Multi-hop Evidence Retrieval for Cross-document Relation Extraction

Keming Lu, **I-Hung Hsu,** **Wenxuan Zhou,** **Mingyu Derek Ma** and **Muhao Chen**
University of Southern California
University of California, Los Angeles
{keminglu,ihunghsu,zhouwenx,muhaoche}@usc.edu; ma@cs.ucla.edu

Abstract

Relation Extraction (RE) has been extended to cross-document scenarios because many relations are not simply described in a single document. This inevitably brings the challenge of efficient open-space evidence retrieval to support the inference of cross-document relations, along with the challenge of multi-hop reasoning on top of entities and evidence scattered in an open set of documents. To combat these challenges, we propose MR.COD (Multi-hop evidence retrieval for Cross-document relation extraction), which is a multi-hop evidence retrieval method based on evidence path mining and ranking. We explore multiple variants of retrievers to show evidence retrieval is essential in cross-document RE. We also propose a contextual dense retriever for this setting. Experiments on CodRED show that evidence retrieval with MR.COD effectively acquires cross-document evidence and boosts end-to-end RE performance in both closed and open settings.[1]

1 Introduction

Relation extraction (RE) is a fundamental task of information extraction (Han et al., 2020) that seeks to identify the relation of entities described according to some context. It is a key task integral to natural language understanding (Liu et al., 2018; Zhao et al., 2020) for inducing the structural perception of unstructured text. Furthermore, it is also an essential step of automated knowledge base construction (Niu et al., 2012; Subasic et al., 2019) and is the backbone of nearly all knowledge-driven AI tasks (Yasunaga et al., 2021; Hao et al., 2017; Lin et al., 2019; Fung et al., 2021; Peters et al., 2019).

Previous works have limited the context of RE within a single sentence (Zhang et al., 2017; Hsu et al., 2022; Zhou and Chen, 2022), a bag of sentences (Zeng et al., 2015; Hsu et al., 2021; Zhu et al., 2020), or a single document (Yao et al., 2019; Zhou et al., 2021; Tan et al., 2022). However, more relations can only be established if multiple documents are considered. For example, more than 57.6% of the relation facts in Wikidata (Erxleben et al., 2014) are not described in individual Wikipedia documents (Yao et al., 2021). In addition, humans also consolidate different steps of a complex event by referring to multiple articles, such as inferring the process of an event from multiple news articles (Naughton et al., 2006) or instructional events (Zhang et al., 2020). To facilitate research in cross-document RE, Yao et al. (2021) constructed the first human-annotated cross-document RE dataset, CodRED, to serve as the starting point of this realistic problem.

Unlike sentence- or document-level RE tasks,

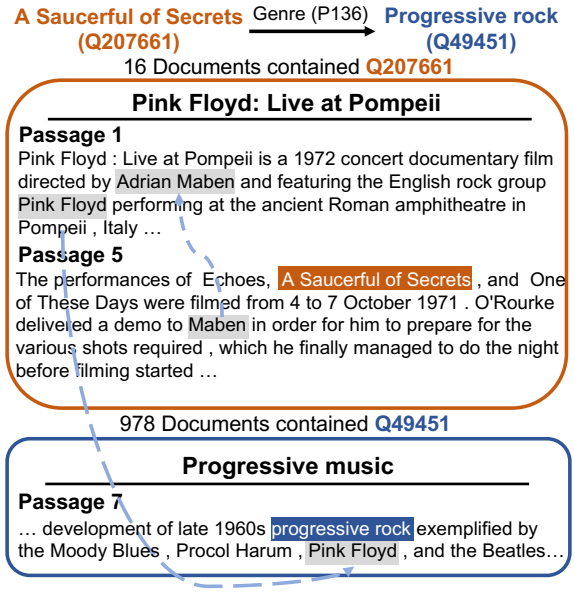

Figure 1: A case for cross-document multi-hop relation reasoning in CodRED. This figure shows a 3-hop evidence path for the triplet ("A Saucerful of Secrets", "Genre", "Progressive rock"), which consists of three passages scattered across two documents from Wikipedia. Arrows in the figure show the bridging entities that link the passages in this evidence path.

[1]Our code is public available at the Github repository: https://github.com/luka-group/MrCoD

Findings of the Association for Computational Linguistics: ACL 2023, pages 10336–10351
July 9-14, 2023

cross-document RE takes a large corpus of documents as input and poses unique challenges (Yao et al., 2021). First, because inferring relations based on the whole corpus is inefficient and impractical, evidence retrieval, which involves extracting evidential context from a large corpus, is crucial for cross-document RE. Second, relations in cross-document RE are usually described by multi-hop reasoning chains with bridging entities. Fig. 1 shows an example of a 3-hop evidence path in CodRED, which spans three related passages in two documents. In this example, the relation between 'A Saucerful of Secrets" and "progressive rock" is described by a reasoning chain containing four bridging entities (marked in grey). On average, cross-document multi-hop reasoning through 4.7 bridging entities is needed in CodRED to infer relations (Yao et al., 2021). Besides, previous work (Zeng et al., 2020; Xu et al., 2021) has shown that intra-document multi-hop reasoning is effective for document-level RE. Therefore, evidence retrieval needs to consider the bridging entities for multi-hop reasoning in cross-document RE.

Against these challenges, we propose a dedicated solution MR.COD (Multi-hop evidence retrieval for Cross-document relation extraction), which extracts evidence from a large corpus by multi-hop dense retrieval. As illustrated in Fig. 2, MR.COD is composed of two steps: evidence path mining and evidence path ranking. In evidence path mining, we first construct a multi-document passage graph, where passages are linked by edges corresponding to shared entities. Then, we use a graph traversal algorithm to mine passage paths from head to tail entities. This step greatly reduces the size of candidate evidential passages. In evidence path ranking, we rank the mined paths based on their relevance. We explore different dense retrievers widely used in open-domain question answering (ODQA) as scorers for evidence paths and further propose a contextual dense retriever better suited for multi-hop relation inference. Finally, the top-K evidence paths are selected as input for downstream relation extraction models. MR.COD is flexible and can be used with any models designed for long-context RE.

The contributions of this work are two-fold. First, we propose a multi-hop evidence retrieval method for cross-document RE and show that high-quality evidence retrieval benefits end-to-end RE performance. Second, we explore multiple widely-used retrievers in our setting and further develop a contextual dense retriever for multi-hop reasoning. Our contributions are verified in both closed and open settings of CodRED (Yao et al., 2021). We observe that MR.COD outperforms other evidence retrieval baselines and boosts end-to-end RE performance with various downstream RE methods.

2 Related Works

We discuss two topics of research that are closely relevant to this study.

Relation Extraction. Recent studies on RE are typically based on datasets with sentence-level (Zhang et al., 2017; Hendrickx et al., 2010) or document-level (Yao et al., 2019; Walker et al., 2006) context. Except for manually annotated datasets, another part of RE focuses on distantly labeled datasets (Riedel et al., 2010), which takes a bag of sentences mentioning the same entity pair as input (Mintz et al., 2009). However, context lengths in these datasets are considerably smaller than that of cross-document RE, an understudied setting that we focus on. Yao et al. (2021) first proposed the cross-document RE task and provided a manually constructed dataset CodRED. Wang et al. (2022) proposed entity-centric snippet selection and cross-path relation attention to enhance performance in cross-document RE. Nevertheless, this work targets the closed setting where evidential context has been given instead of the more challenging and realistic open setting we investigate in this paper.

Evidence retrieval for RE. Evidence retrieval has shown to be effective in document-level RE. Huang et al. (2021b) proposed a simple heuristic method to select evidence sentences from documents. Huang et al. (2021a) used evidence as auxiliary supervision to guide the model in finetuning. Xie et al. (2022) developed a lightweight model to extract evidence sentences. However, these works limit evidence retrieval to a single document and are infeasible to scale up to the cross-document setting. In cross-document RE, Yao et al. (2021) proposes a heuristic simple way to extract evidence, which selects text paths based on the occurrence count of head and tail entities and selects snippets around entity mentions as evidence. Wang et al. (2022) enhances evidence selection in the closed setting as entity-centric snippet selection. However, neither method considers multi-hop reasoning, which is vital for cross-document RE.

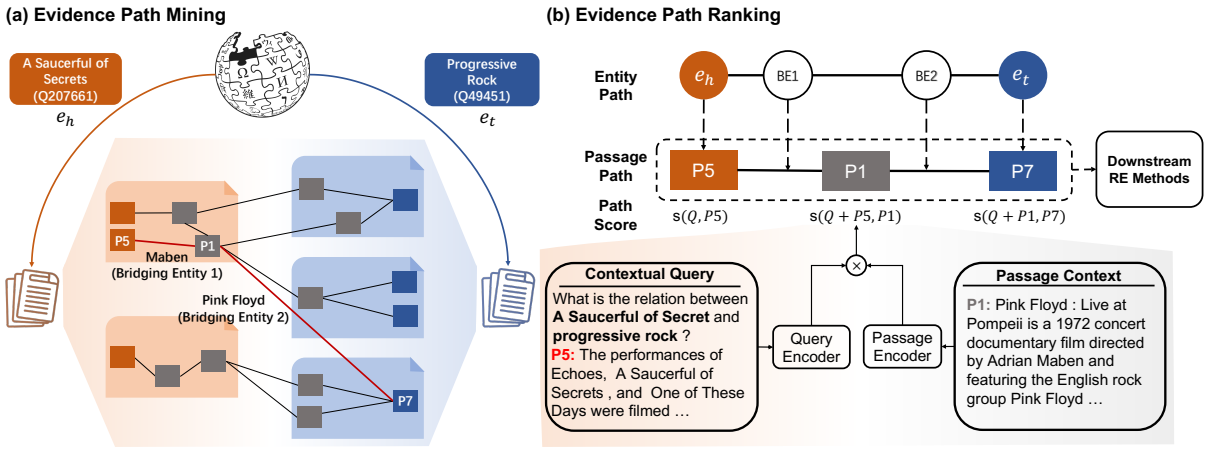

Figure 2: Overview of the evidence retrieval method MR.COD. Given head and tail entities *A Saucerful of Secrets* and *Progressive Rock*, MR.COD extracts documents with mentions of them and then builds a multi-document passage graph as shown in subfigure (a). A 3-hop candidate evidence path is marked in red, linked by two bridging entities, *Maben* and *Pink Floyd*. This candidate evidence path is then scored by a contextual dense retriever, as shown in subfigure (b). The sequential scoring process takes a contextual query and the next hop passage as input. Evidence paths ranked as top-K will be further adjusted in length and used as evidence for downstream RE methods.

Dense Retrieval. Dense retrieval is a fast-developing research topic summarized adequately by the latest surveys, Zhao et al. (2022) and Zhu et al. (2021). Therefore, we only provide a highly selected review. Karpukhin et al. (2020) proposed a dense passage retriever (DPR) with bi-encoder encoding queries and contexts separately. Lee et al. (2021) extended DPR to phrase retrieval and conducts passage retrieval based on phrases. However, these two methods do not directly support multi-hop retrieval. Xiong et al. (2021a) proposed a multi-hop retriever with a shared encoder and query-augmented methods. Nevertheless, its experiments are constrained to two-hop reasoning, although it can theoretically be extended to more hops. Besides, generative retrieval methods can potentially serve as retrievers in our method. We leave this direction as a feature study. However, dense retrievers designed for open-domain question answering (ODQA) can not directly apply to evidence retrieval in cross-document RE because the semantics of queries in this setting are much sparser. In this work, we adapt representative dense retrieval methods and further develop a variant of DPR specifically for evidence retrieval in cross-document RE.

3 Methods

In this section, we describe MR.COD, a multi-hop evidence retrieval method for cross-document RE. We will introduce the preliminaries (§3.1), proposed evidence retrieval (§3.2 - §3.4), and downstream RE methods we explored (§3.5).

3.1 Preliminaries

Problem Definitions. The input of cross-document RE consists of a head entity e_h, a tail entity e_t, and a corpus of documents. The documents are annotated with mentions of entities. Cross-document RE aims to infer the relation r between the head and tail entities from a candidate relation set $\mathbf{R}$. Following Yao et al. (2021), cross-document RE has two settings. In the **closed setting**, only the related documents are provided to the models, and the relations are inferred from the provided documents. While in the more challenging and realistic **open setting**, the whole corpus of documents is provided, and the model needs to efficiently and precisely retrieve related evidence from the corpus. We conduct experiments in both settings.

Method Overview. We divide cross-document RE into two phases: evidence retrieval and relation inference. **Evidence retrieval** aims to retrieve evidential context that is short enough to meet the input length constraint of downstream RE models while providing sufficient information for relation inference. **Relation inference** determines the relations between pairs of entities based on the retrieved evidence. Our main contribution is a multi-hop evidence retrieval method consisting of a graph-based evidence path mining algorithm and a path ranking method with multi-hop dense retriev-

ers as scorers. Fig. 2 shows our proposal in detail with the same input example in Fig. 1. We assume that the evidence is represented as an **evidence path**, i.e., a chain of passages linked by bridging entities. This evidence path begins with passages containing head entities (i.e., head passages) and ends with passages containing tail entities (i.e., tail passages). This assumption is also widely adopted in multi-hop reasoning in ODQA (Feldman and El-Yaniv, 2019; Feng et al., 2022). Subfigure (b) in Fig. 2 displays an evidence path with four passages linked by three entities. We build a multi-document passage graph and run the graph traversal algorithm to find candidate evidence paths, shown in the subfigure (a) in Fig. 2 (§3.2). We rank all candidate evidence paths using multi-hop dense retrievers as scorers, shown in subfigure (b) in Fig. 2 (§3.3). The top-K evidence paths are selected and further prepared as input of downstream RE models (§3.4). Our evidence retrieval method is agnostic to downstream RE models. Therefore, we adopt previous RE models for relation inference (§3.5).

3.2 Evidence Path Mining

Evidence path mining aims to efficiently extract multi-hop evidence paths that align with our assumptions from an open set of documents.

For head-to-head comparison, we follow the text path assumption in Yao et al. (2021), i.e., a candidate evidence path can only go across two documents containing head and tail entities, respectively. Therefore, we only keep documents containing at least one mention of head and tail entities to build a multi-document passage graph consisting of three types of nodes: head passages, tail passages, and other passages that do not contain head or tail mentions. If two passages share mentions of one entity, we create an edge marked for this entity between them. There may be multiple edges between two passages if they share multiple entities. Given this graph as an input, our algorithm finds all paths from head passages to tail passages as evidence paths with the graph traversal method.

Specifically, we employ depth-first search, an efficient unsupervised path mining algorithm, to find evidence paths on the previously obtained passage graph. Graph traversal begins at a head passage and ends at a tail passage. The detailed algorithm is described in Alg. 1 in Appx. §A. To eliminate repetition and ensure that an evidence path is mined from a text path as the same setting by Yao et al. (2021), we enforce several additional constraints:

- An evidence path should not contain head or tail entities in the middle of the path.
- There should be no repeated passages or repeated bridging entities in the path.
- Max lengths of paths should be less than a predefined length H.

These constraints encourage our algorithm to prioritize shorter paths. They also help to improve efficiency and mine more meaningful evidence paths, as two entities are more directly related within a shorter evidence path. A probing analysis in Tab. 7 shows that most of the evidence can be recalled by paths with less than five hops. This insight is also exacerbated in other works focusing on multi-hop reasoning (Yang et al., 2018; Xiong et al., 2021b).

3.3 Adapting Retrievers for Path Ranking

We adapt dense retrievers to rank evidence paths and develop a contextual variant to overcome sparse query semantics for evidence retrieval.

3.3.1 Dense passage retriever (DPR)

DPR is a bi-encoder model first developed as the retrieval component in ODQA. It uses a dense encoder $E_P(\cdot)$ to map text passages into offline low-dimension vector indices. During runtime, it uses another query encoder $E_Q(\cdot)$ to retrieve the top-K most similar passages based on maximum inner product search (MIPS):

$$\text{sim}(q, p) = E_Q(q)^T E_P(p). \quad (1)$$

Two encoders are independent BERT models (Devlin et al., 2019), and the representations of `[CLS]` tokens are used to represent the query or passage.

DPR is trained with a contrastive loss. Let $T = \{\langle q_i, p_i^+, \{p_{i,j}^-\}_{j=1}^n \rangle\}_{i=1}^N$ be the training corpus where q, p^+, p^-, n, N are queries, positive, negative passages, number of negative passages and samples, the loss function is formulated as:

$$l(T_i) = -\log \frac{e^{\text{sim}(q_i, p_i^+)}}{e^{\text{sim}(q_i, p_i^+)} + \sum_{j=1}^n e^{\text{sim}(q_i, p_{i,j}^-)}}.$$

The negative passages can be other passages in the same batch or hard ones mined using BM25.

3.3.2 Adapting DPR in Cross-document RE

We employ a dense retriever to measure similarities between entity pairs and potential evidence

passages. However, although DPR is widely used and proven effective in ODQA, using it directly for cross-document RE has two limitations. First, queries in ODQA are richer in semantics, while queries in cross-document RE only focus on identifying the relations between head and tail entities. Accordingly, we use the following simple template to transform entity pairs into semantic queries:

> What is the relation between **Head Entity** and **Tail Entity**?

Second, DPR does not consider more than one positive passage, while evidence paths in cross-document RE consist of multiple passages. To address this issue, we extend the training corpus of DPR to multiple positive scenarios, where $T = \{\langle q_i, \{p^+_{i,k}\}^m_{k=1}, \{p^-_{i,j}\}^n_{j=1}\rangle\}^N_{i=1}$ and m denotes to the number of positive evidence, and the loss function is formulated as:

$$l(T_i) = -\sum_{k=1}^{m} \log \frac{e^{\text{sim}(q_i, p^+_{i,k})}}{e^{\text{sim}(q_i, p^+_{i,k})} + \sum_{j=1}^{n} e^{\text{sim}(q_i, p^-_{i,j})}}. \quad (2)$$

As for inference, we use DPR as a scorer to rank evidence paths. For an evidence path $P = [p_i]^H_{i=1}$ and a query $q(e_h, e_t)$, the ranking score is calculated as the average similarity between the query and all passages:

$$s(q(e_h, e_t), P) = \frac{1}{|P|} \sum_{p_i \in P} \text{sim}(q(e_h, e_t), p_i) \quad (3)$$

3.3.3 Contextual DPR

Although DPR can be adapted as a scorer for ranking evidence paths in cross-document RE, it is not specifically designed for multi-hop retrieval. Besides, queries in our setting are significantly briefer than those in ODQA and may not be sufficient for retrieving relevant passages. Therefore, we develop a contextual DPR as a multi-hop retriever.

Training. To enrich the semantics of queries and enable multi-hop reasoning, we augment the original training corpus by data augmentation, where we concatenate the original queries and positive evidence to form new queries. Specifically, for training data $T = \{\langle q(e_h, e_t), \{p^+_{i,k}\}^m_{k=1}, \{p^-_{i,j}\}^n_{j=1}\rangle\}^N_{i=1}$ where N is the size of data, we augment the query with one of the positive passage. Therefore, the augmented sample is $T' = T \cup \{\langle q(e_h, e_t) \oplus p_{i,l}, \{p^+_{i,k}\}^m_{k=1,k\neq l}, \{p^-_{i,j}\}^n_{j=1}\rangle\}^m_{l=1}$, where $\oplus$ denotes the string concatenation. We follow the same negative sampling strategy in the original DPR and train this contextual variant with the same loss function as Eq. 2.

Inference. We conduct a sequential scoring process with contextual DPR as the scorer for evidence path ranking. Denoting $P = [p_i]^H_{i=1}$ as an evidence path consisting of H passages, this scoring function calculates sequential similarities between augmented queries and the next hop of passage:

$$s(q(e_h, e_t), P) = \frac{1}{|P|}(\text{sim}(q(e_h, e_t), p_1) + \sum_{i=2}^{|P|} \text{sim}(q(e_h, e_t) \oplus p_{i-1}, p_i)). \quad (4)$$

This sequential scoring function Eq. 4 can take advantage of additional context in the query. Furthermore, embeddings of all augmented queries can be calculated offline to ensure efficiency.

3.4 Input Preparation

To make the evidence path suitable as input for a downstream RE model with maximum input sequence length L, we need to transform it into a token sequence that fits within L. If the length of all evidence exceeds L, we iteratively drop sentences containing the least number of mentions until the total length fits in L while avoiding dropping sentences containing mentions of head or tail entities. If the length of all evidence is smaller than L, we augment each passage in the evidence path by evenly adding more tokens from the preceding and succeeding snippets until the total length meets L, which is the same strategy adopted in Yao et al. (2021). After this length adjustment, all passages in the evidence path are concatenated in order as input for downstream RE methods.

3.5 Downstream RE Methods

Downstream RE methods are the last component in cross-document RE, which takes head and tail entities and evidence context extracted by previous steps as input and conducts relation inference. As our method focuses on evidence retrieval that is agnostic to the RE methods, we use the same RE methods in the previous cross-document RE benchmarks for head-to-head comparison, which are described in detail in §4.1.

4 Experiments

This section presents an experimental evaluation of MR.COD for evidence retrieval and end-to-end RE performance. We introduce the experimental setup (§4.1), main results (§4.2), and ablation study on incorporated techniques (§4.3).

4.1 Experimental Setup

Datasets. We conduct our experiments on the cross-document RE benchmark CodRED (Yao et al., 2021) built from the English Wikipedia and Wikidata. CodRED contains 5,193,458 passages from 258,079 documents with mention annotations of 11,971 entities. There are 4,755 positive relational facts annotated for 276 relations and 25,749 NA (Not Available) relational facts. We experiment in both the closed and open settings discussed in §3.1. In the closed setting, the context is organized in text paths, i.e., a pair of documents containing head and tail entities. A relational fact corresponds to multiple text paths. In the open setting, the context is a subset of Wikipedia documents. A subset of CodRED also has human-annotated sentence-level evidence annotations, which can be used as fine-tuning data for evidence retrieval. More detailed statistics can be found in Tab. 6.

Metrics. We report the F1 score and the area under the precision and recall curve (AUC) as end-task RE metrics which are the same as Yao et al. (2021). Scores on the test set are obtained from the official CodaLab competition of CodRED. We report path- and passage-level recall to evaluate the evidence retrieval component. The path-level recall is the proportion of evidence paths fully extracted by methods, while passage-level recall only considers the proportion of evidence passages recalled by methods. The path-level recall is more strict but has closer correlations to downstream RE performance. To further investigate the different performances of evidence retrieval with short and long paths, we provide recall among evidence paths with ≤ 3 and > 3 hops, respectively.

End-to-end Baselines. We compare MR.COD with the retrieval baseline **Snippet** proposed by Yao et al. (2021). This method first retrieves text paths based on the rank of counts of head and tail entities, then extracts 256-token snippets around the first head and tail mentions in the text path as evidence for downstream RE methods. Wang et al. (2022) proposes a bridging entity-centric method **BridgingCount**, which first retrieves sentences with a count of bridging entities and then reorders them with SentenceBERT (Reimers and Gurevych, 2020). However, this method only works for the close setting, so we only compare end-to-end RE performance in the close setting with it. We then compare end-to-end RE performance of MR.COD with the Snippets baseline in both closed and open settings based on three RE methods:[2] (1) **EntGraph** (Yao et al., 2021) is a graph-based method that first infers intra-document relations and then aggregates them to obtain cross-document relations. This method is named as Pipeline in Yao et al. (2021). (2) **BERT+ATT** (Yao et al., 2021) uses a BERT encoder and selective attention to encode evidence. This method is named as End2End in Yao et al. (2021). (3) **BERT+CrossATT** (Wang et al., 2022) enhances **BERT+ATT** via introducing a cross-path entity relation attention. (4) **Longformer+ATT** is a variant that replaces the BERT encoder in the BERT+ATT baseline with Longformer (Beltagy et al., 2020) so that it can encode two documents at once without evidence retrieval.

Scorer Ablations. We compare the proposed contextual DPR with four scorers in evidence path ranking: (1) **Random** is a baseline randomly selecting top-K evidence paths without ranking. (2) **BM25** (Robertson et al., 1996) is a widely-used sparse information retrieval function based on the bag-of-words model. (3) **DPR** (Karpukhin et al., 2020) is a dense passage retriever that we adapt to evidence retrieval as described in §3.3.2. (4) **MDR** (Xiong et al., 2021a) is a multi-hop dense retriever with query augmentation.

Configurations. We initialize dense retrievers with pretrained checkpoints on ODQA tasks and then finetune them on the evidence dataset of CodRED. We conduct a probing analysis to decide the proper maximum hop number for Alg. 1 and found only 1.6% of the cases required more than four-hop reasoning in the training split of evidence dataset in CodRED. Similar conclusions about maximum hop number are also found in multi-hop ODQA, where two-hop reasoning is set as default (Yang et al., 2018). Therefore, we conduct experiments on both three- and four-hop evidence retrieval to evaluate the effectiveness of our method since a larger hop number will only marginally improve performance but significantly increases the computational com-

[2] We rename methods in Yao et al. (2021) and Wang et al. (2022) to avoid name confusion in this paper.

Method		Closed Setting				Open Setting			
Evidence Retriever	*RE model*	F1 (D)	AUC (D)	F1 (T)	AUC (T)	F1 (D)	AUC (D)	F1 (T)	AUC (T)
/	Longformer+ATT	48.96	45.77	49.94	45.27	44.38	37.04	42.79	37.11
BridgingCount	BERT+CrossATT	61.12 †	60.91†	62.48†	60.67†	−−‡	−−‡	−−‡	−−‡
Snippets	EntGraph	30.54†	17.45†	32.29†	18.94†	26.45†	14.07†	28.70†	16.26†
	BERT+ATT	51.26†	47.94†	51.02†	47.46†	47.23†	40.86†	45.06†	39.05†
	BERT+CrossATT	59.40	55.95	60.92	59.47	51.69	49.59	55.90	51.15
MR.COD	BERT+ATT	57.16	54.43	57.47	53.18	51.83	46.39	52.59	47.08
	BERT+CrossATT	61.20	59.22	62.53	61.68	53.06	51.00	57.88	53.30

† indicates results collected from Yao et al. (2021) and Wang et al. (2022).
‡ BridgingCount is designed for the closed setting and is unable to scale up to the open setting

Table 1: End-to-end RE results of MR.COD with contextual DPR and baselines on dev and test sets of CodRED. We report F1 and AUC in closed and open settings. (D) and (T) refer to the results on dev and test set splits respectively. Results on the test set are obtained by the official competition of CodRED on the CodaLab.

		Path-level Recall†			Passage-level Recall‡		
		All	$H_T < 3$	$H_T \geq 3$	All	$H_T < 3$	$H_T \geq 3$
	Snippets	17.53	28.73	11.22	51.98	60.64	49.72
MR.COD (H=3)	All paths	53.74	−−	−−	−−	−−	−−
	Random	14.07	22.46	9.34	44.47	50.29	42.96
	w/ BM25	22.73	35.48	15.56	51.78	60.55	49.49
	w/ MDR	22.71	**44.52**	8.18	41.57	47.10	40.12
	w/ DPR	23.93	37.30	16.41	53.52	62.35	51.22
	w/ Contextual DPR	**25.88**	39.68	**18.10**	**55.04**	**65.48**	**52.31**
MR.COD (H=4)	All paths	67.79	−−	−−	−−	−−	−−
	Random	13.93	20.32	10.32	44.78	49.23	43.62
	w/ BM25	23.64	36.27	16.54	54.25	62.00	52.22
	w/ MDR	23.80	**45.01**	9.00	43.99	48.23	41.54
	w/ DPR	24.85	37.86	17.52	56.02	64.27	53.87
	w/ Contextual DPR	**27.18**	41.92	**18.53**	**57.12**	**67.73**	**54.02**

† The proportion of fully extracted evidence paths.
‡ The proportion of recalled evidence passages.

Table 2: Evidence retrieval results of MR.COD compared with baselines and different variants on the dev set of the CodRED evidence dataset. H and H_T denote the maximum hop number of path mining and hop number in gold evidence, respectively. The best scores are identified in **bold**, and the second best scores are underlined.

plexity of Alg. 1. For consistency with (Yao et al., 2021), we select the top 16 evidence paths from all paths given by Alg. 1 in the open setting. We use grid search to find optimal hyperparameters in all experiments. Detailed implementation configurations are described in Appx. §D.

4.2 Results

We report end-to-end RE results of MR.COD and evidence retrieval performance in this section.

End-to-end RE. We employ MR.COD with contextual DPR as the scorer when conducting end-to-end RE evaluation according to ablation study in §4.3. Comparison results are shown in Tab. 1. We focus on the *open setting* since it is more realistic and challenging. MR.COD significantly benefits all RE models compared with Snippets on the open setting. For example, MR.COD outperforms Snippets by 7.53% in test F1 and 8.03% in test AUC when using BERT+ATT as the RE model. And MR.COD with BERT+CrossATT achieves the best scores in test F1 (57.88%) and test AUC (53.30%), which improves about 2 percent compared with Snippets. The results illustrate the necessity of multi-hop evidence retrieval for inducing cross-document relations in the open setting.

At the same time, MR.COD also leads to improvements in the *closed setting*, showing that MR.COD helps RE models ping-point relevant evidence in the limited context. The most notable improvement is witnessed in comparison between MR.COD and Snippets with BERT+ATT, where MR.COD improves by 6.45% in test F1 and 5.72% in test AUC. BridgingCount is a retrieval baseline specifically designed for the closed setting, which is slightly outperformed by MR.COD in the

Context	F1	AUC
Gold evidence	52.44	51.37
w/ random augmentation	48.93	46.85
w/ random drop bridging	48.79	46.85
MR.COD evidence	50.61	49.30
w/ input preparation	51.01	49.52
w/ random augmentation	48.19	45.50
w/ random drop bridging	48.93	49.12

Table 3: End-to-end RE results on evidence dev set in CodRED with BERT+ATT. We consider gold and MR.COD evidence as input and develop two variants to show the importance of precise evidence retrieval.

Method		Closed		Open	
Evidence Retriever		F1	AUC	F1	AUC
Snippets		47.23†	40.86†	45.06†	39.05†
H=3	Random	52.12	48.34	48.92	42.13
	w/ BM25	52.41	49.02	49.88	43.31
	w/ MDR	51.40	48.27	49.11	42.43
	w/ DPR	55.08	51.66	51.37	46.51
	w/ Contextual DPR	**57.65**	51.08	52.32	46.56
H=4	Random	51.78	47.09	48.22	41.65
	w/ BM25	52.63	49.21	50.32	44.17
	w/ MDR	50.98	48.01	48.97	42.59
	w/ DPR	53.21	48.75	51.41	45.25
	w/ Contextual DPR	57.47	**53.18**	**52.59**	**47.08**

† indicates results collected from Yao et al. (2021).

Table 4: End-to-end RE results with MR.COD and BERT+ATT based on retriever variants on the test set of CodRED. We report F1 and AUC in closed and open settings. The best scores are identified in **bold**, and the second best scores are underlined.

closed setting with the same RE model. However, MR.COD achieves the highest performance in the open setting while BridgingCount cannot scale up to the open setting. As for baselines, the Longformer+ATT performs worse than BERT+ATT and other language model methods adopting evidence retrievers. These observations show that even in the closed setting, evidence retrieval benefits by targeting the supporting evidence in the given context.

Evidence Retrieval. We also analyze the performance of MR.COD with 3- and 4-hop path mining and multiple scorer variants in Tab. 2. Path-level recalls of the 3- and 4-hop evidence path mining are 53.74% and 67.79%, which indicates the best recall a 3-hop (or 4-hop) evidence path mining model can get without any path filtering. The random baselines are outperformed by MR.COD with retrievers, demonstrating the effectiveness of path ranking. DPR outperforms BM25 and MDR in all settings. Contextual DPR with 4-hop path mining further improves performance by around 2 percent on average compared with the original DPR and surpasses Snippets by 9.65% and 5.14%, contributing to multi-hop evidence retrieval by enriching query context. We also observe recalling evidence paths with more hops is more challenging, while 4-hop path mining consistently improves on longer paths. However, path- and passage-level recalls are not always consistent. For example, MDR performs extraordinarily well in recalling short paths but fails on most of the longer ones.

4.3 Ablation Study

We provide the following analyses to further evaluate core components of MR.COD.

Importance of Precise Evidence Retrieval. We conduct end-to-end evaluations on evidence dev set with BERT+ATT that uses gold and MR.COD evidence as input. We also build variants by randomly augmenting evidence to 512 tokens or dropping bridging evidence. Tab. 3 shows random augmentation and bridging drop decreases end-to-end RE performance with both gold and MR.COD evidence as input, which suggests the importance of precise evidence retrieval. We also found input preparation, a local context augmentation, in MR.COD will not damage performance since it helps with recall.

Effectiveness of Evidence Path Mining. Tab. 7 in Appx. §C shows an analysis of evidence path mining with different maximum hop numbers H. First, the number of passage and entity paths mined by Alg. 1 increases exponentially when H increases, suggesting an exponential complexity on H. However, a small H will be sufficient since the recall of evidence paths increases marginally as $H \geq 3$. The failure rates are also less than 7.9% when $H \geq 3$. In summary, evidence path mining is effective and efficient under these settings.

Scorers. Tab. 4 shows an ablation study of scorer variants in both settings on BERT+ATT. Both 3-hop and 4-hop evidence path mining enhance RE performance even with random selection compared with the Snippets baseline, showing our assumption of evidence paths and path mining can benefit end-to-end RE. We also witness marginal improvements when H increases. Contextual DPR with 4-hop evidence path mining achieves the best performance on most metrics. Comparing Tab. 4 along with Tab. 2, we found evidence retrieval methods with higher recalls tend to perform better in end-

to-end RE, which supports the claim that evidence retrieval is crucial for cross-document RE.

4.4 Case Study

We provide a case of retrieved evidence to demonstrate the interpretability of MR.COD. As the example in Tab. 5, MR.COD correctly finds evidence describing the head entity and bridging entity Columbia Pictures. At the same time, Snippets and BridgingCount fail to retrieve evidence that can form a reasoning chain. This example demonstrates that Mr. Cod can perform evidentially supported and precise RE. We will add a thorough version of this analysis to the final version paper to help readers understand how evidence retrieval works.

5 Conclusion

We study efficient and effective ways to extract multi-hop evidence for cross-document RE and propose MR.COD. MR.COD extracts evidence paths from an open set of documents and ranks them with adapted dense retrievers as scorers. To overcome the gap between retrieval in ODQA and evidence retrieval for RE, we develop a contextual DPR that augments sparse queries with passage context. Extensive experiments show high-quality evidence retrieved by MR.COD boosts end-to-end cross-document RE performance. Future works include extending MR.COD to more retrieval methods, such as generative dense retrievers.

Acknowledgement

We appreciate the reviewers for their insightful comments and suggestions. And we appreciate Yuan Yao and other authors of CodRED dataset (Yao et al., 2021) for updating the dataset and sharing baseline codes. I-Hung Hsu was supported part by the Intelligence Advanced Research Projects Activity (IARPA), via Contract No. 2019-19051600007, and a Cisco Research Award. Wenxuan Zhou and Muhao Chen were supported by the NSF Grant IIS 2105329, the Air Force Research Laboratory under agreement number FA8750-20-2-10002, an Amazon Research Award and a Cisco Research Award. Mingyu Derek Ma was supported by the AFOSR MURI grant #FA9550-22-1-0380, the Defense Advanced Research Project Agency (DARPA) grant #HR00112290103/HR0011260656, and a Cisco Research Award. Computing of this work was partly supported by a subaward of NSF Cloudbank 1925001 through UCSD.

Limitations

Limitations of this work include that we only investigate MR.COD on a set of representative single- and multi-hop dense retrievers. Recent works have proposed more variants of dense retrievers, such as generative retrievers (Lee et al., 2022; Izacard and Grave, 2020) and multi-hop retrievers (Das et al., 2019; Khattab et al., 2021), that can be adapted to use as alternative scorers in MR.COD. Furthermore, we only conduct experiments on three- and four-hop settings. Although this choice is reasonable and supported by various works, we admit that more hops could be needed in real-world application scenarios, which is understudied in this paper.

Ethics Statement

MR.COD is designed for retrieving evidence that supports cross-document RE. However, the evidence retrieved by MR.COD is not always factually correct. This evidence can only be considered as potential context describing facts between given entities.

References

Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.

Rajarshi Das, Ameya Godbole, Dilip Kavarthapu, Zhiyu Gong, Abhishek Singhal, Mo Yu, Xiaoxiao Guo, Tian Gao, Hamed Zamani, Manzil Zaheer, and Andrew McCallum. 2019. Multi-step entity-centric information retrieval for multi-hop question answering. In *Proceedings of the 2nd Workshop on Machine Reading for Question Answering*, pages 113–118, Hong Kong, China. Association for Computational Linguistics.

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

Fredo Erxleben, Michael Günther, Markus Krötzsch, Julian Mendez, and Denny Vrandečić. 2014. Introducing wikidata to the linked data web. In *International semantic web conference*, pages 50–65. Springer.

Method	Evidence
Mr.COD	In the 1934 film, "[One Night of Love](Head Entity)", her first film for [Columbia](Bridging Entity), she portrayed a small-town girl who aspires to sing opera. [Harry Cohn](Tail Entity), president and head of [Columbia Pictures](Bridging Entity), took an 18% ownership in Hanna and Barbera's new company.
Snippets	She was nominated for the Academy Award for Best Actress for her performance in "[One Night of Love](Head Entity)". In 1947, Moore died in a plane crash at the age of 48. [Harry Cohn](Tail Entity), president and head of [Columbia Pictures](Bridging Entity), took an 18% ownership in Hanna and Barbera's new company.
BridgingEnt	In 1947, Moore died in a plane crash at the age of 48. [Harry Cohn](Tail Entity), president and head of [Columbia Pictures](Bridging Entity), took an 18% ownership in Hanna and Barbera's new company. She was nominated for the Academy Award for Best Actress for her performance in "[One Night of Love](Head Entity)".

Table 5: Cases of evidence from different methods.

Yair Feldman and Ran El-Yaniv. 2019. Multi-hop paragraph retrieval for open-domain question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2296–2309, Florence, Italy. Association for Computational Linguistics.

Yue Feng, Zhen Han, Mingming Sun, and Ping Li. 2022. Multi-hop open-domain question answering over structured and unstructured knowledge. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 151–156, Seattle, United States. Association for Computational Linguistics.

Yi R. Fung, Christopher Thomas, Revanth Gangi Reddy, Sandeep Polisetty, Heng Ji, Shih-Fu Chang, Kathleen R. McKeown, Mohit Bansal, and Avi Sil. 2021. Infosurgeon: Cross-media fine-grained information consistency checking for fake news detection. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP*.

Xu Han, Tianyu Gao, Yankai Lin, Hao Peng, Yaoliang Yang, Chaojun Xiao, Zhiyuan Liu, Peng Li, Maosong Sun, and Jie Zhou. 2020. More data, more relations, more context and more openness: A review and outlook for relation extraction. *arXiv preprint arXiv:2004.03186*.

Yanchao Hao, Yuanzhe Zhang, Kang Liu, Shizhu He, Zhanyi Liu, Hua Wu, and Jun Zhao. 2017. An end-to-end model for question answering over knowledge base with cross-attention combining global knowledge. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 221–231, Vancouver, Canada. Association for Computational Linguistics.

Iris Hendrickx, Su Nam Kim, Zornitsa Kozareva, Preslav Nakov, Diarmuid Ó Séaghdha, Sebastian Padó, Marco Pennacchiotti, Lorenza Romano, and Stan Szpakowicz. 2010. SemEval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 33–38, Uppsala, Sweden. Association for Computational Linguistics.

I-Hung Hsu, Xiao Guo, Premkumar Natarajan, and Nanyun Peng. 2021. Discourse-level relation extraction via graph pooling. *arXiv preprint arXiv:2101.00124*.

I-Hung Hsu, Kuan-Hao Huang, Shuning Zhang, Wenxin Cheng, Premkumar Natarajan, Kai-Wei Chang, and Nanyun Peng. 2022. A simple and unified tagging model with priming for relational structure predictions. *arXiv preprint arXiv:2205.12585*.

Kevin Huang, Peng Qi, Guangtao Wang, Tengyu Ma, and Jing Huang. 2021a. Entity and evidence guided document-level relation extraction. In *Proceedings of the 6th Workshop on Representation Learning for NLP (RepL4NLP-2021)*, pages 307–315, Online. Association for Computational Linguistics.

Quzhe Huang, Shengqi Zhu, Yansong Feng, Yuan Ye, Yuxuan Lai, and Dongyan Zhao. 2021b. Three sentences are all you need: Local path enhanced document relation extraction. 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 998–1004, Online. Association for Computational Linguistics.

Gautier Izacard and Edouard Grave. 2020. Leveraging passage retrieval with generative models for open domain question answering. *arXiv preprint arXiv:2007.01282*.

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. Association for Computational Linguistics.

Omar Khattab, Christopher Potts, and Matei Zaharia. 2021. Baleen: Robust multi-hop reasoning at scale via condensed retrieval. *Advances in Neural Information Processing Systems*, 34:27670–27682.

Hyunji Lee, Sohee Yang, Hanseok Oh, and Minjoon Seo. 2022. Generative multi-hop retrieval.

Jinhyuk Lee, Alexander Wettig, and Danqi Chen. 2021. Phrase retrieval learns passage retrieval, too. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3661–3672, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Bill Yuchen Lin, Xinyue Chen, Jamin Chen, and Xiang Ren. 2019. KagNet: Knowledge-aware graph networks for commonsense reasoning. 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 2829–2839, Hong Kong, China. Association for Computational Linguistics.

Shuman Liu, Hongshen Chen, Zhaochun Ren, Yang Feng, Qun Liu, and Dawei Yin. 2018. Knowledge diffusion for neural dialogue generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1489–1498, Melbourne, Australia. Association for Computational Linguistics.

Mike Mintz, Steven Bills, Rion Snow, and Daniel Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 1003–1011, Suntec, Singapore. Association for Computational Linguistics.

Martina Naughton, Nicholas Kushmerick, and Joseph Carthy. 2006. Event extraction from heterogeneous news sources. In *proceedings of the AAAI workshop event extraction and synthesis*, pages 1–6.

Feng Niu, Che Zhang, Christopher Ré, and Jude W Shavlik. 2012. Deepdive: Web-scale knowledge-base construction using statistical learning and inference. *VLDS*, 12:25–28.

Matthew E. Peters, Mark Neumann, Robert Logan, Roy Schwartz, Vidur Joshi, Sameer Singh, and Noah A. Smith. 2019. Knowledge enhanced contextual word representations. 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 43–54, Hong Kong, China. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2020. Making monolingual sentence embeddings multilingual using knowledge distillation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Sebastian Riedel, Limin Yao, and Andrew McCallum. 2010. Modeling relations and their mentions without labeled text. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 148–163. Springer.

SE Robertson, S Walker, MM Beaulieu, M Gatford, and A Payne. 1996. Okapi at trec-4. in proceedings of the 4th text retrieval conference (trec-4): pp. 73-96.

Pero Subasic, Hongfeng Yin, and Xiao Lin. 2019. Building knowledge base through deep learning relation extraction and wikidata. In *AAAI Spring Symposium: Combining Machine Learning with Knowledge Engineering*.

Qingyu Tan, Ruidan He, Lidong Bing, and Hwee Tou Ng. 2022. Document-level relation extraction with adaptive focal loss and knowledge distillation. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1672–1681, Dublin, Ireland. Association for Computational Linguistics.

Christopher Walker, Stephanie Strassel, Julie Medero, and Kazuaki Maeda. 2006. Ace 2005 multilingual training corpus. *Philadelphia: Linguistic Data Consortium*.

Fengqi Wang, Fei Li, Hao Fei, Jingye Li, Shengqiong Wu, Fangfang Su, Wenxuan Shi, Donghong Ji, and Bo Cai. 2022. Entity-centered cross-document relation extraction. *arXiv preprint arXiv:2210.16541*.

Yiqing Xie, Jiaming Shen, Sha Li, Yuning Mao, and Jiawei Han. 2022. Eider: Empowering document-level relation extraction with efficient evidence extraction and inference-stage fusion. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 257–268.

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

Wenhan Xiong, Xiang Lorraine Li, Srinivasan Iyer, Jingfei Du, Patrick Lewis, William Yang Wang, Yashar Mehdad, Wen-tau Yih, Sebastian Riedel, Douwe Kiela, and Barlas Oğuz. 2021b. Answering complex open-domain questions with multi-hop dense retrieval. *International Conference on Learning Representations*.

Wang Xu, Kehai Chen, and Tiejun Zhao. 2021. Discriminative reasoning for document-level relation extraction. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1653–1663, Online. Association for Computational Linguistics.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium. Association for Computational Linguistics.

Yuan Yao, Jiaju Du, Yankai Lin, Peng Li, Zhiyuan Liu, Jie Zhou, and Maosong Sun. 2021. CodRED: A cross-document relation extraction dataset for acquiring knowledge in the wild. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4452–4472, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Yuan Yao, Deming Ye, Peng Li, Xu Han, Yankai Lin, Zhenghao Liu, Zhiyuan Liu, Lixin Huang, Jie Zhou, and Maosong Sun. 2019. DocRED: A large-scale document-level relation extraction dataset. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 764–777, Florence, Italy. Association for Computational Linguistics.

Michihiro Yasunaga, Hongyu Ren, Antoine Bosselut, Percy Liang, and Jure Leskovec. 2021. QA-GNN: Reasoning with language models and knowledge graphs for question answering. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 535–546, Online. Association for Computational Linguistics.

Daojian Zeng, Kang Liu, Yubo Chen, and Jun Zhao. 2015. Distant supervision for relation extraction via piecewise convolutional neural networks. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1753–1762, Lisbon, Portugal. Association for Computational Linguistics.

Shuang Zeng, Runxin Xu, Baobao Chang, and Lei Li. 2020. Double graph based reasoning for document-level relation extraction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1630–1640, Online. Association for Computational Linguistics.

Hongming Zhang, Muhao Chen, Haoyu Wang, Yangqiu Song, and Dan Roth. 2020. Analogous process structure induction for sub-event sequence prediction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1541–1550, Online. Association for Computational Linguistics.

Yuhao Zhang, Victor Zhong, Danqi Chen, Gabor Angeli, and Christopher D. Manning. 2017. Position-aware attention and supervised data improve slot filling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 35–45, Copenhagen, Denmark. Association for Computational Linguistics.

Wayne Xin Zhao, Jing Liu, Ruiyang Ren, and Ji-Rong Wen. 2022. Dense text retrieval based on pretrained language models: A survey. *arXiv preprint arXiv:2211.14876*.

Xueliang Zhao, Wei Wu, Can Xu, Chongyang Tao, Dongyan Zhao, and Rui Yan. 2020. Knowledge-grounded dialogue generation with pre-trained language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3377–3390, Online. Association for Computational Linguistics.

Wenxuan Zhou and Muhao Chen. 2022. An improved baseline for sentence-level relation extraction. In *Proceedings of the Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (AACL)*.

Wenxuan Zhou, Kevin Huang, Tengyu Ma, and Jing Huang. 2021. Document-level relation extraction with adaptive thresholding and localized context pooling. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 14612–14620.

Fengbin Zhu, Wenqiang Lei, Chao Wang, Jianming Zheng, Soujanya Poria, and Tat-Seng Chua. 2021. Retrieving and reading: A comprehensive survey on open-domain question answering. *arXiv preprint arXiv:2101.00774*.

Tong Zhu, Haitao Wang, Junjie Yu, Xiabing Zhou, Wenliang Chen, Wei Zhang, and Min Zhang. 2020. Towards accurate and consistent evaluation: A dataset for distantly-supervised relation extraction. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6436–6447, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Appendices

A Evidence Path Mining Algorithm

The core algorithm of proposed evidence path mining is a depth-first search described in Alg. 1. This is a search algorithm based on backtracking on the multi-document passage graph. In this algorithm, path searching begins with a set of nodes, i.e., passages, with head mentions S_{e_h}. These nodes are pushed into a stack and the search algorithm extends this stack via adding neighbors of them into this stack. The algorithm starts backtracking when the current path finds any passages with tail mentions S_{e_t} or the length of current path meets the maximum hop number H. After the graph traversal, all paths linking S_{e_h} and S_{e_t} with a length less than H will be mined and collected as the candidate evidence paths. This algorithm may fail in some cases. For example, there may not exist a path within H hop between S_{e_h} and S_{e_t}. The proportion of failure in the CodRED dataset is analyzed in Appx. §C. In these cases, we introduce a redemption way described in Appx. §D.3 to collect candidate paths.

Algorithm 1: Depth-first search for evidence path mining

Input: Head entity e_h, Tail entity e_t, Maximum hop number H, A multi-document passage graph $G(N = (d_i, p_j), E = [(d_i, p_m), e_j, (p_k)])$, A set of passages (nodes) with head mentions S_{e_h}, A set of passages (nodes) with tail mentions S_{e_t}

Output: All evidence paths $\mathbf{P} = \{P_l\}_{l=1}^{n_p}$

```
/* Initilize containers                                   */
paths = list()// output
path_entities = list()// briding entities
stack = stack()// backtracking stack
visited = set()// visited passage set
seen_path = dict() // visited passage with a
    specific start node
/* Initialize searching                                   */
stack.append(e_h)
visited.add(e_h)
seen_path[e_h] = list()
path_entities.add(None)
/* Search with backtracking                               */
while length(stack) > 0 do
    start = stack[-1]
    neighbor_nodes = get_neighbor(G, start) if start
      not in seen_path then
        seen_path[start] = list()
    g = 0
    for passage, entity in neighbor_nodes do
        if passage not in visited and w not in
          seen_path[start] then
            g = g + 1
            stack.append(passage)
            visited.add(passage)
            seen_path[start].append(passage)
            path_entities.append(entity)
            if entity in S_{e_t} then
                paths.append(stack)
                latest_pop = stack.pop()
                path_entities.pop()
                visited.remove(latest_pop)
            break
    if g == 0 or length(stack) > H then
        latest_pop = stack.pop()
        path_entities.pop()
        if latest_pop in seen_path then
            del seen_path[latest_pop]
        visited.remove(latest_pop)
return paths
```

B CodRED Dataset

We collect the CodRED dataset from its official Github repository[3], which includes relation triplets, evidence dataset, and documents for closed and open settings. This repo is licensed under the *MIT* license. Furthermore, CodRED dataset contains processed Wikidata and Wikipedia. Wikidata is licensed under the *CC0* license. The text of Wikipedia is licensed under multiple licenses, such as *CC BY-SA* and *GFDL*.

We process the raw data of CodRED as the recipe described in the official Github repository, which includes transforming raw documents and appending them to a Redis server for downstream RE methods. The original evidence is annotated at the sentence-level while we transform them into passage-level by simply annotating passages containing evidence sentences as evidence passages. And we use the same evaluation metric and obtain evaluation scores of the test set from the official CodaLab competition for CodRED.

Tab. 6 shows the statistics of the CodRED dataset. We use two parts of the dataset, including a full dataset and the subset with evidence. We use the evidence subset to finetune our retriever models.

Split	Triplets	Text paths
Train	19,461	129,548
Dev	5,568	40,740
Test	5,535	40,524
Train (Evidence)	3,566	12,013
Dev (Evidence)	1,093	3497

Table 6: Statistics of the CodRED dataset

C Analysis of Evidence Path Mining

Tab. 7 shows evidence path mining analysis results on the CodRED evidence dataset. This analysis focuses on observing recall, failure rate, lengths, and mining speeds of evidence paths along with the increase of maximum hop number.

H	Recall	Fail	P. Path	E. Path	Speed
2	30.4	33.6	2.3	1.1	7358
3	53.7	7.9	32.6	9.3	2735
4	67.8	4.7	390	71	284
5	73.1	4.4	4309	528	26

Table 7: Analysis of evidence path mining algorithm (Alg. 1) on maximum hop number on the CodRED evidence dataset. We report the recall and fail rate (%) of evidence paths, the average number of passage paths (P. Path) and entity paths (E. Path), and processing speed (iter/s).

D Detailed Implementation

We describe the detailed implementation of all components in this work.

[3]CodRED Github repository: https://github.com/thunlp/CodRED

D.1 Retrieval methods

All supervised retrieval methods are trained on the training split of CodRED evidence dataset and tuned hyperparameters on the dev split.

BM25. We use the Python implementation of BM25 algorithm *Rank_BM25*[4]. We first remove all characters that are neither English characters and numbers from all passages. And then we tokenize the passages into bags of words with the *word_tokenize* function in *NLTK*. We further preprocess by removing stopwords collected in *gensim* and then stem all words with the *PorterStemmer* in *NLTK*. The queries are preprocessed in the same way. We use Okapi BM25 to rank the top-K evidence paths for each query and use it as evidence for the next steps. All third-party APIs we used in this section are run with default parameters. The ranking score is calculated as the average BM25 score between the query and each passage in the evidence path.

MDR. We develop our adapted MDR based on the MDR official Github repository.[5] We train the MDR from the public checkpoint shared in the repository, which is trained from roberta-base. We use the same shared encoder setting as the original MDR. The batch sizes for training and inference are 16; the learning rate is 2e-5; the maximum lengths of queries, contexts, and augmented queries are 70,300,350, respectively; The warm-up rate is 0.1. We train this model for 5 epochs on 3 NVIDIA RTX A5000 GPUs for 8 hours. Then we encode all passages with 8 NVIDIA RTX A5000 GPUs. The queries are first generated with augmented passages and encoded offline before evidence retrieval.

DPR. We develop our adapted DPR based on the DPR official Github repository.[6] We train the DPR from the public checkpoint trained on the NQ dataset with the single adversarial hard negative strategy. We follow the configuration *biencoder_local* in the original DPR training configurations: batch size is 4; learning rate is 2e-5; warmup steps are 1237; number of training epochs is 50; maximum gradient norm is 2.0. We run the training on 8 NVIDIA RTX A5000 GPUs for 12 hours.

Contextual DPR. We use the same setting of DPR to train the contextual DPR, except we prolong the training epochs to 70 and reduce the learning rate to 1e-5. We run the training on 8 NVIDIA RTX A5000 GPUs for 12 hours.

D.2 Downstream RE methods

We train downstream RE methods on the training split of CodRED and tune hyperparameters on the dev split. We strictly follow the CodRED recipe so we also introduce path-level and intra-document supervision when training following RE models.

BERT+ATT. We use the same code collected from the official repository[7]. We train BERT+ATT from the sketch on 8 NVIDIA RTX A5000 GPUs for 10 hours. The training and inference batch size is 1; the learning rate is 3e-5; the number of training epochs is 8; The base model is *bert-base-cased* from Huggingface Transformers.

BERT+CrossATT. We use the same code collected from the official repository[8]. We train BERT+CrossATT from the sketch on 4 NVIDIA RTX A5000 GPUs for 20 hours. The training and inference batch size is 1; the learning rate is 3e-5; the number of training epochs is 10; The base model is *bert-base-cased* from Huggingface Transformers.

D.3 Simple redemption of MR.COD

We assume we can find a chain of passages linked by bridging entities. However, this assumption does not always hold for the fixed maximum number of hops. As Tab. 7 shown, the fail rate increases when the maximum hop number decreases. We use a simple redemption that selects passages containing head and tail mentions as an evidence path when evidence path mining fails. The fail rate is considerably low ($\leq 5\%$) when the hop number is more than 3. Therefore, this simple redemption will not affect prove the effectiveness of the evidence path mining algorithm. Besides, a small portion of entities may appear in many documents, leading to low efficiency of Alg. 1. Therefore, we propose another simple redemption that selects top 50 documents based on entity count when an entity appears in more than 50 documents. The text-path selection in (Yao et al., 2021) inspires this redemption.

[4]Rank-BM25 Github Repository: https://github.com/dorianbrown/rank_bm25

[5]MDR Github Repository: https://github.com/facebookresearch/multihop_dense_retrieval

[6]DPR Github Repository: https://github.com/facebookresearch/DPR

[7]BERT+ATT Github Repository:https://github.com/thunlp/CodRED

[8]BERT+CrossATT Github Repository:https://github.com/MakiseKuurisu/ecrim

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
Limitations section at the beginning of page 9

☑ A2. Did you discuss any potential risks of your work?
Ethics Statement section on page 9

☑ A3. Do the abstract and introduction summarize the paper's main claims?
abstract and Introduction section

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☑ Did you use or create scientific artifacts?

Section 4

☑ B1. Did you cite the creators of artifacts you used?
Section 4.1 and Appendix B, and Appendix D

☑ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Appendix B

☑ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
Appendix B and Appendix D

☐ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
Not applicable. Left blank.

☐ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Not applicable. Left blank.

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
Section 4.1 and Appendix B

C ☑ Did you run computational experiments?

Section 4 and 5

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Appendix D

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
Section 4.1 and Appendix D

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
Section 4.2

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
Appendix D

D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

☐ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
No response.

☐ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
No response.

☐ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
No response.

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
No response.

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
No response.