Virtual Knowledge Graph Construction for Zero-Shot Domain-Specific Document Retrieval

Yeon Seonwoo[†*],
Seunghyun Yoon[‡], **Franck Dernoncourt**[‡], **Trung Bui**[‡],
Alice Oh[†]
[†]KAIST, [‡]Adobe Research,
yeon.seonwoo@kaist.ac.kr
{syoon, dernonco, bui}@adobe.com
alice.oh@kaist.edu

Abstract

Domain-specific documents cover terminologies and specialized knowledge. This has been the main challenge of domain-specific document retrieval systems. Previous approaches propose domain-adaptation and transfer learning methods to alleviate this problem. However, these approaches still follow the same document representation method in previous approaches; a document is embedded into a single vector. In this study, we propose VKGDR. VKGDR represents a given corpus into a graph of entities and their relations (known as a virtual knowledge graph) and computes the relevance between queries and documents based on the graph representation. We conduct three experiments 1) domain-specific document retrieval, 2) comparison of our virtual knowledge graph construction method with previous approaches, and 3) ablation study on each component of our virtual knowledge graph. From the results, we see that unsupervised VKGDR outperforms baselines in a zero-shot setting and even outperforms fully-supervised bi-encoder. We also verify that our virtual knowledge graph construction method results in better retrieval performance than previous approaches. [1]

1 Introduction

In domain-specific QA, building retrievers is challenging since queries and documents in a specific domain cover terminologies and specialized knowledge, which are not well covered in general documents. (Zhang et al., 2020; Ma et al., 2021; Yu et al., 2020, 2021). Another problem is the difficulty in building datasets for training retrievers. This problem comes from 1) the complexity of knowledge treated in the documents, and 2) costly dataset maintenance; recall that domain-specific documents are frequently updated (e.g., software manuals are updated whenever there is a version update) (Castelli et al., 2020; Nandy et al., 2021; Voorhees et al., 2021; Maia et al., 2018).

Recent domain-specific document retrieval studies propose domain-adaptation and transfer learning methods (Thakur et al., 2021; Ma et al., 2021; Beltagy et al., 2019; Gururangan et al., 2020; Chalkidis et al., 2020). However, these methods still use the conventional document representation method, embedding a document into a single vector. This is problematic because a single vector is insufficient to cover complex knowledge in a domain-specific document. Semi-structured knowledge representation methods effectively address this problem, but they have only been applied to open-domain documents. (Dhingra et al., 2020; Sun et al., 2021; Zhang et al., 2018; Sun et al., 2018; Yasunaga et al., 2021; Talmor and Berant, 2018).

In this paper, we propose an automatic virtual knowledge graph construction method for zero-shot domain-specific document retrieval. A virtual knowledge graph (VKG) is a graph representation of a corpus that consists of entities and their relations. In VKG, the relations are represented by relation vectors (Dhingra et al., 2020; Sun et al., 2021). This semi-structured representation enables explicit reasoning over the corpus. We apply this framework to domain-specific document retrieval. One of the key components of the VKG construction method is a relation encoder, which computes relation vectors of two entities. This study shows that previous supervision methods for relation encoders are insufficient for domain-specific documents, and we propose a novel distant-supervision method.

We validate VKGDR in three types of experiments. First, we conduct zero-shot domain-specific document retrieval on two domain-specific QA datasets: TechQA (Castelli et al., 2020) and PhotoshopQuiA (Dulceanu et al., 2018). The results show that VKGDR outperforms domain-adaptation and

[*] This work was done during an internship at Adobe Research.

[1] We provide the implementation of VKGDR at https://github.com/yeonsw/VKGDR

Proceedings of the 29th International Conference on Computational Linguistics, pages 1169–1178
October 12–17, 2022.

transfer learning methods. From this experiment, we also verify that unsupervised VKGDR outperforms a fully-supervised dense retriever. Second, we show that our distant-supervision method for training the relation encoder outperforms previous approaches. In this experiment, we construct VKGs with our relation encoders and baselines' encoders. Then, we measure the retrieval performance of each VKG. Third, we conduct an ablation study on two main components of a VKG, graph representation of a corpus and relation vectors. The results show that each component increases the retrieval performance of VKGDR by a large margin.

2 Related Work

A virtual knowledge graph is a graph representation of a corpus that consists of entities and their relations. The relations of entities are represented by relation vectors. Dhingra et al. (2020) propose a differentiable VKG for multi-hop QA. Their VKG is trained by the end-to-end supervision method on question-answer pairs. Sun et al. (2021) use VKG for knowledge graph QA. They apply relation encoders used in relation extraction studies to VKG construction and follow distant-supervision proposed by Soares et al. (2019). Our work provides a novel distant-supervision method for building a virtual knowledge graph for domain-specific documents. In section 5.1, we compare our methods with Sun et al. (2021) to validate the efficacy of our method.

Domain-specific documents cover complex knowledge and require advanced representation methods. Previous approaches in domain-specific document retrieval focus on a document encoder training method and data scarcity problem but still follow conventional document representation methods. Ma et al. (2021); Liang et al. (2020) augment domain-specific question-answer pairs from an external corpus and train their encoders on the dataset. Yu et al. (2020); Zhang et al. (2020) provide a pre-training method on domain-specific documents. We propose a novel virtual knowledge graph construction method and apply our method to domain-specific document retrieval.

3 Method

We propose a novel domain-specific document retrieval method, VKGDR, based on a virtual knowledge graph (VKG). VKGDR consists of two modules: a VKG construction module and a document retrieval module. A VKG is a graph representation of a given corpus that connects mentions with directed edges, and each directed edge has a relation vector. (Dhingra et al., 2020; Sun et al., 2021). The document retrieval module computes the similarity between queries and documents with the mention pairs and their relation vectors. We describe notations and details of each module in the following sections.

3.1 Notations

In this section, we define notations and terms used in our paper and VKG research (Dhingra et al., 2020). VKGDR takes a corpus and outputs a virtual knowledge graph. The corpus, $\mathcal{C}$ is a set of documents; $\mathcal{C} := \{d_1, ..., d_n\}$. A document is defined as a sequence of tokens; $d_k := [d_k^1, ..., d_k^{L_k}]$, where d_k^j is the j'th token of document d_k and L_k is the number of tokens in document d_k. VKGDR's entity extractor builds a set of entities [2], $\mathcal{E}$ and a set of mentions, $\mathcal{M}$. The definition of an entity is a named entity in the corpus, $\mathcal{C}$, and the definition of mention is a text segment in the corpus, $\mathcal{C}$, that corresponds to an entity in $\mathcal{E}$. Formally, the mention is defined as $m_i = \{d_k, a, b, e_j\}$; the mention m_i is a text segment starting from index a and end at index b in document d_k, which corresponds to entity e_j. Figure 1 shows the difference between mentions and entities. In the figure, the highlighted text segments are the mentions, and there are multiple mentions for each entity. For instance, entity “TRC 5011" appears multiple times in this document, and each text segment that refers to entity “TRC5011" is a mention of the entity.

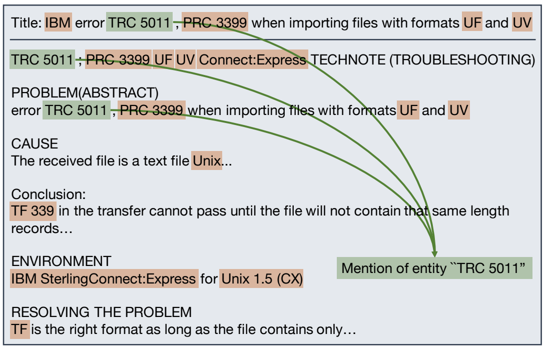
3.2 Virtual Knowledge Graph

A virtual knowledge graph is a directed graph consisting of mentions and their relations. The relations are represented by relation vectors (Dhingra et al., 2020). Formally, we define an edge of the VKG as follows:

$$(m_a, m_b, \vec{r}_j).$$

This represents that there exists an edge directed from mention m_a to m_b, and $\vec{r}_j$ is the relation vector of the mention pair. Mention m_a is called the head, and m_b is called the tail. VKGDR constructs a virtual knowledge graph with the following steps: 1) connecting all relevant mentions and 2) computing relation vectors of edges. In our study, we

[2] We use the NER model provided by spaCy.



Title: IBM error TRC 5011 , PRC 3399 when importing files with formats UF and UV

TRC 5011 ; PRC 3399 UF UV Connect:Express TECHNOTE (TROUBLESHOOTING)

PROBLEM(ABSTRACT)
error TRC 5011 , PRC 3399 when importing files with formats UF and UV

CAUSE
The received file is a text file Unix...

Conclusion:
TF 339 in the transfer cannot pass until the file will not contain that same length records…

ENVIRONMENT
IBM SterlingConnect:Express for Unix 1.5 (CX)

RESOLVING THE PROBLEM
TF is the right format as long as the file contains only…

Figure 1: An example document of the TechQA dataset and mentions in the document. Orange and green highlights are the mentions. Mention is a text segment that refers to a certain entity. For example, entity “TRC 5011” appears multiple times in this document and each text segment that refers to “TRC 5011” is a mention of entity “TRC 5011”.

assume two mentions are relevant if they appeared in the same document and connect the two mentions with directed edges in both directions. Thus, for a given document with n mentions, there are n^2 combinations of mention pairs and n^2 more mentions pairs since we connect mentions with directed edges in both directions.

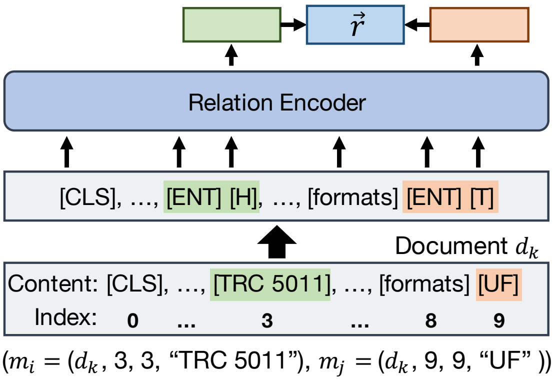
3.3 Relation Embedding

Relation encoders compute relation vectors of mention pairs connected in a virtual knowledge graph. Relation encoders aim to embed mention pairs into a similar vector space if they are in similar relation. Previous approaches distantly-supervise relation encoders since training data is often unavailable. One of the previous approaches assumes that mention pairs referring to the same entity-pair have the same relation (Sun et al., 2021), and they train their relation encoder to maximize the similarity of these similar mention pairs. In this study, we propose a novel distant-supervision method for domain-specific documents.

Model Architecture: VKGDR's relation encoder (RE) takes a mention pair and computes the relation vector.

$$\vec{r}_{i,j} = \mathrm{RE}(m_i = \{d_k, a, b, e_u\}, m_j = \{d_k, c, d, e_v\})$$

In previous relation embedding studies, relation encoders take preprocessed mention pairs as an input. The preprocessing steps are: 1) adding special tokens to the head and the tail mentions to indicate their direction and 2) masking the mentions (Mintz et al., 2009; Soares et al., 2019; Sun et al., 2021). Our relation embedding method is based on the previous approaches and proceeds following steps on mention pairs. For a given mention pair, $(m_i = (d_k, a, b, e_u), m_j = (d_k, c, d, e_v))$, we represent the two mentions in document d_k as follows:



$\vec{r}$
Relation Encoder
[CLS], ..., [ENT] [H], ..., [formats] [ENT] [T]
Document d_k
Content: [CLS], ..., [TRC 5011], ..., [formats] [UF]
Index: 0 ... 3 ... 8 9
(m_i = (d_k, 3, 3, “TRC 5011”), m_j = (d_k, 9, 9, “UF”))

Figure 2: The inference process of the relation encoder of VKGDR. Two mentions are given to the relation encoder. Mention, m_i and m_j are text spans in document d_k located from index 3 to 3 and index 9 to 9. We mask mention tokens with [ENT] and indicate the head and the tail with [H] and [T]. The relation encoder takes this input and computes a relation vector of the two mentions.

$$(m_i, m_j) = [d_k^1, ..., \boldsymbol{e_u}, ..., \boldsymbol{e_v}, ..., d_k^{L_k}].$$

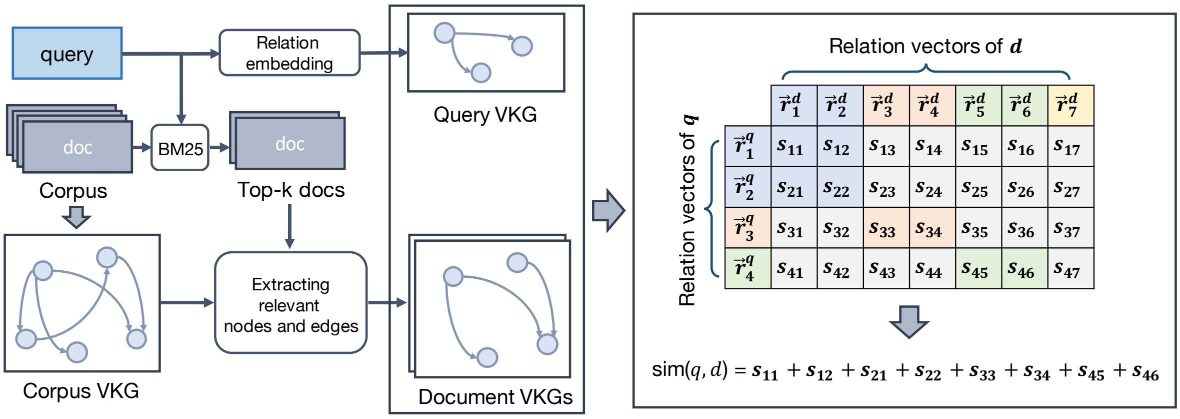
$\mathbf{e_v}$ and $\mathbf{e_u}$ are the tokens in document d_k corresponding to the two mentions. Next, we put special tokens, [H] and [T], to the mentions as follows:

$$[d_k^1, ...\boldsymbol{e_u}, \textbf{[H]}, ..., \boldsymbol{e_v}, \textbf{[T]}, ..., d_k^{L_k}].$$

Now, the above sequence of tokens represents the direction between the two mentions; without the special tokens, the relation encoder predicts the same relation vector for the opposite input, (m_j, m_i). Mention masking enables the relation encoder to compute the relation vector based on the context of the mention pairs, not based on their textual representation. We mask the mentions as follows:

$$[d_k^1, ..., \textbf{[ENT]}, \textbf{[H]}, ..., \textbf{[ENT]}, \textbf{[T]}, ..., d_k^{L_k}].$$

Figure 2 shows the input preprocessing step and the relation vector computation step. In this example, the green token is the head, and the red token is the tail. The head and tail tokens are inserted into the document, and the entities are masked with the special token. The relation encoder takes the whole sequence of tokens and computes contextualized



$$
\begin{array}{c|ccccccc}
 & \vec{r}_1^{\,d} & \vec{r}_2^{\,d} & \vec{r}_3^{\,d} & \vec{r}_4^{\,d} & \vec{r}_5^{\,d} & \vec{r}_6^{\,d} & \vec{r}_7^{\,d} \\
\hline
\vec{r}_1^{\,q} & s_{11} & s_{12} & s_{13} & s_{14} & s_{15} & s_{16} & s_{17} \\
\vec{r}_2^{\,q} & s_{21} & s_{22} & s_{23} & s_{24} & s_{25} & s_{26} & s_{27} \\
\vec{r}_3^{\,q} & s_{31} & s_{32} & s_{33} & s_{34} & s_{35} & s_{36} & s_{37} \\
\vec{r}_4^{\,q} & s_{41} & s_{42} & s_{43} & s_{44} & s_{45} & s_{46} & s_{47}
\end{array}
$$

$$\text{sim}(q,d) = s_{11} + s_{12} + s_{21} + s_{22} + s_{33} + s_{34} + s_{45} + s_{46}$$

Figure 3: The document retrieval process of VKGDR. First, VKGDR pre-indexes the corpus VKG. Second, we extract relevant nodes and edges from the VKG for a given query since comparing the query with the entire VKG is computationally inefficient. Third, VKGDR transforms the query to a query VKG with the same VKG construction method used for the corpus VKG. Finally, we compare the two VKGs and compute their similarity. In the right part of this figure, we use a color-coding method to indicate the relation vectors that share the same entity pairs.

vector representations of the head and tail tokens. Then, we compute the relation vector from the two vectors with an additional MLP layer.

Training Process: This study proposes a novel distant-supervision method for building a virtual knowledge graph from domain-specific documents. Distant-supervision methods in previous relation embedding approaches proceed following steps: 1) heuristically annotate mention pairs in the same relation and 2) train the relation encoder to maximize the similarity between mention pairs in the same relation.

We propose a novel distant-supervision method for domain-specific documents. One of the previous approaches assumes mention pairs are in the same relation if they share the same entity pair (Soares et al., 2019; Sun et al., 2021). In domain-specific QA, mention pairs are often in different relations, and the relation varies depending on the context of the document. With the previous assumption, relation encoders predict similar relation vectors for mention pairs with the same entity pair even they are in different relations. We assume that the context of mention pairs is more important than the entities they refer to. In our approach, mention-pairs are in the same relation if they appeared in the same document. Formally, we train our relation encoder with the following method. For a given mention-pair, $p = (m_i, m_j)$, the positive sample (p^+) is a mention pair appeared in the same document, and the negative sample is mention pair in the different document. The loss function of our relation encoder is as follows:

$$
\begin{aligned}
&L(p, p^+, p_1^-, \ldots, p_{\#\text{neg}}^-) = \\
&\quad -\log\Big(\frac{e^{\text{sim}(p,p^+)}}{e^{\text{sim}(p,p^+)} + \sum_{i=1}^{\#\text{neg}} e^{\text{sim}(p,p_i^-)}}\Big),
\end{aligned} \tag{1}
$$

where sim function is the dot product of the two relation vectors; $\text{sim}(p_1, p_2) = \vec{r}_{p1}^{\top}\vec{r}_{p2}$.

3.4 Document Retrieval Process

VKGDR uses a virtual knowledge graph to find the document most relevant to a given query. The document retrieval process of VKGDR follows four steps: 1) selecting top-k relevant documents with BM25, 2) extracting mention pairs appeared in the top-k documents and their relation vectors from the VKG, 3) constructing a VKG of a given query, and 4) finding the most relevant document by comparing the document VKGs and the VKG of the input query. In the first step, we select top-k documents relevant to the query. This is because our VKG consists of a huge number of mention pairs; comparison between the query and the VKG is computationally costly. Then, we construct document VKGs. A document VKG is a graph of mention pairs that appeared in the document. Since we retrieved top-k documents, we get k number of document VKGs. In the third step, we transform the query to a VKG; queries cannot be directly compared with document VKGs since they are in textual form. In this step, we use the same relation encoder used for computing the VKG of the given corpus. In the last step, VKGDR finds the most relevant document by comparing the query VKG and the document VKGs. We describe the details of this VKG comparison process in the following section. Figure 3 shows an overall illustration of VKGDR.

Comparing two VKGs: The query VKG and the document VKG consist of several mention pairs as follows:

$$\begin{aligned}\mathrm{VKG}_q &= \{(m_h, m_t, \vec{r})_i\}_{i=1}^{k} \\ \mathrm{VKG}_d &= \{(m_h, m_t, \vec{r})_i\}_{i=1}^{n},\end{aligned}$$

where VKG_q is the query VKG and VKG_d is the document VKG. VKGDR computes the similarity between two VKGs with the following equation:

$$\mathrm{similarity}(q, d) = \sum_{\substack{(m_h^q, m_t^q, \vec{r}^{\,q}) \in \mathrm{KTq} \\ (m_h^d, m_t^d, \vec{r}^{\,d}) \in \mathrm{KTd}}} \mathbb{1}((m_h^d, m_t^d) = (m_h^q, m_t^q)) \vec{r}^{\,q\,\intercal} \vec{r}^{\,d},$$

where $\mathbb{1}$ is an indicator function that maps true condition to one and zero for false condition.

4 Experimental Setup

We validate domain-specific document retrieval performance of VKGDR in three experimental settings. In the first experiment, we evaluate VKGDR and baselines in a zero-shot setting. The zero-shot setting emulates the real-world problem of domain-specific document retrieval; training data is insufficient or absent. Additionally, we conduct the same experiment in a fully-supervised setting and show the efficacy of VKGDR. In the second experiment, we verify the efficacy of our proposed distant-supervision method by comparing our method with previous methods. We construct three VKGs with relation encoders trained with three different distant-supervision methods. The third experiment is an ablation study that evaluates each component in VKGDR. A virtual knowledge graph consists of two main components: 1) graph representation of a given corpus and 2) relation vectors of mention pairs. In this experiment, we evaluate 1) VKGDR without graph representation and 2) VKGDR without relation vectors and show the efficacy of each component. All experiments are conducted on two domain-specific QA datasets, TechQA and PhotoshopQuiA, and evaluated with document retrieval metrics, R@K and MRR. We describe details of the datasets and baselines in the following sections and describe evaluation metrics and hyper-parameter settings in Appendix A.1.

	Train	Dev	Test
TechQA	600	310	490
PhotoshopQuiA	2001	571	284

Table 1: The number of instances in the TechQA dataset and the PhotoshopQuiA dataset.

4.1 Datasets

TechQA: TechQA is a question answering dataset in the domain of IT support (Castelli et al., 2020). The questions ask about IBM products and applications running in computational environments supported by IBM. This dataset provides question-answer pairs and 800,000 technical notes that provide descriptions of IBM's products. Each question is annotated with 50 documents retrieved by BM25, and one of the 50 documents is the ground truth document. Thus, the task of this dataset is to find the correct document among the 50 documents. The numbers of question-answer pairs of TechQA is 1,400. Table 1 shows the detailed statistics of the TechQA dataset.

PhotoshopQuiA: PhotoshopQuiA is a non-factoid question-answering dataset on Adobe Photoshop (Dulceanu et al., 2018). The questions and answers are users' questions and answers from several web forums related to Adobe Photoshop. This dataset provides question-answer pairs but not the corpus. So, we have built a corpus with all answer text in this dataset and built question-document pairs as TechQA; each question is annotated with 50 documents retrieved by BM25, and the 50 documents contain the ground truth document. Table 1 shows the detailed statistics of the PhotoshopQuiA dataset.

4.2 Baselines

There are two types of document retrievers: lexical retrievers and dense retrievers. We compare VKGDR with a lexical retriever and three dense retrievers.

Lexical Retriever: We use BM25 as the lexical retriever. BM25 has a better or similar performance than dense retrievers when training data is insufficient and the questions are domain-specific (Thakur et al., 2021). Thus, BM25 is a strong baseline in our problem setting.

Dense retriever: **DPR** is a dense retriever for open-domain QA (Karpukhin et al., 2020). We use

DPR trained on NaturalQuestions (Kwiatkowski et al., 2019), an open-domain QA dataset. Domain-adaptation (**Adapt**) is another approach for training dense retrievers in a zero-shot setting. We pre-train BERT-large (Devlin et al., 2019) on the corpus of each dataset and compare with VKGDR. We use a CLS vector of BERT-large for document representation. The performance of fully-supervised models provides an approximation of the performance of unsupervised models. We train a bi-encoder with the same supervision method used in DPR on the question-answer pairs of each dataset and compare this model (“**DPR***”) with VKGDR. The encoder of DPR* is initialized with RoBERTa-large (Liu et al., 2019).

5 Results

In this section, we verify the efficacy of VKGDR with the experiments described in the previous section. The experimental results demonstrate three findings. First, VKGDR outperforms baselines in a zero-shot setting and a fully-supervised setting. Furthermore, VKGDR without fine-tuning outperforms a fully-supervised bi-encoder. Second, our distant-supervision method for the relation encoder outperforms the previous method. Third, the two main components of VKGDR, graph representation of a corpus and relation vectors, are essential to achieve the zero-shot performance of VKGDR. We describe details of the experimental results in the following sections.

5.1 Zero-Shot Domain-Specific Document Retrieval

Table 2 and Figure 4 show zero-shot domain-specific document retrieval performance of VKGDR and baselines. These experiments support the following findings: 1) constructing a VKG is more effective than transfer learning methods when training data is unavailable, and 2) our distant-supervision method for the relation encoder outperforms the previous method. We describe details of experimental results in the following paragraphs.

Type	Model	S	TechQA		
			R@1	R@5	MRR
L	BM25	✗	43.7	63.7	54.2
D	Adapt	✗	5.0	11.8	12.1
	DPR	✗	16.8	40.6	28.6
	VKGDR	✗	**39.3**	**63.7**	**50.2**
D+L	Adapt	✗	9.3	28.7	34.5
	DPR	✗	28.7	55.6	47.2
	VKGDR	✗	**44.3**	**68.7**	**55.8**
D	DPR*	✓	36.8	73.1	52.3

Table 2: The zero-shot domain-specific document retrieval performance of VKGDR and baselines on TechQA. In the first column, “L” represents that the model type is a lexical retriever. “D” represents dense retrievers. “D+L” is an ensemble model of a dense model and BM25. The “S” column indicates whether each model is trained on the question-document pairs of TechQA. The results show that VKGDR outperforms baselines of the same model type and even outperforms the fully-supervised model in R@1 and MRR.

Efficacy of VKG in a Zero-Shot Setting: Table 2 shows the performance of three types of models. The first column indicates the type of each model. Type “L” represents lexical retrievers, type “D” represents dense retrievers, and type “D+L” represents ensemble models of type “D” and type “L.” The ensemble models compute a similarity score of each document with the following formula:

$$\mathrm{Score}(d_i) = -(\mathrm{Rank}_{\mathrm{Dense}}(d_i) + \lambda \cdot \mathrm{Rank}_{\mathrm{BM25}}(d_i)),$$

where $\mathrm{Rank}_{\mathrm{Dense}}(d_i)$ and $\mathrm{Rank}_{\mathrm{BM25}}(d_i)$ are ranks of document d_i predicted by a dense retriever and BM25. λ is a weight for BM25, and we set λ to 1.0. The column “S” in Table 2 (“S” stands for supervision) indicates whether each model is a zero-shot model or a fully-supervised model. “✗” represents that the model is an unsupervised model, and “✓” represents the model is trained on the question-document pairs of the TechQA train set.

The results of type “D” models show that VKG construction brings better retrieval performance than other approaches. We show that the domain-adaptation method (Adapt) significantly underperforms than VKGDR by 34.3%p in R@1. Training retrievers on data in another domain (DPR) results in 22.5%p lower performance than VKGDR in R@1. From these results, we show the efficacy of constructing a VKG.

From previous literature, we see that BM25 outperforms dense retrievers when insufficient question-document pairs are provided and when the questions are domain-specific (Ma et al., 2021; Thakur et al., 2021). The results of BM25 in Table 2 are aligned with previous research on document retrievers; BM25 outperforms dense retrievers (“D” models) in Table 2. We combine BM25

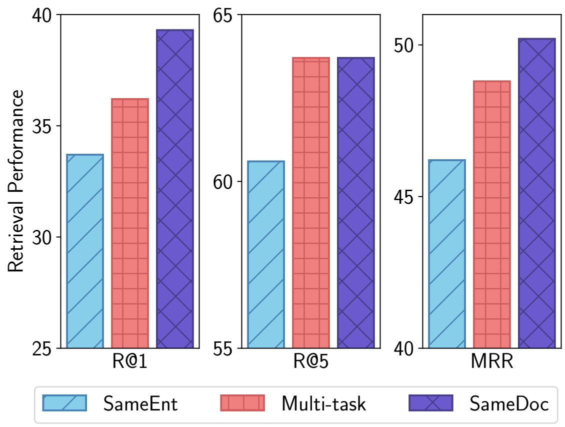


Figure 4: The zero-shot domain-specific document retrieval performance of VKGDR in three different relation encoder distant-supervision methods. SameEnt is the previous method that assumes two mention pairs have the same relation if the mention pairs share the same entity pair. SameDoc is our proposed supervision. Multi-task is multi-task learning of SameEnt and SameDoc. The results show the efficacy of our distant-supervision method.

and dense retrievers (type "D+L") to improve the retrieval performance in a zero-shot setting. As a result, VKGDR achieves 5%p higher retrieval performance, 44.3 R@1, and this outperforms other baselines, including BM25.

VKG construction enables unsupervised retrievers to overcome a fully-supervised bi-encoder. We report the performance of a bi-encoder trained on question-document pairs of TechQA in Table 2 (DPR*). We see that only type "D" VKGDR outperforms DPR* in R@1 and MRR. These results support that the VKG is a key component in achieving a better performance than a fully-supervised model when training data is unavailable.

VKGDR outperforms previous approaches in building a VKG: Figure 4 shows the zero-shot document retrieval performance of three different VKG construction methods on TechQA. The three methods are "SameEnt", "SameDoc", and "Multi-task." The relation encoder in each method uses a different distant-supervision. "SameEnt" assumes that mention pairs sharing the same entity pair have the same relation (Sun et al., 2021). "SameDoc" is our distant-supervision method. We conduct multi-task learning of "SameEnt" and "SameDoc" ("Multi-task"). Multi-task learning combines multiple object functions and achieves better performance than the models trained by only one of the object functions.

Type	Model	S	PhotoshopQuiA R@1	PhotoshopQuiA R@5	PhotoshopQuiA MRR
L	BM25	✗	4.9	10.5	8.8
D	Adapt	✗	2.1	14.4	11.1
	DPR	✗	9.1	26.7	19.9
	VKGDR	✗	**22.5**	**45.0**	**33.3**
D+L	Adapt	✗	1.4	11.6	9.8
	DPR	✗	6.3	**16.9**	**14.1**
	VKGDR	✗	**8.8**	15.8	12.9
D	DPR*	✓	12.3	36.2	24.5

Table 3: The zero-shot domain-specific document retrieval performance of VKGDR and baselines on PhotoshopQuiA. This table shares the symbols used in the first column and the meaning of the "S" column with Table 2. The results show that VKGDR outperforms baselines in a zero-shot setting and even outperforms the fully-supervised model.

Figure 4 shows that our distant-supervision, "SameDoc", outperforms the previous approach, "SameEnt". Also, we see that the performance of "SameEnt" increases when "SameEnt" is jointly trained with our distant-supervision. However, the performance of "SameDoc" decreases in this multi-task setting. This result indicates that the previous approach and our method are not complementary in the multi-task setting. From these results, we show that the context of mention pairs provides a better supervision signal than the textual form of mention pairs (entities of the mentions).

5.2 Zero-Shot Domain-Specific Answer Retrieval

Table 3 shows the zero-shot answer retrieval performance of VKGDR and baselines on PhotoshopQuiA. Type "D" retrievers show similar results as Table 2; VKGDR outperforms other type "D" baselines. VKGDR also outperforms the fully-supervised bi-encoder, DPR*. These results show that using a VKG brings better answer retrieval performance than the domain-adaptation method and the transfer learning method when training data is unavailable.

In Table 3, we show the performance of BM25 and dense retrievers ensembled with BM25. The lexical retriever underperforms dense retrievers on PhotoshopQuiA, whereas BM25 is a strong baseline on TechQA. Also, using lexical matching degenerates the overall retrieval performance of dense

Model	TechQA		
	R@1	R@5	MRR
DPR*	36.8	73.1	52.3
VKGDR	**48.7**	**76.8**	**60.1**

Model	PhotoshopQuiA		
	R@1	R@5	MRR
DPR*	12.3	36.2	24.5
VKGDR	**25.3**	**52.1**	**38.1**

Table 4: The document retrieval performance of VKGDR and DPR supervised on question-document pairs of TechQA and PhotoshopQuiA.

retrievers. This is because of the inconsistent use of terminologies between the corpus and the questions. The corpus of PhotoshopQuiA consists of answers written by users, not the official manual of the product, and this makes PhotoshopQuiA more difficult than TechQA.

5.3 Fully-Supervised Domain-Specific Document Retrievers

Fully-supervised VKGDR outperforms the fully-supervised bi-encoder (DPR*). Table 4 shows the retrieval performance of VKGDR trained on question-document pairs of TechQA and PhotoshopQuiA. We train the relation encoder with the following assumption: mentions pairs that appeared in the same question-document pair are in similar relation. The fully-supervised relation encoder is then used to compute the relation vectors of the VKG, and VKGDR uses the new VKG for document retrieval. The relation encoder trained on question-document pairs increase the retrieval performance of VKGDR; R@1 of VKGDR in Table 4 are 4.4%p and 2.8%p higher than the R@1 of VKGDR in Table 2 and 3. Also, fully-supervised VKGDR significantly outperforms DPR* by 11.9%p and 13.0%p in R@1 on TechQA and PhtoshopQuiA, respectively; we see the same pattern in other evaluation metrics.

5.4 Ablation Study

	R@1	R@5	MRR
VKGDR	39.3	63.7	50.2
- w/o relation embedding	32.5	59.3	44.9
- w/o mention pairs	31.8	53.7	42.9

Table 5: This table shows the performance of VKGDR in three different settings: without any modification on the VKG, using the VKG without relation vectors ("w/o relation embedding"), and using the VKG without the graph structure ("w/o mention pairs"). The results indicate that both components are essential to achieve the previous experimental results.

VKG consists of two components: graph representation of a corpus and relation vectors. In this section, we verify the importance of each module. Table 5 shows the performance of VKGDR on TechQA in three different settings: VKGDR, VKGDR without using the relation vectors (w/o relation embedding), and VKGDR without using the graph structure (w/o mention pairs). We describe each setting with the example in Figure 3. "w/o relation embedding" is a model that uses $\vec{1}$ (a vector that all elements are one) for all relation vectors in the VKG; all relation vectors in Figure 3 are replaced with $\vec{1}$. This is equivalent to using the number of overlapping mention pairs as the similarity between a question and a document. "w/o mention pairs" is a model without mention pair matching. For instance, all values in the similarity matrix (right part of Figure 3) are used to compute the question-document similarity. Table 5 shows that "w/o relation embedding" has better performance than "w/o mention pairs". This indicates that the graph structure is slightly more important than the relation embedding. However, the gap is not significant in R@1 and MRR. So, we see that both components are essential to achieve the document retrieval performance of VKGDR.

6 Conclusion

The main challenge in domain-specific document retrieval is the difficulty of specialized knowledge and terminologies appearing in the documents. In this study, we propose VKGDR to resolve this problem. VKGDR consists of two modules: 1) the model that represents a given corpus into a graph of mentions and their relations and 2) a document retriever based on the VKG. We showed that VKGDR outperforms previous retrievers in zero-shot domain-specific document retrieval. When insufficient training data is provided, unsupervised VKGDR shows even better results than a fully-supervised dense retriever. Also, we compared our VKG construction method with a previous method and showed that our method performs better on domain-specific documents.

References

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. SciBERT: A pretrained language model for scientific text. In *EMNLP-IJCNLP*.

Vittorio Castelli, Rishav Chakravarti, Saswati Dana, Anthony Ferritto, Radu Florian, Martin Franz, Dinesh Garg, Dinesh Khandelwal, J Scott McCarley, Michael McCawley, et al. 2020. The techqa dataset. In *ACL*.

Ilias Chalkidis, Manos Fergadiotis, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos. 2020. LEGAL-BERT: The muppets straight out of law school. In *Findings of the Association for Computational Linguistics: EMNLP*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*.

Bhuwan Dhingra, Manzil Zaheer, Vidhisha Balachandran, Graham Neubig, Ruslan Salakhutdinov, and William W Cohen. 2020. Differentiable reasoning over a virtual knowledge base. In *ICLR*.

Andrei Dulceanu, Thang Le Dinh, Walter Chang, Trung Bui, Doo Soon Kim, Manh Chien Vu, and Seokhwan Kim. 2018. Photoshopquia: A corpus of non-factoid questions and answers for why-question answering. In *LREC*.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. In *ACL*.

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 *EMNLP*.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. 2019. Natural questions: A benchmark for question answering research. *TACL*.

Davis Liang, Peng Xu, Siamak Shakeri, Cicero Nogueira dos Santos, Ramesh Nallapati, Zhiheng Huang, and Bing Xiang. 2020. Embedding-based zero-shot retrieval through query generation. *arXiv*.

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

Ji Ma, Ivan Korotkov, Yinfei Yang, Keith Hall, and Ryan McDonald. 2021. Zero-shot neural passage retrieval via domain-targeted synthetic question generation. In *EACL*.

Macedo Maia, Siegfried Handschuh, André Freitas, Brian Davis, Ross McDermott, Manel Zarrouk, and Alexandra Balahur. 2018. Www'18 open challenge: financial opinion mining and question answering. In *Companion Proceedings of the The Web Conference 2018*.

Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *ACL*.

Abhilash Nandy, Soumya Sharma, Shubham Maddhashiya, Kapil Sachdeva, Pawan Goyal, and Niloy Ganguly. 2021. Question answering over electronic devices: A new benchmark dataset and a multi-task learning based qa framework. In *Findings of the Association for Computational Linguistics: EMNLP*.

Livio Baldini Soares, Nicholas FitzGerald, Jeffrey Ling, and Tom Kwiatkowski. 2019. Matching the blanks: Distributional similarity for relation learning. In *ACL*.

Haitian Sun, Bhuwan Dhingra, Manzil Zaheer, Kathryn Mazaitis, Ruslan Salakhutdinov, and William Cohen. 2018. Open domain question answering using early fusion of knowledge bases and text. In *EMNLP*.

Haitian Sun, Patrick Verga, Bhuwan Dhingra, Ruslan Salakhutdinov, and William W Cohen. 2021. Reasoning over virtual knowledge bases with open predicate relations. In *ICML*.

Alon Talmor and Jonathan Berant. 2018. The web as a knowledge-base for answering complex questions. In *NAACL-HLT*.

Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. BEIR: A heterogeneous benchmark for zero-shot evaluation of information retrieval models. In *NeurIPS Datasets and Benchmarks Track*.

Ellen Voorhees, Tasmeer Alam, Steven Bedrick, Dina Demner-Fushman, William R Hersh, Kyle Lo, Kirk Roberts, Ian Soboroff, and Lucy Lu Wang. 2021. Trec-covid: constructing a pandemic information retrieval test collection. In *SIGIR*.

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 *NAACL-HLT*.

Wenhao Yu, Lingfei Wu, Yu Deng, Ruchi Mahindru, Qingkai Zeng, Sinem Guven, and Meng Jiang. 2020. A technical question answering system with transfer learning. In *EMNLP: System Demonstrations*.

Wenhao Yu, Lingfei Wu, Yu Deng, Qingkai Zeng, Ruchi Mahindru, Sinem Guven, and Meng Jiang. 2021. Technical question answering across tasks and domains. In *NAACL-HLT: Industry Papers*.

Rong Zhang, Revanth Gangi Reddy, Md Arafat Sultan, Vittorio Castelli, Anthony Ferritto, Radu Florian, Efsun Sarioglu Kayi, Salim Roukos, Avirup Sil, and Todd Ward. 2020. Multi-stage pretraining for low-resource domain adaptation. In *EMNLP*.

Yuyu Zhang, Hanjun Dai, Zornitsa Kozareva, Alexander J Smola, and Le Song. 2018. Variational reasoning for question answering with knowledge graph. In *AAAI*.

A Appendices

A.1 Experimental Setup

Evaluation Metrics: Recall@k (R@k) and mean reciprocal rank (MRR) are evaluation metrics for document retrieval tasks. R@k measures the proportion of the model's predictions where top-k retrieved documents contain the ground truth document. MRR is defined with the predicted rank of the ground truth document as follows:

$$\text{MRR} = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{r_i},$$

where n is the number of predictions, r_i is the predicted rank of the ground truth document of i'th query.

Hyper-parameter Settings: We use Adam optimizer with a warmup ratio of 0.1 and set the learning rate to 2×10^{-5} for VKGDR and baselines. We use the validation score to get the best checkpoint for all models. VKGDR's relation encoder is trained on the pre-trained BERT-large model. We train the relation encoder with a batch size of 128 for two epochs. The max length of the relation encoder is set to 128, and the number of negative samples in (1) is set to 2. We train RoBERTa (Bi-Encoder) with a batch sizes of 32 for twenty epochs and Adapt with a batch size of 80 for ten epochs. For both baselines, we set the max sequence length to 512. We use a machine with eight A100 GPUs. We report the result of a single trial.

A.2 License or Terms of Artifacts

We use BERT whose license is under the Apache License 2.0 free with modification and distribution. Also, we use RoBERTa whose license is under the GNU GENERAL PUBLIC LICENSE free with modification and distribution. All models we used are publicly available.