Natural Logic-guided Autoregressive Multi-hop Document Retrieval for Fact Verification

Rami Aly
University of Cambridge
Department of Computer Science
and Technology
rami.aly@cl.cam.ac.uk

Andreas Vlachos
University of Cambridge
Department of Computer Science
and Technology
andreas.vlachos@cl.cam.ac.uk

Abstract

A key component of fact verification is the evidence retrieval, often from multiple documents. Recent approaches use dense representations and condition the retrieval of each document on the previously retrieved ones. The latter step is performed over all the documents in the collection, requiring storing their dense representations in an index, thus incurring a high memory footprint. An alternative paradigm is retrieve-and-rerank, where documents are retrieved using methods such as BM25, their sentences are reranked, and further documents are retrieved conditioned on these sentences, reducing the memory requirements. However, such approaches can be brittle as they rely on heuristics and assume hyperlinks between documents. We propose a novel retrieve-and-rerank method for multi-hop retrieval, that consists of a retriever that jointly scores documents in the knowledge source and sentences from previously retrieved documents using an autoregressive formulation and is guided by a proof system based on natural logic that dynamically terminates the retrieval process if the evidence is deemed sufficient. This method is competitive with current state-of-the-art methods on FEVER, HoVer and FEVEROUS-S, while using 5 to 10 times less memory than competing systems. Evaluation on an adversarial dataset indicates improved stability of our approach compared to commonly deployed threshold-based methods. Finally, the proof system helps humans predict model decisions correctly more often than using the evidence alone.

1 Introduction

With the growing volume of potentially misleading and false claims (Graves, 2018), automated fact verification (Hardalov et al., 2022; Guo et al., 2022) is of growing interest. A key component of open-domain fact verification systems is the retrieval of relevant documents from a knowledge base (KB) which provide the necessary evidence supporting or refuting a claim. Evidence retrieval accuracy correlates strongly with fact-checking accuracy, as observed in a recent shared task (Aly et al., 2021b).

Claim: The 66th Primetime Emmy Awards was hosted by an Iraqi comedian born in 1973.

Evidence Documents:

66th Primetime Emmy Awards

The 66th Primetime Emmy Awards honored the best in U.S. prime time television programming from June 1, 2013 until May 31, 2014, as chosen by the Academy of Television Arts & Sciences. Comedian and Late Night host Seth Meyers hosted the ceremony for the first time.

Seth Meyers

Seth Adam Meyers (born December 28 , 1973) is an American comedian , writer , producer , political commentator , actor , media critic , and television host. He hosts Late Night with Seth Meyers, a late-night talk show on NBC. Prior to that, he was a cast member and head writer for NBC's Saturday Night Live (2001–2014).

Verdict: Refuted

Figure 1: A FEVER example where multiple documents are required for verification (relevant evidence in red).

Document retrieval for fact verification can be complex, as required evidence is often found in multiple documents, with each document containing partial information needed to assess the veracity of a claim. An example is shown in Figure 1. Given the claim "*The 66th Primetime Emmy Awards was hosted by an Iraqi comedian born in 1973.*", information from two documents has to be considered to verify the claim: one mentions the person who hosted the Awards, and another provides information about this person. The claim itself is often not leading to the second document, and it is rather misleading as it mentions incorrect information that would retrieve irrelevant documents (i.e. about Iraqi comedians). Instead we need to condition the retrieval of some evidence pieces on other evidence documents, (i.e. that *Seth Meyers* is the host of the *66th Primetime Emmy Awards*).

Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pages 6123–6135
December 7-11, 2022 ©2022 Association for Computational Linguistics

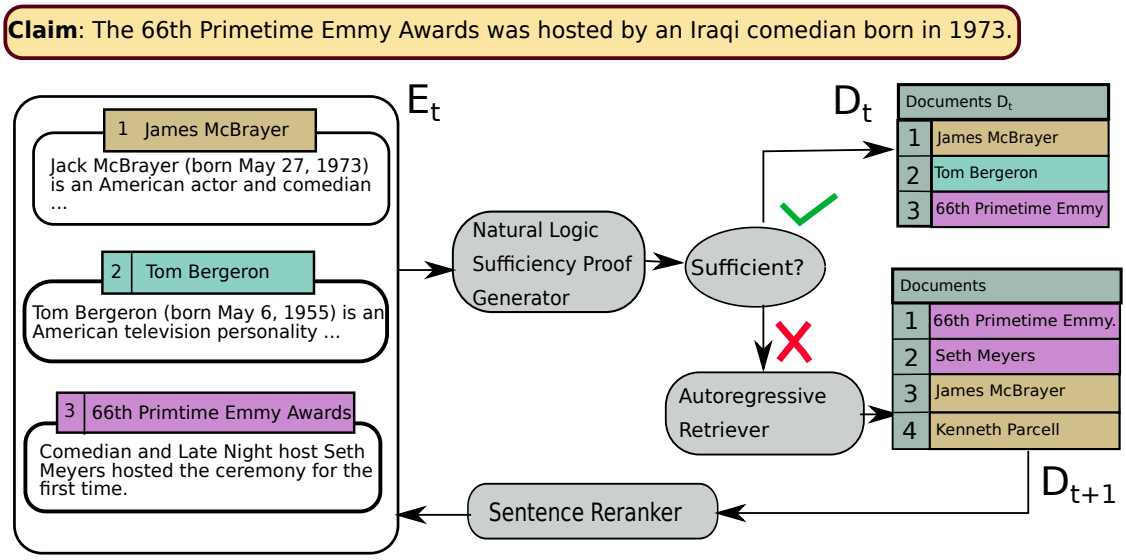

Figure 2: The AdMIRaL pipeline. At hop t, given the claim and sentences E_t from documents D_t, a proof is generated to predict whether the evidence E_t is sufficient for verification or whether additional evidence is needed. If sufficient, the retriever terminates, otherwise, our autoregressive retriever scores documents in the KB jointly with E_t updating the documents to D_{t+1}, before they are passed to the sentence reranker to obtain E_{t+1}.

Recent approaches to multi-hop retrieval for fact verification commonly use dense representations for both claim and documents, and condition the retrieval of each document on the previously retrieved ones (Xiong et al., 2021; Khattab et al., 2021). After each iteration (hop), the claim representation is modified and compared against the entire KB, necessitating to store all document representations in a dense index. However, typically millions of documents are considered, thus the dense index has a large memory footprint (see Table 1). An alternative paradigm is Retrieve and Rerank (RnR) (Nie et al., 2019; Stammbach, 2021; Malon, 2021), where candidate documents are retrieved, sentences from them are reranked, and then conditioned on the top-ranked sentences additional documents are retrieved. By retrieving the candidate documents using sparse retrievers (e.g. BM25) a dense index becomes unnecessary, while a dense reranker can still be used to take advantage of dense representations. Yet, to reach competitive multi-hop performance, RnR systems assume links between documents and rely on heuristics, such as down-weighting hyperlinked documents by a fixed factor, assuming evidence from the first iteration is more important. Heuristics might not generalize well across datasets, and while links between documents are beneficial when available (e.g. in Wikipedia), many textual KBs do not have them.

Model	Datasets		
	HoVer	FEVER	FEVEROUS-S
BM25	3.0GB	6.1GB	20.4GB
MDR	32.1GB	72.8GB	–
ColBERT	81.3GB	–	–
ColBERTv2	16.0GB	34.4GB	124.2GB
AdMIRaL (Ours)	3.3GB	6.4GB	20.7GB

Table 1: Memory footprint for different datasets of several sparse/dense retrieval models. FEVER/HoVer consider only Wikipedia introductions, while FEVEROUS-S consists of entire Wikipedia pages.

To address these challenges we propose AdMIRaL (Autoregressive document Multi-hop Information Retrieval with Natural Logic-guidance), a novel multi-hop document retriever for RnR that consists of two components: i) a retriever that jointly scores documents in the KB and sentences reranked from previously retrieved documents using an autoregressive formulation (De Cao et al., 2021), ii) a proof system using Natural Logic (MacCartney and Manning, 2014) to assess the sufficiency of the evidence retrieved to verify a given claim, and terminate the retrieval of further documents. The method is illustrated in Figure 2. By retrieving using an autoregressive formulation, generating document and sentence identifiers jointly token by token and conditioned

only on the context, AdMIRaL does not need to store dense representations in an index. The proof system controls the merging of evidence documents between hops while being faithful and interpretable with regards to system's operation in each hop.

We improve document recall and F_1 by 1.4% and 4.6% over the state-of-the-art performance on FEVER (Thorne et al., 2018) respectively, and are competitive with state-of-the-art performance on HoVer (Jiang et al., 2020) and the sentence-only version of FEVEROUS (i.e. excluding tables) (Aly et al., 2021a), while using 5 to 10 times less memory than competing dense retrieval systems and a runtime complexity more favourable when scaling to large KBs. We further assess the robustness of AdMIRaL on an adverserial version of FEVER (Hidey et al., 2020), and show performance gains using various initial retrievers. Finally, human evaluation indicates that the natural logic proofs help humans predict model decisions correctly more often than using the evidence directly.[1]

2 Related Work

Early approaches to multi-hop document retrieval for automated fact verification are based on the RnR paradigm. They use sparse or entity-linking based retrievers to find candidate documents (e.g. (Hanselowski et al., 2018)), rerank sentences in a classification formulation (such as ESIM (Thorne et al., 2018) or pre-trained encoders (Liu et al., 2020; Zhong et al., 2020)), and use hyperlinks to find additional documents (Nie et al., 2019; Stammbach and Neumann, 2019). The aforementioned approaches are limited to two iterations, using hyperlinks extracted from the initial list of candidate sentences to be considered as additional documents in a second iteration. Assuming that the evidence from the first iteration (i.e. initial retrieval) is more important, Stammbach (2021) down-weight hyperlinked documents by a fixed factor. Malon (2021) proposes the use of a generative model that imagines missing evidence sentences and selects new sentences based on word overlap. Indirect improvements to multi-hop RnR are achieved through stronger document retrieval models, such as GENRE (De Cao et al., 2021), or more accurate rerankers (Stammbach, 2021; Jiang et al., 2021a). GENRE in particular produced state-of-the-art results on document retrieval for fact verification, by generating documents using an autoregressive formulation, not necessitating a dense index.

Multi-hop dense passage retrieval (MDR) (Xiong et al., 2021) iteratively retrieves evidence documents using dense passage retrieval (DPR) (Karpukhin et al., 2020), a bi-encoder that encodes claim and documents separately and uses efficient maximum inner-product search to score each document in the KB. Since the search space grows exponentially with each iteration in the number of documents in the KB, MDR uses beam search to aggressively prune the search space which reduces scalability to many hops. In contast, Khattab et al. (2021)'s Baleen retriever performs multi-hop document retrieval by condensing retrieved documents after each iteration into a condensed context (i.e. sentence(s)) that is used to update the dense representation of the claim, reducing the search space by omitting all other candidates. While condensing is similar to the reranking step of RnR, Baleen then still scores each document in the KB at each hop, like MDR. They further propose late interaction (FLIPR), which allows different parts of the claim to match different relevant parts of documents. Indicative of the challenge of scaling to large KBs is that out of 12 systems submitted to the FEVEROUS shared task (Aly et al., 2021b), with FEVEROUS being the fact verification dataset with the largest KB) not a single one opted to use a dense retrieval index.

Beyond fact verification, in question answering (QA), sophisticated graph based retrieval models have been proposed that make explicit use of links in the KB (Asai et al., 2020; Li et al., 2021). Of particular relevance is the approach of (Qi et al., 2021), that uses the feedback of the reader after each iteration to determine whether the answer has been retrieved and is sufficient, thus determining the number of hops dynamically instead of fixing it in advance. In their QA benchmark, all answers to the questions can be found in the source; however in many fact verification datasets a key challenge included are claims that cannot be verified using the KB. Moreover, as explained false information mentioned in claims can be misleading, making the stopping criterion based on sufficiency for fact checking more challenging.

3 AdMIRaL

Given a KB consisting of documents $\mathbb{D}$, the task of document retrieval for fact verification is to find the

[1] https://github.com/Raldir/AdMIRaL

set documents $D = \{d_1, \ldots, d_n\} \subset \mathbb{D}$ that is sufficient to support or refute a claim c.[2] We assume that each document is associated with a unique document title, following previous work (De Cao et al., 2021). In the case of multi-hop retrieval n must be larger than 1. In the RnR paradigm the retrieval is done in three steps: (i) find and return the k best-scoring document sets $D_t = \{D_t^1, \ldots D_t^i \ldots D_t^k\}$, with t being the current retrieval iteration, with $t \geq 1$ and $|D_t^i| = t$, and D_1^i the ith best-scoring single-hop document set of length 1 (i.e. a single document), (ii) rerank sentences from the k document sets in D_t into the top l sentences E_t, iii) use E_t to update the set of document sequences to D_{t+1}. Steps two and three are then repeated for a total of n hops. Since n is not known a priori for a given claim previous work sets n to an upper bound of the dataset.

AdMIRaL performs the third step of RnR in two stages: (i) the retrieval of D_{t+1} by jointly scoring $\mathbb{D}$ and E_t using a generative model in an autoregressive framework that cross-attends over the claim c and E_t, (ii) a dynamic retrieval termination criterion formulated as an evidence sufficiency task by generating a proof of Natural Logic (MacCartney and Manning, 2014) based on Proofver (Krishna et al., 2022), hence the number of hops n_dyn for AdMIRaL is being determined dynamically for each claim given the evidence E_t, with $n_dyn \leq n$.

3.1 Autoregressive Document Retrieval

Given a claim c and the top-ranked sentences E_t, we formulate AdMIRaL as a pointwise reranker for document sets D_{t+1}^i of length $t+1$, i.e. sets containing one more document (hop) than the ones in D_t. The scoring function for D_{t+1}^i is defined jointly over D_t^i and d_{t+1}. We consider the sentences E_t to be an approximation of all relevant information in D_t regarding c, yet in much more compact form. Hence, we define the score of a set of documents D_t^i as the sum of the scores of all its possible underlying sentence combinations :

$$\begin{aligned}
&\mathrm{score}(D_{t+1}^i|c, E_t) \\
&= \mathrm{score}_d(d_1, \ldots, d_t, d_{t+1}|c, E_t) \\
&= \sum_{s_1 \in d_1, \ldots, s_t \in d_t} \mathrm{score}_s(s_1, \ldots s_t, d_{t+1}|c, E_t),
\end{aligned} \tag{1}$$

[2]In FEVER, if a claim is labelled with not enough evidence (*NEI*) it has no documents associated with it, unlike in FEVEROUS and HoVer (HoVer merges *refuted* and *NEI* instances in one class, *not supported*).

with $d_{t+1} \in \mathbb{D}$. Thus Eq.1 scores D_{t+1}^i jointly on $\mathbb{D}$ and E_t. Since we assume all relevant information of D_t is in E_t, the scoring function $\mathrm{score}_s(\mathcal{S}, d_{t+1}|c, E_t)$, with $\mathcal{S} = \{s_1, \ldots, s_t\}$, only scores sets $\{\mathcal{S}, d_{t+1}\}$ where all sentences of $\mathcal{S}$ are in E_t. The scoring function is computed using a generative model with an autoregressive formulation over the unique document titles, conditioning the score of d_{t+1} on $\mathcal{S}$:

$$\begin{aligned}
&\mathrm{score}(\mathcal{S}, d_{t+1}|c, E_t) \\
&= p_\theta(\mathcal{S}|c, E_t) \cdot p_\theta(d_{t+1}|c, E_t, \mathcal{S}) \\
&= \underbrace{\prod_{u=1}^{M} p_\theta(q_u|q_{<u}, c, E_t)}_{p_\theta(\mathcal{S}|c, E_t)} \\
&\cdot \underbrace{\prod_{m=1}^{N} p_\theta(y_m|y_{<m}, c, E_t, \mathcal{S})}_{p_\theta(d_{t+1}|c, E_t, \mathcal{S})},
\end{aligned} \tag{2}$$

, where $\mathbf{y}$ is the sequence of tokens representing the title of document d_{t+1}, $\mathbf{q}$ is the sequence of tokens representing the sentence identifiers (i.e. unique sentence ids encoded in E_t) of $\mathcal{S}$, and θ are the parameters of the model.

The scoring model is a generative pre-trained transformer-based architecture, namely BART (Lewis et al., 2020), allowing us to cross-encode claim c and sentences E_t (Petroni et al., 2019; Radford et al., 2019). Since the document sequences are scored using only the claim and the information E_t, no document representations have to be pre-computed and stored in an index. However, since $|\mathbb{D}|$ is very large, it is infeasible to compute a score for each set $\{\mathcal{S}, d_{t+1}\}$, instead we use beam search to efficiently navigate the search space, searching only for the q top-ranked sequences. Note that the beam search for generation differs substantially from the one used for iterative retrieval by Xiong et al. (2021): our search is over the model's vocabulary and using a softmax operation for scoring, not over the entire KB with a MIPS comparison between all dense representations, the former being substantially more efficient with a much smaller search space. Since in traditional decoding any token from the vocabulary can be generated at any position, we might generate sequences that are non-existing document/sentence identifiers. We follow De Cao et al. (2021) by constraining the generation using a prefix tree (trie). In practice, we have to switch between two search spaces: the sentence

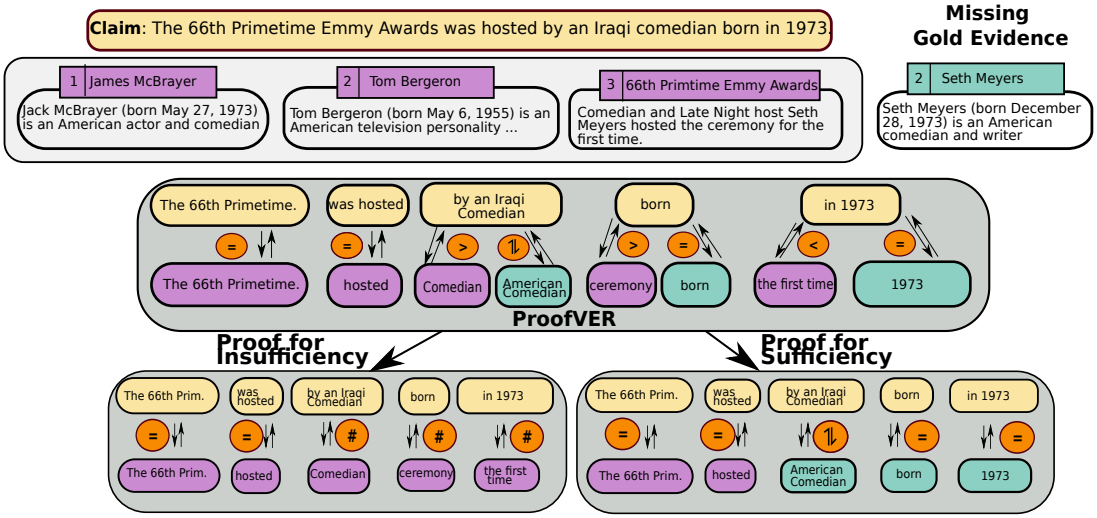

Figure 3: Left: Illustration of the process of generating sufficiency proofs for training. For a given claim two proofs are generated: one given sufficient and insufficient evidence. Insufficiency is predicted iff any mutation in the proof sequence is assigned the independence operator.

and document identifiers, as we want to ensure that sentence identifiers are generated first to condition the generation of d. We achieve this by employing dynamically constrained markup decoding (De Cao et al., 2021), where markups are used during encoding to switch between search spaces. The q top-ranked document sets are then returned as D_{t+1}.

Training We train a separate model for each hop t using maximum likelihood estimation, following the Neural Machine Translation fine-tuning of BART (Lewis et al., 2020), computing the log probability of a document title (as a sequence of tokens) given E_g and the claim c. Given an ordered list of gold evidence sentences E_g of length m, with $m \leq n$, we consider as input during training the first $m-1$ evidence elements, and the document title of the remaining evidence sentence to be y. In cases where an explicit ordering of evidence sentences is not available, we generate all $t-1$ combinations of splitting E_g into input and output data. See appendix A.1 for details.

3.2 Evidence Sufficiency with Natural Logic

To determine whether the retrieved evidence is sufficient for the claim being verified, we generate a proof using natural logic (MacCartney and Manning, 2014), inspired by recent work who used it for verification (Krishna et al., 2022). Given a claim c and the evidence sentences E_t, a seq2seq model generates a proof sequentially in an autoregressive formulation, from left to right. Each part of the claim c is sequentially mutated into an evidence span of E_t, with a natural logic operation (NatOp) defining the nature of the mutation. We consider four out of seven NatOps defined in (MacCartney and Manning, 2014) for the sufficiency proof: equivalence ($\equiv$), negation ($\neg$), alternation ($\downarrow\uparrow$), and independence (#). See Figure 3 for an example (bottom left). Mutations between semantically equivalent spans are assigned the equivalence NatOp ($\equiv$), such as *The 66th Primetime Emmy Awards*. The mutation of the claim span *by an Iraqi comedian* is assigned the independence NatOp (#), indicating that no related evidence span exist in the Evidence E_t. We do not consider the cover NatOp ($\smile$), forward entailment NatOp ($\sqsubseteq$) and reverse entailment NatOp ($\sqsupseteq$) as they are not conclusive indicators for sufficiency and can be replaced with independence for our purposes. For instance, Africa $\sqsubseteq$ Tunisia holds, yet given a claim "Ryan Gosling has been to Africa." and evidence "Ryan Gosling has been to Tunisia.", further evidence that links Tunisia to Africa is required for the evidence to be sufficient.

To determine the sufficiency based on the generated proof, we consider the sequence of operators assigned to each mutation. We predict insufficiency iff any mutation in the proof sequence is assigned the independence operator. The proof-based sufficiency prediction is faithful by construction and provides an and explainable sufficiency prediction for multi-hop systems.

Since claim-evidence pairs annotated with natural logic proofs for sufficiency prediction are not available, we generate them. For each claim we generate two proofs: one based on insufficient and one on sufficient evidence (see Figure 3). Given incomplete and complete evidence for a claim, we first use ProofVER (Krishna et al., 2022) to generate the respective initial proofs. However, since ProofVER has been trained on data annotated with heuristics targeting the assessment of a claims veracity, ProofVER's NatOps specifying mutations are not suitable for our purpose of predicting the sufficiency of evidence. Hence, we reassign NatOps in each initial proof to ensure its consistency with the sufficiency of the input evidence. First, all forward/reverse entailment NatOps ($\sqsubseteq$)/($\sqsupseteq$) are replaced with independence NatOps (#). We then modify the fewest NatOPs in a proof possible to reach the correct (in-)sufficiency prediction. For proofs that indicate sufficiency of evidence which is insufficient, we assign the independence NatOp to the mutation with the most dissimilar claim and evidence spans, measured using cosine similarity of the mean-pooled contextual representation of a pre-trained language model. If a proof indicates insufficiency but is sufficient, we first search for claim/evidence terms in multiple lexicons (Wordnet (Miller, 1995), Synonym Antonym pairs of (Roth and Schulte im Walde, 2014), and PPDB (Pavlick et al., 2015)) to find suitable matches, and then replace all remaining independence NatOps with equivalence if the claim is supported, or negation if refuted.

4 Experimental setup

Datasets We evaluate our multi-hop document retriever on FEVER (Thorne et al., 2018), FEVEROUS (Aly et al., 2021a), and HoVer (Jiang et al., 2020). FEVER consists of claims that predominantly require a single evidence sentence (87%). Contrary to Xiong et al. (2021) who evaluate only on the multi-hop part on FEVER, we report results for multi-hop and the entire dataset; this is more realistic as in practice it is unknown in advance whether a claim requires multi-hop document retrieval. HoVer contains of 46%, 36%, and 18% two-hop, three-hop, and four-hop claims, respectively. Contrary to FEVER, and HoVer that only consider the introductory section of Wikipedia pages, FEVEROUS considers entire Wikipedia articles, including semi-structured evidence in the form of table (cells) and contains of 16% multi-hop claims. From FEVEROUS we consider only the claims that require exclusively sentence evidence, as we focus on text retrieval (FEVEROUS-S), which constitutes about 41% of claims in FEVEROUS.

Implementation Details The autoregressive model for both retrieval and proof generation is BART (Lewis et al., 2020), which are trained independently from each other. For the initial retrieval, we first retrieve document candidates D_1 for all three datasets using GENRE, fine-tuned on the KILT version of FEVER (Petroni et al., 2021), and BM25 based on Pyserini (Lin et al., 2021). To rerank the sentences of these documents and keeping the top $l = 5$ in E_1, we use the token-level evidence selection model of (Stammbach, 2021) for FEVER, and the pointwise T5 reranker of (Jiang et al., 2021a) for HoVer, and FEVEROUS-S. For FEVER and FEVEROUS-S we consider the top 10 documents for reranking while for HOVER we focus on the top 100, to keep scores comparable to previous work.

5 Results

5.1 Multi-hop Document Retrieval

Document retrieval results on each dataset's dev set are shown in Table 2. Results include single-hop retrievers, covering sparse retrieval (BM25), entity-based (GENRE), and dense passage retrieval (DPR). We further show the scores of the single-hop retriever used for AdMIRaL, namely AdMIRaL single-hop. We further compare AdMIRaL against state-of-the-art multi-hop retrieval approaches, including MDR, Baleen, and ColBERT-Hop (Khattab et al., 2021), which all necessitate a dense index. We further compare against an RnR retriever that makes explicit use of hyperlinks in sentences to retrieve new documents, as done in the multi-hop setting of Stammbach (2021). The memory and computational requirements to run dense retrievers on FEVEROUS-S exceeded our resources (see Appendix A.2), hence results on FEVEROUS-S are only computed for RnR, i.e. Jiang et al. (2021a) with hyperlinks, and single-hop approaches. As seen in Table 2, AdMIRaL achieves the highest 2-hop recall score on all datasets, and the highest overall recall on FEVER and FEVEROUS-S, falling only behind Baleen on HoVer. AdMIRaL improves 2-hop recall of the initial retrieval by 34.8% percentage points on

	Model/Datasets	Recall@5				Recall@100	
		FEVER		FEVEROUS-S		HoVer	
		2-hop	Overall	2-hop	Overall	2-hop	Overall
Single-Hop	BM25	0.150	0.658	0.410	0.752	0.789	0.397
	GENRE	0.191	0.892	0.330	0.705	0.382	0.107
	AdMIRaL single-hop	0.357	0.928	0.441	0.799	0.886	0.470
	DPR	0.191	0.754	–	–	–	–
Multi-Hop	Hyperlinks	0.667	<u>0.945</u>	<u>0.506</u>	<u>0.822</u>	0.904	0.641
	MDR†	<u>0.691</u>	–	–	–	–	–
	ColBERT-Hop*	–	–	–	–	<u>0.958</u>	0.748
	Baleen*	–	–	–	–	**0.977**	**0.922**
	AdMIRaL (Ours)	**0.705**	**0.956**	**0.610**	**0.847**	**0.977**	<u>0.817</u>

Table 2: Document retrieval scores for 2-hop, and overall scores. To compare with previous work on HoVer, we report recall@100 for *supported* claims on dev. † and * indicate results taken from Xiong et al. (2021) and Khattab et al. (2021), respectively. **Bold** numbers indicate best and <u>underline</u> the second-best score.

FEVER, 16.9% on FEVEROUS-S and by 9.1% on HoVer. For the latter, overall scores increased by even larger 34.7%, as AdMIRaL improves retrieval substantially for HoVer claims requiring more than 2 hops. Furthermore, AdMIRaL is more precise than state-of-the-art models, achieving an F_1 improvement over MDR on FEVER by 4.6% and over hyperlinks on FEVEROUS-S by 14.2% (see Appendix A.3).

5.2 Efficiency

Memory Footprint AdMIRaL achieves overall competitive performance while being an order of magnitude more space efficient. AdMIRaL's memory footprint is composed of the inverted index for the initial BM25 retrieval and the prefix tree of the document titles (excluding the model itself), resulting in a footprint of 3.3GB, 6.4GB and 20.7GB for HoVer, FEVER, and FEVEROUS-S, respectively. This is about 10 times less than MDR, 5 times less than Baleen (ColBERTv2), and 27 times less than ColBERT (see Table 1), since these necessitate a dense index of all documents in the KB. The footprint of AdMIRaL is comparable to RnR approaches (Stammbach, 2021), as the inverted index is only a few hundred Megabytes of size, negligible to the size of the inverted index.

Runtime Efficiency The runtime complexity of a retriever consists of two components: step (i) the indexing of the KB and (ii) the retrieval itself.[3] Since AdMiRaL only builds a BM25 index for the initial retrieval - step (i) - is substantially faster than for dense retrievers such as Baleen. Specifically, it takes 2 minutes and 32s to build the HoVer index for AdMIRaL (and for Stammbach's RnR retriever), compared to 290 minutes and 12s for Baleen's dense index. For step (ii), AdMIRaL scales better than dense retrievers with respect to both KB size and the number of iterations. Dense retrievers such as Baleen or MDR scale by $\mathcal{O}(n|\mathbb{D}|)$, with $|\mathbb{D}|$ being the number of documents in the KB, as they do a comparison to all documents in the KB at every iteration for a total of n iterations, with n being the number of iterations set to an upper bound. In contrast, for AdMIRaL, only the initial retrieval depends on the size of the KB (i.e. BM25), while the autoregressive retrieval at each hop depends on the model's vocabulary size, hence $\mathcal{O}(n_{dyn} + |\mathbb{D}|)$ (including initial retrieval, otherwise $\mathcal{O}(n_{dyn})$), with $n_{dyn} \leq n$ being the number of iterations according to the dynamic termination of AdMIRaL. [4] However, the underlying constant of AdMIRaL is large as it relies on two Encoder-Decoder models (autoregressive retrieval + proof generation) and a sentence reranker which makes it computationally more expensive when used on relatively small KB's such as HoVer. We measure 2.87s on average for a single HoVer query on AdMIRaL, 1.94s for Baleen, and 0.69s for Stammbach (2021). However, on large KBs (such as FEVEROUS-S with 7x the size of HoVer), the runtime is more favourable

[3]The indexing efficiency is relevant as a KB's content frequently updates in the real world and so then must the index.

[4]Reducing runtime through dynamic termination with AdMIRaL would be challenging as the computational cost of the proof generator itself is at least as high as the autoregressive retriever. Investigating how dynamic termination can improve multi-hop retrieval efficiency is an interesting future direction to explore but is outside of our focus of AdMIRaL.

for AdMIRaL than Baleen. While the memory requirements to run Baleen on FEVEROUS-S exceed our resources, already on a KB with twice the size of HoVer, Baleen takes 2.30s, while AdMiRaL's runtime is nearly unchanged (2.88s).

6 Discussion

Autoregressive Generation We compare the auto-regressive document scoring method of AdMIRaL to some variants, namely a model that i) considers only the top sentence of E_t for scoring and generation (Top-1) ii) does not score documents and sentences jointly, instead it only scores documents, and the top-ranking documents are concatenated to the t highest-ranked documents of D_t (Not-joint) iii) scores a ranked set of documents directly (i.e. scoring $\mathbb{D}^t$) (Joint-docs). We also evaluate a AdMIRaL model that exploits hyperlink information by concatenating a sentence's hyperlinks to its the end before being passed as input. Results are shown in Table 3. While *Top-1* achieves a comparable exact match score to AdMIRaL (even slightly higher for two-hop claims), its recall is considerably lower. On the contrary, *Not-joint* achieves competitive recall, yet, lags behind in terms of exact match accuracy, as the original order of the top-ranked documents in D_t are largely unchanged. Finally, *Joint-docs* performs worst overall, likely due to the difficulty of evidence ordering during training, as also observed by (Xiong et al., 2021). Incorporating hyperlink information into AdMIRaL improves recall substantially, resulting to a FEVER multi-hop state-of-the-art improvement of 0.16 percentage points.

Model	FEVER			
	R@5		EM	
	Two-hop	Overall	Two-hop	Overall
Initial	0.35	0.93	0.26	0.87
Top-1	0.67	0.94	0.52	0.88
Not-Joint	0.71	0.96	0.32	0.87
Joint-docs	0.59	0.94	0.43	0.87
Ours	0.71	0.96	0.51	0.89
Ours w/ hyperlinks	**0.85**	**0.97**	**0.51**	**0.89**

Table 3: Document retrieval scores for several variations to proposed autoregressive retriever, R@5: Recall@5, EM: Exact Match Accuracy.

Robustness We further evaluate the robustness of our model by evaluating AdMIRaL on an adversarial fact verification dataset DeSePtion (Hidey et al., 2020), which consists adversarial attacks generated as part of the FEVER2.0 adversarial shared task (Thorne et al., 2019). The attacks consider lexical variations/substitutions, entity disambiguation, (multi-hop) temporal reasoning, multiple prepositions and multi-hop reasoning. Document level results for models trained on FEVER and evaluated on DeSePtion are shown in Table 4. AdMIRaL achieves substantial increases over the initial retriever and a BM25 baseline. Moreover, while *Not-joint* achieves similar recall to AdMIRaL on FEVER, it performs worse on the adversarial dataset. This highlights the brittleness of adding new documents statically to the top-ranking documents using a fixed position or threshold, as commonly done.

	Model	Datasets			
		FEVER		Adversarial-FEVER	
		Two-hop	Overall	Two-hop	Overall
R	Initial	0.36	0.93	0.54	0.77
	BM25	0.15	0.658	0.22	0.55
	Not-Joint	**0.71**	**0.96**	0.72	0.84
	Ours	**0.71**	**0.96**	**0.74**	**0.86**
EM	Initial	0.26	0.87	0.48	0.69
	BM25	0.06	0.40	0.13	0.36
	Not-Joint	0.32	0.87	0.50	0.71
	Ours	**0.51**	**0.89**	**0.56**	**0.73**

Table 4: Document retrieval scores on the adversarial dataset, R: Recall@5, EM: Exact Match accuracy.

Results with different initial retrievers Another aspect we analyze is the stability of AdMIRaL with different initial retrievers, i.e. the retrieval of E_1. Results are shown in Table 5. The relative improvements achieved by AdMIRaL are consistent across retrievers, improving recall@5 for BM25, KGAT (Liu et al., 2020), and (Jiang et al., 2021b) (and (Stammbach, 2021) as used in AdMIRaL) on FEVER, by an average of 33% percentage points with a variance of 0.0004.

Initial Retriever	Single-hop		AdMIRaL	
	Two-hop	Overall	Two-hop	Overall
BM25	0.065	0.486	0.370	0.780
KGAT	0.470	0.955	0.790	0.968
(Jiang et al., 2021a)	0.356	0.925	0.701	0.953
AdMIRaL	0.357	0.928	0.705	0.956

Table 5: Document retrieval scores using various methods for retrieving the initial evidence sentences E_1, R: Recall@5. Note that KGAT includes gold documents before sentence re-ranking and hence has not been considered in our main experiments.

Sufficiency Proof with Natural Logic To evaluate the effectiveness of our proof-based approach to determine evidence sufficiency, we compare AdMIRaL against four baselines: i) a model that always considers the evidence to be complete/incomplete ii) a binary classifier (BART with linear head) trained to distinguish complete from incomplete input, iii) a ProofVER generated proof, iv) a Natlog proof generated solely by using lexical resources and assigning unmatched mutations the independence NatOp. We further compare our approach against an oracle that always correctly decides whether additional evidence is required. Results on FEVER are shown in Table 6. In addition to retrieval recall@5 the table shows insufficiency precision and recall for multi-hop claims. Considering all evidence to be insufficient is equivalent to running AdMIRaL's retriever for a fixed number of iterations n. AdMIRaL's sufficiency check improves notably on it, also outperforming alternative sufficiency prediction methods. However, we also note that there is substantial room for improvement as the oracle outperforms our current approach in terms of precision, which translates to a substantially higher retrieval recall@5.

Model	FEVER		Insufficiency	
	Two-hop	Overall	P	R
All insuf	0.69	0.95	0.59	1.0
All suf	0.65	0.94	0.0	0.0
Classifier	0.69	0.94	0.61	0.87
ProofVER	0.68	0.96	0.61	0.76
Lexicon/KBs only	0.69	0.96	0.60	**0.96**
Ours	**0.71**	**0.96**	**0.70**	0.93
Ours w/ Oracle Merger	0.74	0.97	1.00	1.00

Table 6: Document retrieval scores, R: Recall@5, EM: Exact Match. We further report scores on insufficiency prediction in terms of recall and precision.

7 Human Evaluation of Sufficiency Proofs

A key advantage of AdMIRaL is the added interpretability of our multi-hop retriever through our proof-based sufficiency prediction. For instance, after an initial retrieval hop our model might predict evidence insufficiency, with the indicated span that information is missing for. If the model is not able to find the relevant information in the next hop, a user could consider a targeted modification of the claim based on the sufficiency proof to enable the model to follow a different retrieval path. Conversely, the model might erroneously indicate evidence sufficiency, so the user can supersede the model's decision in an informed manner.

To explore the interpretability of our sufficiency proofs, we conduct a forward prediction experiment (Doshi-Velez and Kim, 2017). Human subjects are asked to predict whether AdMIRaL considers the evidence for a given claim to be sufficient or not, by using the generated sufficiency proof. The comparison baseline provides only the evidence sentences instead. Since we are evaluating the proof as an explanation mechanism to humans, we ensured that no subject was familiar with the deterministic nature of our approach. To enable non-experts to make use of the proof, we replaced the NatOps with English phrases, similar to (Krishna et al., 2022) (see Appendix A.4).

The evaluation consists of 60 annotations from 6 subjects. Ten claims, each paired with an AdMIRaL NatLog proof and baseline explanation are annotated by three subjects. No subject annotates the same claim for both AdMIRaL and baseline explanation, as otherwise a subject might be influenced by the explanation it has seen before for the same claim. Using the sufficiency proofs, subjects correctly predict the model's decision in 70% of cases, compared to the baseline's 50%. The inter-annotator agreement for both AdMIRaL's and baseline's explanation is 0.80 Fleiss κ (Fleiss, 1971). Moreover, annotators predict a system's behavior using AdMIRaL's explanation 20% faster than with the baseline, taking an average time of 51 seconds, reduced to 24 seconds after the first 5 annotations.

8 Conclusion

This paper explored an auto-regressive Retrieval and Rerank model for multi-hop document retrieval that is guided by a proof system based on natural logic that dynamically terminates the retrieval process if the retrieved evidence is deemed sufficient. Our model does only cause minimal memory footprint compared to current state-of-the-art retrieval models while achieving competitive retrieval recall and F_1. Human evaluation indicates that the generated proof as a sufficiency condition is interpretable, enabling a human-in-the loop in the model's retrieval process. Future work aims to investigate to which extent a verification model (i.e. ProofVER) could inform the retrieval of evidence directly, creating an end-to-end closed loop system, as well as human-in-the-loop approaches.

Acknowledgements

This work was supported by the Engineering and Physical Sciences Research Council Doctoral Training Partnership (EPSRC). Andreas Vlachos is supported by the ERC grant AVeriTeC (GA 865958) and the EU H2020 grant MONITIO (GA 965576). We thank Amrith Krishna for giving us access to ProofVER, both him and Nicola De Cao for useful comments and suggestions, and Dominik Stammbach for helping reproducing their multi-hop retriever. Further, the authors would like to thank the 6 subjects who volunteered to be part of the human evaluation, namely Youmna Farag, Zhijiang Guo, Sana Kidwai, Pietro Lesci, Nedjma Ousidhoum, and Andre Schurat, as well as Christopher Bryant and Christoph Hüter for early feedback on the survey. We finally thank the anonymous reviewers for their time and effort giving us feedback on our paper.

Limitations

All benchmarks explored in the paper use Wikipedia as the KB, which is homogeneous compared to heterogeneous sources professional fact-checkers use (e.g. news articles, encyclopedias, scientific documents). Our retrieval methods also focus solely on unstructured evidence in the form of sentences, however, as indicated, recent datasets also consider other modalities. Moreover, the datasets were constructed explicitly using hyperlinks on Wikipedia, thus our approach appears to be particularly suited to these benchmarks. However, we are not aware of a large-scale fact verification dataset that refrains from annotating data that way. Moreover, while natural logic is interpretable, its expressiveness is limited. More complex reasoning e.g. involving time ranges or numbers is not suited for Natural Logic.

Ethics Statement

We anticipate that our retrieval system will be used in fact checking systems. Our retrieval system does not make any judgements about the truth of a statement in the real-world but only consider Wikipedia as the source of evidence to be used as the entire experimental environment has been confined to it. Wikipedia is a great collaborative resource, yet it has mistakes and noise of its own similar to any encyclopedia or knowledge source. Thus we discourage users of using our retrieval system to make absolute statements about the claims being verified, i.e. avoid using it to develop truth-tellers.

References

Rami Aly, Zhijiang Guo, Michael Schlichtkrull, James Thorne, Andreas Vlachos, Christos Christodoulopoulos, Oana Cocarascu, and Arpit Mittal. 2021a. FEVEROUS: Fact Extraction and VERification Over Unstructured and Structured information. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)*.

Rami Aly, Zhijiang Guo, Michael Sejr Schlichtkrull, James Thorne, Andreas Vlachos, Christos Christodoulopoulos, Oana Cocarascu, and Arpit Mittal. 2021b. The fact extraction and VERification over unstructured and structured information (FEVEROUS) shared task. In *Proceedings of the Fourth Workshop on Fact Extraction and VERification (FEVER)*, pages 1–13, Dominican Republic.

Akari Asai, Kazuma Hashimoto, Hannaneh Hajishirzi, Richard Socher, and Caiming Xiong. 2020. Learning to retrieve reasoning paths over wikipedia graph for question answering. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*.

Nicola De Cao, Gautier Izacard, Sebastian Riedel, and Fabio Petroni. 2021. Autoregressive Entity Retrieval. In *International Conference on Learning Representations*.

Finale Doshi-Velez and Been Kim. 2017. Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608*.

Joseph L Fleiss. 1971. Measuring nominal scale agreement among many raters. *Psychological bulletin*, 76(5):378.

Lucas Graves. 2018. Understanding the Promise and Limits of Automated Fact-Checking. Technical report, Reuters Institute, University of Oxford.

Zhijiang Guo, Michael Schlichtkrull, and Andreas Vlachos. 2022. A Survey on Automated Fact-Checking. *Transactions of the Association for Computational Linguistics*, 10:178–206.

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.

Momchil Hardalov, Arnav Arora, Preslav Nakov, and Isabelle Augenstein. 2022. A survey on stance detection for mis- and disinformation identification. In

Findings of the Association for Computational Linguistics: NAACL 2022, pages 1259–1277, Seattle, United States.

Christopher Hidey, Tuhin Chakrabarty, Tariq Alhindi, Siddharth Varia, Kriste Krstovski, Mona Diab, and Smaranda Muresan. 2020. DeSePtion: Dual sequence prediction and adversarial examples for improved fact-checking. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8593–8606, Online. Association for Computational Linguistics.

Kelvin Jiang, Ronak Pradeep, and Jimmy Lin. 2021a. 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.

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.

Yichen Jiang, Shikha Bordia, Zheng Zhong, Charles Dognin, Maneesh Singh, and Mohit Bansal. 2020. HoVer: A dataset for many-hop fact extraction and claim verification. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3441–3460, Online.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online.

Omar Khattab, Christopher Potts, and Matei Zaharia. 2021. Baleen: Robust multi-hop reasoning at scale via condensed retrieval. In *Advances in Neural Information Processing Systems*, online.

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.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. 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.

Shaobo Li, Xiaoguang Li, Lifeng Shang, Xin Jiang, Qun Liu, Chengjie Sun, Zhenzhou Ji, and Bingquan Liu. 2021. Hopretriever: Retrieve hops over wikipedia to answer complex questions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 13279–13287.

Jimmy Lin, Xueguang Ma, Sheng-Chieh Lin, Jheng-Hong Yang, Ronak Pradeep, and Rodrigo Nogueira. 2021. Pyserini: An easy-to-use python toolkit to support replicable ir research with sparse and dense representations. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '21).*, online.

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.

Bill MacCartney and Christopher D Manning. 2014. Natural logic and natural language inference. In *Computing meaning*, pages 129–147. Springer.

Christopher Malon. 2021. Team Papelo at FEVEROUS: Multi-hop evidence pursuit. In *Proceedings of the Fourth Workshop on Fact Extraction and VERification (FEVER)*, pages 40–49, Dominican Republic. Association for Computational Linguistics.

George A. Miller. 1995. Wordnet: A lexical database for english. *Commun. ACM*, 38(11):39–41.

Yixin Nie, Haonan Chen, and Mohit Bansal. 2019. Combining fact extraction and verification with neural semantic matching networks. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, the Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, the Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 6859–6866.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32.

Ellie Pavlick, Pushpendre Rastogi, Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. 2015. PPDB 2.0: Better paraphrase ranking, fine-grained entailment relations, word embeddings, and style classification. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 425–430, Beijing, China.

Fabio Petroni, Aleksandra Piktus, Angela Fan, Patrick Lewis, Majid Yazdani, Nicola De Cao, James Thorne, Yacine Jernite, Vladimir Karpukhin, Jean Maillard, Vassilis Plachouras, Tim Rocktäschel, and Sebastian

Riedel. 2021. KILT: a benchmark for knowledge intensive language tasks. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2523–2544, Online.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. 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)*, pages 2463–2473, Hong Kong, China.

Peng Qi, Haejun Lee, Tg Sido, and Christopher Manning. 2021. Answering open-domain questions of varying reasoning steps from text. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3599–3614, Online and Punta Cana, Dominican Republic.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Michael Roth and Sabine Schulte im Walde. 2014. Combining word patterns and discourse markers for paradigmatic relation classification. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 524–530, Baltimore, Maryland.

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.

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.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. 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*, pages 809–819, New Orleans, Louisiana.

James Thorne, Andreas Vlachos, Oana Cocarascu, Christos Christodoulopoulos, and Arpit Mittal. 2019. The FEVER2.0 shared task. In *Proceedings of the Second Workshop on Fact Extraction and VERification (FEVER)*, pages 1–6, Hong Kong, China.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online.

Wenhan Xiong, Xiang Li, Srini Iyer, Jingfei Du, Patrick Lewis, William Yang Wang, Yashar Mehdad, Scott Yih, Sebastian Riedel, Douwe Kiela, and Barlas Oguz. 2021. Answering complex open-domain questions with multi-hop dense retrieval. In *International Conference on Learning Representations*, online.

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.

A Appendix

A.1 Autoregressive Document Retrieval

Generative Model We use a pre-trained seq2seq model, namely BART (Lewis et al., 2020), which allows the model to capture both surface-level information and semantic aspects between the claim and candidate sentences using cross-attention in its encoder while its decoder is attending over the hidden sequence during generation. The input is structured so that the claim is followed by the sentences, each separated by end-of-sentence tokens: c </s> e_0, </s> ... </s> e_k. Each evidence sentence in E_i is preceded by the corresponding document title in square brackets, e.g. "*[James McBrayer] Jack McBrayer (born May 27... [Tom Bergeron] Tom Bergeron (born May 6, 1955) ...*". For decoding the sentence identifier are generated first, followed by the document identifier. For the dynamic markup decoding we use square brackets to swap between search spaces while each sentence identifier is separated by a space: $e_{p1}e_{p2}[d_p]$. The input to the proof sufficiency module is formatted according to the requirements of ProofVER. Thy dynamic markup of the proof's output is structured by using braces to surround a claim span, square brackets to cover the evidence span, and the token after the closing square bracket to be the natural logic operator.

Training We train a separate model for each hop t using maximum likelihood estimation, following the Neural Machine Translation fine-tuning of BART (Lewis et al., 2020). Given an ordered list of gold evidence sentences E_g, we generate training data by considering only samples with gold evidence sentences equal to the number of hops t.

	Model/Datasets	FEVER		FEVEROUS-S		HoVer	
		2-hop	Overall	2-hop	Overall	2-hop	Overall
Single-Hop	BM25	0.101	0.367	0.327	0.548	0.385	0.141
	GENRE	0.195	0.609	0.330	0.0.367	0.396	0.152
	AdMIRaL-1hop	0.359	0.643	0.456	0.612	0.653	0.289
Multi-Hop	Hyperlinks	0.412	<u>0.647</u>	<u>0.488</u>	<u>0.617</u>	<u>0.711</u>	<u>0.441</u>
	MDR†	<u>0.550</u>	–	–	–	–	–
	AdMIRaL (Ours)	**0.596**	**0.667**	**0.579**	**0.634**	**0.783**	**0.559**

Table 7: F_1 document retrieval scores for 2-hop, and overall scores. To compare with previous work on HoVer, we report recall@100 for *supported* claims on dev. † indicate results taken from Xiong et al. (2021). Results from Khattab et al. (2021) excluded as computation of F_1 in HoVer is unclear and script not accessible. **Bold** numbers indicate best and <u>underline</u> the second-best score.

Since the model is trained as a pointwise reranker, we keep the top $t-1$ gold evidence sentences in the input, i.e. in E_t. and the output is subsequently comprised of a single sequence $p \in P_t$. We generate p as the output label by concatenating the sentence ids of the $t-1$ gold evidence sentences with the document title the remaining gold sentence is representing. Since we consider the top l sentences during inference, with $l \geq t$, we further sample $l-t$ negative sentences from the document candidates D_t and add them to the input. We further shuffle the input randomly, forcing the model to learn to attend to all input sentences. For instance, given the example in Figure 1, the training data for hop $t = 2$ would contain as input the claim, "*Seth Meyers hosted the ceremony*", and negative samples, and the output label could be *E2 [Seth Meyers]*, with 2 varying depending on its position in the input. In cases where an explicit ordering of evidence sentences is not available, we generate all $t-1$ possible ways of keeping $t-1$ gold sentences in the input and using the other as the output.

A.2 Implementation Details

All models are implemented using PyTorch (Paszke et al., 2019). The autoregressive model for both retrieval and proof generation are based on the `Huggingface` (Wolf et al., 2020) implementation of BART (Lewis et al., 2020) and GENRE (De Cao et al., 2021). For all experiments we use a beam size of 25 for the autoregressive generation, and a beam size of 5 for the generation of the sufficiency proof. We used default hyperparameters of BART on all experiments. In case D_t contains less documents than considered by the metric (e.g. recall@5 but number of documents $k < 5$) we add additional documents from D_{t-1}. All experiments were run on a machine with a *single* Quadro RTX 8000 and 64GB RAM memory. Krishna et al. (2022) kindly provided us access to their ProofVER model. For BM25 we set $k1 = 0.6$ and $b = 0.4$, following recommendations of Pyserini.

A.3 Further Results

Table 7 shows results F_1 scores on FEVER, FEVEROUS-S, and HoVer.

A.4 Human Evaluation

All subjects in the human evaluation are undergraduate/graduate/postgraduates students in either computer science or linguistics. 4 subjects are male, 2 female. None of the subjects had prior knowledge on natural language inference.

NatOP	Paraphrase
$\equiv$	Equivalent Spans
$\neg$	Evidence span refutes claim span
$\downarrow\uparrow$	Evidence span contradicts the claim span
#	Unrelated claim span and evidence span

Table 8: NatOPs and their corresponding paraphrases.