# Entity Retrieval for Answering Entity-Centric Questions

Anonymous ACL submission

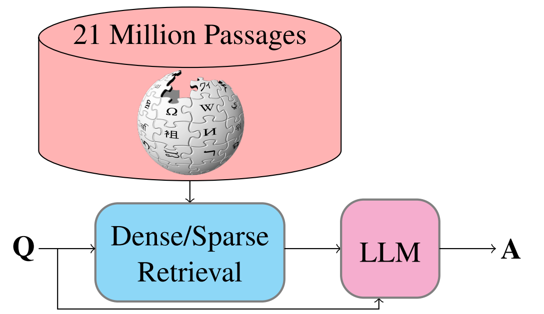
## Abstract

The similarity between the question and indexed documents is a crucial factor in document retrieval for retrieval-augmented question answering. Although this is typically the only method for obtaining the relevant documents, it is not the sole approach when dealing with entity-centric questions. In this study, we propose *Entity Retrieval*, a novel retrieval method which rather than relying on question-document similarity, depends on the salient entities within the question to identify the retrieval documents. We conduct an in-depth analysis of the performance of both dense and sparse retrieval methods in comparison to *Entity Retrieval*. Our findings reveal that our method not only leads to more accurate answers to entity-centric questions but also operates more efficiently.

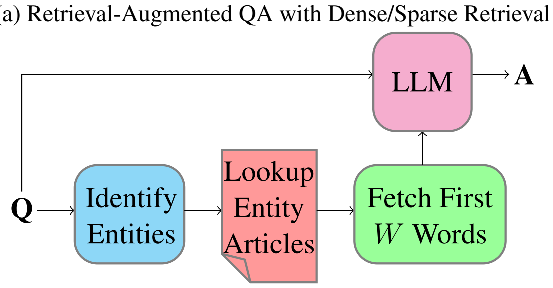
* We have included our source code implementation of the project, along with the generated model answers, in the Software section of our submission.

## 1 Introduction

Information retrieval has significantly enhanced the factual reliability of large language model (LLM) generated responses (Shuster et al., 2021) in question answering (Zhu et al., 2021; Zhang et al., 2023). This improvement is particularly notable in a research area known as retrieval-augmented generation (RAG; Lewis et al., 2020b; Izacard and Grave, 2021; Singh et al., 2021). RAG systems typically employ the *Retriever-Reader* architecture (Chen et al., 2017), with retrievers being either sparse (Peng et al., 2023), dense (Karpukhin et al., 2020), or a hybrid (Glass et al., 2022). The reader, which is a generative language model (e.g., BART; Lewis et al., 2020a, T5; Raffel et al., 2020, GPT-3; Brown et al., 2020), conditions its generated answers on the documents deemed relevant by the retriever. Recent RAG methodologies exploit the in-context learning capabilities of LLMs to incorporate the retrieved documents into the prompt (Shi et al., 2023; Peng et al., 2023; Yu et al., 2023).




(a) Retrieval-Augmented QA with Dense/Sparse Retrieval




(b) Retrieval-Augmented QA with *Entity Retrieval*

Figure 1: *Entity Retrieval* simplifies the process of obtaining augmentation documents by replacing the need to search through large indexed passages with a straightforward lookup. For **Q**: `What is the capital of Seine-Saint-Denis?` *Entity Retrieval* considers the first few sentences of `Seine-Saint-Denis` Wikipedia article which states "`Its prefecture is` **`Bobigny`**`.`" and returns **A = Bobigny** where the other retrieval methods return **A = Saint-Denis** or **A = Paris**.

Kandpal et al. (2023) demonstrate that retrieval-augmentation improves LLMs' performance in answering entity-centric questions that seek factual information about real-world entities[1]. They show that this technique is particularly helpful for questions about rare entities, which appear infrequently in LLM training and fine-tuning data.

[1]Entity-centric questions typically have answers that are concise single words or short phrases. These answers often reference or directly stem from a knowledge base entity (Ranjan and Balabantaray, 2016).

But is there a correlation between the quality of the retrieved documents and the generated response quality? Sciavolino et al. (2021) demonstrate that dense retrievers retrieve less relevant documents for answering entity-centric questions than simpler sparse retrievers. Additionally, Cuconasu et al. (2024) show that the presence of irrelevant documents leads to worse answers.

In this paper, we propose *Entity Retrieval* (Figure 1b), which uses salient entities in the question to lookup knowledge base (e.g., Wikipedia) articles that correspond to each entity, and uses the first $W$ words of the articles as augmentation documents for the question passed to the LLM.

Our contributions are as follows: (1) we propose *Entity Retrieval*, a novel method of acquiring augmentation documents using salient entities in the questions, (2) we compare the retrieval performance quality of several retrieval techniques (both dense and sparse) to *Entity Retrieval* for questions within two entity-centric question answering datasets, (3) we study the retrieval-augmentation quality of the compared techniques and *Entity Retrieval*, using salient entity annotations of the questions, and (4) we examine the application a recent state-of-the-art entity linking method for *Entity Retrieval* in the absence of entity annotations in entity-centric questions.

## 2 Retrieval for Retrieval-Augmentation

Retrieval-augmentation (Lewis et al., 2020b) is a method of converting Closed-book question answering[2] (Roberts et al., 2020) into extractive question answering (Abney et al., 2000; Rajpurkar et al., 2016), where the answers can be directly extracted from the retrieved documents. Despite the abundance of effective retrieval techniques for retrieval-augmented question answering in existing literature (Zhan et al., 2020a,b; Yamada et al., 2021; Izacard et al., 2022; Santhanam et al., 2022; Ni et al., 2022, *inter alia.*), this section will concentrate on a select few methods[3] utilized to study answering entity-centric questions in this paper.

**BM25** (Robertson et al., 1994, 2009) is a probabilistic retrieval method that ranks documents based on the frequency of query terms appearing in each document, adjusted by the length of the document and overall term frequency in the collection. It operates in the sparse vector space, relying on precomputed term frequencies and inverse document frequencies to retrieve documents based on keyword matching.

**DPR** (Dense Passage Retrieval; Karpukhin et al., 2020) leverages a bi-encoder architecture, wherein the initial encoder processes the question and the subsequent encoder handles the passages to be retrieved. The similarity scores between the two encoded representations are computed using a dot product. Typically, the encoded representations of the second encoder are fixed and indexed in FAISS (Johnson et al., 2019; Douze et al., 2024), while the first encoder is optimized to maximize the dot-product scores based on positive and negative examples.

**ANCE** (Xiong et al., 2021) is another dense retrieval technique similar to DPR. It employs an encoder to transform both the questions and passages into dense representations. These representations are compared using dot product similarity. The key distinction from DPR is that ANCE uses hard negatives generated by periodically updating the passage embeddings during training, which helps the model learn more discriminative features, thereby enhancing retrieval performance over time.

## 3 Entity Retrieval for Question Answering

While quite powerful, most retrieval-augmented systems are notably time and resource-intensive, necessitating the storage of extensive lookup indices and the need to attend to all retrieved documents to generate a response (see Section 4.7). This attribute renders such methods less desirable, particularly given the drive to run LLMs locally and on mobile phones (Alizadeh et al., 2023).

Entity recognition has been an integral component of statistical question answering systems (Aghaebrahimian and Jurčíček, 2016, *inter alia*). Additionally, the extensively studied field of Knowledge Base Question Answering (Cui et al., 2017, *inter alia*) has underscored the significance of entity information from knowledge bases in question answering (Salnikov et al., 2023). A traditional neural question answering pipeline may contain entity detection, entity linking, relation prediction, and evidence integration (Mohammed et al., 2018; Lukovnikov et al., 2019), where entity detection can employ LSTM-based (Hochreiter and

[2]Closed-book QA focuses on answering questions without additional context during inference.

[3]We selected the methods supported by `pyserini.io` for the similarity between the underlying modules, minimizing discrepancies across different implementations.

Schmidhuber, 1997) or BERT-based (Devlin et al., 2019) encoders. Inspired by this body of work, we investigate the relevance of retrieval based on entity information as an alternative strategy to the proposed retrieval methods of Section 2, especially for answering entity-centric questions with LLMs.

Our proposed method *Entity Retrieval*, leverages the salient entities within the questions to identify and retrieve their corresponding knowledge base articles. We will then use the first $W$ words of these articles as the documents augmenting entity-centric questions when prompting LLMs. Figure 1 presents a schematic comparison between *Entity Retrieval* and dense retrieval methods in identifying retrieval documents to enhance question answering with LLMs.

## 4 Experiments and Analysis

### 4.1 Setup

We focus on Wikipedia as the knowledge base and utilize the pre-existing BM25, DPR, and ANCE retrieval indexes in Pyserini (Lin et al., 2021). These indexes, follow established practices (Chen et al., 2017; Karpukhin et al., 2020) and segments the articles into non-overlapping text blocks of 100 words, resulting in 21,015,300 passages. For dense retrievers, the passages are processed with a pre-trained context encoder, generating fixed embedding vectors stored in a FAISS index (Douze et al., 2024). Our experimental entity-centric questions are encoded using the question encoder, and the top $k$ relevant passages to the encoded question are retrieved from the FAISS index. For BM25 sparse retriever, the passages are stored in a Lucene index and the questions are keyword matched to this index.

As outlined in Section 3, the document retrieval process will require loading the entire index (as well as the question encoder for dense retrieval) into memory which entails significant time and memory consumption. To address this challenge, following Ram et al. (2023), we treat document retrieval as a pre-processing step, caching the most relevant passages for each question before conducting the question answering experiments.

For *Entity Retrieval*, similar to BM25, DPR, and ANCE, we maintain document lengths at 100 words. However, our approach diverges in sourcing documents: rather than drawing from a large index of 21 million passages, we employ the salient entities within the question and retrieve their corresponding Wikipedia articles, which we then truncate to the initial 100 words. Nonetheless, to explore the impact of document size, beyond the standard 100-word segment aligned with comparable methods, we investigate *Entity Retrieval* across varied lengths, including the first 50, 300, and 1000 words from the retrieved Wikipedia articles.

We conduct our retrieval-augmented question answering experiments using LLaMA 3 model[4], and in all such experiments[5], we prevent it from generating sequences longer than 10 subwords.

We do not use any training question-answer pairs in the prompts of our models. In other words, aside from a simple instruction for answering the question, in the Closed-book setting, the prompt solely comprises the question, while in the retrieval-augmented settings using BM25, DPR, and ANCE, it includes the pre-fetched retrieved documents from the corresponding retrieval index along with the question. Similarly, for the *Entity Retrieval* settings, the prompt consists of the first $W$ words of the Wikipedia pages corresponding to the salient entities in the question. We follow Ram et al. (2023) for question normalization and prompt formulation.

### 4.2 Data

We use the following datasets in our experiments:

**EntityQuestions** (Sciavolino et al., 2021) is created by collecting 24 common relations (e.g., 'author of' and 'located in') and transforming fact triples (subject, relation, object) that contain these relations, into natural language questions using predefined templates. The dataset comprises 176,560 train, 22,068 dev, and 22,075 test question-answer pairs. To expedite our analytical experiments in this paper, given the extensive size of the dev and test sets, we constrain the question-answer pairs in these subsets to those featuring salient entities within the top 500K most linked Wikipedia pages, as suggested by (Shavarani and Sarkar, 2023). Thus, the dev and test subsets of EntityQuestions considered in our experiments consist of 4,710 and 4,741 questions, respectively.

**FactoidQA**[6] (Smith et al., 2008) contains 2,203 hand crafted question-answer pairs derived from Wikipedia articles, with each pair accompanied by

[4] https://llama.meta.com/llama3/.

[5] We run our experiments on one server containing 2 RTX A6000s with 49GB GPU memory each.

[6] https://www.cs.cmu.edu/~ark/QA-data/data/Question_Answer_Dataset_v1.2.tar.gz.

its corresponding Wikipedia source article included in the dataset.

**StrategyQA**[7] (Geva et al., 2021) is a complex boolean question answering dataset, constructed by presenting individual terms from Wikipedia to annotators. Its questions contain references to more than one Wikipedia entity, and necessitate implicit reasoning for binary (Yes/No) responses. The dataset comprises 5,111 answered questions initially intended for training question answering systems, with the system later tested on test set questions with unreleased answers. This training set is split into two subsets, based on the perceived challenge of questions by adversarial annotation models (Dua et al., 2019), resulting in `train` and `train_filtered` subsets containing 2,290 and 2,821 questions, respectively.

### 4.3 Evaluation

We evaluate the performance of the retrieval methods using the following metrics:

- nDCG@$k$ (normalized Discounted Cumulative Gain at rank $k$; Järvelin and Kekäläinen, 2002) evaluates the quality of a ranking system by considering both the relevance and the position of documents in the top $k$ results. Mathematically, it is represented as

$$\text{nDCG@}k = \frac{\sum_{i=1}^{k} \frac{2^{r_i}-1}{\log_2(i+1)}}{\sum_{i=1}^{|REL_k|} \frac{2^{r_i}-1}{\log_2(i+1)}}$$

  Where, $r_i$ denotes the relevance score of a document for a question, with relevance score $r_i = 1$ if the document contains a normalized form of the expected answer to the question, and $r_i = 0$, otherwise. And, $REL_k$ refers to a subset of the retrieved documents that contain a normalized form of the expected answer. nDCG@$k$ scores range between 0 and 1, where a score of 1 signifies an optimal ranking with the most relevant documents positioned at the top.

- MRR (Mean Reciprocal Rank; Voorhees and Harman, 1999) is the average of the reciprocal ranks of the first relevant document for each question. Mathematically, it is represented as

$$\text{MRR} = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \frac{1}{r_i}$$

  where $|Q|$ represents the total number of questions and $r_i$ denotes the rank of the first relevant document for the $i$-th question.

- Top-$k$ Retrieval Accuracy, as reported by Sciavolino et al. (2021), is calculated as the number of questions with at least one relevant document in the top $k$ retrieved documents divided by the total number of questions in the dataset.

We evaluate the performance of the retrieval-augmented question-answering models with each retrieval method as follows:

- For FactoidQA and EntityQuestions datasets, we use `OpenQA-eval` (Kamalloo et al., 2023) scripts to evaluate model performance, and report exact match (EM) and F1 scores by comparing expected answers to normalized model responses.

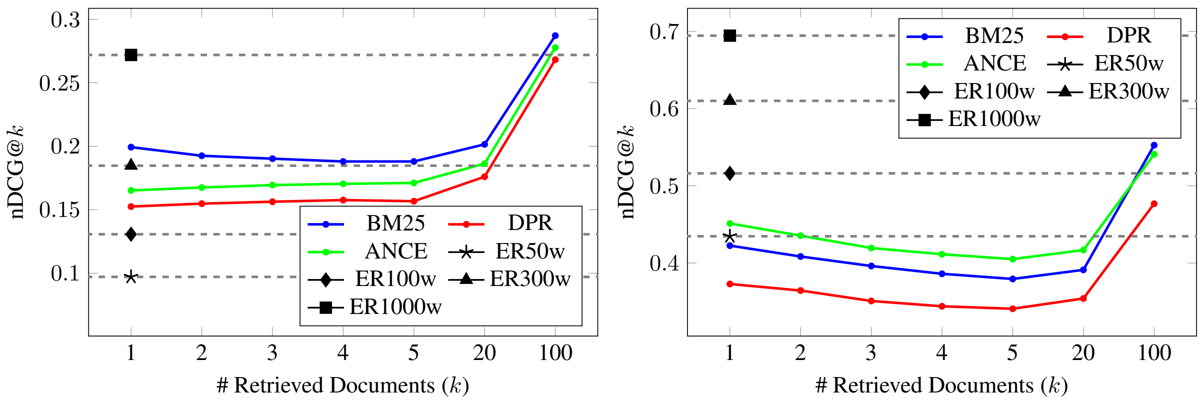
- For StrategyQA, we present accuracy scores by comparing model responses to the expected boolean answers in the dataset. As well, to assess model comprehension of the task, we count the number of answers that deviate from Yes or No and report this count in a distinct column labeled "`Inv #`" for each experiment.

### 4.4 Entity Retrieval Performance using Question Entity Annotations

We begin our analysis by comparing *Entity Retrieval* performance using BM25, DPR, and ANCE. For this experiment, we calculate nDCG with various retrieved document sets of size $k$ = 1, 2, 3, 4, 5, 20, and 100 documents. We use the entity annotations provided with the questions from FactoidQA and the dev set of EntityQuestions to fetch their corresponding Wikipedia articles, excluding StrategyQA from our analysis as it does not include entity annotations. On average, the FactoidQA and EntityQuestions datasets contain one salient entity per question.

To evaluate the effect of document length, we compare *Entity Retrieval* with the first 100 words (equivalent to the size of documents returned by BM25, DPR, and ANCE; noted as *ER100w*) and also consider the first 50, 300, and 1000 words of the retrieved Wikipedia articles (noted as *ER50w*, *ER300w*, and *ER1000w*). An *Entity Retrieval* document with 300 words has the same word count as three documents returned by BM25 or DPR.

[7] https://allenai.org/data/strategyqa.




(a) FactoidQA

(b) EntityQuestions (dev)

Figure 2: nDCG@$k$ scores comparing the quality of BM25, DPR, ANCE, and *Entity Retrieval* by considering both the relevance and the position of documents in the top $k$ retrieved passages for each question.

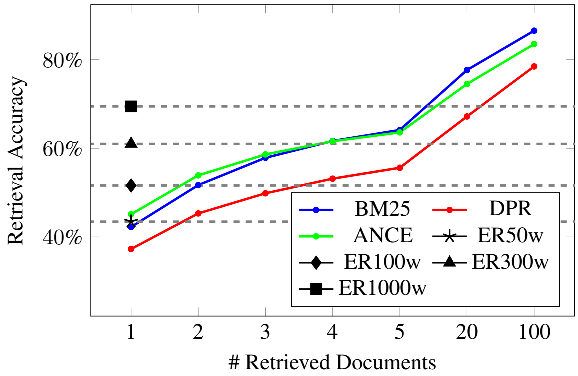



Figure 3: Retrieval Accuracy scores showcasing the correlation between the number of retrieved documents and the expected answers' coverage in EntityQuestions (dev) subset.

| | **FactoidQA** | **EntityQuestions (dev)** |
|---|---|---|
| BM25 | 0.245 | 0.522 |
| DPR | 0.209 | 0.456 |
| ANCE | 0.222 | 0.536 |
| ER50w | 0.097 | 0.435 |
| ER100w | 0.131 | 0.516 |
| ER300w | 0.185 | 0.610 |
| ER1000w | **0.272** | **0.695** |

Table 1: MRR scores comparing the retrieval quality of BM25, DPR, ANCE, and *Entity Retrieval* through the average of the reciprocal ranks of the first relevant document for each question.

Figure 2 presents the computed nDCG@$k$ scores across varying document sizes, highlighting the superior performance of *Entity Retrieval* over other retrieval methods in the context of the entity-centric datasets under study. Notably, *ER1000w*, which corresponds to ten BM25 retrieved passages in terms of word count, exhibits a retrieval performance on par with 100 retrieved documents in FactoidQA and surpasses BM25, the top-performing retriever on EntityQuestions, by 25%. This impressive performance by *Entity Retrieval* can be attributed to its ability to retrieve fewer, yet more relevant, documents. This observation aligns with the conclusion drawn by Cuconasu et al. (2024), which emphasizes that the retrieval of irrelevant documents can negatively impact performance. *Entity Retrieval* effectively minimizes the retrieval of such documents. Further insights can be gleaned from the comparison of nDCG scores along the x-axis of the plots in Figure 2. As the number of retrieved documents increases, the likelihood of retrieving irrelevant documents also rises, leading to a decline in retrieval performance when moving from 1 to 5 retrieved documents.

Table 1 showcases the calculated MRR scores, emphasizing the quicker attainment of relevant retrieval documents in *Entity Retrieval* compared to other retrieval methods. Concurrently, Figure 3 illustrates the impact of incrementing the number of retrieved documents on the expansion of the expected answers' coverage for the EntityQuestions dev subset.

While it may be appealing to consider 100 or more documents to simultaneously enhance both nDCG and Retrieval Accuracy, it is important to note that 100 retrieved documents would comprise 10,000 words. This could potentially overwhelm the model with excessive noise (irrelevant documents), and as well, could make it extremely costly to execute retrieval-augmented question answering, especially when the cost of API calls is calculated per token. We would need at least 10,000 tokens (optimistically, assuming each word equates to only one token) in addition to the tokens in the question. These factors suggest that retrieving a few documents for each question is more beneficial.

Taking these considerations into account, along with the nDCG@$k$, MRR, and Retrieval Accuracy results from this section, we gain a comprehensive understanding of the trade-off between the quality of the retrieved documents, which diminishes as we consider more documents, and the answer coverage, which increases as the model has a higher chance of encountering the right document with the correct hint for the answer. Consequently, we opt for $k = 4$ as a default, and we will always retrieve the top-4 documents in our retrieval-augmented question answering experiments.

### 4.5 Retrieval-Augmented Question Answering

Next, we shift our focus to study the effectiveness of our proposed *Entity Retrieval* method compared to other retrieval methods in enhancing the quality of responses to entity-centric questions. In this section, we examine three distinct scenarios: (1) the Closed-book setting, where we use "`Answer these questions:`" as the task instruction, followed by the question, (2) the Retrieval-Augmented setting, where we use retrieved documents as a basis, followed by "`Based on these texts, answer these questions:`", and then the question, and (3) the *Entity Retrieval* with question entity annotations, which uses the same prompt as the retrieval-augmented setting. The only difference lies in the documents retrieved, as we have previously discussed.

The initial eight rows of Table 2 present the results of our experiments using LLaMA 3 (8B) model. Upon examining these results, it is evident that *ER100w*, the most analogous *Entity Retrieval* setting to other retrieval methods, outperforms in terms of both EM and F1 scores. This setting returns identical 100-word documents as the other retrieval methods. Furthermore, our dense retrieval results align with the observations of Sciavolino et al. (2021), asserting that entity-centric questions indeed challenge dense retrievers. Although the BM25 method proves successful in enhancing the results compared to the Closed-book setting, it is noteworthy that even *Entity Retrieval* with the initial 50 words of the articles corresponding to the salient entities within questions yields superior results. This is particularly significant when compared to other retrieval methods which necessitate indexing the entire knowledge base on disk and loading the index into memory; a process required in inference time where caching is not an option.

| **LLaMA3 (8B)** | **FactoidQA** | | **EntityQuestions** | | | |
|---|---|---|---|---|---|---|
| | | | **dev** | | **test** | |
| | **EM** | **F1** | **EM** | **F1** | **EM** | **F1** |
| Closed-book | 30.7 | 39.3 | 22.7 | 37.8 | 22.8 | 38.1 |
| Retrieval-Augmented QA | | | | | | |
| BM25 | 32.2 | 42.4 | 23.8 | 38.6 | 23.3 | 38.5 |
| DPR | 29.4 | 38.5 | 22.0 | 36.2 | 20.5 | 35.3 |
| ANCE | 30.5 | 40.0 | 23.1 | 37.9 | 22.7 | 37.9 |
| *Entity Retrieval* w/ Question Entity Annotations | | | | | | |
| ER50w | 34.2 | 43.5 | 24.9 | 41.2 | 23.9 | 41.0 |
| ER100w | 33.6 | 42.8 | **26.2** | **42.8** | **25.7** | **42.4** |
| ER300w | 33.7 | 43.0 | **26.2** | **42.8** | 25.3 | **42.4** |
| ER1000w | **35.1** | **44.9** | 25.2 | 41.9 | 24.5 | 41.3 |
| *Entity Retrieval* w/ SpEL Entity Annotations | | | | | | |
| ERSp50w | 29.7 | 38.6 | 24.3 | 39.2 | 24.0 | 39.7 |
| ERSp100w | 28.3 | 37.4 | 25.0 | 40.1 | 24.2 | 39.8 |
| ERSp300w | 26.8 | 35.6 | 24.4 | 39.7 | 24.6 | 40.2 |
| ERSp1000w | 21.3 | 30.4 | 24.4 | 39.7 | 23.0 | 39.2 |

Table 2: Question answering efficacy comparison between Closed-book and Retrieval-augmentation using BM25, DPR, ANCE, and *Entity Retrieval*. EM refers to the exact match between predicted and expected answers, disregarding punctuation and articles (a, an, the). Results represent the average of two runs with the margin of error values provided in Table 6 in the Appendix.

### 4.6 Entity Retrieval in absence of Question Entity Annotations

In this section, we concentrate on the most crucial component of the *Entity Retrieval* method: the salient entities within entity-centric questions. We explore a scenario where the entities are not explicitly provided in the question, suggesting the use of an entity linking method to extract these entities. Ideally, we would like to evaluate all recent entity linking methods to identify the most effective one. However, due to time and budget limitations, we depend on the recent benchmarking studies by Ong et al. (2024) to choose an entity linking method. They examine the latest entity linking methods in terms of performance against unseen data and endorse SpEL (Shavarani and Sarkar, 2023) as the top performer. Consequently, we investigate *Entity Retrieval* using entities identified with SpEL, while reserving the examination of other entity linking

| | | |
|---|---|---|
| Question | Who performed Alexis Colby? | What is the capital of Seine-Saint-Denis? |
| Answer | Joan Collins | Bobigny |
| Closed-Book | Diana Ross | Paris |
| BM25 | Linda Evans | Saint-Denis |
| DPR | Alexis Cohen | Saint-Denis |
| ANCE | Nicollette Sheridan performed Alexis Colby. | Saint-Denis |
| ERSp100w | Joan Collins | Bobigny |
| Question | Where did John Snetzler die? | Where was Brigita Bukovec born? |
| Answer | Schaffhausen | Ljubljana |
| Closed-Book | He died in London, England, in 178 | Brigita Bukovec was born in Slovenia |
| BM25 | John Snetzler died in London. | Slovenia |
| DPR | John Snetzler died in London | in Slovakia |
| ANCE | in England | Rîbnița |
| ERSp100w | Schaffhausen | Ljubljana |

Table 3: Example questions from EntityQuestions (dev) to demonstrate the performance of *Entity Retrieval*.

techniques for *Entity Retrieval* for future research.

We maintain the *Entity Retrieval* settings as before, defining *ERSp50w*, *ERSp100w*, *ERSp300w*, and *ERSp1000w* for performing entity linking with SpEL, then retrieving the Wikipedia articles corresponding to the SpEL identified entities, and using the first 50, 100, 300, and 1000 words of these articles as documents to augment the question when prompting the LLM.

Passing the questions from our datasets to SpEL for analysis, we find that it generates a maximum of 8, 3, and 4 annotations for FactoidQA, EntityQuestions, and StrategyQA, respectively. On average, it produces 0.8, 0.7, and 1.1 annotations per question for these same datasets. SpEL successfully identifies and links entities in 56.5% of FactoidQA questions (1244/2203), 66.0% of EntityQuestions (dev) questions (3108/4710), 65.3% of EntityQuestions (test) questions (3095/4741), 75.8% of StrategyQA (train) questions (1735/2290), and 74.2% of StrategyQA (train_filtered) questions (2094/2821).

The final four rows of Table 2 showcase the comparative results of utilizing entities identified by SpEL for *Entity Retrieval*. Given that one-third of EntityQuestions and approximately half of FactoidQA lack identified annotations, the exact match scores reveal that *Entity Retrieval* performs robustly and surpasses BM25, the top-performing competitor retrieval method, for the entity-centric question-answering datasets under examination. This underscores the potential of *Entity Retrieval* within this paradigm. In addition, the disparity between the results with and without question entity annotations strongly indicates the necessity for further research in the Entity Linking domain, which could enhance entity-centric question answering as a downstream task. Table 3 provides some example questions where *Entity Retrieval* has led to better answers.

| **LLaMA3 (8B)** | **train** | | **train_filtered** | |
|---|---|---|---|---|
| | **Acc.** | **Inv #** | **Acc.** | **Inv #** |
| BM25 | 43.8 | 601 | 49.1 | 679 |
| ANCE | 47.0 | 550 | 51.8 | 637 |
| ERSp50w | 49.7 | 378 | 56.2 | 417 |
| ERSp100w | **50.5** | **367** | **56.6** | **389** |
| ERSp300w | 46.2 | 508 | 53.9 | 538 |
| ERSp1000w | 40.2 | 778 | 43.2 | 924 |

Table 4: Comparison of *Entity Retrieval* using SpEL identified entities to the best-performing dense and sparse retrieval methods of Table 2 on the StrategyQA dataset. Given the expected boolean results for StrategyQA questions, we restricted LLaMA 3 to generate only one token. *Acc.* indicates the fraction of answers that correctly match the expected Yes or No responses in the dataset, while *Inv #* represents the count of labels that are neither Yes nor No, but another invalid answer. Results represent the average of two runs with the margin of error values provided in Table 7 in the Appendix.

Table 4 presents a comparison of the performance of *Entity Retrieval* using SpEL identified entities against other retrieval methods on the StrategyQA dataset. The results clearly demonstrate the superior performance of *Entity Retrieval* over the top-performing retrieval methods as shown in Table 2. It is important to note that the 100-word setting (*ERSp100w*) is the most analogous to other retrieval methods, given that the size of their retrieved documents is also 100 words. Interestingly, the results from the 1000-word setting suggest that longer documents do not necessarily enhance the

| | Total Time | Disk Storage | Main Memory |
|---|---|---|---|
| BM25 | 45min | 11GB | 2.3GB |
| ANCE | 960min | 61.5GB | 64.2GB |
| ERSp100w | 34min | 9.4GB | 6.3GB |

Table 5: Comparison of the required resources for each retrieval method in real-time execution. The reported total time values exclude the time taken to load the indexes and models, focusing solely on the time used to answer the questions.

model's recall. In fact, beyond a certain length, the model may become overwhelmed by the sheer volume of noise, leading to confusion. Lastly, the invalid count values suggest that *Entity Retrieval* is more effective in assisting the model to comprehend the boolean nature of expected responses, eliminating the need to rely on retrieval from millions of passages.

### 4.7 Real-time Efficiency Analysis

Our analysis thus far has primarily focused on the retrieval performance, without consideration for the time and memory efficiency; crucial factors in retrieval method selection. In this section, we shift our focus to these aspects.

We begin by replacing the pre-built cache with the original retrieval modules that were used in creating the retrieval cache document sets. We load the indexes and the necessary models for fetching the retrieval documents. We then record the peak main memory requirement of each method during the experiment. It is important to note that all retrieval methods primarily rely on main memory, with minimal differences in GPU memory requirements. Therefore, we report an average GPU memory requirement of 35GB for the LLaMA 3 (8B) setting and exclude it from our results table. We then feed all 2,203 FactoidQA questions into the BM25, ANCE, and *Entity Retrieval* (using SpEL identified entities) to fetch the top-4 documents. We report the total time taken to generate answers to all the questions. Additionally, we keep track of all the pre-built models and indexes that each method requires for download and storage. We report the total size of all downloaded files to disk.

Table 5 presents our findings on time and memory requirements. It is evident that ANCE requires significantly more time to fetch and provide documents, six times more disk space to store its indexes, and over ten times higher main memory demands to load its dense representations. In contrast, BM25 and *Entity Retrieval* are more resource-friendly. Notably, *Entity Retrieval* is 25% faster than BM25 in response generation while demanding the total memory and disk space of a standard personal computer. Future research can be directed towards reducing the memory requirements of *Entity Retrieval*; a direction which we find quit promising.

## 5 Related Work

Similar to our studies, Kandpal et al. (2023) investigate the impact of salient entities on question answering, and propose constructing oracle retrieval documents as the 300-word segment surrounding the ground-truth answer from the Wikipedia page that contains the answer (entity name). Our approach leverages salient entities from questions without directly involving answers. Additionally, they primarily use entities to classify questions into those concerning frequent knowledge base entries versus those about rare entries on the long-tail, whereas our approach assigns a more substantial role to entities, treating them as pointers guiding the retrieval of relevant documents to augment questions.

Sciavolino et al. (2021) compare DPR and BM25 retrievers for entity-centric questions, and demonstrate that DPR greatly underperforms BM25. They attribute this to dense retrievers' difficulty with infrequent entities, which are less represented in training data. In contrast, BM25's frequency-based retrieval is not sensitive to entity frequency. We take a parallel approach and propose a simple yet effective method that leverages salient entities in the question for identifying augmentation documents.

## 6 Conclusion

In this study, we focused on retrieval-augmented question answering, and explored various retrieval methods that rely on the similarity between the question and the content of the passages to be retrieved. We introduced a novel approach, *Entity Retrieval*, which deviates from the conventional similarity mechanism. Instead, it capitalizes on the salient entities within the question to identify retrieval documents. Our findings indicate that our proposed method is not only more accurate but also faster in the context of entity-centric question answering.

## Limitations and Ethical Considerations

Our proposed *Entity Retrieval* method is specifically tailored for answering entity-centric questions, with its performance heavily reliant on the presence of question entities. In scenarios where entity annotations are absent, the method's effectiveness is directly tied to the performance of external entity linking methods. We acknowledge that our exploration of potential entity linking methods has not been exhaustive, and further investigation may yield insights that could enhance the *Entity Retrieval* method, even in the absence of question entity annotations.

Furthermore, we recognize that entity linking can occasionally result in ambiguous entities. Our research has not delved into the impact of such ambiguities on the *Entity Retrieval* method, and we propose that future studies should focus on ensuring the selection of the most contextually appropriate entities for retrieval.

Our research is primarily centered on Wikipedia as the knowledge base, a choice heavily influenced by previous studies for the sake of comparability. However, we acknowledge the importance of exploring other knowledge bases and ontologies, particularly in different domains, such as UMLS (Bodenreider, 2004) in the medical field.

In terms of benchmarking, we have compared the *Entity Retrieval* method against a limited selection of existing retrieval methods, guided by our judgement, experience, and considerations of implementation availability. We concede that our comparison has not been exhaustive, and this reasoning extends to our comparison using different LLMs and their available sizes.

Our research is on English only, and we acknowledge that entity-centric question answering in other languages is also relevant and important. We hope to extend our work to cover multiple languages in the future. We inherit the biases that exist in the data used in this project, and we do not explicitly de-bias the data. We are providing our code to the research community and we trust that those who use the model will do so ethically and responsibly.

## References


Steven Abney, Michael Collins, and Amit Singhal. 2000. Answer extraction. In *Sixth Applied Natural Language Processing Conference*, pages 296–301, Seattle, Washington, USA. Association for Computational Linguistics.

Ahmad Aghaebrahimian and Filip Jurčíček. 2016. Open-domain factoid question answering via knowledge graph search. In *Proceedings of the Workshop on Human-Computer Question Answering*, pages 22–28, San Diego, California. Association for Computational Linguistics.

Keivan Alizadeh, Iman Mirzadeh, Dmitry Belenko, Karen Khatamifard, Minsik Cho, Carlo C Del Mundo, Mohammad Rastegari, and Mehrdad Farajtabar. 2023. Llm in a flash: Efficient large language model inference with limited memory. *arXiv preprint arXiv:2312.11514*.

Olivier Bodenreider. 2004. The unified medical language system (umls): integrating biomedical terminology. *Nucleic acids research*, 32(suppl_1):D267–D270.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading Wikipedia to answer open-domain questions. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1870–1879, Vancouver, Canada. Association for Computational Linguistics.

Florin Cuconasu, Giovanni Trappolini, Federico Siciliano, Simone Filice, Cesare Campagnano, Yoelle Maarek, Nicola Tonellotto, and Fabrizio Silvestri. 2024. The power of noise: Redefining retrieval for rag systems. *arXiv preprint arXiv:2401.14887*.

Wanyun Cui, Yanghua Xiao, Haixun Wang, Yangqiu Song, Seung-won Hwang, and Wei Wang. 2017. Kbqa: Learning question answering over qa corpora and knowledge bases. *Proceedings of the VLDB Endowment*, 10(5).

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.

Matthijs Douze, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvasy, Pierre-Emmanuel Mazaré, Maria Lomeli, Lucas Hosseini, and Hervé Jégou. 2024. The faiss library. *arXiv preprint arXiv:2401.08281*.

Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. 2019.

DROP: A reading comprehension benchmark requiring discrete reasoning over paragraphs. 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 2368–2378, Minneapolis, Minnesota. Association for Computational Linguistics.

Mor Geva, Daniel Khashabi, Elad Segal, Tushar Khot, Dan Roth, and Jonathan Berant. 2021. Did aristotle use a laptop? a question answering benchmark with implicit reasoning strategies. *Transactions of the Association for Computational Linguistics*, 9:346–361.

Michael Glass, Gaetano Rossiello, Md Faisal Mahbub Chowdhury, Ankita Naik, Pengshan Cai, and Alfio Gliozzo. 2022. Re2G: Retrieve, rerank, generate. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2701–2715, Seattle, United States. Association for Computational Linguistics.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2022. Unsupervised dense information retrieval with contrastive learning. *Transactions on Machine Learning Research*.

Gautier Izacard and Edouard Grave. 2021. Leveraging passage retrieval with generative models for open domain question answering. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 874–880, Online. Association for Computational Linguistics.

Kalervo Järvelin and Jaana Kekäläinen. 2002. Cumulated gain-based evaluation of ir techniques. *ACM Transactions on Information Systems (TOIS)*, 20(4):422–446.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*, 7(3):535–547.

Ehsan Kamalloo, Nouha Dziri, Charles Clarke, and Davood Rafiei. 2023. Evaluating open-domain question answering in the era of large language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5591–5606, Toronto, Canada. Association for Computational Linguistics.

Nikhil Kandpal, Haikang Deng, Adam Roberts, Eric Wallace, and Colin Raffel. 2023. Large language models struggle to learn long-tail knowledge. In *International Conference on Machine Learning*, pages 15696–15707. PMLR.

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.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020a. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020b. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474.

Jimmy Lin, Xueguang Ma, Sheng-Chieh Lin, Jheng-Hong Yang, Ronak Pradeep, and Rodrigo Nogueira. 2021. Pyserini: A Python toolkit for reproducible information retrieval research with sparse and dense representations. In *Proceedings of the 44th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2021)*, pages 2356–2362.

Denis Lukovnikov, Asja Fischer, and Jens Lehmann. 2019. Pretrained transformers for simple question answering over knowledge graphs. In *The Semantic Web–ISWC 2019: 18th International Semantic Web Conference, Auckland, New Zealand, October 26–30, 2019, Proceedings, Part I 18*, pages 470–486. Springer.

Salman Mohammed, Peng Shi, and Jimmy Lin. 2018. Strong baselines for simple question answering over knowledge graphs with and without neural networks. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 291–296, New Orleans, Louisiana. Association for Computational Linguistics.

Jianmo Ni, Chen Qu, Jing Lu, Zhuyun Dai, Gustavo Hernandez Abrego, Ji Ma, Vincent Zhao, Yi Luan, Keith Hall, Ming-Wei Chang, and Yinfei Yang. 2022. Large dual encoders are generalizable retrievers. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9844–9855, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Nicolas Ong, Hassan S. Shavarani, and Anoop Sarkar. 2024. Unified examination of entity linking in absence of candidate sets. In *Proceedings of the 2024 Conference of the North American Chapter of the*

*Association for Computational Linguistics: Human Language Technologies*, Mexico. Association for Computational Linguistics.

Baolin Peng, Michel Galley, Pengcheng He, Hao Cheng, Yujia Xie, Yu Hu, Qiuyuan Huang, Lars Liden, Zhou Yu, Weizhu Chen, et al. 2023. Check your facts and try again: Improving large language models with external knowledge and automated feedback. *arXiv preprint arXiv:2302.12813*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Ori Ram, Yoav Levine, Itay Dalmedigos, Dor Muhlgay, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. 2023. In-context retrieval-augmented language models. *Transactions of the Association for Computational Linguistics*, 11:1316–1331.

Prakash Ranjan and Rakesh Chandra Balabantaray. 2016. Question answering system for factoid based question. In *2016 2nd International Conference on Contemporary Computing and Informatics (IC3I)*, pages 221–224. IEEE.

Adam Roberts, Colin Raffel, and Noam Shazeer. 2020. How much knowledge can you pack into the parameters of a language model? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5418–5426, Online. Association for Computational Linguistics.

Stephen Robertson, Hugo Zaragoza, et al. 2009. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389.

Stephen E. Robertson, Steve Walker, Susan Jones, Micheline Hancock-Beaulieu, and Mike Gatford. 1994. Okapi at trec-3. In *Text Retrieval Conference*.

Mikhail Salnikov, Hai Le, Prateek Rajput, Irina Nikishina, Pavel Braslavski, Valentin Malykh, and Alexander Panchenko. 2023. Large language models meet knowledge graphs to answer factoid questions. In *Proceedings of the 37th Pacific Asia Conference on Language, Information and Computation*, pages 635–644, Hong Kong, China. Association for Computational Linguistics.

Keshav Santhanam, Omar Khattab, Jon Saad-Falcon, Christopher Potts, and Matei Zaharia. 2022. ColBERTv2: Effective and efficient retrieval via lightweight late interaction. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3715–3734, Seattle, United States. Association for Computational Linguistics.

Christopher Sciavolino, Zexuan Zhong, Jinhyuk Lee, and Danqi Chen. 2021. Simple entity-centric questions challenge dense retrievers. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6138–6148, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Hassan S. Shavarani and Anoop Sarkar. 2023. SpEL: Structured prediction for entity linking. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 11123–11137, Singapore. Association for Computational Linguistics.

Weijia Shi, Sewon Min, Michihiro Yasunaga, Minjoon Seo, Rich James, Mike Lewis, Luke Zettlemoyer, and Wen-tau Yih. 2023. Replug: Retrieval-augmented black-box language models. *arXiv preprint arXiv:2301.12652*.

Kurt Shuster, Spencer Poff, Moya Chen, Douwe Kiela, and Jason Weston. 2021. Retrieval augmentation reduces hallucination in conversation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3784–3803, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Devendra Singh, Siva Reddy, Will Hamilton, Chris Dyer, and Dani Yogatama. 2021. End-to-end training of multi-document reader and retriever for open-domain question answering. *Advances in Neural Information Processing Systems*, 34:25968–25981.

Noah A Smith, Michael Heilman, and Rebecca Hwa. 2008. Question generation as a competitive undergraduate course project. In *Proceedings of the NSF Workshop on the Question Generation Shared Task and Evaluation Challenge*, volume 9.

Ellen M Voorhees and Donna Harman. 1999. Overview of the eighth text retrieval conference (trec-8). In *Proceedings of the Eighth Text REtrieval Conference (TREC-8), NIST Special Publication*.

Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul N. Bennett, Junaid Ahmed, and Arnold Overwijk. 2021. Approximate nearest neighbor negative contrastive learning for dense text retrieval. In *International Conference on Learning Representations*.

Ikuya Yamada, Akari Asai, and Hannaneh Hajishirzi. 2021. Efficient passage retrieval with hashing for open-domain question answering. 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 979–986, Online. Association for Computational Linguistics.

Wenhao Yu, Zhihan Zhang, Zhenwen Liang, Meng Jiang, and Ashish Sabharwal. 2023. Improving language models via plug-and-play retrieval feedback. *arXiv preprint arXiv:2305.14002*.

Jingtao Zhan, Jiaxin Mao, Yiqun Liu, Min Zhang, and Shaoping Ma. 2020a. Learning to retrieve: How to train a dense retrieval model effectively and efficiently. *arXiv preprint arXiv:2010.10469*.

Jingtao Zhan, Jiaxin Mao, Yiqun Liu, Min Zhang, and Shaoping Ma. 2020b. Repbert: Contextualized text embeddings for first-stage retrieval. *arXiv preprint arXiv:2006.15498*.

Qin Zhang, Shangsi Chen, Dongkuan Xu, Qingqing Cao, Xiaojun Chen, Trevor Cohn, and Meng Fang. 2023. A survey for efficient open domain question answering. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14447–14465, Toronto, Canada. Association for Computational Linguistics.

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*.

## A Margin of Error Results

| LLaMA3 (8B)* | FactoidQA | | EntityQuestions | | | |
|---|---|---|---|---|---|---|
| | | | dev | | test | |
| | EM | F1 | EM | F1 | EM | F1 |
| Closed-book | 30.7±0.1 | 39.3±0.0 | 22.7±0.5 | 37.8±1.0 | 22.8±0.1 | 38.1±0.6 |
| Retrieval-Augmented QA | | | | | | |
| BM25 | 32.2±1.1 | 42.4±0.2 | 23.8±0.3 | 38.6±0.8 | 23.3±0.0 | 38.5±0.1 |
| DPR | 29.4±1.0 | 38.5±1.2 | 22.0±0.1 | 36.2±0.2 | 20.5±0.4 | 35.3±0.6 |
| ANCE | 30.5±0.4 | 40.0±0.4 | 23.1±0.7 | 37.9±0.6 | 22.7±0.7 | 37.9±0.9 |
| *Entity Retrieval* w/ Question Entity Annotations | | | | | | |
| ER50w | 34.2±0.7 | 43.5±0.6 | 24.9±0.2 | 41.2±0.0 | 23.9±0.5 | 41.0±0.1 |
| ER100w | 33.6±0.5 | 42.8±0.5 | **26.2**±0.0 | **42.8**±0.1 | **25.7**±0.1 | **42.4**±0.0 |
| ER300w | 33.7±1.4 | 43.0±1.7 | **26.2**±0.4 | **42.8**±0.0 | 25.3±1.0 | **42.4**±1.1 |
| ER1000w | **35.1**±0.4 | **44.9**±0.7 | 25.2±0.4 | 41.9±0.6 | 24.5±0.9 | 41.3±0.6 |
| *Entity Retrieval* w/ SpEL Entity Annotations | | | | | | |
| ERSp50w | 29.7±0.3 | 38.6±0.7 | 24.3±0.2 | 39.2±0.1 | 24.0±0.1 | 39.7±0.0 |
| ERSp100w | 28.3±0.9 | 37.4±1.2 | 25.0±0.4 | 40.1±0.3 | 24.2±0.2 | 39.8±0.1 |
| ERSp300w | 26.8±0.6 | 35.6±0.7 | 24.4±0.0 | 39.7±0.1 | 24.6±0.3 | 40.2±0.5 |
| ERSp1000w | 21.3±0.5 | 30.4±0.8 | 24.4±0.1 | 39.7±0.1 | 23.0±0.7 | 39.2±0.7 |

Table 6: Question answering efficacy comparison between Closed-book and Retrieval-augmentation using BM25, DPR, ANCE, and *Entity Retrieval*. EM refers to the exact match between predicted and expected answers, disregarding punctuation and articles (`a`, `an`, `the`).
* Results represent the average of two runs, accompanied by a margin of error based on a 99% confidence interval.

| LLaMA3 (8B)* | train | | train_filtered | |
|---|---|---|---|---|
| | Acc. | Inv # | Acc. | Inv # |
| BM25 | 43.8±0.1 | 601±4 | 49.1±1.0 | 679±7 |
| ANCE | 47.0±1.2 | 550±15 | 51.8±1.0 | 637±42 |
| ERSp50w | 49.7±1.2 | 378±34 | 56.2±1.3 | 417±31 |
| ERSp100w | **50.5**±2.0 | **367**±21 | **56.6**±0.5 | **389**±1 |
| ERSp300w | 46.2±1.9 | 508±22 | 53.9±1.9 | 538±14 |
| ERSp1000w | 40.2±0.4 | 778±3 | 43.2±0.3 | 924±13 |

Table 7: Comparison of *Entity Retrieval* using SpEL identified entities to the best-performing dense and sparse retrieval methods on the StrategyQA dataset.
* Results represent the average of two runs, accompanied by a margin of error based on a 99% confidence interval.