are sceptical towards the adoption of automatic detection systems, could easily exploit, instead, a tool which checks in real-time if their interviewees are referring to (telling) inaccurate (false) data (claims). It is worth mentioning that popular search engines do not represent an effective solution because they do not report verified information and thus they could activate a dangerous and time-consuming verification cascade, i.e., the experts should in turn verify whether the retrieved evidences are actually reliable. In addition, we highlight that the task of detecting previously fact-checked information does not depend on the claim veracity; it supports the fact-checking process in a preliminary phase w.r.t. retrieving the evidences which the truthfulness assessment should be based on.

Despite having been proposed to integrate the checking of the input claim against a knowledge base of verified information in the fact-checking pipeline [4], the problem of detecting previously fact-checked information has been considered only recently by [5] which formulates the information retrieval task of ranking a list of verified documents according to the relevance with the input claim. Under these settings, the task aims at filtering out already verified claims, thus allowing professionals to focus on brand new claims which should be carefully checked, also by retrieving other evidences. However, as opposed to classical ad-hoc retrieval problems, the documents corpus, i.e. the verified information, is not static and, in principle, for each truthfulness assessment an update should be triggered on it.

Inspired by the great advancements transformers architectures have been bringing to the natural language processing field [6,7], competitors at CheckThat!2020 [8] dealt with the task and showed that different transformers' fine tuning strategies lead to promising performance improvements w.r.t. standard information retrieval baselines (e.g. BM25). However, the efficiency of the proposed approaches has not been analyzed yet, neither the different requirements of top-k retrieval and reranking models has been considered within a two-stages pipeline. Consequently, since the retriever-reranker architecture has been widely studied for information retrieval and question answering systems [9], it seems profitable to explore how the most recent methods and models perform on the above-mentioned task. In this paper, we conduct an extensive benchmark, especially considering the recent advances that neural ranking models and transformer-based systems have brought to both retriever and reranker stages [9,10]. We evaluate the models on a real-world tweets' dataset considering both the effectiveness and the efficiency of the system. Our results indicate that the integration of conventional and neural methodologies holds considerable potential as a research avenue for enhancing the performance of retrievers. Additionally, we find that complex neural rerankers have the potential to be efficient in practical settings as they do not require a high volume of document processing to improve ranking performance.

Overall, these findings unveil the practical utility of conventional and neural methodologies from relevant literature in the context of detecting previously fact-checked information, thereby highlighting the potential for their effective application in real-world settings.

The paper is organized as follows. After having presented related works regarding fact-checking methods and ranking models proposed in the recent literature (Section 2), we present the benchmarking framework in Section 3 and define our research objectives in Section 4.1. The experimental evaluation using a Twitter dataset is presented in Section 4. Finally, Section 5 discuss the theoretical and practical implications of our research and Section 6 discusses several conclusions and possible future works to focus on.

2. Related works

2.1. Fact-checking panorama

The fact-checking problem, i.e. predicting the veracity of a claim, has been studied for long time from different perspectives and under disparate scenarios. Recently, researchers are increasingly focusing on evidence-aware fact-checking, i.e. extracting the veracity of an input claim based on retrieved evidences, which can support or refute it. Under these settings, [11] releases FEVER dataset aiming at fact-checking mutated claims generated from Wikipedia pages. [12,13] exploit web search engines to find real-time potential evidences and compute their stance w.r.t. the input claim. In addition, [14] leverages LSTM models and attention mechanisms to retrieve documents and to capture their most relevant sentences, respectively. [15] first employs neural semantic matching networks to address the document retrieval and the evidence selection problems. Inspired by the unprecedented performance transformer architectures are achieving in many NLP tasks, [16] adopts BERT model to compute the evidences' relevance and the veracity of the input claim. In addition, [17,18] leverage reasoning elements over an entity-graph and a hierarchical hypergraph, respectively, to perform the verification process with fine-grained evidences.

Another research direction performs fact-checking relying on knowledge base. To this end, [19] builds a knowledge graph of fact-checked information which can be queried in order to assess the veracity of an input claim. In addition, [20] encodes background knowledge in the form of Horn rules and generates rule-based explanations supporting the veracity prediction of the claim. [20] determines the claim truthfulness treating the knowledge graph as a flow network. More recently, [21] proposes to use language models as knowledge base, exploiting their factual knowledge acquired during the pretraining process.

Our work analyzes the fact-checking panorama from the perspective of detecting previously fact-checked information: assuming most of the claims are repeated over time, especially on social media platforms, we aim to detect if an input claim has been already checked and stored in a predefined knowledge base. It is worth noting that evidence-based fact-checking approaches [11,15] differ from the considered task because while the former aims at predicting the claim's veracity by understanding whether some evidences support or refute it, the latter does not depend at all on the claim veracity. Indeed, if an input claim has been already fact-checked there is no point in verifying it again regardless of its truthfulness. In other words, detecting previously fact-checked information supports the fact-checking process in a preliminary phase, and in principle, is complementary to evidence-based approaches.

Despite having been proposed to integrate the checking of the input claim against a knowledge base of verified information in the fact-checking pipeline [4], only during the last year, some initial works proposed their solution. [5] ranked verified information according to their relevance to the input claim. Specifically, they use standard information retrieval algorithms (e.g. BM25) and compute cosine similarity over the embedding produced by a not fine-tuned BERT model. In addition, competitors at CheckThat!2020 [8] showed that different fine tuning strategies lead to promising performance improvements. Finally, [22] addresses the problem using multimodal data, i.e. the texts and the images of the claim and of the verified information.

Despite the overall ranking performance, no one has analyzed the efficiency of the proposed approaches. For instance, winners at CheckThat!2020 [23] explicitly declares that their approach is unfeasible with large-scale documents' corpus because it would take hours to retrieve the top-k element for an input claim. From this perspective, we consider a more realistic scenario where not only retrieval performance but also execution times should be considered. In other words, we fully explore the trade-off between effectiveness and efficiency to understand the best operating settings for such systems.

Furthermore, given the information retrieval nature of the task, there is a wide range of powerful, yet not explored, architectures and models [9,10] which could be used to optimize the overall performance. We try to bridge this gap considering a retriever-reranker architecture and benchmark a broad range of models considering both their efficiency and effectiveness.

2.2. Multi-stage ranking models

Ranking list of documents according to some queries is a common problem when performing information retrieval tasks. Specifically, when the document corpus is very large, multi-stage pipelines are the de-facto standard to solve the problem [9]. In other words, the first stage *retriever* performs top-k document retrieval, i.e. the potential set of documents relevant to the query; the second (and, in case, its successors) stage *reranker* aims at reordering that set of candidates with more powerful and computationally expensive models.

Retriever. The first-stage retrieval task has long been dominated by the classical term-based probabilistic models (e.g. BM25 [24]) due to their efficiency and effectiveness even with million-scale corpus of documents. Nevertheless, they still suffer from the vocabulary mismatch problem [25] and do not model the document semantics which is essential when considering text's meaning. While in the past decades term dependency and topic models [26–28] have addressed the former problem, the unprecedented performance improvements that transformer architectures and representation learning strategies are achieving in NLP, have determined an explosive growth of works proposing their neural network-based semantic first-stage retriever. [9] classifies neural retriever into two categories – *sparse retrieval methods* and *dense retrieval methods*. The former strategies adopt efficient sparse representation for query and documents and essentially improve the weighting scheme of the classical term-based methods (e.g. DeepCT [29], docT5query [30]).

On the other hand, *dense retrieval methods* usually consist of a dual-encoder architecture which embed queries and documents independently, the final relevance score is computed through a similarity function f. These methods can be further categorized into *term-level representation learning* and *document-level representation learning* [9]. The former models represent queries and documents with the sequence of their terms' embeddings and f performs term-level matching and aggregates the result to compute the final score (e.g. DC-BERT [31], ColBERT [32]). *Document-level representation learning* approaches find one global representation for each query and document (e.g. Sentence-BERT [33] DPR [34]).

It is worth to note that even if the above-mentioned methods are categorized as first-stage retriever for their efficiency, they can still be used for end-to-end retrieval, performing jointly the retrieval and reranking tasks.

In this work, we benchmark the wide range of the above-mentioned retrievers discussing which category is more promising in the context of retrieving fact-checked information. In addition, we also assess whether the most advanced (neural) models could be exploited as one-stage retrievers without any reranking.

Reranker. Even if some of the retriever models have proven discrete ranking performance [24,32], researchers are working hard to design specialized learning-to-rank systems. In fact, during the last decade, we have witnessed a strong growth in applying deep neural networks for building ranking models, also referred to as neural ranking models (NRMs). Specifically, they can categorized into two classes – *representation-based* and *interaction-based* approaches [10]. The former methods leverage the same bi-encoder plus matching layer architecture adopted by *dense retrieval methods*. Some leading examples are DSMN [35] and ESIM [36], which exploit fully-connected networks and chained LSTMs, respectively, to perform Natural Language Inference tasks. In the domain of fact-checking, NSMN [15], first, combines bidirectional LSTMs and pooling strategy in order to perform jointly evidence retrieval and fact verification.

On the other hand, *interaction-based* NRMs aim to capture relevant matching signals between a query and a document based on word interactions. While pioneering works, i.e. MatchPyramid [37] and KNRM [38], applied deep neural networks to represent the words interaction matrix, more recently pre-trained transformers [6,7] have achieved the new state-of-the-art performance on any ranking-related tasks. In particular, [39] shows the effectiveness of using ensemble of different BERT-models and combining point-wise, pair-wise and list-wise loss functions. Similarly, [40] proposes a two-stages re-ranking pipeline with a point-wise (monoBERT) and a pair-wise (duoBERT) classification models, respectively.

Finally, some hybrid architectures have been proposed (e.g. DUET [41] combining the outputs of models from different categories to produce the relevance score.

Whilst interaction-based approaches leads to better ranking performance compared to representation-based ones, their application for end-to-end retrieval is still limited due to their low efficiency in online ranking scenarios [10].

In this work, we select models from both categories and apply them to rerank previously retrieved top-k fact-checked documents according to their relevance to the input claim. Specifically, we assess to what extent one model should be preferred to another considering their efficiency and effectiveness.

3. Method

3.1. Problem formulation

The task of detecting previously fact-checked information aims at improving the fact-checking process by filtering out all information that has been already verified. Thus, the task deals with an input claim c and a (large) corpus of fact-checked documents $\mathcal{D} = \{d_1, d_2, \dots, d_N\}$. It is worth to note that while c does not have a predefined structure (e.g. statements from a political debate or social network posts), a document d_i represents a formal assessment of the claim under verification, indeed it provides its context and all arguments which should be considered in the evaluation.

The information retrieval formulation of this task follows the retriever-reranker paradigm: the first-stage retriever should learn a function $s : \{ (c, d_i) \mid d_i \in \mathcal{D} \} \rightarrow \mathbb{R}$ which assigns high scores to relevant (c, d) pairs and low scores to irrelevant ones. In other words, the retriever aims at finding the set $\bar{\mathcal{D}} = \{\bar{d_1}, \bar{d_2}, \dots, \bar{d_M} \mid M \ll N\}$ of potentially relevant documents with respect to c, thus making $\bar{\mathcal{D}} \subset \mathcal{D}$. Consequently, the second-stage reranker should learn a function $f : \{ (c, \bar{d_i}) \mid \bar{d_i} \in \bar{\mathcal{D}} \} \rightarrow \mathbb{R}$ which reorders the elements in $\bar{\mathcal{D}}$ according to how much similar they are to the input claim c. We observe that learning s is not trivial for at least two reasons. First, it has a strong efficiency requirement dependent on the million/billion scale documents' corpus it deals with. Second, it should be flexible enough to integrate additional knowledge about new events. By the same token, learning f could be difficult since input claims can be phrased differently with respect to the corresponding fact-checked information, even if they refer to the same concepts [5]. Another problem to deal with is that complex claims might be the conjunction of different verified claims, thus making even partial matches relevant for the task.

It is worth to note that this task does not depend on the claim truthfulness but can support its estimation since we assume that the documents in the corpus have been already fact-checked and thus can be used as evidences for the verification task.

3.2. Benchmarking architecture

As mentioned in the previous section, ranking problems are widely common in information retrieval tasks, and machine learning approaches are more and more studied to propose effective solutions. In order to integrate and compare the most recent advancements of neural ranking and retrieval models with the classical information retrieval approaches, we considered the two-stages learning-to-rank model depicted in Fig. 1.

The first-stage retriever aims at selecting the subset $\bar{\mathcal{D}}$ of the documents corpus. Specifically, it assumes that the input claim and the most related documents share some basic properties such as mentioning

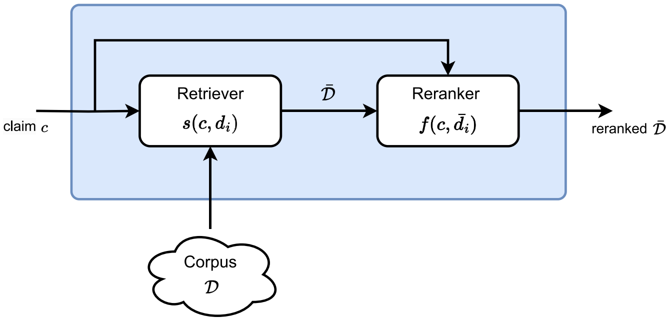


Fig. 1. Pipeline of our benchmarking architecture.

the same entities, having similar statistical representation (e.g. TF-IDF features) or referring to the same concepts/topics. In other words, the retriever filters out the completely unrelated verified information in $\mathcal{D}$. We do not expect that it has high ranking performance; but we require that it has good recall scores in order to not affect the reranking performed by the second step. To put it differently, a pre-selection algorithm which leaves out too many relevant documents, would become a bottleneck for the performance of the overall system. Moreover, since the retriever deals with huge quantities of documents, it should be efficient and scalable with respect to the size of the corpus. We will evaluate the effect of the algorithm's choice in the experiment section.

The second-stage reranker is an advanced NRM which models the intrinsic semantics of the claim and the (subset of) documents in order to perform an high performance re-ranking. In other words, once the retriever has filtered documents which are correlated with the input claim at high level, the reranker performs semantic matching trying to assess whether the input query and document, i.e. the $(c, \bar{d}_i)$ pair, convey, even partially, the same meaning/concepts.

It is worth to note that the chosen multi-stage pipeline allows us to benchmark both interaction-based and representation-based rerankers without affecting too much the system's efficiency since even if the NRM is computationally expensive, it should only predict the relevance between the input claim and a much smaller set of verified documents, i.e. those selected by the retriever algorithm. The extent to which this re-ranking process affects the effectiveness and the efficiency of the framework will be evaluated though our experiments.

Despite some recent attempts to build end-to-end neural retrieval systems [32,42], we conjecture that the multi-stage pipeline, besides improving efficiency, might also increase the performance of the reranker due to the simpler problem it has to solve when combined with the retriever. In other words, if we use the ranking model alone, it should learn to distinguish between the input claim's semantic and all possible knowledge contained in the documents' corpus, while in our settings it has to work in a more controllable environment where the training procedure could assume a certain degree of pertinence between the claim and the documents to (re-)rank.

Finally, even if the framework is thought to work with corpus represented as lists of verified documents, we point out its flexibility towards knowledge graph (KG) representations: specifically, without any alteration of the reranker model, the retriever algorithm is replaced by an inference procedure on the KG using the entity and the relationships extracted from input claim. We leave the exploration of this scenario to future works.

4. Experiments

All experiments have been performed on Google Colab equipped with one single core hyper threaded Xeon Processor @2.2 GHz, 12 GB of RAM and a Tesla T4 GPU. The code will be made available on Github.

4.1. Research questions

Trying to merge the wide literature on retrieval and ranking models to detect previously fact-checked documents, we design our research objectives in order to assess which methods are better suited for our two-stages pipeline. Furthermore, we want to evaluate both the effectiveness and efficiency of the framework with respect to the performance of single models and the actual applicability in real scenarios. Concretely, we want to answer the following research questions:

- (RQ1) Which are the best retrievers? Can modern neural semantic techniques replace the standard term-based approaches?
- (RQ2) Which are the best neural (re-)ranking models?
- (RQ3) What is the benefit of combining retrievers and rerankers with respect to the overall performance?

4.2. Dataset & Metrics

We considered the dataset provided by [5], consisting of 1000 tweets retrieved from Snopes[6] fact-checking articles, and of 10 396 verified claims extracted from ClaimsKG dataset [19]. Specifically, data refers to multiple domains, including politics and gossip, and the tweet and its verified document may be phrased similarly, thus allowing a simpler approximate matching algorithm to work properly, or with different terms, thus requiring more refined and semantic-based strategies to perform the correct match.

We used the standard split 60%/20%/20% split, the authors provided, for training, validation and testing sets. As in most information retrieval tasks, many verified claims never appear as related to any of the original tweets.

For the ranking formulation, we adopted Mean Reciprocal Rank (MRR), Mean Average Precision truncated at k (MAP@k) and the hit ratio [43] truncated at k (HasPositives@k), as evaluation metrics. While the first two metrics take into account the ranking order, the last one evaluates the capability of the system to retrieve correct matches. It is worth to note that since most of the tweets have only one relevant document, HasPositives@k is almost equal to Recall@k. In addition, we performed the statistical t-test between top-ranked models to assess the reliability of our results.

From the application perspective, metrics on lower values of k (e.g. $k \in \{1, 3, 5\}$) might be indicative of the system utility in easing manual fact-checkers works, i.e. experts would spot in real time if the top ranked results are relevant to the input claim. On the other hand, metrics on higher values of k (e.g. $k \in \{10, 20\}$) should be considered in offline settings and/or in an automated fact-checking pipeline where results should be further processed as evidences for the veracity prediction.

4.3. Models & Training details

In the following subsection we detail which are the retrievers/rerankers considered in the benchmark, explaining how they have been trained and configured in order to promote reproducibility.

We select a wide range of retrievers dividing them in four groups.

First, we consider classical probabilistic approaches including BM25 [24], TF-IDF [44] and Language Model with Dirichlet smoothing [45]. These algorithms assign a score to each tweet-claim pair based on exact matching between the words in the tweet and the words in a target verified claim. They have been long studied and applied in various information retrieval tasks, thus representing the baseline for the other retrievers. We adopted the Elasticsearch[7] (version 7.10.1)

[6] https://www.snopes.com/.

[7] https://www.elastic.co/.

implementation for BM25 and LM Dirichlet, with default parameters, and used Haystack library[8] for TF-IDF.

Second, we select docT5query [46] as neural sparse retrieval models because expanding the documents with auto-generated queries seems profitable in this context because the query, i.e., the (false) information, is often repeated with a few differences over times. Specifically, we adapted the official code[9] and use the provided *T5-base* model to generate three queries for each document. We then used BM25 to reindex the expanded documents.

Third, we choose ColBERT [32] as neural sparse retrieval models. It is worth noting that ColBERT can be used for reranking as well, due to the interaction mechanism it performs between query and document terms. In particular, we used the official repository[10] and retrained the *bert-based-uncased* model using the default hyper-parameters.

Fourth, we picked SentenceBERT [33] and DPR [34] as neural document-based dense retrieval techniques. The former is the first attempt to leverage transformer-based models to perform text similarity and thus represents our "neural" baseline. Specifically, we used the sentence-transformer library,[11] fine tuning (for 4 epochs and a batch size of 16) the *stsb-distilbert-base* model using cosine similarity loss.

On the other hand, the latter adopts the in-batch negative strategy to reuse negative examples already in the training batch rather than creating new ones. In particular, we used the Haystack library,[8] fine tuning (for 10 epochs and a batch size of 16) the *bert-base-uncased* model.

With the exception of DPR which customizes the batch generation strategy, the training dataset has always been built considering the positive query-document pair and 10 random negative ones.

Considering the second stage of the pipeline, we considered 9 rerankers, divided in the categories mentioned in Section 2.2. Specifically, we choose MatchPyramid [37], KNMR [38], ConvKNMR [47] and BERT models [6], as interaction-based algorithms; ESIM [36] and HAR [48], as representation-based algorithms; DUET [41] as hybrid model.

For HAR we used the official implementation,[12] and for all others methods we adopted the Pytorch implementation of the Matchzoo framework [49]. All hyper-parameters have been set to default with the exception of the number of kernels in KNRM and ConvKNRM which was set to 11. All models have been trained until convergence on the validation set. Finally, for BERT, we adopted the *stsb-distilroberta-base* cross-encoder provided by the sentence-transformer library,[11] fine tuning (for 4 epochs and a batch size of 16) using the cross-entropy loss.

When training rerankers, we need to select k negative samples for each tweet-claim pair. The choice of k might be decisive for the performance of the model: low values might determine poor performance because the model would see few pairs representing non-matching knowledge. On the other hand, since there is just one verified claim matching most of the tweets in our dataset, increasing k too much might lead to imbalanced training set, making the learning task more difficult. We select 50 random negative documents from the top-100 ones retrieved in the first stage. However, in the experiments we also evaluate the effect of a completely random choice.

4.4. (RQ1) Which are the best retrievers?

Table 1 reports the results of the retriever models. We do not report latency performance since the documents' corpus is too short to observe significant differences between the chosen models.

While not reaching BM25 performance, it is evident to notice the progress of neural retrievers which overcome the TF-IDF baseline and perform comparably with the LM Dirichlet one.

The sparse model docT5Query [46] is the first runner up and exhibits great improvements with respect to BM25 on which it relies. In other words, we conjecture that expanding fact-checked documents with artificially-generated queries and then indexing through standard techniques (BM25 in our case) is an effective approach because the generation process clearly extracts the subjects, the topics and/or the events increasing the probability of detecting matching queries citing those concepts. Unfortunately, we point out how the queries' generation procedure, relying on the T5 transformer [50], is computationally intensive and might not be usable in online scenarios where the documents' corpus should be often updated.

On the other hand, ColBERT [32] achieves interesting performance without requiring any pre-processing step. Moreover, the late interaction mechanism it implements between query and document words seems efficient enough to be scalable with million scale corpora.

Finally, document-level neural retrievers (SentenceBERT [33] and DPR [34]) are one step behind the other approaches, probably because representing the whole document/query with just one embedding provides a coarse representation, which does not capture the necessary details to infer the relation between the claim and its verified document. Concretely, fact-checked documents are usually characterized by longer texts which cite several concepts and entities to assess the claim veracity. Under such scenario, it is difficult to provide an insightful representation looking at the document as a whole, instead of considering more granular information (e.g., text's terms and/or sentences).

To sum up, the recent advancements in neural information retrieval seem to be bridging the gap with classical retrieval approaches but we have shown that even the most modern retrievers still cannot replace them in practice. In addition, combining the two approaches and designing more efficient interaction functions are the most promising research directions to follow.

4.5. (RQ2) Which are the best neural re-ranking models?

Table 2 reports the rerankers' performance, considering just those queries which have at least one relevant article in the top 50 documents retrieved in the first stage by BM25. Not surprisingly, interaction-based approaches perform generally better than representation-based ones since they explicitly look for relevant matching signals in query-document pairs.

Apart from the reranker's category, the most important insight is that transformer-based models (BERT [6] and colBERT [32] outperform by far other algorithms. Specifically, they reach good results already when considering rankings truncated at top positions, meaning that they could effectively catch the relation between the fact-checked document's and the input claim. Despite ranking performance, the execution time of these models strongly affects the number of documents they could practically rerank, we will analyze this aspect in the next section.

When observing the huge performance difference between transformer-based systems and other NRMs, we conjecture that it depends on the transformers' pre-training procedure, which allows these models to acquire not only language syntax and semantics but also factual and relational knowledge [21]. By contrast, other NRMs (e.g. MatchPyramid [37], KNRM [38] are trained from scratch, thus requiring more training (labeled) data and time to achieve good reranking performance.

Finally, BERT [6] performs better than colBERT [32] because of the more complex interaction mechanism it implements to capture matching signals between the input claim and the verified document. Indeed, although colBERT's late interaction mechanism prioritizes (computational) efficiency, it cannot compete with the full self-attention mechanism BERT relies on.

To sum up, employing and fine tuning pre-trained language models seem to be the best and easiest solution to obtain high-quality rankings. A fairer comparison with other NRMs will be possible when a million-scale dataset of fact-checked information will be released.

[8] https://github.com/deepset-ai/haystack.

[9] https://github.com/castorini/docTTTTTquery.

[10] https://github.com/stanford-futuredata/ColBERT.

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

[12] https://github.com/mingzhu0527/HAR.

Table 1
Performance of retrievers (bold indicates the best results, underline the first runner up).

Category	Model	MRR	HasPositives@k						
		all	$k=1$	$k=3$	$k=5$	$k=10$	$k=20$	$k=50$	$k=100$
Classical	TF-IDF	0.681	0.593	0.739	0.789	0.829	0.869	0.914	0.924
	LM Dirichlet [51]	0.799	0.770	0.825	0.860	0.890	0.915	0.95	0.960
	BM25 [24]	**0.817**	**0.785**	**0.865**	**0.880**	**0.895**	**0.915**	**0.950**	**0.960**
Neural sparse	docT5query [30]	0.786	0.754	0.834	0.844	0.894	0.919	0.945	0.960
Term-based	ColBERT [32]	0.765	0.708	0.793	0.819	0.874	0.904	0.944	0.949
Document-level	SentenceBERT [33]	0.669	0.592	0.713	0.763	0.804	0.834	0.884	0.924
	DPR [34]	0.624	0.547	0.673	0.718	0.753	0.788	0.859	0.909

Table 2
Performance of Neural Ranking Models (NRMs) (bold indicates the best results, underline the first runner up).

Category	Model	MRR	MAP@k				
		all	$k=1$	$k=3$	$k=5$	$k=10$	$k=20$
Interaction -based	BERT [6]	**0.968***	**0.942***	**0.968***	**0.968***	**0.968***	**0.968***
	ColBERT [32]	0.903	0.847	0.893	0.901	0.902	0.903
	MAN [22]	0.509	0.386	0.470	0.484	0.501	0.509
	MatchPyramid [37]	0.495	0.413	0.444	0.462	0.479	0.489
	KNRM [38]	0.319	0.212	0.272	0.287	0.298	0.307
	ConvKNRM [52]	0.744	0.677	0.721	0.729	0.738	0.742
Representation -based	ESIM [36]	0.507	0.370	0.451	0.482	0.498	0.504
	HAR [48]	0.602	0.331	0.508	0.557	0.557	0.560
Hybrid-based	DUET [41]	0.392	0.233	0.302	0.313	0.323	0.330

*Statistical significance at $p = 0.001$ w.r.t. the second best

Table 3
Performance of the overall pipeline (bold indicates the best results, underline the first runner up).

Model	HasPositive@k				
	$k=1$	$k=3$	$k=5$	$k=10$	$k=20$
BM25 [24]	0.785	0.865	0.880	0.895	0.915
BERT [6]	**0.865**	**0.935**	**0.960**	**0.970**	**0.985**
ColBERT [32]	0.793	0.819	0.874	0.904	0.944
BM25 (100) + BERT	0.862	0.925	0.935	0.945	0.955
BM25 (100) + ColBERT	0.779	0.794	0.804	0.804	0.804

Table 4
Performance of the overall pipeline (bold indicates the best results, underline the first runner up).

Model	MRR	MAP@k					
	all	$k=1$	$k=3$	$k=5$	$k=10$	$k=20$	all
BM25 [24]	0.817	0.816	0.819	0.821	0.822	0.817	0.785
BERT [6]	0.901	0.865	0.895	0.901	0.902	0.903	0.903
ColBERT [32]	0.709	0.749	0.754	0.762	0.765	0.765	0.708
BM25 (100) + BERT	**0.906**	**0.873**	**0.905**	**0.908**	**0.908**	**0.908**	**0.909**
BM25 (100) + ColBERT	0.739	0.756	0.760	0.761	0.761	0.762	0.738

Table 5
Effect of negative pairs' selection during reranker training.

Model	MAP@k				
	$k=1$	$k=3$	$k=5$	$k=10$	$k=20$
BERT (random negatives)	0.365	0.525	0.556	0.573	0.575
BERT (top-k negatives)	0.865	0.895	0.901	0.902	0.903

4.6. (RQ3) What is the benefit of combining retrievers and rerankers?

As mentioned in the previous section, the two steps of our framework capture different kinds of information and thus it is worth exploring how their combination performs. Tables 3 and 4 outlines the performance of the overall system, in terms of HasPositive@k and MAP@k respectively, considering the combination of two transformers reranker (BERT and ColBERT) with the best retriever algorithm (BM25). We configure the system so as the latter model selects the top-100 verified claims from the documents' corpus.

It is evident how the combination is effective when combining strong retriever with stronger reranker (BERT), in fact, the overall combination overcomes both its components, considered in isolation. On the other hand, weaker rerankers (ColBERT), as well as weak retrievers, might compromise the performance becoming a bootleneck for the whole system. In complete fairness, we highlight that the results of the two best models are not statistically different but still meaningful because, even if the two solutions perform comparably, the combination between BM25 and BERT is much more efficient than BERT alone, as we will show afterwards.

Furthermore, Table 5 clearly proves that retraining the reranker on the claims retrieved by the first stage positively affects the ranking performance confirming our hypothesis that the use of the prefiltering information retrieval algorithm simplifies the learning task since the NRM does not have to match the input tweet's semantic with all knowledge encoded in the verified claims.

Finally, we evaluate the effect of the number of documents selected by the first-stage retriever. Specifically, we assess how this parameter affects the efficiency and the effectiveness of the overall system. Table 6 reports the runtimes, and their 95% confidence intervals, of the system resulting from the combination of the BM25 retriever and each transformer model, the scenario where no re-ranker is applied is considered as well. While colBERT, as all representation-based models, scales better (even better than the BM25 baseline alone), BERT model runtimes strongly increase (up to 30 s per query) with the number of retrieved documents. Concretely, assuming the system should ease manual fact-checker's effort, we convey that a profitable response time should be within five seconds thus making BERT algorithm not tractable when there are more than 1000 documents to rerank. As a result, the application of complex transformer is tightly constrained with the adoption of a high-recall retriever which filters out the greatest part of the documents' corpus.

In addition, Fig. 2 depicts the MAP metrics varying the top-k fact-checked documents retrieved by the BM25 baseline: not surprisingly, performance generally increase (decrease) when considering BERT (colBERT) reranker. This behavior depends on the fact that when increasing the number of retrieved documents, we are converging on the performance of the second-stage reranker applied in isolation, thus not exploiting the retriever anymore. However, we observe that for BERT, performance no longer improves when retrieving more than 100 documents.

To sum up, information retrieval literature brings a wide range of methods and models, which could be exploited to efficiently solve the problem of detecting previously fact-checked information. Specifically, the multi-stage ranking pipeline seems to achieve acceptable quality performance integrating efficient retrievers with most complex rerankers, making the trade-off between ranking performance and runtimes smoother. Concretely, in the context of this benchmark, we conclude that the best system is composed by the BM25 model,

Table 6
Runtimes (in seconds) varying the number of claims to rerank.

	Without rerank	colBERT	BERT
BM25 (10)	0.0170 ± 0.0017	0.0634 ± 0.0014	0.0500 ± 0.0100
BM25 (100)	0.0233 ± 0.0010	0.0703 ± 0.0014	0.3483 ± 0.0153
BM25 (1000)	0.1156 ± 0.0054	0.1688 ± 0.0053	3.3851 ± 0.1709
BM25 (10000)	0.6122 ± 0.0900	0.7225 ± 0.0908	30.8846 ± 1.5110

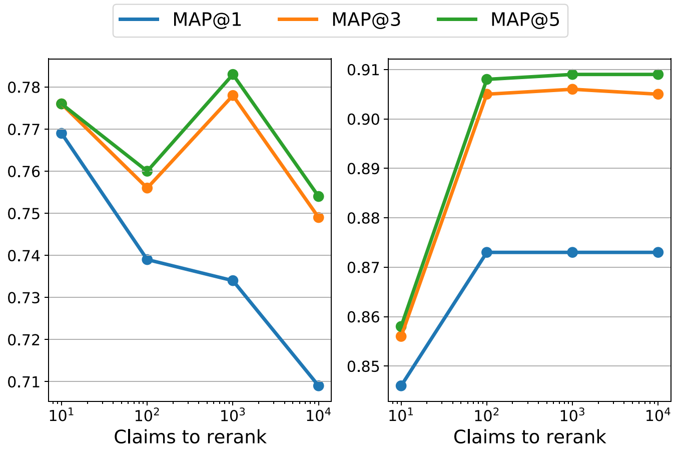


Fig. 2. Performance varying the number of claims retrieved by the first stage and reranked by ColBERT (left) and BERT (right).

retrieving up to 100 fact-checked documents, followed by the BERT model which performs a high performance (re-)ranking.

4.7. Error analysis

In order to better assess the behavior of the framework in real scenarios, we conducted an error analysis aiming at understanding when the best retriever-reranker combination, i.e. the BM25 retriever followed by a BERT reranker, performs better than its components and what kinds of input tweet may still lead to wrong results.

First, we consider the tweet "#SixFlagsBaltimore closed parks to all non-muslims for the Muslim family day. How nice. When is CHRISTIAN day?" whose corresponding verified claim is "Six Flags is temporarily closing one or more of their theme parks to the public to host Muslim Family Day.". In this case, neither the BM25 model nor the BERT model can rank the corresponding verified claim in the first position. In particular, the former algorithm is able to retrieve the right verified claim but ranks it on the fifth position along with other topical-related claims such as "First Lady Michelle Obama proposed a Hug A Muslim day to replace Columbus Day.", while the BERT model applied in isolation cannot retrieve the correct match in the top-20 ranking, since it gets confused by other information which regard the same subject (e.g. Muslims) but in total different contexts (e.g. war, crime). On the other hand, the combination of the two models is effective since it exploits the partial good results of the first stage and exploits the semantic knowledge acquired by the second one to re-rank the correct claim in the first position.

Second, in order to understand the limits of our approach we consider a scenario where both the stages applied in isolation and their combination are not able to perform the correct ranking. Taking the tweet "You should get one for your house! #PizzaVendingMachine", the ES baseline cannot retrieve the corresponding verified claim in the top-100 results, thus penalizing the results of the reranking step as well. On the other hand, the BERT model alone is able to retrieve the correct verified information but, again, it gets confused with other pizza-related claim regarding totally different contexts (e.g. satire, cooking recipes). We believe that this behavior depends on the fact that the considered tweet does not express directly the entities it refers to, thus making the information extraction and semantic understanding much more difficult.

5. Theoretical and practical implications

The misinformation phenomenon can have adverse societal effects, threatening democracies, journalism and freedom of expression. Despite their increasing effort, fact-checking organizations cannot keep the pace of the huge amount of false information spreading on social media. Within this context, detecting previously fact-checked information could improve the verification process increasing fact-checkers productivity and providing more reliable evidence which the assessment should be based on.

Theoretically, our study bridges the gap between the recent ad-hoc proposals dealing with the above-mentioned task and the vast amount of techniques and models that have been proposed in the information retrieval and Q&A literature. To this end, we have defined a retriever-reranking framework to assess the efficiency and the effectiveness of the analyzed techniques, and, as opposed to existing works, we have explored the best trade-off between ranking performance and execution times.

Our results show that combining standard and neural approaches is the most promising research direction to improve retrieval performance; fine-tuned transformer architectures provide high-quality (re-)ranking performance. In addition, these complex rerankers might still be efficient in practice since there is no need to process a high number of documents to improve ranking performance. Finally, our error analysis shows the limitation of standard retrieval algorithms when the inputs texts contain too many entities or do not cite those entities directly.

6. Conclusions and future directions

In this paper, we addressed the problem of detecting previously fact-checked information through a multi-stage ranking pipeline. We have benchmarked state-of-the-art retriever and reranker models, considering also how the combination of standard information retrieval algorithms and modern semantic models might improve the overall performance. The experimental results prove that the integration of standard term-based and neural-based retrievers is the most promising direction to improve top-k document retrieval. In addition, stronger transformer-based rerankers seem to be the most effective solution to perform a high performance reranking due to the extensive knowledge acquired during their pre-training process.

There are a number of avenues of future work that we would like to explore. First, after having considered textual claims and documents, we wonder whether multimodal data might improve performance, i.e. tweets' and articles' images might be helpful in analyzing the semantic relation between the input data and the verified claims. Simultaneously, we would like to explore the possibility to use knowledge graph inference algorithms as first-stage retrievers. In addition, some works should be devoted to collect a much larger dataset regarding claims (and their corresponding verification documents) related to different topics and from different sources. Finally, since the proposed approaches are opaque with respect to the decisions they make during top-k retrieval as well as during reranking, we would like to enhance the interpretability of the system using explainable artificial intelligence techniques in order to unveil the relations that models recognize between the input claim and the verified information.

CRediT authorship contribution statement

Tanmoy Chakraborty: Conceptualization, Methodology, Formal analysis, Writing – original draft, Writing – review & editing, Software. **Valerio La Gatta:** Conceptualization, Methodology, Formal analysis, Writing – original draft, Writing – review & editing, Software. **Vincenzo Moscato:** Conceptualization, Methodology, Formal analysis, Writing – original draft, Writing – review & editing, Software. **Giancarlo Sperlì:** Conceptualization, Methodology, Formal analysis, Writing – original draft, Writing – review & editing, Software.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The authors do not have permission to share data.

Acknowledgements

T. Chakrabrorty would like to acknowledge the support of Ramanujan Fellowship, CAI, IIIT-Delhi and ihub-Anubhuti-iiitd Foundation set up under the NM-ICPS scheme of the Department of Science and Technology, India.

References

[1] H. Allcott, M. Gentzkow, Social Media and Fake News in the 2016 Election, Working Paper 23089, National Bureau of Economic Research, 2017, http://dx.doi.org/10.3386/w23089, URL: http://www.nber.org/papers/w23089.

[2] M. Cinelli, W. Quattrociocchi, A. Galeazzi, C.M. Valensise, E. Brugnoli, A.L. Schmidt, P. Zola, F. Zollo, A. Scala, The COVID-19 social media infodemic, 2020, CoRR;abs/2003.05004. URL: https://arxiv.org/abs/2003.05004. arXiv:2003.05004.

[3] S. Vosoughi, D. Roy, S. Aral, The spread of true and false news online, Science 359 (6380) (2018) 1146–1151, http://dx.doi.org/10.1126/science.aap9559, URL: https://science.sciencemag.org/content/359/6380/1146. arXiv:https://science.sciencemag.org/content/359/6380/1146.full.pdf.

[4] N. Hassan, G. Zhang, F. Arslan, J. Caraballo, D. Jimenez, S. Gawsane, S. Hasan, M. Joseph, A. Kulkarni, A.K. Nayak, V. Sable, C. Li, M. Tremayne, ClaimBuster: The first-ever end-to-end fact-checking system, Proc. VLDB Endow. 10 (12) (2017) 1945–1948, http://dx.doi.org/10.14778/3137765.3137815.

[5] S. Shaar, N. Babulkov, G. Da San Martino, P. Nakov, That is a known Lie: Detecting previously fact-checked claims, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Online, 2020, pp. 3607–3618, http://dx.doi.org/10.18653/v1/2020.acl-main.332, URL: https://www.aclweb.org/anthology/2020.acl-main.332.

[6] L. Zhuang, L. Wayne, S. Ya, Z. Jun, A robustly optimized BERT pre-training approach with post-training, in: Proceedings of the 20th Chinese National Conference on Computational Linguistics, Chinese Information Processing Society of China, Huhhot, China, 2021, pp. 1218–1227, URL: https://aclanthology.org/2021.ccl-1.108.

[7] J. Devlin, M. Chang, K. Lee, K. Toutanova, BERT: pre-training of deep bidirectional transformers for language understanding, 2018, CoRR;abs/1810.04805. URL: http://arxiv.org/abs/1810.04805. arXiv:1810.04805.

[8] A. Barron-Cedeno, T. Elsayed, P. Nakov, G.D.S. Martino, M. Hasanain, R. Suwaileh, F. Haouari, N. Babulkov, B. Hamdan, A. Nikolov, S. Shaar, Z.S. Ali, Overview of CheckThat! 2020: Automatic identification and verification of claims in social media, 2020, arXiv:2007.07997.

[9] Y. Cai, Y. Fan, J. Guo, F. Sun, R. Zhang, X. Cheng, Semantic models for the first-stage retrieval: A comprehensive review, 2021, CoRR;abs/2103.04831. URL: https://arxiv.org/abs/2103.04831. arXiv:2103.04831.

[10] J. Guo, Y. Fan, L. Pang, L. Yang, Q. Ai, H. Zamani, C. Wu, W.B. Croft, X. Cheng, A deep look into neural ranking models for information retrieval, Inf. Process. Manage. 57 (6) (2020) 102067, http://dx.doi.org/10.1016/j.ipm.2019.102067.

[11] J. Thorne, A. Vlachos, C. Christodoulopoulos, A. Mittal, FEVER: a large-scale dataset for fact extraction and VERification, in: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), Association for Computational Linguistics, New Orleans, Louisiana, 2018, pp. 809–819, http://dx.doi.org/10.18653/v1/N18-1074, URL: https://aclanthology.org/N18-1074.

[12] R. Baly, M. Mohtarami, J. Glass, L. Màrquez, A. Moschitti, P. Nakov, Integrating stance detection and fact checking in a unified corpus, in: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers), Association for Computational Linguistics, New Orleans, Louisiana, 2018, pp. 21–27, http://dx.doi.org/10.18653/v1/N18-2004, URL: https://www.aclweb.org/anthology/N18-2004.

[13] M. Nadeem, W. Fang, B. Xu, M. Mohtarami, J. Glass, FAKTA: An automatic end-to-end fact checking system, in: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations), Association for Computational Linguistics, Minneapolis, Minnesota, 2019, pp. 78–83, http://dx.doi.org/10.18653/v1/N19-4014, URL: https://www.aclweb.org/anthology/N19-4014.

[14] K. Popat, S. Mukherjee, A. Yates, G. Weikum, Declare: Debunking fake news and false claims using evidence-aware deep learning, in: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Brussels, Belgium, 2018, pp. 22–32, http://dx.doi.org/10.18653/v1/D18-1003, URL: https://www.aclweb.org/anthology/D18-1003.

[15] Y. Nie, H. Chen, M. Bansal, Combining fact extraction and verification with neural semantic matching networks, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 33, 2019, pp. 6859–6866, http://dx.doi.org/10.1609/aaai.v33i01.33016859.

[16] A. Soleimani, C. Monz, M. Worring, BERT for evidence retrieval and claim verification, in: J.M. Jose, E. Yilmaz, J.a. Magalhães, P. Castells, N. Ferro, M.J. Silva, F. Martins (Eds.), Advances in Information Retrieval, Springer International Publishing, Cham, 2020, pp. 359–366.

[17] C. Chen, F. Cai, X. Hu, J. Zheng, Y. Ling, H. Chen, An entity-graph based reasoning method for fact verification, Inf. Process. Manage. 58 (3) (2021) 102472, http://dx.doi.org/10.1016/j.ipm.2020.102472, URL: https://www.sciencedirect.com/science/article/pii/S0306457320309614.

[18] C. Chen, F. Cai, X. Hu, W. Chen, H. Chen, HHGN: A hierarchical reasoning-based heterogeneous graph neural network for fact verification, Inf. Process. Manage. 58 (5) (2021) 102659, http://dx.doi.org/10.1016/j.ipm.2021.102659, URL: https://www.sciencedirect.com/science/article/pii/S0306457321001473.

[19] A. Tchechmedjiev, P. Fafalios, K. Boland, M. Gasquet, M. Zloch, B. Zapilko, S. Dietze, K. Todorov, Claimskg: A knowledge graph of fact-checked claims, in: C. Ghidini, O. Hartig, M. Maleshkova, V. Svátek, I. Cruz, A. Hogan, J. Song, M. Lefrançois, F. Gandon (Eds.), The Semantic Web – ISWC 2019, Springer International Publishing, Cham, 2019, pp. 309–324.

[20] P. Shiralkar, A. Flammini, F. Menczer, G.L. Ciampaglia, Finding streams in knowledge graphs to support fact checking, in: 2017 IEEE International Conference on Data Mining, ICDM, 2017, pp. 859–864, http://dx.doi.org/10.1109/ICDM.2017.105.

[21] F. Petroni, T. Rocktäschel, P.S.H. Lewis, A. Bakhtin, Y. Wu, A.H. Miller, S. Riedel, Language models as knowledge bases? 2019, CoRR;abs/1909.01066. URL: http://arxiv.org/abs/1909.01066. arXiv:1909.01066.

[22] N. Vo, K. Lee, Where are the facts? Searching for fact-checked information to alleviate the spread of fake news, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP, Association for Computational Linguistics, Online, 2020, pp. 7717–7731, http://dx.doi.org/10.18653/v1/2020.emnlp-main.621, URL: https://www.aclweb.org/anthology/2020.emnlp-main.621.

[23] M. Bouziane, H. Perrin, A. Cluzeau, J. Mardas, A. Sadeq, Team buster.ai at CheckThat! 2020 insights and recommendations to improve fact-checking, in: CLEF, 2020.

[24] S. Robertson, H. Zaragoza, The probabilistic relevance framework: BM25 and beyond, Found. Trends Inf. Retr. 3 (4) (2009) 333–389, http://dx.doi.org/10.1561/1500000019.

[25] G.W. Furnas, T.K. Landauer, L.M. Gomez, S.T. Dumais, The vocabulary problem in human-system communication, Commun. ACM 30 (11) (1987) 964–971, http://dx.doi.org/10.1145/32206.32212.

[26] D. Metzler, W.B. Croft, A Markov random field model for term dependencies, in: Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '05, Association for Computing Machinery, New York, NY, USA, 2005, pp. 472–479, http://dx.doi.org/10.1145/1076034.1076115.

[27] D.M. Blei, A.Y. Ng, M.I. Jordan, Latent Dirichlet allocation, J. Mach. Learn. Res. 3 (null) (2003) 993–1022.

[28] D.D. Lee, H.S. Seung, Algorithms for non-negative matrix factorization, in: Proceedings of the 13th International Conference on Neural Information Processing Systems, NIPS '00, MIT Press, Cambridge, MA, USA, 2000, pp. 535–541.

[29] Z. Dai, J. Callan, Context-aware term weighting for first stage passage retrieval, in: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '20, Association for Computing Machinery, New York, NY, USA, 2020, pp. 1533–1536, http://dx.doi.org/10.1145/3397271.3401204.

[30] R. Nogueira, W. Yang, J. Lin, K. Cho, Document expansion by query prediction, 2019, CoRR;abs/1904.08375.URL: http://arxiv.org/abs/1904.08375. arXiv:1904.08375.

[31] P. Nie, Y. Zhang, X. Geng, A. Ramamurthy, L. Song, D. Jiang, DC-BERT: decoupling question and document for efficient contextual encoding, in: J. Huang, Y. Chang, X. Cheng, J. Kamps, V. Murdock, J. Wen, Y. Liu (Eds.), Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25-30, 2020, ACM, 2020, pp. 1829–1832, http://dx.doi.org/10.1145/3397271.3401271.

[32] O. Khattab, M. Zaharia, Colbert: Efficient and effective passage search via contextualized late interaction over BERT, in: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '20, Association for Computing Machinery, New York, NY, USA, 2020, pp. 39–48, http://dx.doi.org/10.1145/3397271.3401075.

[33] N. Reimers, I. Gurevych, Sentence-BERT: Sentence embeddings using siamese BERT-networks, in: K. Inui, J. Jiang, V. Ng, X. Wan (Eds.), EMNLP/IJCNLP (1), Association for Computational Linguistics, 2019, pp. 3980–3990, URL: http://dblp.uni-trier.de/db/conf/emnlp/emnlp2019-1.html#ReimersG19.

[34] V. Karpukhin, B. Oguz, S. Min, P. Lewis, L. Wu, S. Edunov, D. Chen, W.-t. Yih, Dense passage retrieval for open-domain question answering, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP, Association for Computational Linguistics, Online, 2020, pp. 6769–6781, http://dx.doi.org/10.18653/v1/2020.emnlp-main.550, URL: https://www.aclweb.org/anthology/2020.emnlp-main.550.

[35] P.-S. Huang, X. He, J. Gao, L. Deng, A. Acero, L. Heck, Learning deep structured semantic models for web search using clickthrough data, in: Proceedings of the 22nd ACM International Conference on Information & Knowledge Management, CIKM '13, Association for Computing Machinery, New York, NY, USA, 2013, pp. 2333–2338, http://dx.doi.org/10.1145/2505515.2505665.

[36] Q. Chen, X. Zhu, Z.-H. Ling, S. Wei, H. Jiang, D. Inkpen, Enhanced LSTM for natural language inference, in: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Vancouver, Canada, 2017, pp. 1657–1668, http://dx.doi.org/10.18653/v1/P17-1152, URL: https://www.aclweb.org/anthology/P17-1152.

[37] L. Pang, Y. Lan, J. Guo, J. Xu, S. Wan, X. Cheng, Text matching as image recognition, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 30, 2016, 1.

[38] C. Xiong, Z. Dai, J. Callan, Z. Liu, R. Power, End-to-end neural ad-hoc ranking with kernel pooling, in: Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '17, Association for Computing Machinery, New York, NY, USA, 2017, pp. 55–64, http://dx.doi.org/10.1145/3077136.3080809.

[39] S. Han, X. Wang, M. Bendersky, M. Najork, Learning-to-rank with BERT in TF-ranking, 2020, CoRR;abs/2004.08476. URL: https://arxiv.org/abs/2004.08476. arXiv:2004.08476.

[40] R. Nogueira, W. Yang, K. Cho, J. Lin, Multi-stage document ranking with BERT, 2019, CoRR;abs/1910.14424. URL: http://arxiv.org/abs/1910.14424. arXiv:1910.14424.

[41] B. Mitra, F. Diaz, N. Craswell, Learning to match using local and distributed representations of text for web search, in: Proceedings of the 26th International Conference on World Wide Web, WWW '17, International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE, 2017, pp. 1291–1299, http://dx.doi.org/10.1145/3038912.3052579.

[42] A. Vakili Tahami, K. Ghajar, A. Shakery, Distilling knowledge for fast retrieval-based chat-bots, in: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '20, Association for Computing Machinery, New York, NY, USA, 2020, pp. 2081–2084, http://dx.doi.org/10.1145/3397271.3401296.

[43] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, T.-S. Chua, Neural collaborative filtering, in: Proceedings of the 26th International Conference on World Wide Web, WWW '17, International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE, 2017, pp. 173–182, http://dx.doi.org/10.1145/3038912.3052569.

[44] H.C. Wu, R.W.P. Luk, K.F. Wong, K.L. Kwok, Interpreting TF-IDF term weights as making relevance decisions, ACM Trans. Inf. Syst. 26 (3) (2008) http://dx.doi.org/10.1145/1361684.1361686.

[45] C. Zhai, Statistical language models for information retrieval, Synth. Lect. Hum. Lang. Technol. 1 (1) (2008) 1–141, http://dx.doi.org/10.2200/S00158ED1V01Y200811HLT001, URL: https://doi.org/10.2200/S00158ED1V01Y200811HLT001. arXiv:https://doi.org/10.2200/S00158ED1V01Y200811HLT001.

[46] D. Cheriton, From doc2query to docttttquery, 2019.

[47] Z. Dai, C. Xiong, J. Callan, Z. Liu, Convolutional neural networks for soft-matching N-grams in ad-hoc search, in: Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining, WSDM '18, Association for Computing Machinery, New York, NY, USA, 2018, pp. 126–134, http://dx.doi.org/10.1145/3159652.3159659.

[48] M. Zhu, A. Ahuja, W. Wei, C.K. Reddy, A hierarchical attention retrieval model for healthcare question answering, in: The World Wide Web Conference, WWW '19, Association for Computing Machinery, New York, NY, USA, 2019, pp. 2472–2482, http://dx.doi.org/10.1145/3308558.3313699.

[49] J. Guo, Y. Fan, X. Ji, X. Cheng, MatchZoo: A learning, practicing, and developing system for neural text matching, in: Proceedings of the 42Nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '19, ACM, New York, NY, USA, 2019, pp. 1297–1300, http://dx.doi.org/10.1145/3331184.3331403, URL: http://doi.acm.org/10.1145/3331184.3331403.

[50] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, P.J. Liu, Exploring the limits of transfer learning with a unified text-to-text transformer, J. Mach. Learn. Res. 21 (2020) 1–67.

[51] C. Zhai, J. Lafferty, A study of smoothing methods for language models applied to ad hoc information retrieval, in: Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '01, Association for Computing Machinery, New York, NY, USA, 2001, pp. 334–342, http://dx.doi.org/10.1145/383952.384019.

[52] Z. Dai, C. Xiong, J. Callan, Z. Liu, Convolutional neural networks for soft-matching N-grams in ad-hoc search, in: Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining, WSDM '18, Association for Computing Machinery, New York, NY, USA, 2018, pp. 126–134, http://dx.doi.org/10.1145/3159652.3159659.

Tanmoy Chakraborty is an Assistant Professor of Computer Science and a Ramanujan Fellow at IIIT-Delhi, India, where he leads a research group, Laboratory for Computational Social Systems (LCS2). His primary research interests include Social Computing and Natural Language Processing. He has received several awards/fellowships including Faculty Awards from Google, IBM and LinkedIn; Early Career Research Award, DAAD Faculty Fellowship. He is a member of ACM and a senior member of IEEE. More details at http://faculty.iiitd.ac.in/∼tanmoy/.

Valerio La Gatta is a Ph.D. student in Information Technology and Electrical Engineering at the Department of Electrical Engineering and Information technology of the University of Naples Federico II. He received the Master degree in Computer engineering from the University of Naples Federico II in 2020. His research interests are focused on Social Network Analysis, eXplainable Artificial Intelligence, Graph Data Mining.

Vincenzo Moscato is an Associate Professor at the Electrical Engineering and Information Technology Department of University of Naples "Federico II". He received the Ph.D. degree in Computer Science from the same University by defending the thesis: 'Indexing Techniques for Image and Video Databases: an approach based on Animate Vision Paradigm". He is one of the leaders of PICUS (Pattern and Intelligence Computation for mUltimedia Systems) departmental research groups and a member of the Big Data and Artificial Intelligence national laboratories within the Consorzio Interuniversitario Nazionale per l'Informatica (CINI). His research activities lay in the area of Multimedia, Big Data, Artificial Intelligence and Social Network Analysis. He was involved in many national and international research projects and coordinated as principal investigator some of the them. He was in the program committees of numerous international conferences and in the editorial boards of several important journals. Finally, he was an author of about 200 publications on international journal, conference proceedings and book chapters.

Giancarlo Sperlì is an assistant professor at the Department of Electrical Engineering and Information Technology of the University of Naples Federico II. He obtained his Ph.D. in Information Technology and Electrical Engineering at the same University defending his thesis: "Multimedia Social Networks". He is a member of the Pattern analysis and Intelligent Computation for mUltimedia Systems (PICUS) departmental research groups. His main research interests are in the area of Cybersecurity, Semantic Analysis of Multimedia Data and Social Networks Analysis. He served as guest editor of different special issues on International Journals. Finally, he has authored about 95 publications in international journals, conference proceedings and book chapters.