# Leveraging Passage Retrieval with Generative Models for Open Domain Question Answering

**Gautier Izacard**[1,2,3]      **Edouard Grave**[1]

[1] Facebook AI Research, Paris
[2] ENS, PSL University, Paris
[3] Inria, Paris
`gizacard|egrave@fb.com`

## Abstract

Generative models for open domain question answering have proven to be competitive, without resorting to external knowledge. While promising, this approach requires to use models with billions of parameters, which are expensive to train and query. In this paper, we investigate how much these models can benefit from retrieving text passages, potentially containing evidence. We obtain state-of-the-art results on the Natural Questions and TriviaQA open benchmarks. Interestingly, we observe that the performance of this method significantly improves when increasing the number of retrieved passages. This is evidence that sequence-to-sequence models offers a flexible framework to efficiently aggregate and combine evidence from multiple passages.

## 1 Introduction

Recently, several works have shown that factual information can be extracted from large scale language models trained on vast quantities of data (Radford et al., 2019; Petroni et al., 2019; Jiang et al., 2019; Talmor et al., 2019). Building on that observation and the advances in pretraining of natural language processing models, Roberts et al. (2020) introduced a generative model for open domain question answering. Without relying on external knowledge, this method obtained competitive results on several benchmarks. However, it requires models containing billions of parameters, since all the information needs to be stored in the weights. This makes models expensive to query and train. In this paper, we investigate how much this method could benefit from having access to an external source of knowledge, such as Wikipedia.

Retrieval based approaches were previously considered in the context of open domain question answering with extractive models (Chen et al., 2017). In that case, systems start by retrieving
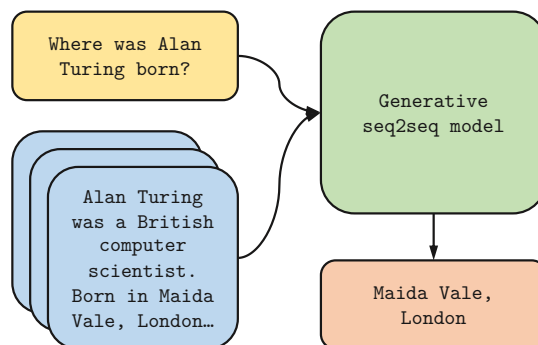


Figure 1: A simple approach to open domain question answering. First, it retrieves support text passages from an external source of knowledge such as Wikipedia. Then, a generative encoder-decoder model produces the answer, conditioned on the question and the retrieved passages. This approach scales well with the number of retrieved passages, as the performance keeps improving when retrieving up to one hundred passages.

support documents, before extracting the answer from these documents. Different retrieval techniques have been considered, either using sparse representations based on TF/IDF or using dense embeddings (Guu et al., 2020; Karpukhin et al., 2020). The models which extract the answers are often based on contextualized word representations such as ELMo or BERT (Peters et al., 2018; Devlin et al., 2019), and predict a span as answer. Aggregating and combining evidence from multiple passages is not straightforward when using extractive models, and multiple techniques have been proposed to address this limitation (Clark and Gardner, 2018; Min et al., 2019a).

In this paper, we explore a simple approach having the best of both worlds, by building on the exciting developments in generative modeling and retrieval for open domain question answering. This method proceeds in two steps, by first retrieving supporting passages using either sparse or dense
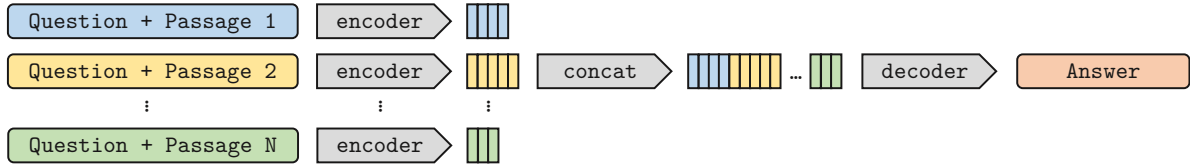
Figure 2: Architecture of the Fusion-in-Decoder method.

representations. Then, a sequence-to-sequence model generates the answer, taking as input the retrieved passages in addition to the question. While conceptually simple, this method sets new state-of-the-art results on the TriviaQA and NaturalQuestions benchmarks. In particular, we show that the performance of our method significantly improves when the number of retrieved passages increases. We believe that this is evidence that generative models are good at combining evidence from multiple passages, compared to extractive ones.

## 2   Related work

**Open domain question answering** is the task of answering general domain questions, in which the evidence is not given as input to the system. While being a longstanding problem in natural language processing (Voorhees et al., 1999), this task has recently regained interest following the work by Chen et al. (2017). In that version of the problem, strong supervision is available to the learning system, in the form of spans corresponding to answers. Chen et al. (2017) proposed to solve the problem by first retrieving support document from Wikipedia, before extracting the answer from the retrieved document. Different methods were proposed to tackle the setting where no gold spans are given to the system, but only the correct answer. Clark and Gardner (2018) proposed to use a global normalization over all the span corresponding to the answer, which was later applied to BERT based models (Wang et al., 2019). Min et al. (2019a) introduced a method based on hard expectation-maximization to tackle noisy supervision from this setting. Wang et al. (2018b) described a technique to aggregate answers from different paragraphs, using confidence and coverage scores.

**Passage retrieval** is an important step in open domain question answering, and is an active area of research to improve QA systems. Initially, sparse representations based on TF/IDF were used to retrieve support documents (Chen et al., 2017). Lee et al. (2018) introduced a supervised learning method to rerank paragraphs based on BiLSTM, while Wang et al. (2018a) trained a ranking system with reinforcement learning. A second approach to improve the retrieval step of QA systems is to used additional information such as the Wikipedia or Wikidata graphs (Min et al., 2019b; Asai et al., 2020). Recently, multiple works show that retrieval systems entirely based on dense representation and approximate nearest neighbors were competitive with traditional approaches. Such models can be trained using weak supervision in the form of question-answer pairs (Karpukhin et al., 2020), or pretrained using a cloze task and finetuned end-to-end (Guu et al., 2020; Lee et al., 2019).

**Generative question answering** was mostly considered in previous work for datasets requiring to generate answers, such as NarrativeQA (Kočiskỳ et al., 2018), CoQA (Reddy et al., 2019) or ELI5 (Fan et al., 2019). These datasets were generated in a way that answers do not correspond to spans in support documents, thus requiring abstractive models. Raffel et al. (2019) showed that generative models are competitive for reading comprehension tasks such as SQuAD (Rajpurkar et al., 2016), where answers are spans. Roberts et al. (2020) proposed to use large pretrained generative models, without using additional knowledge, for open domain question answering. Closest to our work, Min et al. (2020) and Lewis et al. (2020) introduced retrieval augmented generative models for open domain question answering. Our approach differs from these works by how the generative model processes the retrieved passages. This allows to scale to large numbers of documents, and to benefit from this large amount of evidence.

## 3   Method

In this section, we describe our approach to open domain question answering. It proceeds in two steps, first retrieving support passages before processing them with a sequence to sequence model.

| Model | NQ EM | TriviaQA EM | TriviaQA EM | SQuAD Open EM | SQuAD Open F1 |
|---|---|---|---|---|---|
| DrQA (Chen et al., 2017) | - | - | - | 29.8 | - |
| Multi-Passage BERT (Wang et al., 2019) | - | - | - | 53.0 | 60.9 |
| Path Retriever (Asai et al., 2020) | 31.7 | - | - | **56.5** | **63.8** |
| Graph Retriever (Min et al., 2019b) | 34.7 | 55.8 | - | - | - |
| Hard EM (Min et al., 2019a) | 28.8 | 50.9 | - | - | - |
| ORQA (Lee et al., 2019) | 31.3 | 45.1 | - | 20.2 | - |
| REALM (Guu et al., 2020) | 40.4 | - | - | - | - |
| DPR (Karpukhin et al., 2020) | 41.5 | 57.9 | - | 36.7 | - |
| SpanSeqGen (Min et al., 2020) | 42.5 | - | - | - | - |
| RAG (Lewis et al., 2020) | 44.5 | 56.1 | 68.0 | - | - |
| T5 (Roberts et al., 2020) | 36.6 | - | 60.5 | - | - |
| GPT-3 few shot (Brown et al., 2020) | 29.9 | - | 71.2 | - | - |
| Fusion-in-Decoder (base) | 48.2 | 65.0 | 77.1 | 53.4 | 60.6 |
| Fusion-in-Decoder (large) | **51.4** | **67.6** | **80.1** | **56.7** | 63.2 |

Table 1: Comparison to state-of-the-art. On TriviaQA, we report results on the open domain test set (left), and on the hidden test set (right), `competitions.codalab.org/competitions/17208#results`).

**Retrieval.** For the retrieval of support passages, we consider two methods: BM25 (Robertson et al., 1995) and DPR (Karpukhin et al., 2020). In BM25, passages are represented as bag of words, and the ranking function is based on term and inverse document frequencies. We use the implementation from Apache Lucene[1] with default parameters, and tokenize questions and passages with SpaCy.[2] In DPR, passages and questions are represented as dense vector representations, computed using two BERT networks. The ranking function is the dot product between the query and passage representations. Retrieval is performed using approximate nearest neighbors with the FAISS library.[3]

**Reading.** Our generative model for open domain QA is based on a sequence-to-sequence network, pretrained on unsupervised data, such as T5 or BART (Raffel et al., 2019; Lewis et al., 2019). The model takes as input the question, as well as the support passages, and generates the answer. More precisely, each retrieved passage and its title are concatenated with the question, and processed independently from other passages by the encoder. We add special tokens `question:`, `title:` and `context:` before the question, title and text of each passage. Finally, the decoder performs atten-

---
[1] `lucene.apache.org`
[2] `spacy.io`
[3] `github.com/facebookresearch/faiss`

tion over the concatenation of the resulting representations of all the retrieved passages. The model thus performs evidence fusion in the decoder only, and we refer to it as *Fusion-in-Decoder*.

By processing passages independently in the encoder, but jointly in the decoder, this method differs from Min et al. (2020) and Lewis et al. (2020). Processing passages independently in the encoder allows to scale to large number of contexts, as it only performs self attention over one context at a time. This means that the computation time of the model grows linearly with the number of passages, instead of quadratically. On the other hand, processing passages jointly in the decoder allows to better aggregate evidence from multiple passages.

## 4  Experiments

In this section, we report empirical evaluations of Fusion-in-Decoder for open domain QA.

**Datasets.** We consider the following datasets, and use the same setting as Lee et al. (2019):

- NaturalQuestions (Kwiatkowski et al., 2019) contains questions corresponding to Google search queries. The open-domain version of this dataset is obtained by discarding answers with more than 5 tokens.

- TriviaQA (Joshi et al., 2017) contains questions gathered from trivia and quiz-league
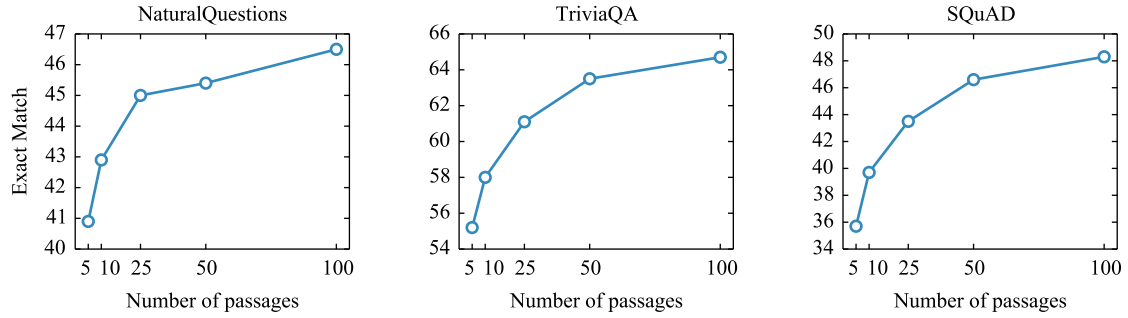
Figure 3: Performance of Fusion-in-Decoder (base) on valid sets as a function of the number of retrieved passages.

websites. The *unfiltered* version of TriviaQA is used for open-domain question answering.

- SQuAD v1.1 (Rajpurkar et al., 2016) is a reading comprehension dataset. Given a paragraph extracted from Wikipedia, annotators were asked to write questions, for which the answer is a span from the corresponding paragraph.

Following Lee et al. (2019) we use the validation as test, and keep 10% of the training set for validation. We use the Wikipedia dumps from Dec. 20, 2018 for NQ and TriviaQA and from Dec. 21, 2016 for SQuAD. We apply the same preprocessing as Chen et al. (2017); Karpukhin et al. (2020), leading to passages of 100 words, which do not overlap.

**Evaluation.** Predicted answers are evaluated with the standard exact match metric (EM), as introduced by Rajpurkar et al. (2016). A generated answer is considered correct if it matches any answer of the list of acceptable answers after normalization. This normalization step consists in lowercasing and removing articles, punctuation and duplicated whitespace.

**Technical details.** We initialize our models with the pretrained T5 models (Raffel et al., 2019), available in the HuggingFace Transformers library.[4] We consider two model sizes, base and large, containing respectively 220M and 770M parameters. We fine-tune the models on each dataset independently, using Adam (Kingma and Ba, 2014) with a constant learning rate of $10^{-4}$ and a dropout rate of 10%. We train the model for 10k gradient steps, with a batch size of 64, using 64 Tesla V100 32Gb. We evaluate models every 500 steps and select the best one on the validation set based on the Exact Match score. During training on NaturalQuestions

and SQuAD, we sample the target among the list of answers, while for TriviaQA, we use the unique human-generated answer. For TriviaQA, answers in uppercase are normalized by converting all letters in lowercase except the first letter of each word, using the `title` Python string method. For both training and testing, we retrieve 100 passages (unless said otherwise), and truncate them to 250 word pieces. Following the results of Karpukhin et al. (2020), passages are retrieved with DPR for NQ and TriviaQA, and with BM25 for SQuAD. We generate answers by using greedy decoding.

**Comparison to state-of-the-art.** In table 1, we compare the results obtained by Fusion-in-Decoder with existing approaches for open domain question answering. We observe that while conceptually simple, this method outperforms existing work on the NaturalQuestion and TriviaQA benchmarks. In particular, generative models seem to perform well when evidence from multiple passages need to be aggregated, compared to extractive approaches. Our method also performs better than other generative models, showing that scaling to large number of passages and processing them jointly leads to improvement in accuracy. Second, we observe that using additional knowledge in generative models by using retrieval lead to important performance gains. On NaturalQuestions, the *closed book* T5 model obtains 36.6% accuracy with 11B parameters, while our approach obtains 44.1% with 770M parameters plus Wikipedia with BM25 retrieval. Both methods use roughly the same amount of memory to store information, indicating that text based explicit memories are competitive for knowledge retrieval tasks.

**Scaling with number of passages.** In Figure 3, we report the performance with respect to the

---

| Training Passages | NaturalQuestions | | TriviaQA | |
|---|---|---|---|---|
| | w/o finetuning | w/ finetuning | w/o finetuning | w/ finetuning |
| 5 | 37.8 | 45.0 | 58.1 | 64.2 |
| 10 | 42.3 | 45.3 | 61.1 | 63.6 |
| 25 | 45.3 | 46.0 | 63.2 | 64.2 |
| 50 | 45.7 | 46.0 | 64.2 | 64.3 |
| 100 | 46.5 | - | 64.7 | - |

Table 2: Performance depending on the number of passages used during training. Exact Match scores are reported on dev sets.

number of retrieved passages. In particular, we observe that increasing the number of passages from 10 to 100 leads to 6% improvement on TriviaQA and 3.5% improvement on NaturalQuestions. On the other hand, the performance of most extractive models seems to peak around 10 to 20 passages (Wang et al., 2019; Yang et al., 2019). We believe that this is evidence that sequence-to-sequence models are good at combining informations from multiple passages.

**Impact of the number of training passages.** In the previous section, the model was trained and evaluated with the same number of passages. To reduce the training computational budget, a simple solution consists in training the model with fewer passages. In Table 2, we report the performance obtained by training with different numbers of passages, while testing with 100 passages. We observe that reducing the number of training passages leads to a decrease of accuracy. Further, we propose to finetune the previous models using 100 passages for 1000 steps. This allows to reduce the accuracy gap, while using significantly less computational resources: we can reach 46.0 EM on NaturalQuestions, using 147 GPU hours, compared to 425 GPU hours when training on 100 passages.

## 5 Conclusion

In this paper, we study a simple approach to open domain question answering, which relies on retrieving support passages before processing them with a generative model. We show that while conceptually simple, this approach is competitive with existing methods, and that it scales well with the number of retrieved passages. In future work, we plan to make this model more efficient, in particular when scaling to large number of support passages. We also plan to integrate the retrieval in our model, and to learn the whole system end-to-end.

## References

Akari Asai, Kazuma Hashimoto, Hannaneh Hajishirzi, Richard Socher, and Caiming Xiong. 2020. Learning to retrieve reasoning paths over wikipedia graph for question answering. In *Proc. ICLR*.

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading Wikipedia to answer open-domain questions. In *Proc. ACL*.

Christopher Clark and Matt Gardner. 2018. Simple and effective multi-paragraph reading comprehension. In *Proc. ACL*.

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

Angela Fan, Yacine Jernite, Ethan Perez, David Grangier, Jason Weston, and Michael Auli. 2019. ELI5: Long form question answering. In *Proc. ACL*.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. 2020. Realm: Retrieval-augmented language model pre-training. *arXiv preprint arXiv:2002.08909*.

Zhengbao Jiang, Frank F Xu, Jun Araki, and Graham Neubig. 2019. How can we know what language models know? *arXiv preprint arXiv:1911.12543*.

Mandar Joshi, Eunsol Choi, Daniel S. Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In *Proc. ACL*.

Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. *arXiv preprint arXiv:2004.04906*.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980.*

Tomáš Kočiský, Jonathan Schwarz, Phil Blunsom, Chris Dyer, Karl Moritz Hermann, Gábor Melis, and Edward Grefenstette. 2018. The NarrativeQA reading comprehension challenge. *TACL.*

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Matthew Kelcey, Jacob Devlin, Kenton Lee, Kristina N. Toutanova, Llion Jones, Ming-Wei Chang, Andrew Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural Questions: a benchmark for question answering research. *TACL.*

Jinhyuk Lee, Seongjun Yun, Hyunjae Kim, Miyoung Ko, and Jaewoo Kang. 2018. Ranking paragraphs for improving answer recall in open-domain question answering. In *Proc. EMNLP.*

Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. 2019. Latent retrieval for weakly supervised open domain question answering. In *Proc. ACL.*

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461.*

Patrick Lewis, Ethan Perez, Aleksandara Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *arXiv preprint arXiv:2005.11401.*

Sewon Min, Danqi Chen, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2019a. A discrete hard EM approach for weakly supervised question answering. In *Proc. EMNLP-IJCNLP.*

Sewon Min, Danqi Chen, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2019b. Knowledge guided text retrieval and reading for open domain question answering. *arXiv preprint arXiv:1911.03868.*

Sewon Min, Julian Michael, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2020. Ambigqa: Answering ambiguous open-domain questions. *arXiv preprint arXiv:2004.10645.*

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proc. NAACL.*

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language models as knowledge bases? In *Proc. EMNLP-IJCNLP.*

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Technical Report.*

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683.*

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proc. EMNLP.*

Siva Reddy, Danqi Chen, and Christopher D Manning. 2019. CoQA: A conversational question answering challenge. *TACL.*

Adam Roberts, Colin Raffel, and Noam Shazeer. 2020. How much knowledge can you pack into the parameters of a language model? *arXiv preprint arXiv:2002.08910.*

Stephen E Robertson, Steve Walker, Susan Jones, Micheline M Hancock-Beaulieu, Mike Gatford, et al. 1995. Okapi at TREC-3. *NIST Special Publication Sp.*

Alon Talmor, Yanai Elazar, Yoav Goldberg, and Jonathan Berant. 2019. oLMpics–on what language model pre-training captures. *arXiv preprint arXiv:1912.13283.*

Ellen M Voorhees et al. 1999. The TREC-8 question answering track report. In *TREC.*

Shuohang Wang, Mo Yu, Xiaoxiao Guo, Zhiguo Wang, Tim Klinger, Wei Zhang, Shiyu Chang, Gerry Tesauro, Bowen Zhou, and Jing Jiang. 2018a. R$^3$: Reinforced ranker-reader for open-domain question answering. In *Proc. AAAI.*

Shuohang Wang, Mo Yu, Jing Jiang, Wei Zhang, Xiaoxiao Guo, Shiyu Chang, Zhiguo Wang, Tim Klinger, Gerald Tesauro, and Murray Campbell. 2018b. Evidence aggregation for answer re-ranking in open-domain question answering. In *Proc. ICLR.*

Zhiguo Wang, Patrick Ng, Xiaofei Ma, Ramesh Nallapati, and Bing Xiang. 2019. Multi-passage BERT: A globally normalized BERT model for open-domain question answering. In *Proc. EMNLP-IJCNLP.*

Wei Yang, Yuqing Xie, Aileen Lin, Xingyu Li, Luchen Tan, Kun Xiong, Ming Li, and Jimmy Lin. 2019. End-to-end open-domain question answering with BERTserini. In *Proc. NAACL (Demonstrations).*