Can a Frozen Pretrained Language Model be used for Zero-shot Neural Retrieval on Entity-centric Questions?

Yasuto Hoshi, Daisuke Miyashita, Yasuhiro Morioka, Youyang Ng, Osamu Torii, Jun Deguchi
Kioxia Corporation
Kawasaki, Kanagawa, Japan
{yasuto1.hoshi, daisuke1.miyashita, yasuhiro.morioka, youyang.ng, osamu.torii, jun.deguchi}@kioxia.com

Abstract

Neural document retrievers, including dense passage retrieval (DPR), have outperformed classical lexical-matching retrievers, such as BM25, when fine-tuned and tested on specific question-answering datasets. However, it has been shown that the existing dense retrievers do not generalize well not only out of domain but even in domain such as Wikipedia, especially when a named entity in a question is a dominant clue for retrieval. In this paper, we propose an approach toward in-domain generalization using the embeddings generated by the frozen language model trained with the entities in the domain. By not fine-tuning, we explore the possibility that the rich knowledge contained in a pretrained language model can be used for retrieval tasks. The proposed method outperforms conventional DPRs on entity-centric questions in Wikipedia domain and achieves almost comparable performance to BM25 and state-of-the-art SPAR model. We also show that the contextualized keys lead to strong improvements compared to BM25 when the entity names consist of common words. Our results demonstrate the feasibility of the zero-shot retrieval method for entity-centric questions of Wikipedia domain, where DPR has struggled to perform.

Introduction

Information retrieval (IR) is the task of finding relevant knowledge or passages corresponding to a given query. Traditional exact lexical-matching approaches, such as TF-IDF (term frequency and inverse document frequency) or BM25 (**??**), have performed well on some IR tasks. Recently, neural retrieval with contextualized dense sentence embedding has shown to be effective for tasks such as open-domain question answering (e.g., **??**).

Dense retriever, or dense passage retrieval (DPR), uses a pair of neural language models as a bi-encoder to obtain latent representations of questions and passages (**?**). Bi-encoders require to be fine-tuned with contrastive learning to embed a question and the relevant passages on semantically similar sentence vectors. Recent works on DPR reported in-domain (tested on the same dataset which was used to fine-tune) retrieval performances exceeding those from sparse retrievers such as BM25 (**?**).

Accepted to Workshop on Knowledge Augmented Methods for Natural Language Processing, in conjunction with AAAI 2023.

However, it has been shown that conventional DPRs struggle with retrieval on BEIR benchmark (**?**) including various domains and even on EntityQuestions (EQ) (**?**) built within a confined domain (Wikipedia), whereas BM25 shows better performance on both of them. This indicates that the generalizability of DPR is limited not only out of domain but even in domain when a named entity in a question is a dominant clue for retrieval. In order to address this, **?** showed that DPR trained with PAQ dataset (**?**) which consists of 65 million of question–answer pairs containing many of named entities in Wikipedia still performs far worse than BM25 in EQ dataset. The results showed that supervised learning with a huge dataset that covers the domain extensively could not lead to good generalization within the domain.

Though BM25 seems to be relatively robust, it may have an inherent drawback: It is not good at searching for common terms in a query, even when the term is important, such as a part of a named entity. Specifically, BM25 uses the inverse document frequency (IDF) weight of the query terms. This property is based on the assumption that rare terms (of higher IDF values) are more informative than common terms (of lower IDF values). However, there are not few cases where named entity consists of common words and a retriever must retrieve relevant passages using it as the dominant clue. For instance, given a question "*Who is the author of Inside Job?*", the entity name "*Inside Job*" consists of common words, and the IDF values of these words are relatively low. In this case, BM25 may have difficulty in retrieving relevant passages.

One of the reasons for the poor generalization of DPRs can be their poor retrieval accuracy on salient phrases such as named entities (e.g., **?**). To address the problem, **?** proposed the Salient Phrase Aware Retriever (SPAR) by combining existing DPR and a dense retriever which is trained so as to imitate BM25 prediction, and SPAR showed better performance than BM25 on EQ dataset. Looking at SPAR from a perspective of modeling, it can be considered as the ensemble of two different dense retrievers. In contrast, our motivation is to investigate if dense retrievers could generalize with a single language model without ensemble.

In this paper, we investigate whether a pretrained language model without supervised fine-tuning can be used for passage retrieval. The approach is inspired by the idea that language models can have a lot more useful knowledge right after pretraining, but it has been forgotten during fine-tuning (e.g., **?**). We propose *Zero-shot Neural Retrieval* (Zero-NeR), a simple dense retrieval method that uses multiple contextualized keys for each passage generated from a frozen pretrained language model. We show our zero-shot method outperforms conventional DPRs and achieves a retrieval accuracy comparable to that of BM25 and SPAR on entity-centric questions in Wikipedia domain. Our system first extracts the named entities in questions and passages. Then, a pretrained language model encodes the recognized entity names into contextualized queries and retrieval keys.

The key contributions of this paper are as follows:

1. We propose a Zero-NeR method that can leverage rich representations for in-domain named entities from a pretrained and frozen language model, and show that its retrieval accuracy is better than the conventional DPRs.
2. We show that using multiple retrieval keys for each passage improves recall especially when a named entity in a question is a dominant clue for retrieval.
3. We demonstrate that our retrieval method is superior to BM25 when entity names in questions consist of common words.

Related Work

Passage Retrieval. Unsupervised sparse retrievers have traditionally been used for IR tasks, including answering open-domain questions (**?**). They are based on bag-of-words exact lexical matching, including TF-IDF and a best-match weighting function called BM25 (**??**). Unlike sparse retrieval, dense retrieval is based on semantic matching in the embedding space. Dense retrievers leverage dense vector representations of sentences embedded by fine-tuned neural language models. We refer the reader to **?** for details on major retrieval methods. It is also reported that re-ranking the retrieved passages can improve recall in EQ dataset (e.g., **?**), but note that re-rankers are outside the scope of this paper.

In-Domain Generalization of Retriever. Dense retrievers still have room for improvement of in-domain generalization, as well as out-of-distribution generalization (**?**). To generalize in Wikipedia domain, **?** trained DPR on PAQ dataset (**?**) which contains many of named entities in Wikipedia. However, the trained model still performed far worse than BM25 in EQ dataset. **?** also attempted to improve recalls for relational questions by training dedicated question encoders for each relation. While this approach somewhat improved recall, the dedicated retrievers did not reach the performance of BM25 on average and did not solve the poor generalization problem. In contrast, our proposed method employs a pretrained language model without fine-tuning for retrieval, allowing us to exploit the rich knowledge including named entities in Wikipedia learned by pretraining.

Difference between Dense Retrieval and Sparse Retrieval. **?** showed that the overlap of results between dense and sparse retrieval was quite small. It has been known empirically that the ensemble results for these relevance scores can exceed the performance of each alone (e.g., **?**). More recently, **?** attempted to train a dense retriever (called as dense Lexical Model $\mathbf{\Lambda}$) to imitate BM25 prediction. Though $\mathbf{\Lambda}$ model underperforms BM25 by itself on EQ dataset, when combined with DPR trained on multiple QA datasets (DPR-multi), the combined model (SPAR) outperforms BM25. SPAR requires two individually trained bi-encoders (thus four BERT models) with tangled architecture, whereas our motivation is to investigate if dense retrievers could generalize with a single language model without ensemble. We also investigate whether dense retrieval has any strengths in areas where sparse retrieval is lacking. We experimentally demonstrate the differences between dense and sparse retrieval by using the IDF value to quantify the generality and rarity of an entity name in a question.

Multiple Keys for Passage Retrieval. Several methods have been proposed for calculating fine-grained interactions between a question and passages using multiple retrieval keys. SPARTA (Sparse Transformer Matching, (**?**)) and COIL (Contextualized Inverted List, (**?**)) produce passage representations for each token embedded by fine-tuned language models and store them in the inverted index. ColBERT (**?**) leverages the embedding of all subword tokens in each passage to form a large collection of keys. However, all of the methods listed here require the encoders for questions and passages to be fine-tuned. During this fine-tuning for retrieval, multiple index updates are required, which is computationally expensive. We seek a retrieval method that works in zero-shot setting, so we propose a method that only requires the inference operation of a pretrained language model together with named-entity recognition (NER).

LUKE model (**?**) is the state-of-the-art language model for tasks related to named entities, including NER. It is easy to suppose that the LUKE model, which seems to have useful knowledge about named entities, would also be useful for passage retrieval. However, experiments using the model to perform passage retrieval have not been conducted. In this paper, we explore the possibility that the rich knowledge contained in a pretrained language model can be used for retrieval tasks. We show the advantage that our method has over existing dense retrievers for simple entity-centric questions.

Zero-Shot Neural Retrieval

Pipeline

Here we describe our proposed zero-shot dense retrieval method using a pretrained and frozen language model. Figure 1 provides a high-level overview of our system. There are three major changes to the DPR: (1) employing embeddings of named entities in passages as retrieval keys, (2) scoring relevance with cosine similarity over multiple keys of each passage and maxpooling the score to output single score, and (3) elimination of encoder fine-tuning for retrieval. The method follows the following steps:

1. Named entities in questions and passages are extracted via the NER procedure. Our implementation limits the number of named entities in a question to one.

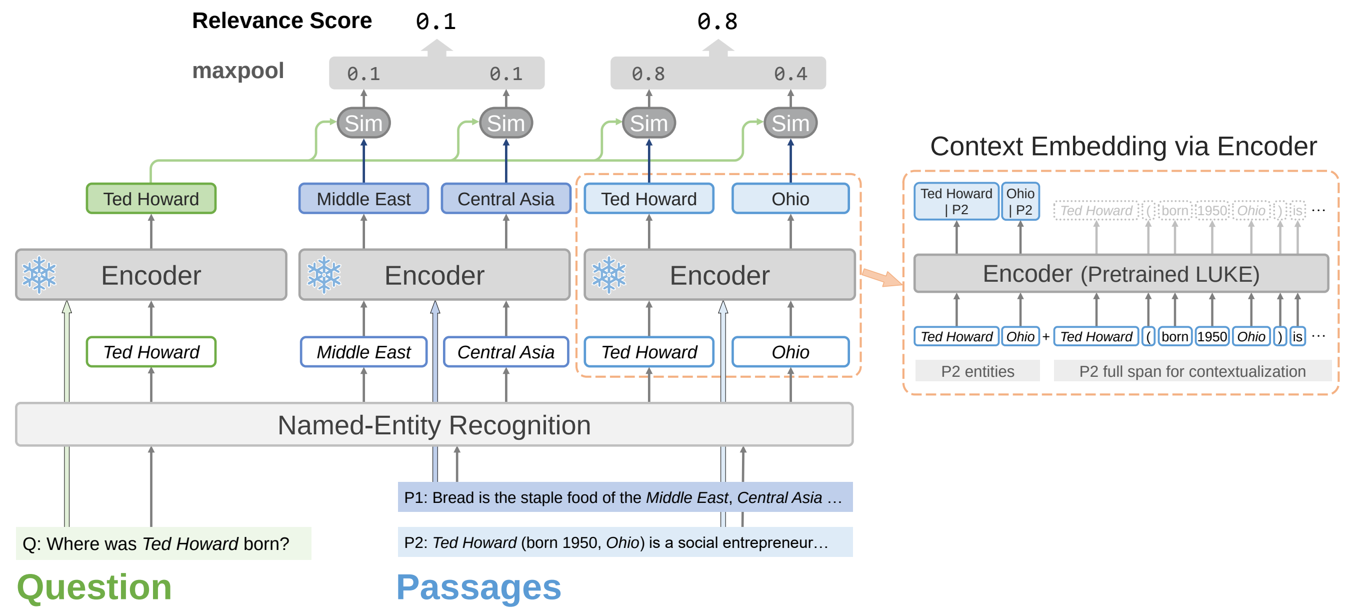


Figure 1: An overview of our proposed zero-shot dense retrieval system. Named entities in a question and passages are extracted via an off-the-shelf named-entity recognition model. Then, the extracted named entities (hollow rectangles) are encoded into dense representations (solid rectangles) with a frozen pretrained language model (denoted as *Encoder*). Along with the entity span, a full sequence of the text is used to condition the semantics of the entire sentence on the embedding (outlined arrows). For instance, the dense representation corresponding to *Ted Howard* is conditioned by P2, shown as "Ted Howard | P2" in this Figure. We also use the embedding of the entire span of a passage title as a retrieval key. The passages can have multiple keys depending on how many entity names the passage has. The similarity between a query and a key is measured using cosine similarity (denoted as *Sim*). For multiple keys in each passage, single relevance score for each passage is calculated via maximum pooling. Note that the proposed system requires no additional training if the encoder language model is pretrained.

2. The named entities in questions and passages are encoded as a query and keys, respectively. When encoding the named entities, the corresponding question and passages are input to the frozen language model to contextualize the named entities. The title of the passage is also encoded and used as a key of the passage. Therefore, a passage with i entity names will have $i+1$ keys.
3. The k highest-scoring passages for a query are retrieved. The cosine similarity is used as a scoring function. For multiple keys in each passage, similarity scores to a query are calculated respectively, then single relevance score for each passage is calculated via maximum pooling. Formally, given a query q, a total of j keys $keys = (k_1, k_2, \dots, k_j)$ of a passage, and the similarity $s_i = \mathrm{sim}(q, k_i)$ where sim is cosine similarity, the relevance score for the passage is

$$\mathrm{score}(q, keys) = \mathrm{maxpool}(s_1, s_2, \dots, s_j).$$

Named Entity Recognition

We use the off-the-shelf LUKE model[2] (?) to extract named entities from passages. This model is trained using entity-annotated Wikidata and is currently state-of-the-art in the CoNLL-2003 NER task (?). The inputs to the model are a text and the candidates of entity span in that text. For each candidate span, the model predicts the types of entity (locations `LOC`, organizations `ORG`, persons `PER`, and miscellaneous `MISC`).

[2]https://huggingface.co/studio-ousia/luke-large-finetuned-conll-2003

Contextualized Representations of Named Entities

To produce queries and keys for retrieval, we use dense representations of the span-level embeddings of entity names in questions and passages. We used the pretrained LUKE model[3] to encode an entity span to a contextualized representation. The LUKE model is based on the RoBERTa model (?) and has an extension called *entity-aware self-attention*. This mechanism allows our system to embed a span-level contextualized representation of the entity name. An entity embedding is output as a single vector from the model, even if it consists of multiple tokens. We also use the embedding of the entire span of a title of a passage as a retrieval key.

Experiments

Here, we describe the datasets and baseline retrievers and explain the preparation for queries and retrieval keys with NER. Then, we report the top-20 retrieval performances on EQ dataset.

[3]https://huggingface.co/studio-ousia/luke-base

Modeling	Retriever	# LM	Retrieval training	Recall@20 (macro average)
Sparse	BM25	-	-	72.0
Dense	DPR-NQ	2	Supervised	45.1
	DPR-multi	2	Supervised	56.7
	DPR-PAQ (**?**)	2	Supervised	59.3
	DPR-PAQ (SPAR checkpoint)	2	Supervised	63.0
	Wiki $\mathbf{\Lambda}$	2	Supervised	69.0
	PAQ $\mathbf{\Lambda}$	2	Supervised	69.9
	SPAR-Wiki (Wiki $\mathbf{\Lambda}$ + DPR-multi)	4	Supervised + Supervised	73.9
	SPAR-PAQ (PAQ $\mathbf{\Lambda}$ + DPR-multi)	4	Supervised + Supervised	74.5
	Contriever	2	Self-supervised	64.7
	Zero-NeR (Ours)	**1**	**None (only LM pre-training)**	**67.1**

Table 1: The macro average of the top-20 retrieval accuracies (denoted as Recall@20) on the 24 relations of EQ test set. *# LM* refers to the total number of language models used in each dense retriever with different weight parameters. *Retrieval training* denotes fine-tuning methods for retrieval of each dense retriever. See text for details of the baseline models.

Datasets

Knowledge Source. We used the Wikipedia passage splits preprocessed by **?**. The passages consist of English Wikipedia passages processed from Dec. 20, 2018 Wikipedia dump data. The corpus contains a total of 3,232,908 articles containing 21,015,324 passages with a length of 100 words.

EntityQuestions[4] (EQ) consists of simple and entity-rich questions (**?**). The dataset contains questions from 24 common relations based on Wikidata (**?**). The questions were generated from relation triplets *(subject, predicate, object)* in the T-REx dataset (**?**) using 24 of manually defined question templates. Each relation in the test set of EQ dataset has at most 1,000 questions (212 questions at minimum) and an average of 919.8 questions. Because the number of questions differs by each relation, we follow the original paper and report the arithmetic macro-average of retrieval accuracies in each relation.

Baseline Models

BM25 (**?**) is a bag-of-words retrieval function based on the term-matching of sparse vectors. We used the Anserini (**?**) implementation with the default parameters of Lucene ($k = 0.9$ and $b = 0.4$).

DPR is a supervised method which employs a bi-encoder structure with a query encoder and a passage encoder (**?**). **DPR-NQ** is fine-tuned on the Natural Questions dataset (**?**). We used the model checkpoint provided in the DPR repo[5]. **DPR-multi** is fine-tuned on the combination of Natural Questions (**?**), TriviaQA (**?**), WebQuestions (**?**), and CuratedTREC (**?**). **DPR-PAQ** is fine-tuned on the PAQ dataset (**?**). In our evaluation, we used DPR-multi and DPR-PAQ checkpoints provided in the SPAR repo[6].

[4]https://github.com/princeton-nlp/EntityQuestions

[5]https://github.com/facebookresearch/DPR

[6]https://github.com/facebookresearch/dpr-scale

SPAR (**?**) is a dense retriever that combines a standard DPR with the dense model called Lexical Model $\mathbf{\Lambda}$ trained to imitate BM25. The SPAR model leverages two bi-encoders (thus four BERT models) to build itself and showed state-of-the-art performance on EQ dataset. We use the **Wiki $\mathbf{\Lambda}$** model trained on Wikipedia corpus, **PAQ $\mathbf{\Lambda}$** model fine-tuned on PAQ dataset (**?**), and models combined with the DPR-multi model, **SPAR-Wiki** (Wiki $\mathbf{\Lambda}$ + DPR-multi) and **SPAR-PAQ** (PAQ $\mathbf{\Lambda}$ + DPR-multi). We used $\mathbf{\Lambda}$ and DPR-multi model checkpoints provided in the SPAR repo. Following the original paper (**?**), we tuned the concatenation weight μ of SPAR on development set of EQ dataset using the value of μ for which macro average of recall@100 is the best. As a result of our tuning, $\mu = 1.25$ is the best for both SPAR-Wiki and SPAR-PAQ.

Contriever[7] (**?**) is trained with self-supervised contrastive learning on Wikipedia and CCNet (**?**). To fine-tune the model with contrastive learning, a positive pair is formed by sampling two spans from a document, that is similar to Inverse Cloze Task (**?**).

Preparing Query and Retrieval Key with NER

Query For the questions in EQ dataset, the named entity in a question can be extracted by using the corresponding question template. For example, the question template for `place of birth` (P19) is "*Where was [E] born?*", where *[E]* is an entity name. Given the question "*Where was Ted Howard born?*", the entity name "*Ted Howard*" can be extracted by comparing the question to the template.

Retrieval Key For Wikipedia passages, following the NER method described in the previous section, a total of 219,258,978 named entities were recognized from 21-million passages. Following the Pipeline section, the retrieval keys were made from named entities in the passages and the Wikipedia article titles. Along with a total of 3,232,908 article titles, the total number of retrieval keys is

[7]https://huggingface.co/facebook/contriever

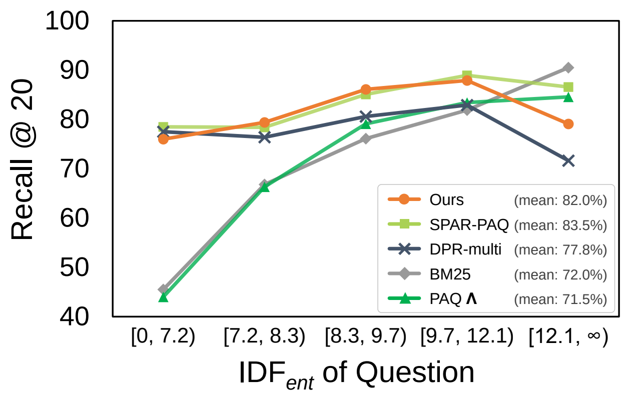


Figure 2: The top-20 retrieval accuracy on `author` (P50, *Who is the author of [E]?*), where the test set questions were grouped by the IDF$_{ent}$ value of the entity name in the questions. The questions are grouped into five buckets according to the maximum IDF value in the entity tokens. The number of questions in each bucket is 200.

222,491,886. This results in each passage having an average of 10.6 retrieval keys for our method.

Main Results: Retrieval Accuracy on the EQ Dataset

Table 1 shows the arithmetic macro average of the top-20 retrieval accuracies on the 24 relations of EQ test set. Our method outperforms DPRs and Contriever and is almost comparable to BM25 and SPAR models. When viewed with respect to each relation, our method outperforms BM25 in retrieval accuracy for seven relations and SPAR-PAQ for three relations. Because the baseline DPRs employs asymmetric bi-encoder (query and key encoders do not share the parameters), # LM of DPRs is two. Note that, our method leverages the same model for the query and key encoders, thus # LM of our method is one.

Contriever employs self-supervised learning method for retrieval training rather than supervised one where knowledge acquired during pretraining is prone to forget (e.g., **?**). However, Contriever performed below BM25 and our method, showing that self-supervised dense retrievers still have difficulties in answering entity-centric questions, even in-domain.

Overall, we find that our method outperforms conventional DPRs, while the performance is below BM25 and SPAR. The results suggest that, by not fine-tuning, the knowledge of frozen language model acquired during pretraining can be applied to passage retrieval and could yield better in-domain generalization of dense retrieval.

Analysis

Characteristics of Our Method Compared with BM25

Here, we examine the advantages our method has over BM25.

Background Our method can have advantages in retrieval, especially when the named entities consist of common words. Given a question like "*Who is the author of Inside Job?*", it is expected that BM25 will have difficulty in searching relevant contexts. This is because the entity name "*Inside Job*" is composed of common words, and the IDF values of these words may be relatively low. On the other hand, regardless of their IDF values, neural language models could distinguish that "*Inside Job*" suggests the title of a work. Such semantics could be useful information in dense retrieval.

Methodology Specifically, we focus on the IDF value, which corresponds to the rarity of a word in a corpus. Following the implementation of BM25, the IDF value of term t is given by

$$\mathrm{IDF}_t = \log \frac{N - n_t + 0.5}{n_t + 0.5},$$

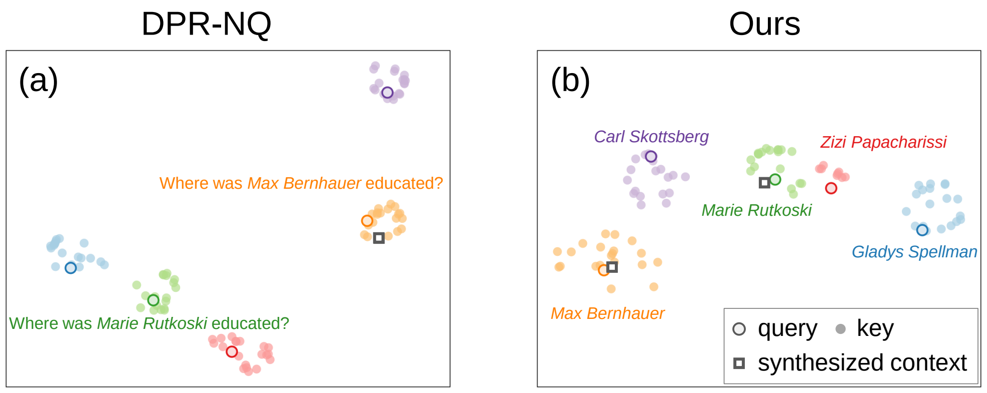
where N denotes the total number of passages and n_t denotes the number of passages containing the term t. For Wikipedia passages used in this study, the value of N is 21,015,324 (see Datasets section). We equally divided the `author` (P50, *Who is the author of [E]?*) questions into five buckets according to the maximum IDF values of the entity words in them. Each bucket has 200 questions. We will henceforth refer to the maximum IDF value of the entity words as IDF$_{ent}$. For example, the IDF values of the words in "*Inside Job*" ("*Inside*", "*Job*") are 5.8 and 6.0, respectively. Therefore, the IDF$_{ent}$ value of "*Inside Job*" is $\mathrm{IDF}_{ent} = \max(5.8, 6.0) = 6.0$.

Result and Discussion Figure 2 shows the top-20 retrieval results as a function of IDF$_{ent}$ of the named entity in the question. Wiki **Λ** and SPAR-Wiki (not shown in the Figure) denote the similar tendency of PAQ **Λ** and SPAR-PAQ, respectively.

Not surprisingly, the retrieval accuracy of BM25 increases monotonically with respect to the IDF$_{ent}$ values. For lower IDF$_{ent}$ values, the retrieval accuracy of BM25 was lower than that of our method. Interestingly, PAQ **Λ** also shows a similar tendency. This tendency is seemingly due to the fine-tuning for retrieval of **Λ** models where they learned to simulate the BM25 prediction. This indicates that, especially for common words, simply learning to imitate BM25 is ineffective and contextualized representations of entities of our method are much more effective for retrieval.

DPR-multi, SPAR-PAQ, and our method perform with less bias for IDF$_{ent}$ values. Figure 2 indicates that the ensemble of PAQ **Λ** and DPR-multi achieves relatively flat IDF$_{ent}$-dependence of SPAR and its high accuracies. Our method achieves similar flatness despite using only one language model, whereas SPAR uses four and DPR-multi uses two different language models.

For the bucket with the largest IDF$_{ent}$, the retrieval accuracy of our method and DPR-multi slightly decreased. For our method, this result might reflect the nature of statistical language modeling: the model may have difficulty in giving meaningful representations for rare words because they have not appeared often enough for their representations to



Synthesized Context (▣):

Max Bernhauer was an Austrian entomologist who specialised in Coleoptera, especially Staphylinidae. He described over 5,000 species. He studied and received a law degree from the University of Vienna in 1899. *Marie Rutkoski* is an American children's writer, and professor at Brooklyn College. She graduated from the University of Iowa with a B.A. in English with a minor in French in 1999, and then her English M.A. and Ph.D. from Harvard University in 2003 and 2006 respectively.

Figure 3: A UMAP projection of the queries and the corresponding top-20 keys retrieved by (a) DPR-NQ and (b) our method. For the queries, we used five randomly selected questions from the `educated at` questions (P69, *Where was [E] educated?*). The retrieved keys with respect to a query (outlined circle) are plotted in the same color as the query. The outlined rectangles represent the embeddings of a synthesized context that contains multiple relational information points for two people, (*Max Bernhauer* and *Marie Rutkoski*). Note that the number of the plotted embeddings of the synthesized context is one for DPR-NQ but two for ours.

be learned (e.g., **?**). For DPR-multi, the result suggests that the rarer the named entity in a question, the less it is included in the training dataset used to fine-tune, and the more difficult it is to retrieve in the test set.

Single Key versus Multiple Keys: Comparison with DPR

Here, we perform a detailed comparison of our method and DPR to assess the effect of using multiple retrieval keys. **?** has shown the advantages of assigning multiple keys for each passage via so-called multi-vector encoding. Our method, likewise, should have an advantage because it uses multiple keys, especially when the passage has multiple relational information.

Result and Discussion Figure 3 shows a UMAP projection (**?**) of the queries and the corresponding keys retrieved by DPR-NQ and our method. The queries consist of five randomly selected questions of the type `educated at` (P69, *Where was [E] educated?*). For DPR-NQ, the query was made from a full sequence of a given question. The query for our method was the embedding of an entity name (in this case, the name of a person) extracted from the question, where the corresponding question was also input to the encoder to contextualize the entity (see Pipeline section for details). The keys in Figure 3 indicate the top-20 retrieval results for DPR-NQ and our proposed method. In addition to the Wikipedia passages, we synthesized a passage that contains multiple relational information for two people, (*Max Bernhauer* and *Marie Rutkoski*).

Note that DPR produces a single key from the synthesized context, whereas our method produces multiple keys. Our method successfully retrieved the synthesized context with both queries (*Max* and *Marie*), but DPR-NQ failed to retrieve it with the query *Marie*. Thus, the multiple keys of our method can help retrieve a passage when it contains multiple relational information. In the next section, we report a detailed ablation that uses one key for each passage selected from the multiple keys.

Ablation Study: Are Multiple Retrieval Keys Required?

Here, we report the retrieval results from using one key for each passage, an equivalent setting to conventional dense retrieval, so that we may better assess the advantages that we have by using multiple keys. As described in Experiments section, the average number of retrieval keys was 10.6 for each 100-word passage from Wikipedia. We chose only one key from the full set of multiple keys for each passage in the following two sampling forms: (1) selected the key randomly (referred to as Random) and (2) selected a key corresponding to an entity span of maximum IDF$_{ent}$ (referred to as Max IDF$_{ent}$).

Results and Discussion In Figure 4, the top-20 retrieval results are shown with the keys sampled using the forms

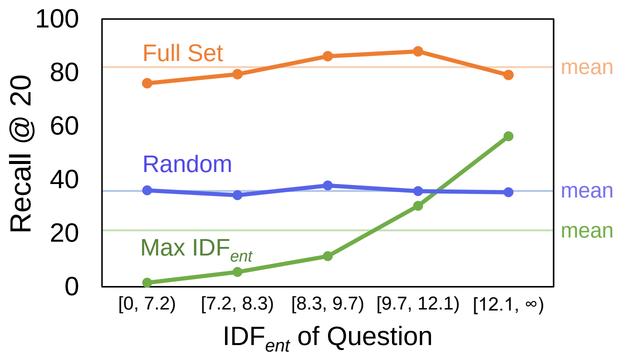


Figure 4: The top-20 retrieval accuracy for `author` (P50, *Who is the author of [E]?*). The format is the same as in Figure 2, and the Full Set is the same result as the one in that figure. For each passage from Wikipedia, only one key was selected using random selection (denoted as Random), along with the key corresponding to the maximum IDF$_{ent}$ (denoted as Max IDF$_{ent}$). The mean accuracies of Random and Max IDF$_{ent}$ are 35.8% and 21.0%, respectively, while the mean accuracy of Full Set is 82.0%.

above. The result with a full set of keys clearly surpasses the results with a single key (Random and Max IDF$_{ent}$). Not surprisingly, the result for Random is flat with no sensitivity to the IDF$_{ent}$ value. However, for Max IDF$_{ent}$, the retrieval accuracy decreases as the IDF$_{ent}$ value decreases. This could be because the corresponding key could not be registered when the IDF$_{ent}$ value for a question was small. For instance, the passages corresponding to the `author` questions can have the title of the work and the author's name. If IDF$_{ent}$ is small for a question (such as "*Inside Job*", the title of the work), then it is easy to imagine that the author's name in the same passage could have a larger IDF$_{ent}$. In this case, the retrieval accuracy can drop off in the absence of keys relevant to the question. We should also mention that even for larger IDF$_{ent}$, the retrieval accuracy of Max IDF$_{ent}$ is below that of a full set of keys. To summarize, we argue that using multiple keys for each passage significantly impacts on passage retrieval.

Ablation Study: Is "Contextualized" Embedding of "Named Entity" Required?

In this section, we investigate the impact that the input to the frozen LUKE encoder model during query generation has on retrieval accuracy.

Background For the entity-aware self-attention mechanism in the LUKE model, a full sentence is input along with entity spans as a condition of the entity spans. The mechanism considers the context for the entity name when computing attention scores. For instance, given the question "*Where was Ted Howard born?*", the contextualized representation of the entity name "*Ted Howard*" is calculated under the condition of a question about the birthplace. We focused on this conditioning and examined the effect of contextualization on queries.

Query	Condition	Top-20	Top-100
Entity span	Full span	**67.1**	**77.5**
Entity span	Entity span	63.2	73.6
Full span	Full span	28.6	44.7

Table 2: The top-20 and top-100 retrieval accuracies (%) on EQ test sets. The highest score is indicated in bold. For example, given a question "*Where was Ted Howard born?*", the entity span is "*Ted Howard*", and the full span of questions is "*Where was Ted Howard born?*".

Result and Discussion Table 2 shows the effectiveness of contextualization. When the query embeddings were generated with only entity spans, the retrieval accuracy degraded about 4 points. Furthermore, the accuracy significantly decreased when the query embeddings were generated for the entire question spans. This indicates that it is necessary to use the contextualized embeddings of named entities in questions.

Conclusion

In this paper, we investigate whether a pretrained language model without supervised fine-tuning can be used for passage retrieval. We proposed *Zero-NeR*, a simple zero-shot neural retrieval method employing rich knowledge in a pretrained and frozen language model for retrieval. Toward in-domain generalization, we investigate the feasibility of the proposed method for entity-centric questions in Wikipedia domain, where existing dense retrievers has struggled to perform. Our method outperforms the conventional supervised and self-supervised bi-encoders, and it is almost comparable to the lexical-matching model and the state-of-the-art dense retriever. We also show that the contextualized keys lead to strong improvements compared to BM25 when the entity names consist of common words. Our findings suggest that zero-shot representations from the frozen language model can be used for retrieval on entity-centric questions and also lead better in-domain generalization.

Delectus in voluptatibus quaerat voluptatum consequuntur, incidunt praesentium expedita optio accusantium, pariatur necessitatibus dicta sunt tempora, ab nemo fuga hic?Alias odit veniam sit animi nulla, tenetur non error architecto.Quam eum tempora impedit illo laborum ullam autem odio ipsam, quod velit illo sint eum assumenda, iure dolore maiores?Nam tenetur error consequuntur, soluta qui reiciendis sed ipsam corporis omnis tempora, amet porro quidem accusantium nam earum veniam consequatur fugit et aliquid impedit, cupiditate minima earum totam distinctio eum optio excepturi sit corporis quos quae.Id numquam consequuntur et laudantium dignissimos libero, sed officia placeat quis saepe aliquam modi commodi, natus deserunt harum nostrum commodi aliquid consequuntur velit beatae.Ducimus debitis omnis tenetur quia eum, reprehenderit veniam dolor, ex similique sunt quaerat quo et mollitia distinctio, commodi amet nostrum culpa voluptates fugit eveniet nemo quos iste perspiciatis vel, unde at dolores.Ratione eos perspiciatis sint ut consequuntur aperiam, fuga amet iure expedita quam, aspernatur libero officiis eligendi totam odit similique velit necessitatibus dolores maiores deleniti?