information

MDPI

Article

GRAAL: Graph-Based Retrieval for Collecting Related Passages across Multiple Documents [†]

Misael Mongiovì [1,2,*,‡] and Aldo Gangemi [1,2,3]

[1] Institute of Cognitive Science and Technology, National Research Council of Italy, 00196 Rome, Italy; aldo.gangemi@cnr.it
[2] Department of Mathematics and Computer Science, University of Catania, 95125 Catania, Italy
[3] Department of Philosophy and Communication, University of Bologna, 40126 Bologna, Italy
[*] Correspondence: misael.mongiovi@unict.it

[†] This article is a revised and expanded version of a paper entitled "Graph-based Retrieval for Claim Verification over Cross-document Evidence", which was presented at "The 10th International Conference on Complex Networks and their Applications", 30 November–2 December 2021, Madrid, Spain.

[‡] Current address: Dipartimento di Matematica e Informatica, Università degli Studi di Catania, Viale Andrea Doria, 6-95125 Catania, Italy.

Abstract: Finding passages related to a sentence over a large collection of text documents is a fundamental task for claim verification and open-domain question answering. For instance, a common approach for verifying a claim is to extract short snippets of relevant text from a collection of reference documents and provide them as input to a natural language inference machine that determines whether the claim can be deduced or refuted. Available approaches struggle when several pieces of evidence from different documents need to be combined to make an inference, as individual documents often have a low relevance with the input and are therefore excluded. We propose GRAAL (GRAph-based retrievAL), a novel graph-based approach that outlines the relevant evidence as a subgraph of a large graph that summarizes the whole corpus. We assess the validity of this approach by building a large graph that represents co-occurring entity mentions on a corpus of Wikipedia pages and using this graph to identify candidate text relevant to a claim across multiple pages. Our experiments on a subset of FEVER, a popular benchmark, show that the proposed approach is effective in identifying short passages related to a claim from multiple documents.

Keywords: passage retrieval; claim verification; knowledge extraction

check for updates

Citation: Mongiovì, M.; Gangemi, A. GRAAL: Graph-Based Retrieval for Collecting Related Passages across Multiple Documents. *Information* **2024**, *15*, 318. https://doi.org/10.3390/info15060318

Academic Editor: María N. Moreno García

Received: 31 January 2024
Revised: 25 May 2024
Accepted: 27 May 2024
Published: 29 May 2024

1. Introduction

Passage retrieval [1] is a fundamental task in *claim verification* [2] and *open-domain question answering* [3] that consists of identifying passages that are related to a given sentence, question, or claim in a large corpus. Although large modern language models can embed a large amount of knowledge in their parameters, and hence do not need to access an external explicit knowledge base [4], retrieval-based approaches are preferable when a human-understandable explanation of results is desirable and when the knowledge base needs to be revised or expanded [5]. These approaches rely on two main steps. A passage retrieval task extracts a series of passages related to the input from the corpus. Then, the extracted passages are given to a language model together with the input to generate the output. Modern language models, such as *Transformers* [6], are effective in reasoning with short text passages, but the size of the input that they can handle is limited and their performance decreases as the size of the input increases. Therefore, the effective retrieval of a small set of related phrases is critical for satisfactory performances.

In this paper, we focus on claim verification since this is a common task that requires reasoning over a large knowledge base. However, we believe that a similar approach can be helpful in other natural language processing (NLP) tasks such as open-domain question answering. With *claim verification* [2], we refer to the task of determining whether a claim is supported or refuted (or neither) by a reference knowledge base. It is a fundamental task in automated fact checking [7], with significant implications in the critical problem of countering disinformation, which has a significant worldwide impact [8]. In addition to automatic fact checking, this task is relevant in any situation where it is necessary to check the consistency of statements against a knowledge base. For example, keeping an updated knowledge base when new information is available requires checking the consistency of the new data with previous knowledge. Integrating a supported statement would introduce redundancy, while a contradicted statement would make the knowledge base inconsistent. This is especially important in robotics, in a scenario where knowledge is kept updated, e.g., through conversation with humans. New statements supported by previous knowledge might increase confidence in codified facts while refuted statements could indicate unreliability of the interlocutor or misunderstanding, or they might suggest that the internal knowledge needs to be rectified.

A general framework for claim verification over a reference corpus has been proposed by Thorne et al. [2]. Given a claim, first, a *document retrieval* component retrieves a series of related documents. Then, a restricted set of sentences related to the claim (the *evidence*) is selected from the retrieved documents (*sentence selection* step). Eventually, the proper *claim verification* is performed by means of a classifier that determines whether the claim or its denial can be inferred from the selected set of sentences and ranks the claim into one of the labels *supported*, *refuted*, or *not enough info*. The recently proposed claim verification tools are mainly based on this framework [9–16]. Other recent work focuses explicitly on document retrieval claim verification and open-domain question answering [5,17].

The approaches described above achieve significant performances on average. However, they struggle when multiple pieces of evidence, some of which having little relevance to the claim, are spread across multiple documents. For instance, the statement "The Beatles were formed in England" can be inferred by the following two sentences: "The Beatles were formed in Liverpool" and "Liverpool is a city and metropolitan borough in Merseyside, England". Considering Wikipedia as the reference corpus, such phrases appear in separate documents, namely the page on "The Beatles" and the page on "Liverpool". However, the relevance of "Liverpool" would be considered negligible by a document retrieval tool as the city is not mentioned in the statement. Without other knowledge, its relevance would not be superior to any other city in England. In principle, the "Liverpool" page can be identified by implicit or explicit background knowledge; however, such knowledge is often unavailable or it might be considered unreliable. Although recent studies [5] consider the distribution of related passages across documents, they do not provide a specific solution to the above problem and thus their performances in retrieving fragmented evidence are limited. For completeness, note that the example above might be solved by a disjunctive query submitted against index-based structures [18–20]. However, in general, concepts can be expressed by multiple verbal forms and hence term-based approaches would not work. Moreover, they cannot determine which entities are in semantic relation with each other, therefore lacking focused targeting of relevant text. On the other hand, using a similar approach that involves semantics to solve the example would require the whole knowledge to be formally represented in a complete graph and the use of a formal query language, which lacks the flexibility and expressivity of natural language. In contrast, our approach capitalizes on recent advances in machine learning models and is able to make use of meaningful knowledge extracted from large corpora.

In a previous conference paper [21], we proposed a graph-based approach for retrieving fragmented evidence. We summarized the reference corpus into a large graph of mentioned *entities*, interconnected by *co-mentions*, i.e., their use in the same sentence. The idea is that exploring this graph can help to identify relevant concepts and corresponding text passages. In the example above, the path through the mentions "The Beatles", "Liverpool", and "England" outlines the evidence for the claim. In this paper, we make a step further by considering co-mentions at the level of *frames*, i.e., utterances that express events or actions, in place of sentences. By considering frames, extracted by means of *Semantic Role Labeling*, we are able to distinguish co-occurrences of entities that participate in the same event or action from unrelated entities in the same sentence. This reduces the connectivity between unrelated entities and, in the end, reduces the amount of unnecessary text retrieved.

With respect to our previous work [21], our contribution can be summarized as follows:

- We propose a novel fine-grained graph-based method for retrieving passages relevant to a claim or question. Our method considers the associations among mentioned entities within a sentence at a finer level compared to previous approaches by examining their semantic relationships.
- We perform extensive experimental analysis and demonstrate that our approach can retrieve a significantly more compact set of passages while still maintaining good performance in accurately targeting the relevant text.
- We perform a qualitative analysis of the results and discuss some real examples, providing useful hints on the impact of our fine-grained graph structure on performance.

In the remainder, we first present some background notions (Section 3). We then describe our graph-based approach for retrieving evidence across documents (Section 4) and report the results of our experimental analysis (Section 5). We discuss related work (Section 2) and eventually conclude the paper and outline future work (Section 6).

2. Related Work

Passage retrieval is a fundamental step for many modern systems of open-domain question answering and claim verification. Recent approaches [1,5,22,23] aim at selecting content relevant to a sentence by means of neural architecture. They employ two encoders for embedding the documents and the query (e.g., a claim) into the same space and perform a cosine similarity search to retrieve candidate documents. Eventually, the search is refined by a cross-encoder classifier that combines each candidate document with the query and decides if it is relevant. The search can be performed at a finer level of granularity by considering short passages in place of complete documents. Although the described retrieval approaches have been proved successful in solving NLP tasks, including claim verification [5,23], they suffer when the evidence is fragmented across several documents, each of them loosely related to the claim. Our approach aims at overcoming this limit by interconnecting sentences of the reference corpus and providing a method for spotting all fragments of candidate evidence at once.

More specific approaches for claim verification, a fundamental step in fact checking, are reviewed in [24]. Recent methods can rely on large annotated datasets to train machine learning models and achieve considerable results. Thorne et al. [2] provided FEVER, the first large-scale dataset with evidence for claim verification over a reference corpus consisting of 185, 445 claims classified as *supported*, *refuted*, or *not enough info* and associated to evidence from a corpus of 5.4 million Wikipedia pages. They described a pipeline that comprises information retrieval and textual entailment components. Recently proposed claim verification systems are mainly based on such a framework [9–16], where a retrieval component extracts sentences related to the claim from the corpus (the evidence) and a textual entailment component classifies the claim based on the retrieved evidence. The retrieval component is usually decomposed into two sub-components: document retrieval, which identifies related documents, and sentence selection, which extracts salient sentences from the retrieved documents. The large size of FEVER enables training machine learning models for the task and obtaining performances that overcome 70% overall accuracy.

The document retrieval step is often shared among different works. A commonly used technique consists of retrieving a set of documents by keyword matching with the document titles [10,13] or calling the MediaWikiAPI (https://www.mediawiki.org/wiki/API, accessed on 7 May 2024) of noun phrases from the claim [9,12,15,16]. Some methods also filter retrieved documents by a classifier based on NSMN [10,13,14], a variant of ESIM [25], a deep learning architecture based on two bidirectional LSTM (long short-term memory) architectures. The claim and its noun phrases are compared with titles of previously retrieved documents to decide its relevance and filter out irrelevant documents. The sequence retrieval step is usually performed by a classifier that decides for every sentence of the retrieved documents whether it is related or not to the claim. Some systems employ ESIM [9,16], NSMN [13], or logistic regression [11] for this step. More recent systems employ transformers, the last-generation language models, such as BERT [12,15] and XLNet [10].

A limit of the described approaches concerns the document retrieval phase. It gives no guarantee that available evidence for or against a claim is retrieved, since such evidence might be contained in documents whose titles might be loosely related or even not related at all to the claim.

With the purpose of finding a solution for the evidence collection across different documents, in a previous conference work, we proposed GraphRetrieve [21], which builds a summarization graph of the corpus for aiding in the evidence collection. In this paper, we extend the work and improve the system by considering a finer-grain concept for defining co-occurrences based on semantic role labeling. Other recent approaches explore the use of graphs for collecting relevant evidence. FarFetched [26] builds a heterogeneous graph that summarizes entities, sections, and articles and looks for paths that connect entities by alternating entity nodes with section nodes. This approach captures the co-occurrence of entity mentions in the same section even if they are unrelated. Our approach encompasses a higher degree of granularity by considering co-mentions only if they belong to the same frame, thus avoiding connecting unrelated entities that happen to be in the same section and significantly reducing the set of candidate evidence passages. Kallipolitis et al. [27] consider a graph that interconnects entities such as patients, encounters, observations, and immunizations with the goal of predicting the risk of a patient's fatality. Giarelis et al. propose employing a "graph-of-docs" model to represent documents and their words to enhance text categorization [28] and feature selection [29]. Jalil et al. [30] employ word graphs to improve text summarization.

3. Background

We consider the task of retrieving sentences *related* to a given statement (or *claim*) c from a corpus of text documents $\mathcal{D}$ (namely the *passage retrieval* task). The definition of "related" is problem-dependent and refers to all sentences that are necessary to formulate a correct answer. In claim verification, our goal is to retrieve a minimal set of sentences that together entail the input claim. Formally, given a corpus $\mathcal{D} = \{D_1, D_2, \ldots, D_n\}$, where D_i (documents) are ordered sets of sequences, and a claim c, we want to identify a set of sentences $S = \{s \in D_1 \cup D_2 \cup \ldots \cup D_n$ with $D_i \in \mathcal{D}\}$ such that S entails c and $|S|$ (the number of sentences in S) is minimum.

In this work, we make use of *Semantic Role Labeling (SRL)*, *Named Entity Resolution (NER)*, and *Entity Linking (EL)* for our passage retrieval solution. SRL [31,32] is the task of identifying utterances (namely *frames*), i.e., predicates that express events or actions, identifying their constituents (namely *arguments*) and inferring the most appropriate relation (namely *semantic role*) between each predicate–argument pair. The set of frames identified by SRL is a finer-grained and more informative representation of text documents with respect to sentences, since multiple frames typically occur in a sentence and they reveal semantic information, i.e., the type of frame and the semantic roles that relate arguments to the predicate. We employ the machine-learning-based SRL tool from Shi and Lin [33], one of the most popular and best-performing tools on out-of-domain corpora.

Entity Linking (EL) [34], referred to also as named entity disambiguation (NED), is the task of linking parts of a text document (mentions) that represent an entity to an external knowledge base. In general, the term entity refers to any object or concept that can be uniquely identified. We consider the set of English Wikipedia pages as our entity library and restrict our focus to entities that have a corresponding Wikipedia page since this allows us to employ off-the-shelf tools freely available (e.g., BLINK [34]). We follow the common practice of performing EL downstream of named entity resolution (NER) [35], i.e., the task of identifying named entities (real-world objects that can be denoted by a proper name, e.g., people, organizations, places, products) in text.

In our graph-based approach, we consider *undirected multi-graphs*, i.e., graphs where the direction of edges is not defined and more than one edge can occur between two vertices. An undirected multi-graph (in the following, simply a graph) $G = (V, E)$ is composed of a set V of *vertices* and a set E of *edges*, where an edge $e = (u, v, p)$ refers to two *endpoints* $u, v \in V$ and a *property* (or label) p. Since the graph is undirected, each edge occurs in both directions ($(u, v, p) \Leftrightarrow (v, u, p)$). Given a graph G, a subgraph of G is a graph $G_s = (V_s, E_s)$ that contains a subset of vertices and edges of G, i.e., $V_s \subseteq V$ and $E_s \subseteq E$.

4. Method

We propose to transpose the problem of identifying related passages to the graph domain by constructing a graph-based summarization of both the corpus and the input statement and finding associations between nodes and edges across such graphs. We first build a large graph G offline, namely the *corpus graph*, which summarizes the entire corpus. To find passages related to a given claim c, we first find all mentions of entities in c (which we call M_c) and then probe G for finding a subgraph G_c that outlines the relevant knowledge according to M_c. Next, we detail the construction of the corpus graph, and then we describe the search process.

4.1. Corpus Graph Construction

The corpus graph $G = (V, E)$ is an undirected multi-graph where vertices are mentioned entities and edges represent co-mentions within the same SRL frame. The construction of this graph is summarized in Algorithm 1.

Algorithm 1: Corpus graph construction

```
Result: Return the graph $G$ that summarizes the corpus $\mathcal{D}$
Compute entity linking over $\mathcal{D}$
    $V \leftarrow$ all entities mentioned in $\mathcal{D}$;
    $M \leftarrow$ all mentions in $\mathcal{D}$
Compute SRL over $\mathcal{D}$
    $F \leftarrow$ all frames in $\mathcal{D}$
for every $f \in F$ do
    for every $(m_1, m_2) \in M \times M$ such that their text span are contained in the span of $f$
     and $entity(m_1) \neq entity(m_2)$ do
        $E \leftarrow E \cup \{(entity(m_1), entity(m_2), f)\}$
    end
end
return $G = (V, E)$
```

First, entity linking and SRL are computed on the whole corpus. Entity linking identifies text spans (mentions) that correspond to entities and links them to a unique identifier that represents the entity. For instance, in the sentence "The Beatles were formed in Liverpool", the entity linker identifies two mentions m_1 = "The Beatles" and m_2 = "Liverpool" and connects them to the Wikipedia unique identifiers of the corresponding pages, namely $entity(m_1)$ and $entity(m_2)$. Note that an entity can have multiple verbal forms and the same text can correspond to multiple entities; it is the responsibility of the entity linker to connect the text span to the correct entity given the context. SRL extracts frames from the documents, as described in Section 3. We consider a frame f as the portion of text corresponding to both the predicate and the arguments of the frame as identified by the SRL tool.

After the entity linker and the SRL tool have been run, for each frame f identified by SRL and for every pair of distinct mentions m_1, m_2 in the text span of frame f, we add to G

an edge $(entity(m_1), entity(m_2), f)$ between their corresponding entities. Note that we use the same notation for referring to both vertices of G and entities. Generally, any concept that can be uniquely identified qualifies as a vertex in G.

In our work, we employ BLINK [34], an off-the-shelf entity-linking tool that links text spans that represent entity mentions to corresponding entries, identified by the URI of the corresponding Wikipedia page. To identify frames, we employ the popular SRL tool from Shi and Lin [33]. We consider two entities as occurring within the same frame f if their text span begins are included in the text span of f. Each frame f is associated to the specific sentence in the corpus $sentence(f)$ where it is evoked. Note that the set of sentences associated to a pair of nodes (u, v) is different from the one in [21] since the former includes only the cases where the two entities are mentioned in the same frame.

4.2. Graph-Based Passage Retrieval

Given a claim c, we first execute BLINK [34] to extract all entity mentions from c. We refer to this set of mentions as M_c. The underlying idea is to consider a suitable subgraph G_c of G that interconnects all entities referred in M_c and retrieve the associated sentences. An example is shown in Figure 1. In the middle, we show a fragment of G, where vertices are entities and edges connect entities that are mentioned in the same frames (in the bottom). The sentence at the top is our claim, which mentions the entities "The Beatles" and "England". The corresponding evidence can be reconstructed by retrieving the frames associated to edges in the path that connects the vertices "The Beatles" and "England", passing through the vertex "Liverpool".

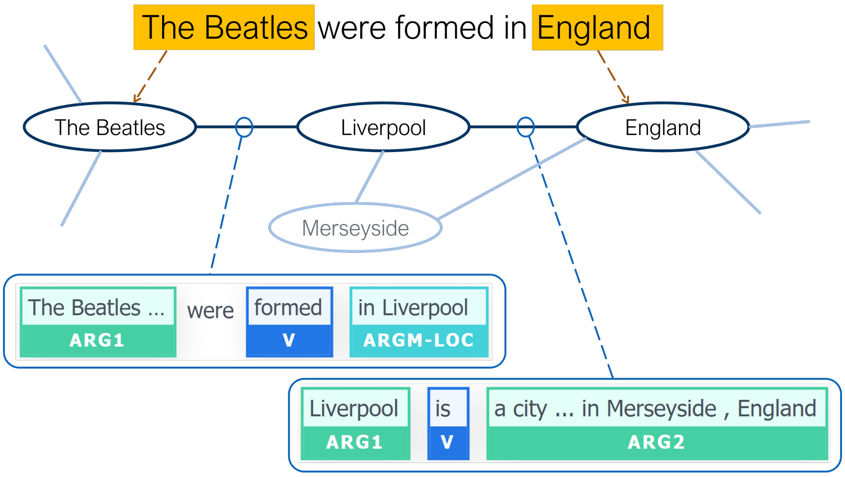

Figure 1. Evidence for a claim (sentence at the top) is outlined by a subgraph of G (graph in the middle). Edges of G represent frames (in the bottom) that mention both endpoints, which serve as candidate evidence for the claim.

Following a parsimony criterion, we try to keep the size of the spanning subgraph small. One option is to take the subgraph that has the fewest edges. Finding this graph corresponds to solving the *minimum Steiner tree*, an optimization problem known to be NP-hard [36]. However, this solution would be expensive and still without guarantees. Indeed, there are potentially many different minimal solutions and even non-minimal variants might sometimes be better fits. A different possibility is to enumerate all the subgraphs that satisfy the connectivity constraints, but this solution would still take exponential time. Our approach combines all paths between mentioned entities below a prefixed length and considers the resulting subgraph G_c as outlining the evidence for claim c.

Details are described in Algorithm 2. To further simplify the method, we limit the length of paths to 2.

```
Algorithm 2: Search candidate evidence employing the corpus graph $G = (V, E)$
    Result: Return a small set of sentences $S$ that entails claim $c$
    Compute entity linking on $c$ and put all mentions in $M_c$
    $V_c^1 \leftarrow \{entity(m) : m \in M_c\}$
    $V_c^2 \leftarrow \{v \in V : (v, u), (v, w) \in E,$ for some $u, w \in V_c^1, u \neq w\}$
    $E_c \leftarrow \{(u, v, f) \in E : u, v \in V_c^1 \cup V_c^2\}$
    $G_c \leftarrow (V_c^1 \cup V_c^2, E_c)$
    $S = \bigcup_{(u,v,f) \in E_c} sentence(f)$
    $S = S \cup \{$all sentences in pages corresponding to some $v \in V_c^1\}$
    return $S$
```

We consider three types of nodes: *mentioned vertices* (V_c^1), i.e., vertices of G that represent entities mentioned in M_c, *between vertices* (V_c^2), i.e., vertices that are connected to at least two mentioned vertices, and all the remaining vertices, namely *unrelated vertices*. The subgraph G_c of G, obtained by taking mentioned vertices, between vertices and all incident edges, outlines the candidate evidence for c. The set of sentences that represent this evidence, S, is constructed by collecting all sentences associated to edges in G_c. To increase recall, we add to S all sentences of pages that correspond to entities mentioned in the claim.

5. Experimental Analysis

We implemented the proposed tool in Python 3.7 and employed the authors' implementation of BLINK (https://github.com/facebookresearch/BLINK/, accessed on 7 May 2024) for entity linking and the AllenAI (https://github.com/allenai/allennlp, accessed on 7 May 2024) implementation (allennlp 2.1.0) of the Shi and Lee SRL tool for detecting frames. We employed BerkeleyDB 5.3. (https://github.com/berkeleydb/libdb, accessed on 7 May 2024), an efficient key-value database, to store the corpus graph. We considered the graph of each Wikipedia page separately and stored it by considering the page ID as the key. To map entity mentions with corresponding graph vertices, we also built an inverted index that maps entities to containing pages. When a claim is given, all pages that contain entities mentioned in the claim are considered, and their corresponding graphs are retrieved, cleaned of unrelated parts, and merged.

We evaluate GRAAL on a subset of statements from the FEVER [2] dataset and its associated reference corpus of Wikipedia abstracts. FEVER contains 185,000 manually annotated statements with information on whether they are supported or disproved (or none of them) by a reference corpus of 5.4 million Wikipedia pages. For each supported or disproved claim, FEVER provides all lines of evidence consisting of all possible combination of sentences supporting or disproving the claim. To assess the ability of collecting cross-document evidence, we select claims whose evidence is distributed across different pages. We also discard statements with fewer than two unique entities and statements with overly general entities (above 1000 mentions in the corpus), resulting in a set of 2580 statements.

We compare GRAAL with *GraphRetrieve* [21] and a baseline, namely *Entity + Mention*, which collects all sentences that mention at least one entity in the claim plus all sentences of Wikipedia pages that correspond to disambiguated entities in the claim. We also consider FarFetched [26], which explores a heterogeneous graph with sections and entities. This approach considers sections in paths that connect entities mentioned in the claim as evidence. The evidence constructor of FarFetched produces identical results to GraphRetrieve when a section corresponds to a sentence (as in their experiments) and when the maximum path length is appropriately set (we consider the maximum path length as $4 * (n - 1)$ to allow for the inclusion of entities not explicitly mentioned in the claim as we do). Considering the equivalence to GraphRetrieve, we do not report FarFetched in the tables. With respect to our previous implementation [21], we increased the effectiveness of entity linking for both GraphRetrieve and Entity + Mention by a more effective management of long sentences. For this reason, the results are slightly different to the ones reported in [21]. The experiments were performed on a machine with 16 CPUs, 16 GB of RAM, and a GPU NVIDIA Quadro P2200.

We first built the corpus graph as described in Section 4.1. This task was expensive, primarily due to entity linking, which required several days of computation. Despite the cost, it is important to note that the evidence is largely static, allowing this task to be performed just once offline. Even if updates to the evidence are necessary, the dataset can be updated incrementally and only occasionally. Additionally, the code has not been optimized for efficiency, which presents an opportunity for improvement. We obtained a graph with 5.4M vertices and 81M edges. We also computed a similar graph for GraphRetrieve, where edges are sentences in place of frames, resulting in 74M edges. The GRAAL graph is slightly larger (about 8%) because of frame overlap, which may occur when the argument of a frame is itself a frame.

Table 1 reports the performance of GRAAL, GraphRetrieve, and the baseline Entity + Mention in terms of number of sentences retrieved, number of documents (Wikipedia pages) covered, hit rate, and overall performance. Considering the size of returned data, on average, GRAAL significantly outperforms the competitors by almost three times (one-third of sentences retrieved) with respect to the baseline and 10% with respect to GraphRetrieve. Note that GRAAL can further reduce the data size by considering sub-sentences corresponding to frames. However, in our experiments, we did not consider this option since we do not have a ground truth at a finer granularity than sentences and therefore we would not be able to compute the hit rate. The hit rate is computed as the percentage of retrieval successes, i.e., the percentage of claims for which all sentences of at least one line of evidence have been retrieved. GRAAL achieves a slightly lower hit rate than its competitors because it uses fewer candidate evidence pieces, slightly reducing the chance of a hit. However, it makes the downstream verification task easier by producing less, but more targeted, candidate evidence. To balance conciseness, indicated by a small number of retrieved sentences, with the hit rate, we evaluated overall performance using the harmonic average. This average is calculated between the reciprocal of the average number of sentences (multiplied by 100 for scaling) and the hit rate. Overall, GRAAL outperforms GraphRetrieve by three percentage points and also significantly outperforms the baseline, Entity + Mention.

Table 1. GRAAL achieves the smallest amount of candidate sentences covering the smallest amount of documents compared to GraphRetrieve and the Entity + Mention baseline. It returns a little more than one-third of retrieved sentences with respect to Entity + Mention and 10% fewer sentences with respect to GraphRetrieve. The lower hit rate of GRAAL is expected since the amount of candidate evidence is strongly reduced. The overall performance of GRAAL, which balances conciseness and hit rate, is the highest.

Method	Avg. #Sentences	Avg. #Documents	Hit Rate	Overall
Entity + Mention	341.2	257.4	**78.9%**	43%
GraphRetrieve	129.9	92.4	70.9%	74%
GRAAL	**116.3**	**81.1**	70.2%	**77%**

To better explain the balance between returned data size and hit rate, we include a scatter plot that shows the hit rate and the average number of sentences retrieved by the three methods (Figure 2). We also split Entity + Mention into its sub-parts, namely Entity (all sentences of Wikipedia pages corresponding to disambiguated entities) and Mentions (all sentences that mention at least one entity in the claim). The best trade-off is given by the proximity to the bottom-right corner, as indicated by the arrow. GRAAL significantly outperforms GraphRetrieve in terms of data size (closer to the bottom) with comparable performances in terms of hit rate. Entity + Mention is the worst in terms of data size (near the top), while Entity (in the bottom) achieves the smallest hit rate (about 40%).

Table 2 reports the running time of the three methods. The average running time for each claim is dominated by the entity-linking task (on average, 4.22 s per claim, not shown).

GRAAL is only 3% slower than GraphRetrieve and 39% slower than Entity + Mention. Note that the implementation is not optimized for efficiency; therefore, the gap can be reduced by a meticulous implementation.

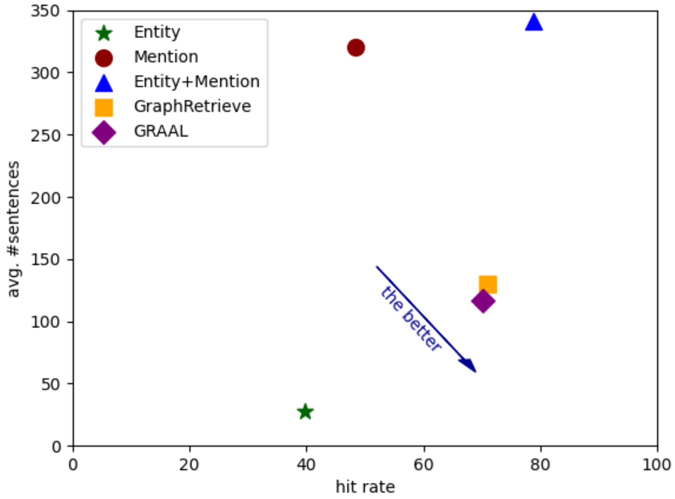

Figure 2. GRAAL achieves an adequate trade-off between hit rate and returned data size (average number of sentences).

Table 2. Running time of GRAAL, GraphRetrieve, and Entity + Mention.

Method	**Avg. Time (s)**
Entity + Mention	**4.25**
GraphRetrieve	5.71
GRAAL	5.90

In Table 3, we report three examples of claims whose cross-document evidence has been correctly retrieved by GRAAL, although such evidence is not contained in the Wikipedia pages linked by BLINK. Compared to GraphRetrieve and GRAAL, Entity + Mention produces a significantly larger number of sentences in all cases since it does not take advantage of the graph-based filtering. GRAAL outperforms GraphRetrieve in two of three cases and performs equivalently in the other case.

To better clarify the evidence collection, we show in Figure 3 the subgraph of G related to the first claim in Table 3, "A singer in Got a Girl starred in Final Destination 3". This claim is true because, according to Wikipedia, Mary Elizabeth Winstead is a singer in *Got a Girl* and has starred in the movie *Final Destination 3*. The evidence for such a claim is not fully contained in pages "Got a Girl" and "Final Destination 3", since part of it is contained in the page "Mary Elizabeth Winstead". The evidence is correctly identified in a group of four frames outlined by the subgraph retrieved by GRAAL, shown in Figure 3. The path that connects vertices "Got a Girl", "Mary Elizabeth Winstead", and "Final Destination 3" represents the evidence for the claim. Its edges are labeled with frames that state the composition of the *Got a Girl* duo, which includes Mary Elizabeth Winstead, and the performance of Mary Elizabeth Winstead in *Final Destination 3*, respectively. The latter is expressed by two different edges, associated to two similar frames that can be seen as alternative evidence that Mary Elizabeth Winstead starred in *Final Destination 3*. Three other edges connected with "United States" appear because both Mary Elizabeth Winstead and *Got a Girl* are stated to be "American". Their occurrence leads GRAAL to retrieve an additional unrelated frame that connects "Got a Girl" to "United States", since it is in a two-hop path between "Got a Girl" and "Final Destination 3".

Table 3. Three example of claims (the first three in the dataset) for which GRAAL finds the correct evidence, where simply collecting sentences from entity-associated pages is insufficient. We report the total number of sentences obtained by Entity + Mention, GraphRetrieve, and GRAAL. The size of the data retrieved by GraphRetrieve and GRAAL is always significantly smaller than Entity + Mention, and GRAAL's performances are always above or the same as GraphRetrieve.

Claim	**#Total Sentences**		
	Entity + Mention	**Graph-Retrieve**	**GRAAL**
A singer in *Got a Girl* starred in *Final Destination 3*	48	**29**	**29**
Mickey Rooney was in a film based on the novel *The Black Stallion* by Walter Farley	459	105	**99**
Emmy Rossum had a prominent role in a movie of which Maggie Greenwald was the director	61	29	**26**

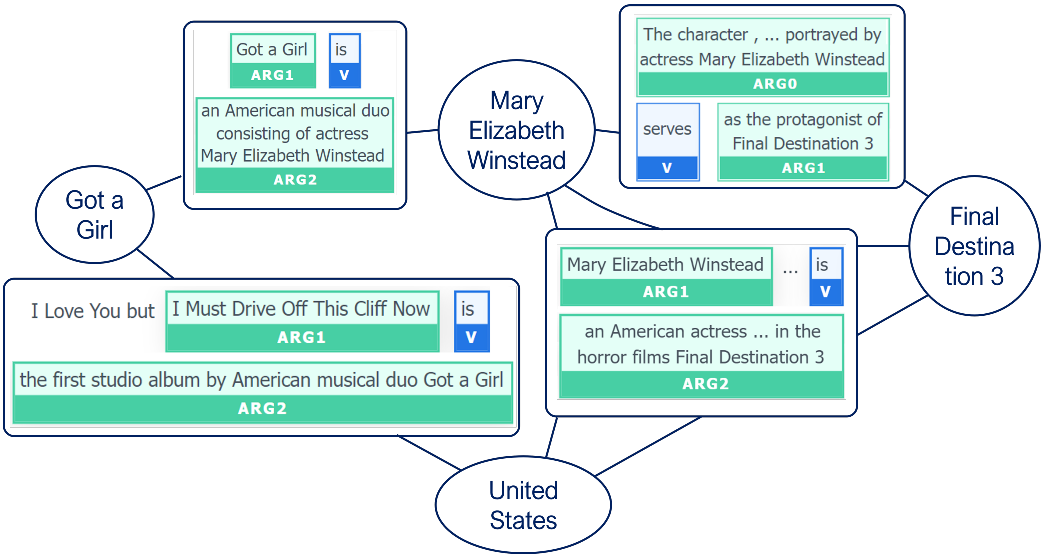
Got a Girl
Mary Elizabeth Winstead
Final Destina tion 3
United States
Got a Girl ARG1
is V
an American musical duo consisting of actress Mary Elizabeth Winstead ARG2
The character , ... portrayed by actress Mary Elizabeth Winstead ARG0
serves V
as the protagonist of Final Destination 3 ARG1
I Love You but
I Must Drive Off This Cliff Now ARG1
is V
the first studio album by American musical duo Got a Girl ARG2
Mary Elizabeth Winstead ARG1
...
is V
an American actress ... in the horror films Final Destination 3 ARG2

Figure 3. A subgraph selected by GRAAL for the claim "A singer in Got a Girl starred in Final Destination 3".

Certainly, GRAAL is not immune to failure. Failures of GRAAL can be classified into three main categories:

- Insufficient text provided by the available entities. This is the most common case of failure. For instance, the claim "Stanley Tucci performed in a television series" cannot be solved because the text contains only one linkable entity, "Stanley Tucci", and the associated text in the reference corpus is not sufficient to validate the claim. The complete evidence requires a passage from the page "Monk (TV series)" containing the information that *Monk* (which Stanley Tucci appeared in) is a television series. Note that this issue might be resolved by a more general entity linker capable of linking broad concepts such as "television series".
- Missing or incorrect entity linking from the input. Failures of the entity linking module in detecting entities from the input (claim or question) prevent identifying the complete

subgraph of the corpus graph, leading to missing important passages. For instance, in the claim "Mickey Rooney was in a film based on the novel The Black Stallion by Walter Farley" the mention "The Black Stallion" is incorrectly linked to "The Black Leather Jacket". Therefore, the retrieved subgraph cannot contain complete evidence.

- Missing or incorrect connections in the corpus graph. This can be due to various reasons, such as the failure of the entity linker to correctly detect an entity from the reference corpus or the inability to detect relations across sentences. For example, the claim "Grace Kelly did not work with Alfred Hitchcock" cannot be contradicted since the information that Grace Kelly worked in *Rear Window* (directed by Alfred Hitchcock) is contained in the reference corpus under "Other notable works include…Rear Window…". In this context, it is clear that the sentence refers to Grace Kelly, but this connection is missed during the generation of the corpus graph.

Note that GRAAL does not consider the semantics of relations between entities. Despite its simplicity, it is able to identify the correct evidence while considerably reducing the volume of data recovered, with a small cost in terms of loss of relevant evidence.

6. Conclusions

We addressed the problem of retrieving passages related to a statement for claim verification and open-domain question answering. We focused on the situation where the passages for formulating the output are spread across multiple documents in a reference corpus. The available methods cannot adequately handle this case since each piece of evidence is retrieved independently. We propose GRAAL, a method consisting of connecting all frames of the reference corpus, extracted by means of semantic role labeling, into one large graph and using such a graph to locate the evidence. Despite the simplicity of the method, we are able to significantly reduce the size of the candidate evidence with respect to a baseline, with a small loss of relevant text. We further improve the performance in terms of size of the retrieved data with regard to a previous work that does not consider frames, maintaining a similar hit rate. A promising research direction that we plan to explore concerns including the semantics of entity relationships to perform a more targeted search and further improve the method.

Author Contributions: Conceptualization, M.M.; Methodology, M.M.; Software, M.M.; Validation, M.M.; Formal analysis, M.M.; Investigation, M.M.; Resources, M.M.; Data curation, M.M.; Writing—original draft, M.M.; Writing—review & editing, M.M. and A.G.; Funding acquisition, A.G. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the projects TAILOR (H2020-ICT-48 GA: 952215) and "SI-ROBOTICS: SocIal ROBOTICS for active and healthy ageing" (PON 676 – Ricerca e Innovazione 2014-2020-G.A. ARS01_01120).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Samarinas, C.; Hsu, W.; Lee, M.L. Latent Retrieval for Large-Scale Fact-Checking and Question Answering with NLI training. In Proceedings of the 2020 IEEE 32nd International Conference on Tools with Artificial Intelligence (ICTAI), Baltimore, MD, USA, 9–11 November 2020; pp. 941–948. [CrossRef]
2. Thorne, J.; Vlachos, A.; Christodoulopoulos, C.; Mittal, A. 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, (Long Papers), New Orleans, LA, USA, 1–6 June 2018; Volume 1, pp. 809–819. [CrossRef]
3. Karpukhin, V.; Oğuz, B.; Min, S.; Lewis, P.; Wu, L.; Edunov, S.; Chen, D.; Yih, W.t. Dense passage retrieval for open-domain question answering. *arXiv* **2020**, arXiv:2004.04906.

4. Petroni, F.; Rocktäschel, T.; Riedel, S.; Lewis, P.; Bakhtin, A.; Wu, Y.; Miller, A. Language Models as Knowledge Bases? 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), Hong Kong, China, 3–7 November 2019; pp. 2463–2473.
5. Lewis, P.; Perez, E.; Piktus, A.; Petroni, F.; Karpukhin, V.; Goyal, N.; Küttler, H.; Lewis, M.; Yih, W.t.; Rocktäschel, T.; et al. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 9459–9474.
6. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. *arXiv* **2017**, arXiv:1706.03762.
7. Thorne, J.; Vlachos, A. Automated Fact Checking: Task Formulations, Methods and Future Directions. In Proceedings of the 27th International Conference on Computational Linguistics, Santa Fe, NM, USA, 20–26 August 2018; pp. 3346–3359.
8. Vosoughi, S.; Roy, D.; Aral, S. The spread of true and false news online. *Science* **2018**, *359*, 1146–1151. [CrossRef] [PubMed]
9. Zhou, J.; Han, X.; Yang, C.; Liu, Z.; Wang, L.; Li, C.; Sun, M. GEAR: Graph-based Evidence Aggregating and Reasoning for Fact Verification. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, 28 July–2 August 2019; pp. 892–901.
10. Zhong, W.; Xu, J.; Tang, D.; Xu, Z.; Duan, N.; Zhou, M.; Wang, J.; Yin, J. Reasoning Over Semantic-Level Graph for Fact Checking. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online, 5–10 July 2020; pp. 6170–6180.
11. Yoneda, T.; Mitchell, J.; Welbl, J.; Stenetorp, P.; Riedel, S. UCL Machine Reading Group: Four Factor Framework For Fact Finding (HexaF). In Proceedings of the First Workshop on Fact Extraction and VERification (FEVER), Brussels, Belgium, November 2018; pp. 97–102. [CrossRef]
12. Soleimani, A.; Monz, C.; Worring, M. BERT for Evidence Retrieval and Claim Verification. *Adv. Inf. Retr.* **2020**, *12036*, 359–366. [CrossRef] [PubMed]
13. Nie, Y.; Chen, H.; Bansal, M. Combining Fact Extraction and Verification with Neural Semantic Matching Networks. In Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence (AAAI-19), Honolulu, HI, USA, 27 January–1 February 2019; Volume 33, pp. 6859–6866, Number: 01. [CrossRef]
14. Ma, J.; Gao, W.; Joty, S.; Wong, K.F. Sentence-level evidence embedding for claim verification with hierarchical attention networks. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL 2019), Florence, Italy, 28 July–2 August 2019; pp. 2561–2571. [CrossRef]
15. Liu, Z.; Xiong, C.; Sun, M.; Liu, Z. Fine-grained Fact Verification with Kernel Graph Attention Network. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online, 5–10 July 2020; pp. 7342–7351.
16. Hanselowski, A.; Zhang, H.; Li, Z.; Sorokin, D.; Schiller, B.; Schulz, C.; Gurevych, I. UKP-Athene: Multi-Sentence Textual Entailment for Claim Verification. In Proceedings of the First Workshop on Fact Extraction and VERification (FEVER), Brussels, Belgium, November 2018; pp. 103–108. [CrossRef]
17. Guu, K.; Lee, K.; Tung, Z.; Pasupat, P.; Chang, M.W. REALM: Retrieval-Augmented Language Model Pre-Training. *arXiv* **2020**, arXiv:2002.08909.
18. Zobel, J.; Moffat, A. Inverted files for text search engines. *ACM Comput. Surv.* **2006**, *38*, 6–es. [CrossRef]
19. Akritidis, L.; Katsaros, D.; Bozanis, P. Improved retrieval effectiveness by efficient combination of term proximity and zone scoring: A simulation-based evaluation. *Simul. Model. Pract. Theory* **2012**, *22*, 74–91. [CrossRef]
20. Cambazoglu, B.B.; Kayaaslan, E.; Jonassen, S.; Aykanat, C. A term-based inverted index partitioning model for efficient distributed query processing. *ACM Trans. Web* **2013**, *7*, 1–23. [CrossRef]
21. Mongiovì, M.; Gangemi, A. Graph-based Retrieval for Claim Verification over Cross-document Evidence. In Proceedings of the International Conference on Complex Networks and Their Applications, Madrid, Spain, 30 November–2 December 2021; pp. 486–495.
22. Chang, W.C.; Yu, F.X.; Chang, Y.W.; Yang, Y.; Kumar, S. Pre-training Tasks for Embedding-based Large-scale Retrieval. *arXiv* **2020**, arXiv:2002.03932.
23. Lee, K.; Chang, M.W.; Toutanova, K. Latent Retrieval for Weakly Supervised Open Domain Question Answering. *arXiv* **2019**, arXiv:1906.00300.
24. Guo, Z.; Schlichtkrull, M.; Vlachos, A. A Survey on Automated Fact-Checking. *arXiv* **2021**, arXiv:2108.11896.
25. Chen, Q.; Zhu, X.; Ling, Z.; Wei, S.; Jiang, H.; Inkpen, D. Enhanced LSTM for Natural Language Inference. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Vancouver, BC, Canada, 30 July–4 August 2017; pp. 1657–1668. [CrossRef]
26. Papadopoulos, D.; Metropoulou, K.; Papadakis, N.; Matsatsinis, N. FarFetched: Entity-centric Reasoning and Claim Validation for the Greek Language based on Textually Represented Environments. In Proceedings of the 12th Hellenic Conference on Artificial Intelligence, Corfu, Greece, 7–9 September 2022; pp. 1–10.
27. Kallipolitis, A.; Gallos, P.; Menychtas, A.; Tsanakas, P.; Maglogiannis, I. Medical Knowledge Extraction from Graph-Based Modeling of Electronic Health Records. In Proceedings of the IFIP International Conference on Artificial Intelligence Applications and Innovations, León, Spain, 14–17 June 2023; pp. 279–290.
28. Giarelis, N.; Kanakaris, N.; Karacapilidis, N. On a novel representation of multiple textual documents in a single graph. In Proceedings of the International Conference on Intelligent Decision Technologies, Virtual Conference, 17–19 June 2020; Springer: Cham, Switzerland, 2020; pp. 105–115.

29. Giarelis, N.; Kanakaris, N.; Karacapilidis, N. An innovative graph-based approach to advance feature selection from multiple textual documents. In Proceedings of the IFIP International Conference on Artificial Intelligence Applications and Innovations, Neos Marmaras, Greece, 5–7 June 2020; pp. 96–106.
30. Jalil, Z.; Nasir, M.; Alazab, M.; Nasir, J.; Amjad, T.; Alqammaz, A. Grapharizer: A Graph-Based Technique for Extractive Multi-Document Summarization. *Electronics* **2023**, *12*, 1895. [CrossRef]
31. Blloshmi, R.; Conia, S.; Tripodi, R.; Navigli, R. Generating Senses and RoLes: An end-to-end model for dependency-and span-based Semantic Role Labeling. In Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21, Montreal, QC, Canada, 19–27 August 2021; pp. 3786–3793.
32. Màrquez, L.; Carreras, X.; Litkowski, K.C.; Stevenson, S. Semantic role labeling: An introduction to the special issue. *Comput. Linguist.* **2008**, *34*, 145–159. [CrossRef]
33. Shi, P.; Lin, J. Simple bert models for relation extraction and semantic role labeling. *arXiv* **2019**, arXiv:1904.05255.
34. Wu, L.; Petroni, F.; Josifoski, M.; Riedel, S.; Zettlemoyer, L. Scalable Zero-shot Entity Linking with Dense Entity Retrieval. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Online, 16–20 November 2020; pp. 6397–6407.
35. Li, J.; Sun, A.; Han, J.; Li, C. A survey on deep learning for named entity recognition. In Proceedings of the 2023 IEEE 39th International Conference on Data Engineering (ICDE), Anaheim, CA, USA, 3–7 April 2023; Volume 34, pp. 50–70.
36. Berman, P.; Ramaiyer, V. Improved approximations for the Steiner tree problem. *J. Algorithms* **1994**, *17*, 381–408. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.