Enhancing Information Retrieval in Fact Extraction and Verification

Daniel Guzman-Olivares[1] **Lara Quijano-Sanchez**[1] **Federico Liberatore**[2]
[1] Autonomous University of Madrid, Spain
[2] Cardiff University, United Kingdom
daniel.guzmano@estudiante.uam.es
lara.quijano@uam.es
liberatoref@cardiff.ac.uk

Abstract

Modern fact verification systems have distanced themselves from the black box paradigm by providing the evidence used to infer their veracity judgments. Hence, evidence-backed fact verification systems' performance heavily depends on the capabilities of their retrieval component to identify these facts. A popular evaluation benchmark for these systems is the FEVER task, which consists of determining the veracity of short claims using sentences extracted from Wikipedia. In this paper, we present a novel approach to the the retrieval steps of the FEVER task leveraging the graph structure of Wikipedia. The retrieval models surpass state of the art results at both sentence and document level. Additionally, we show that by feeding our retrieved evidence to the best-performing textual entailment model, we set a new state of the art in the FEVER competition.

1 Introduction

The two-year Coronavirus pandemic and the recent war in Ukraine have evidenced how easily disinformation spreads among the general public and the social consequences this can have. In the information era's day-to-day, we live in a super-connected media ecosystem that provides us with an endless stream of facts and hoaxes alike but no immediate tools to separate them (Olan et al., 2022; Barua et al., 2020). Moreover, the rapid development of larger and more capable language models has made disinformation detection significantly harder since traditional fact-verification systems, usually framed as textual entailment classification problems, are now vulnerable to synthetic disinformation attacks (Du et al., 2022; Stiff and Johansson, 2022). Therefore, modern high-performing fact-verification systems include a previous information retrieval step to condition the posterior veracity judgment on the extracted evidence (Lewis et al., 2020b; de Jong et al., 2022; Glass et al., 2022).

The FEVER task (Thorne et al., 2018a) consists in retrieving relevant evidence from Wikipedia given a claim and labeling it as either *Supports*, *Refutes*, or *Not enough info*. Traditionally, systems participating in the FEVER challenge have divided the task into three steps (Thorne et al., 2018b), each corresponding to a part of their pipeline: the document retrieval step, the sentence retrieval step, and the textual entailment step. In contrast with the last two steps, for most top-performing systems, the document retrieval module is directly inherited or slightly modified from previous work (Hanselowski et al., 2018; Nie et al., 2018). Therefore, for this step, the majority of systems follow one of these two strategies:

The MediaWiki API + span-matching system (Hanselowski et al., 2018). Filtering relevant documents by querying the MediaWiki API for each entity mentioned and discarding results if the entity is not present in the page's title.

Keyword matching + semantic similarity system (Nie et al., 2018). Keyword matching search for initial filtering and Neural Semantic Matching Model (NSMN) for scoring candidate documents using a concatenation of their title and first sentence along with the claim.

These approaches, although proven effective, pose three important limitations:

L1. The usage of MediaWiki API as a first document retrieval step limits the usability of the models outside Wikipedia's scope.

L2. The precision of representing an entire document using only its title and first sentence may prove insufficient to correctly assess semantical relevance.

L3. Discarding a document based on exact keyword matching can be excessively conser-

Proceedings of the Sixth Fact Extraction and VERification Workshop (FEVER), pages 38–48

vative considering query-reference flexibility (e.g. *Michael Jackson-The King of Pop*).

Having identified the above research gaps, we pose the following research hypotheses:

H1. An encoder used for asymmetric semantic search eliminates the MediaWiki API dependency and can more effectively represent semantic relations between queries and documents.

H2. Considering parts of documents as a connected network of path-related pieces of information improves the retrieval quality (specially on queries requiring evidence from more than one document).

Hence, to test the above hypotheses, in this paper we present a novel approach to the document retrieval step for the FEVER task[1]; independent of external resources and capable of retrieving multi-hop evidence while handling partial and even misspelled references in claims. Although our work is mainly focused on the document retrieval step, we also provide a complementary model for sentence retrieval. Our approach establishes a new state of the art in both information retrieval steps and the textual entailment step.

2 Background

The vast majority of systems participating in the FEVER task challenge divide their pipelines into three steps and import their document retrieval step from previous work (Zhou et al., 2019; Stammbach, 2021; Krishna et al., 2022). It is worth mentioning that although some systems (Liu et al., 2020; Zhong et al., 2020; Soleimani et al., 2020) have embedded the baseline document retrieval strategies directly into their architectures, more recent models (Stammbach, 2021; Jiang et al., 2021b) have shown better results by concatenating the retrieved documents from the two baseline models (i.e., Hanselowski et al. (2018); Nie et al. (2018)) with other classical information retrieval techniques such as TF-IDF (Ramos, 2003) or BM25 (Robertson and Zaragoza, 2009).

The second step of most FEVER pipelines consists of performing sentence retrieval from the previously obtained documents. Unlike the previous step, this task has been explored from various perspectives. In the early days of the FEVER task, systems used ESIM-based architectures (Hanselowski et al., 2018; Nie et al., 2018). However, motivated by maximizing recall, the research focus changed to target the multi-hop evidence problem leading to the first iterative sentence retrieval models (Stammbach and Neumann, 2019; Subramanian and Lee, 2020). These models use transformers (Vaswani et al., 2017) to fine-tune large pre-trained language models (LM) used as backbone, such as BERT (Devlin et al., 2019), ALBERT, (Lan et al., 2020) or RoBERTa (Liu et al., 2019). Specifically, to target the multi-hop evidence problem, these models conceive the sentence retrieval step as an iterative process in which they assess the importance of new sentences by considering both the claim and the relevant sentences already retrieved.

Parallel to the iterative retrieval models, another variety of models leverage not only direct connections but the complete graph structure of Wikipedia to rank sentences (Zhong et al., 2020; Liu et al., 2020; Zhou et al., 2019) using graph neural networks (GNNs) (Scarselli et al., 2009). State-of-the-art models (Jiang et al., 2021b; Stammbach, 2021; Krishna et al., 2022) generally fall under one of these categories but have pivoted to more refined token-level representations or bigger LMs such as BigBird (Zaheer et al., 2020), T5 (Raffel et al., 2020) or DeBERTa (He et al., 2021). A recent approach, Claim-Dissector (Fajcik et al., 2022) proposes to divide the retrieved documents into blocks instead of individual sentences and encode each block individually.

The final step of the FEVER task involves recognizing textual entailment (TE). This subtask has traditionally been treated as a multi-class classification problem and tackled by fine-tuning from scratch some LM making use of transformers, alignment and concatenation of the retrieved evidence (Zhou et al., 2019; Liu et al., 2020; Subramanian and Lee, 2020). Top-performing systems in FEVER's public leaderboard (Fajcik et al., 2022; Stammbach, 2021) use DeBERTa-based models already trained over the Multi-Genre Natural Language Inference (MNLI) task (Williams et al., 2018) as backbone.

3 Formal task

The FEVER task consists in performing evidence-backed claim verification. Formally, the knowledge

[1]Results, intermediary files and code will be released on https://github.com/DanielGuzmanOlivares/fever-retrieval.

base, $\mathcal{D}$, is a collection of more than 5 million documents each corresponding to a Wikipedia page, $\mathcal{D} := \{d_i\}_i$, where each document d_i is itself a variable-size collection of sentences, $d_i := \{s_j^i\}_j$. Given the collection of documents $\mathcal{D}$ and a query (a statement) q , a valid system $\mathcal{S}$ must return a veracity assessment $\widetilde{v}$ for q along with a subset $\widetilde{\mathcal{E}}$, of at most five sentences supporting or refuting q:

$$\mathcal{S}(q; \mathcal{D}) \longrightarrow (\widetilde{v}, \widetilde{\mathcal{E}}) \quad \text{s.t.}$$

$$\begin{cases} \widetilde{\mathcal{E}} \subset \bigsqcup_{\mathcal{D}} d_i \\ |\hat{\mathcal{E}}| \leq 5 \\ \widetilde{v} \in \{\text{Supports, Refutes, Not Enough Info}\} \end{cases}$$

Datasets. The FEVER task, as of today, has three associated datasets: the training dataset, the shared task dev dataset, and the shared task test dataset (open competition) (Thorne et al., 2018b). The training dataset is the largest of the three containing 145,449 claims and is unbalanced towards the "*Supports*" class, which represents more than half of the examples. The dev and test datasets are widely used as the evaluation benchmarks for a FEVER pipeline. They are equal in size (19,998) and balanced between the three classes.

Metrics. Following previous work, for evaluating performance we use accuracy (ACC) in the textual entailment step, the FEVER score (FS)[2] for the whole pipeline, and Recall@K (R@K) for the retrieval steps. Additionally, we also consider the Mean Reciprocal Rank (MRR) and the proportion of claims where the system returns at least one relevant item (AND) in the retrieval tasks.

4 Model

Following the traditional pipeline organization, we propose a three-step architecture (see Figure 1) where: i) The document retrieval step uses partial references in the claim and document-level encoding to select an initial collection of documents that is later expanded (if necessary) for addressing the multi-hop evidence problem; ii) The sentence retrieval step combines the sentence retrieval part of $LF_{\text{2-iter}} + D_{XL}$ model (Stammbach, 2021) which is the current best-performing system with a DeBERTa-based cross-encoder (Reimers and Gurevych, 2019); iii) The textual entailment step uses the MNLI-trained DeBERTa model used in $LF_{\text{2-iter}} + D_{XL}$ with our retrieved evidence.

4.1 Data processing

The whole data ecosystem associated with our proposed system is graph-based[3] and consists of:

A reference lookup table. Where all the references to documents are stored, the indexing format is (document title -> list of references) (e.g., Obama -> [Barack Obama, President Obama ...]).

A graph database. Implemented as a Neo4J database, mimics the graph structure needed to get neighbours and references from the given collection of documents.

An embedding database. Pre-computed document embeddings indexed by title to ease the workload of GPU computations.

A sentence database. Containing all the sentences for each document in the provided collection.

We implement the data interface transforming the given Wikipedia dump. This process can be summarized in the following steps for every record in the dump: i) Extract all relevant information from plain-text Wikipedia entry, this includes separated sentences, and the links to external articles; ii) The second step is querying the reference lookup table (initially empty) to check if the linked references already exist in the table. Should any of the references not be present in the table, we query the WikiMedia API to update the records; iii) Once the references are updated in the lookup table, the connections are added to the graph database as new edges; iv) The embedding database is updated with the embedding obtained from the encoder model; v) The sentence database is updated as well with the associated article sentences. This process is represented in Figure 2. In Figure 1, the data interface that the pipeline uses consists of all the aforementioned parts and is represented as *Graph-ref database*.

4.2 Document Retrieval

To the best of the authors' knowledge, the top-performing architectures use a baseline model (Hanselowski et al., 2018) approach combined with

[2]FS is the central metric for the FEVER task. A prediction is only deemed correct if the label is correct and the evidence is sufficient.

[3]The system expects a graph of interconnected documents where connections represent references between documents.

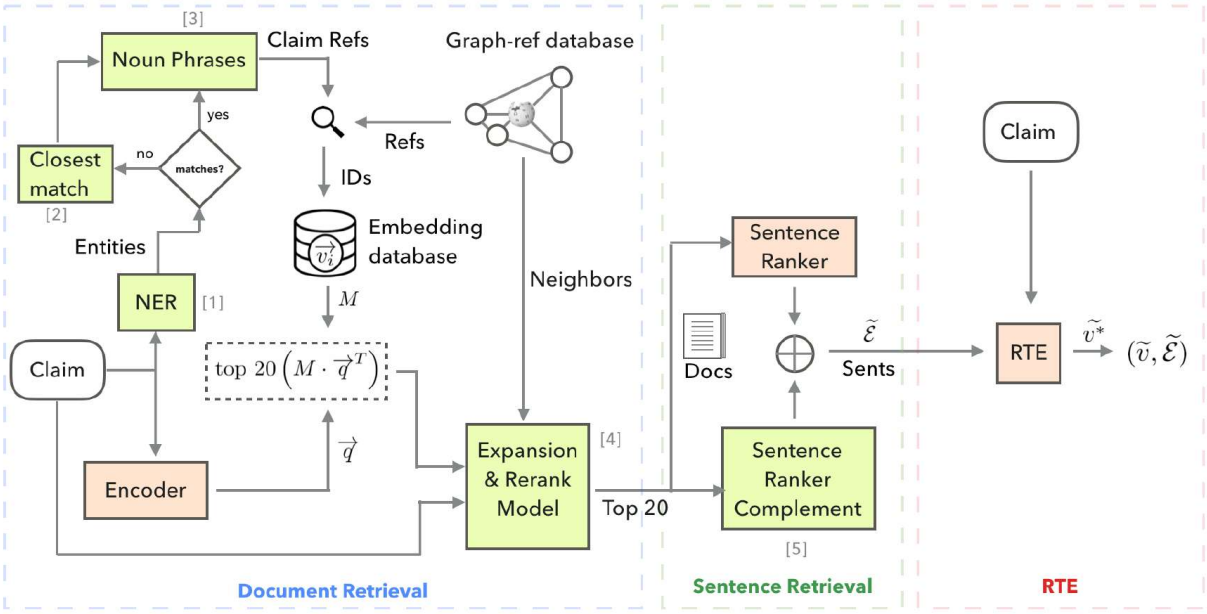

Figure 1: Full proposed pipeline. The pipeline is divided into the three traditionally used sub-architectures. The acid yellow components have been implemented from scratch for this work, whereas the salmon ones are imported from previously existing architectures. Note that each developed component has an index [x] later referenced in the corresponding module description and ablation study.

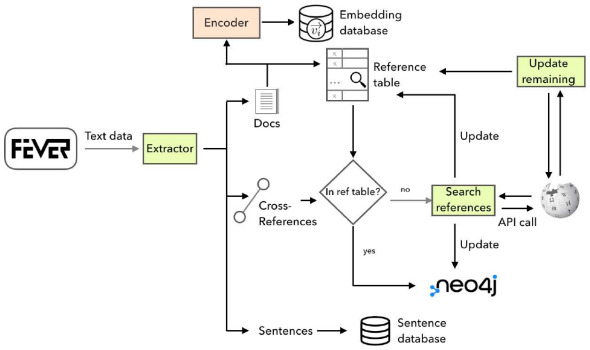

Figure 2: Data processing for the proposed solution. The green components have been implemented from scratch for this work whereas the salmon ones are imported from previously existing architectures.

TF-IDF or BM25 to return an average of 20 documents. On the other hand, the document retrieval part of our architecture consists of various modules sequentially interconnected to output a selection of document paths associated with the input query. In order, these components are the following:

Name Entity Recognition [box [1] in Figure 1]**.** We have trained a token-level classification module using BERT (base) as backbone, for balance between complexity and performance (F_1=0.95). Specifically, we have framed this task as a three-class classification problem (see Appendix A) using BIO labels (Ramshaw and Marcus, 1995) following the traditional approach in NER architectures (Li et al., 2020; Jiang et al., 2021a; Xia et al., 2022).

Although many existing pre-trained models exist to perform this particular task, after trying some publicly available models in random samples for NER, we found that some entities were missed. Therefore, we have opted for training our own model since a considerable number of entities in Wikipedia differ from the classical form of an entity (e.g., a country, a person, a place, a work of art). Such Wikipedia entities usually resemble something like *history of something*, *presidency of someone*, or even concepts that are not considered as entities per se like *water* or *banana*.

Closest Match [box [2] in Figure 1]**.** This module is motivated by observed annotation errors in the FEVER dataset (e.g., *Mellila* - *Melilla*). Since the document retrieval pipeline uses the reference lookup table for finding documents indexed by references, if one of these is not grammatically correct, it would not yield any matches. To avoid this particular case, a conditional path bifurcation has been added in between the NER module and the Noun Phrases selection (see Figure 1). In case a reference yields no results, the closest-match search triggers. The closest-match search takes as input the retrieved sequence from the NER step

and finds the closest (normalized edit distance) reference to it, effectively ensuring there is always at least one associated reference for every sequence[4].

Once references have been associated to sequences of tokens, there are various plausible candidates for relevant pages. At this point, the approximate position of the entities within the claim is known thanks to the token-level classification of the NER module. However, detecting the entities' extension can be especially problematic for the cases in which an entity includes a modifier, which makes it hard for the NER system to fully recognize it as part of the relying entity. For example, in '*Charles **II** of England was born on Thursday 29 May 1630*,' the **II** directly impacts the evidence that should be retrieved.

Noun-phrases selection [box [3] in Figure 1]**.** This module addresses this problem by using three different language planes in order to capture entities:

The semantic plane. Uses the NER pipelines from Flair (Akbik et al., 2019) and SpaCy (Honnibal et al., 2020) since they are trained for a wider variety of entities and can retrieve information that the proposed NER system might miss.

The syntactical plane. Uses the AllenAI Open Information Extraction (OEI) system (Stanovsky et al., 2018) to extract the syntactical subject and direct object of a claim. Relevant to the cases that are not associated with an object or an event (e.g. *Water is part of the **History of Earth.***)

The ontology plane. Rule-based parsing built on top of SpaCy's dependency parsing. Essentially retrieves modifiers not included in the entities provided by the NER module.

Finally, the information retrieved by the three planes and the already predicted references (from the NER module) are combined and later joined with the lemmatized version of the NER references (see Figure 3). This process allows us to accurately extract multiple candidate entities given a claim, mitigating the WikiMedia dependence from previously proposed solutions (see **L1** in Section 1).

Encoding. At this point, the complete set of references is available, and, by using the lookup table,

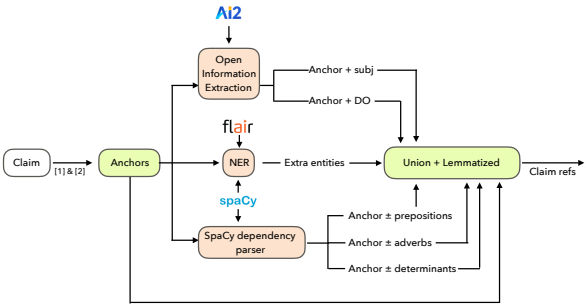

Figure 3: The Noun Phrase module internally. We use green for custom architectures and salmon for imported ones. Anchors represent the references obtained by the pipeline till this point since it obtains the probable points of the claim where entities are.

the associated documents are retrieved. Since, on average, there are too many documents to move on to the next step of the FEVER pipeline, the system uses semantic relatedness to assess the importance of documents conditioned to the claim. In practice, this means encoding the claim (Hofstätter et al., 2021) to obtain the query vector $\overrightarrow{q}$. Then the vectors associated with the selected documents are retrieved from the embeddings database[5] and stacked in a matrix M. Finally, we multiply $M \cdot \overrightarrow{q}^T$ obtaining the vector of semantic closeness for every query-document pair. We select the top $\mathcal{K} = 20$ documents corresponding to the largest entries of the vector.

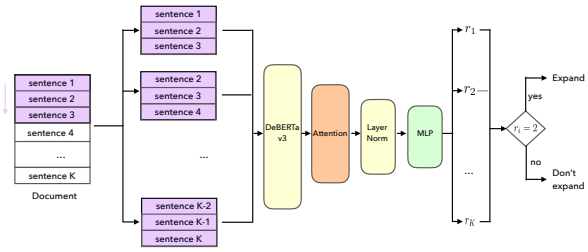

Figure 4: The Expansion model. Note the expansion window sliding (in purple).

Expansion and Rerank [box [4] in Figure 1]**.** This module consists of a two-step architecture that leverages the graph structure of the built database to improve recall. While the previous modules provide twenty documents directly related to the references in the claim, this module goes one step further and explores the neighbours of the provided documents (expansion) to estimate the importance of the second-order documents

[4]This makes the system more flexible than those discarding non-exact matches (see **L3** in Section 1).

[5]Note that the full documents are pre-encoded in contrast with just the title and first sentence (see **L2** in Section 1).

(neighbours) given the claim and the first-order documents.

In the expansion stage, instead of considering every neighbour of every document provided by the previous component in the pipeline, a model has been developed to decide which documents are worth expanding to optimize performance and ease the workload in the sentence retrieval step (since we only expand relevant parts of the initial document). For training this model we divide a document in consecutive overlapping (context) windows and treat the problem as a 3-way classification in which each window's class correspond to the amount of relevant information contained on it (none, some, or all) (see Appendix B).

Preliminary experiments showed that context windows of three consecutive sentences offer the best performance. For each of these, the sentences are concatenated (separated by the [SEP] special token) along with the document's title for helping coreference disambiguation (Malon, 2018). Then for every concatenation, the DeBERTa V3 model is used to obtain the context embedding from both concatenation and claim. Afterward, both embeddings are concatenated and fed to a custom attention head (see Figure 4). Finally, the document is expanded if any of the context-concatenations is evaluated as SOME INFORMATION PRESENT. Following the expansion, we group the resulting collection of documents in paths according to expansion results (i.e. for a given document d_1, if d_1 is expanded obtaining neighbours $n_1, n_2, ..., n_m$ we group paths $(d_1, n_1), (d_1, n_2), ..., (d_1, n_m)$, otherwise if $d1$ is not expanded, only (d_1) is considered as a single path).

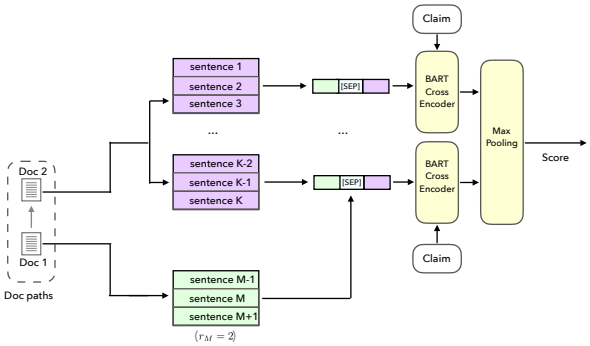

Figure 5: The Rerank model. For a case with a path of length 2, the first-order document's context window (in green) is used as a complement to every context window (sliding) for the second-order document (in purple)

Following the above module we have to assess the semantic relatedness to the query of a rather large set of interlinked documents. The Rerank model is an efficient way to accomplish this task. Internally, the model modifies the classical cross-encoder architecture to use a linked source. We can distinguish two cases regarding the input path's length:

Path of length 2. First, we start by using the context windows again, fully sliding for the second order document and just using the context window that originated the expansion in the first order document. We concatenate the sentences from the first order window with every sentence of the second order window and create a large concatenation of sentences (see Figure 5).

Path of length 1. In this case, we only have one document, so we separate every sentence from the document instead of creating concatenations.

We then feed the concatenations, along with the claim to a BART-based[6] (Lewis et al., 2020a) cross-encoder that outputs a score. We take the maximum score from all concatenations and output it as the relevance score. Finally, we sort the document paths by given score and take the maximum number of paths possible, ensuring that the total number of documents does not exceed $\mathcal{K} = 20$.

4.3 Sentence Retrieval

The sentence retrieval step of our pipeline uses a combination of the current state-of-the-art model, $LF_{\text{2-iter}} + D_{XL}$ (Stammbach, 2021), and a simple DeBERTav3-based cross-encoder combining all possible sentences from first and second order context windows for every path. For the input of both models, we use the document path collection outputted from our document retrieval pipeline. Originally, the $LF_{\text{2-iter}} + D_{XL}$ model uses UKP's (Hanselowski et al., 2018) document retrieval step combined with TF-IDF and a (query, sentence) pair evaluation based on a token-level BigBird (Zaheer et al., 2020) model for the sentence retrieval step. Particularly, the $LF_{\text{2-iter}} + D_{XL}$ sentence retrieval architecture works in two stages. On the first one, the query and all the sentences from first-order documents are evaluated and given a score. Every pair given a score greater than 0 is expanded. Finally, every expanded sentence is evaluated conditioned not only on the query but also on the first-order

[6]Preliminary tests showed that BART offered the best results among several LMs.

sentence from which it comes from (again using the BigBird model).

Although using the documents retrieved from our solution in $LF_{\text{2-iter}} + D_{XL}$ performs reasonably well on its own, we found that combining the sentence rankings from this model with the rankings from our own cross-encoder boosts global performance (see Table 4). Directly combining the rankings from both models is possible since both are based on retrieving connected (first-second order) sentences. Formally, given the definition of a ranking:

$$\widetilde{\mathcal{R}} := \{p_i\}_i$$

$$p_i = \begin{cases} (s_k^i, s_m^j), & \text{if } d_j \text{ expanded from } d_i. \\ (s_k^i) & \text{if } d_i \text{ has no expansion.} \end{cases}$$

We can define the *order* of p_i, a path in ranking $\widetilde{\mathcal{R}}$, as

$$\varphi_{\widetilde{\mathcal{R}}}(p_i) = \begin{cases} i & \text{if } p_i \subset \widetilde{\mathcal{R}} \\ 0 & \text{otherwise} \end{cases}$$

In this context, the combination of the rankings $\widetilde{\mathcal{R}}_{LF}$ and $\widetilde{\mathcal{R}}_{CE}$ corresponding to the $LF_{\text{2-iter}} + D_{XL}$ system and our cross encoder respectively, is defined as:

$$\widetilde{\mathcal{E}} = \underset{A \subset \widetilde{\mathcal{R}}_{LF} \cup \widetilde{\mathcal{R}}_{CE}}{\operatorname{argmax}} \sum_A \varphi_{\widetilde{\mathcal{R}}_{CE}}(p) + \varphi_{\widetilde{\mathcal{R}}_{LF}}(p) \quad \text{s.t.}$$

$$\sum_A |p| \leq 5$$

5 Experimental Evaluation and Results

We present our results for the development dataset at every stage and the FEVER challenge competition (test set) results:

Document retrieval. As previously commented, some of the most recent approaches add the documents retrieved from classic techniques such as TF-IDF and BM25 to the results retrieved from their main document retrieval architectures. In doing so, the retrieved documents lose the ranking order, and it would be inaccurate to directly compare recall@K since the results from these combined systems are not rankings but rather collections of documents. Therefore, we compare our results with the unaltered baseline systems in Table 1 and establish a new state of the art for this stage, surpassing UKP's results by 3.07%. A comparison of our approach's performance varying the number of documents can be found in Table 2. We observe a high and steady MRR metric, which means that in most cases, there is a relevant document within the top 5 documents. Hence, most of the recall errors are likely claims that are not correctly interpreted (i.e., no relevant document in all ranking) or multi-hop evidence cases in which not all the evidence was retrieved. Finally, we perform an ablation study regarding all the modules in the document retrieval step of our solution (see Table 3), from which it can be inferred that the Noun Phrases (box [3] in Figure 1) and the Expansion & Rerank (box [4] in Figure 1) modules are the parts that have a higher impact on performance. Additionally, it is worth noting that the Closest Match module (box [2] in Figure 1) does not have a significant impact on general performance, meaning that although some examples exist, there are not many instances with grammatical errors within the FEVER dev dataset.

System	R@10
UKP (Hanselowski et al., 2018)	93.55
UNC (Nie et al., 2018)	92.82
Ours	**96.62**

Table 1: Comparison of document retrieval system's recall with existing architectures

Nº Docs	Recall	AND	MRR
5	95.54	97.26	0.935
10	96.62	97.96	0.935
15	97.08	98.20	0.935
20	97.20	98.29	0.935

Table 2: Document retrieval metrics of our proposed solution considering different number of documents.

Sentence retrieval. In Table 4, we report the results with and without combining the $LF_{\text{2-iter}} + D_{XL}$ system to our cross-encoder for this stage, along with a performance comparison with the existing architectures. Our proposed solution outperforms the current state of the art by 1.05 %. Note that $LF_{\text{2-iter}} + D_{XL}$ system also surpasses the state of the art when given the documents selected from our document retrieval step. Indicating that our document retrieval strategy potentially improves the effectiveness of a sentence retrieval module.

Textual entailment. In the test set (competition),

Combination	R@20
[1]	94.50
[1] + [2]	94.74
[1] + [3]	95.03
[1] + [2] + [3]	95.27
[1] + [4]	95.30
[1] + [2] + [4]	95.80
[1] + [3] + [4]	97.01
[1] + [2] + [3] + [4]	97.20

Table 3: Ablation study for the proposed system. Note that every component is referred to as an index [x] which is depicted in Figure 1.

	System	R@5	Acc	FS
Development dataset	(Hanselowski et al., 2018)	86.02	68.49	64.74
	(Nie et al., 2018)	86.79	69.72	66.49
	(Subramanian and Lee, 2020)	90.50	75.77	73.44
	(Stammbach and Neumann, 2019)	89.80	72.10	-
	(Zhou et al., 2019)	86.72	74.84	70.69
	(Liu et al., 2020)†	94.37	78.29	76.11
	(Zhong et al., 2020)	90.50	79.16	-
	(Jiang et al., 2021b)	90.54	**81.26**	77.75
	(Krishna et al., 2022)	-	80.74	79.07
	(Stammbach, 2021)	93.62	-	-
	(Chen et al., 2022)	79.61	79.44	77.38
	(Fajcik et al., 2022)	93.30	80.80	78.00
	Ours [1-4]	93.93	80.03	78.36
	Ours [1-5] (Full)	**94.67**	80.95	**79.12**
Test dataset	(Zhou et al., 2019)	-	71.60	67.10
	(Liu et al., 2020)	-	74.07	70.38
	(Zhong et al., 2020)	-	74.64	71.48
	(Jiang et al., 2021b)	-	79.35	75.87
	(Krishna et al., 2022)	-	79.47	76.82
	(Stammbach, 2021)	-	79.16	76.68
	(Chen et al., 2022)	-	75.24	71.17
	(Fajcik et al., 2022)	-	79.27	76.45
	(Izacard et al., 2022)‡	-	**80.06**	21.29
	Ours[1-5] (Full)	-	79.69	**76.91**

Table 4: Performance for the second and third stages in the development and test datasets. † The system uses gold evidence when reporting these results. ‡ The system was not specifically designed for FEVER, trained with the whole Wikipedia for performing fact verification, hence the disparity in Acc and FS.

regarding the Fever Score, our proposal achieves a new state of the art by using our retrieved evidence with the approach followed in $LF_{\text{2-iter}}+D_{XL}$. Additionally, we report the second-highest accuracy score, 79.69%, only surpassed by the Atlas system (Izacard et al., 2022). In the development dataset, we report a competitive 80.95% accuracy while our Fever Score (FS), 79.12%, outperforms the current state of the art.

6 Conclusions

In this paper, we have proposed a retrieval architecture that combined with a textual entailment model outperforms the state of the art in all stages of the FEVER task. Our architecture starts by leveraging document-level semantic representation to narrow an initial collection of documents to 20 candidates. Filtered results are later expanded using the graph structure inherent to the built database. Once expansion is completed, our model scores the context windows inside documents, ranks the link paths, and takes the top elements from the ranking, ensuring that no more than 20 documents are retrieved. Then, the documents are passed on to the sentence retrieval model that combines the prediction of the $LF_{\text{2-iter}}+D_{XL}$ system with a simple cross-encoder to obtain a sentence-paths ranking. Finally, following the approach in $LF_{\text{2-iter}}+D_{XL}$, a pre-trained DeBERTa-based MNLI model is used and later post-processed based on the output logits.

Regarding our initial research hypotheses; considering the results obtained in the ablation study (see Table 3) and the sentence retrieval steps (see Table 1, Table 4) we can conclude that: i) We can use semantic encoding as an alternative to keyword matching to build a retrieval system independent of external resources (**H1**); ii) Expanding and reranking connected paths of information using small context windows inside documents improve retrieval quality (**H2**).

Limitations

The main limitation of our model concerns the expansion operation in the retrieval steps. In particular the system assumes a constant maximum length of two hops. This decision leads to some recall errors, however, in the FEVER development dataset, more than 99% of the evidence can be retrieved with at most two sentences. Another limitation of our model is relying on a cascade-based architecture i.e., the performance of one step is always bounded by the performance on the previous step. Additionally, although not directly dependent on

external resources, we expect a graph structure between documents for the model to work and this could prove complicated to manage depending on environments different than Wikipedia.

Ethics Statement

The presented work could help to more accurately extract information to verify statements. However, the system relies on contrasting facts using a "truth" database. The existence of such a resource is not a trivial assumption to make, especially if we consider open sources of information such as social networks in which virtually anyone can add content. Consequently, and in addition to the fact that no system is perfect, we discourage the usage of our work as any kind of ground truth for any fact verification task if the reference database cannot be checked by experts both in terms of accuracy and possible biases.

Acknowledgements

We sincerely thank the anonymous reviewers for their thorough reviewing and valuable suggestions. The research of Guzman-Olivares and Quijano-Sanchez was conducted with financial support from the Spanish Ministry of Science and Innovation, grant PID2019-108965GB-I00. The research of Liberatore was partially funded by the grant PID2019-108679RB-I00 of the Spanish Ministry of Science and Innovation.

References

Alan Akbik, Tanja Bergmann, Duncan Blythe, Kashif Rasul, Stefan Schweter, and Roland Vollgraf. 2019. FLAIR: An easy-to-use framework for state-of-the-art NLP. In *NAACL 2019, 2019 Annual Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 54–59.

Zapan Barua, Sajib Barua, Salma Aktar, Najma Kabir, and Mingze Li. 2020. Effects of misinformation on covid-19 individual responses and recommendations for resilience of disastrous consequences of misinformation. *Progress in Disaster Science*, 8:100119.

Jiangui Chen, Ruqing Zhang, Jiafeng Guo, Yixing Fan, and Xueqi Cheng. 2022. Gere: Generative evidence retrieval for fact verification. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '22, page 2184–2189, New York, NY, USA. Association for Computing Machinery.

Michiel de Jong, Yury Zemlyanskiy, Nicholas FitzGerald, Fei Sha, and William W. Cohen. 2022. Mention memory: incorporating textual knowledge into transformers through entity mention attention. In *International Conference on Learning Representations*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Yibing Du, Antoine Bosselut, and Christopher D. Manning. 2022. Synthetic disinformation attacks on automated fact verification systems. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(10):10581–10589.

Martin Fajcik, Petr Motlécek, and Pavel Smrz. 2022. Claim-dissector: An interpretable fact-checking system with joint re-ranking and veracity prediction. *CoRR*, abs/2207.14116v1.

Michael Glass, Gaetano Rossiello, Md Faisal Mahbub Chowdhury, Ankita Naik, Pengshan Cai, and Alfio Gliozzo. 2022. Re2G: Retrieve, rerank, generate. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2701–2715, Seattle, United States. Association for Computational Linguistics.

Andreas Hanselowski, Hao Zhang, Zile Li, Daniil Sorokin, Benjamin Schiller, Claudia Schulz, and Iryna Gurevych. 2018. UKP-athene: Multi-sentence textual entailment for claim verification. In *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*, pages 103–108, Brussels, Belgium. Association for Computational Linguistics.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2021. Debertav3: Improving deberta using electra-style pretraining with gradient-disentangled embedding sharing.

Sebastian Hofstätter, Sheng-Chieh Lin, Jheng-Hong Yang, Jimmy Lin, and Allan Hanbury. 2021. Efficiently teaching an effective dense retriever with balanced topic aware sampling. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '21, page 113–122, New York, NY, USA. Association for Computing Machinery.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spaCy: Industrial-strength Natural Language Processing in Python.

Gautier Izacard, Patrick Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and Edouard

Grave. 2022. Few-shot learning with retrieval augmented language models.

Haoming Jiang, Danqing Zhang, Tianyu Cao, Bing Yin, and Tuo Zhao. 2021a. Named entity recognition with small strongly labeled and large weakly labeled data. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1775–1789, Online. Association for Computational Linguistics.

Kelvin Jiang, Ronak Pradeep, and Jimmy Lin. 2021b. Exploring listwise evidence reasoning with t5 for fact verification. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 402–410, Online. Association for Computational Linguistics.

Amrith Krishna, Sebastian Riedel, and Andreas Vlachos. 2022. ProoFVer: Natural logic theorem proving for fact verification. *Transactions of the Association for Computational Linguistics*, 10:1013–1030.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. ALBERT: A lite BERT for self-supervised learning of language representations. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020a. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Patrick S. H. Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020b. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Jing Li, Aixin Sun, Ray Han, and Chenliang Li. 2020. A survey on deep learning for named entity recognition. *IEEE Transactions on Knowledge and Data Engineering*, PP:1–1.

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

Zhenghao Liu, Chenyan Xiong, Maosong Sun, and Zhiyuan Liu. 2020. Fine-grained fact verification with kernel graph attention network. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7342–7351, Online. Association for Computational Linguistics.

Christopher Malon. 2018. Team papelo: Transformer networks at FEVER. In *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*, pages 109–113, Brussels, Belgium. Association for Computational Linguistics.

Yixin Nie, Haonan Chen, and Mohit Bansal. 2018. Combining fact extraction and verification with neural semantic matching networks.

Femi Olan, Uchitha Jayawickrama, Emmanuel Ogiemwonyi Arakpogun, Jana Suklan, and Shaofeng Liu. 2022. Fake news on social media: the impact on society. *Information Systems Frontiers*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Juan Ramos. 2003. Using tf-idf to determine word relevance in document queries.

Lance Ramshaw and Mitch Marcus. 1995. Text chunking using transformation-based learning. In *Third Workshop on Very Large Corpora*.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Stephen Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends in Information Retrieval*, 3:333–389.

Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. 2009. The graph neural network model. *IEEE Transactions on Neural Networks*, 20(1):61–80.

Amir Soleimani, Christof Monz, and Marcel Worring. 2020. Bert for evidence retrieval and claim verification. In *Advances in Information Retrieval*, pages 359–366, Cham. Springer International Publishing.

Dominik Stammbach. 2021. Evidence selection as a token-level prediction task. In *Proceedings of the Fourth Workshop on Fact Extraction and VERification (FEVER)*, pages 14–20, Dominican Republic. Association for Computational Linguistics.

Dominik Stammbach and Guenter Neumann. 2019. Team DOMLIN: Exploiting evidence enhancement for the FEVER shared task. In *Proceedings of the Second Workshop on Fact Extraction and VERification (FEVER)*, pages 105–109, Hong Kong, China. Association for Computational Linguistics.

Gabriel Stanovsky, Julian Michael, Luke Zettlemoyer, and I. Dagan. 2018. Supervised open information extraction. In *NAACL-HLT*.

Harald Stiff and Fredrik Johansson. 2022. Detecting computer-generated disinformation. *International Journal of Data Science and Analytics*, 13(4):363–383.

Shyam Subramanian and Kyumin Lee. 2020. Hierarchical Evidence Set Modeling for automated fact extraction and verification. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7798–7809, Online. Association for Computational Linguistics.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018a. 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)*, pages 809–819, New Orleans, Louisiana. Association for Computational Linguistics.

James Thorne, Andreas Vlachos, Oana Cocarascu, Christos Christodoulopoulos, and Arpit Mittal. 2018b. The fact extraction and VERification (FEVER) shared task. In *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*, pages 1–9, Brussels, Belgium. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

Yu Xia, Quan Wang, Yajuan Lyu, Yong Zhu, Wenhao Wu, Sujian Li, and Dai Dai. 2022. Learn and review: Enhancing continual named entity recognition via reviewing synthetic samples. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2291–2300, Dublin, Ireland. Association for Computational Linguistics.

Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, and Amr Ahmed. 2020. Big bird: Transformers for longer sequences. In *Advances in Neural Information Processing Systems*, volume 33, pages 17283–17297. Curran Associates, Inc.

Wanjun Zhong, Jingjing Xu, Duyu Tang, Zenan Xu, Nan Duan, Ming Zhou, Jiahai Wang, and Jian Yin. 2020. Reasoning over semantic-level graph for fact checking. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6170–6180, Online. Association for Computational Linguistics.

Jie Zhou, Xu Han, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. 2019. GEAR: Graph-based evidence aggregating and reasoning for fact verification. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 892–901, Florence, Italy. Association for Computational Linguistics.

A Synthetic NER Dataset

The synthetic dataset for the NER problem has been built as follows: i) Given a claim, separate it by words and extract the associated pages from the gold evidence; ii) Use edit distance at token level to perform keyword matching with the previously separated words; iii) Discard the matchings having an edit distance smaller than a threshold (we used .4); iv) Use a BERT-based tokenizer to separate the sentence. For each matched sequence, label the first belonging token as *B* (begin) and every other as *I* (intermediary); v) Any token that is not either *I* or *B* is labeled as *O* (Null).

B Synthetic Rerank Dataset

The rank dataset has been built as follows: i) Divide the claims into two groups regarding the number of evidence pieces (one or two) needed for the veracity judgment to be valid; ii) Balance the groups by under-sampling the group with only one piece of evidence needed; iii) Join the groups and randomly create sequences of context windows from first and second-order documents; iv) Give these sequences a score according to the information they present regarding information completeness: 0 for unrelated content, 0.5 for related but incomplete (second-order case in which only one of the context windows is correct), and 1 for complete evidence.