A Copy-Augmented Generative Model for Open-Domain Question Answering

Shuang Liu[1]*, Dong Wang[2], Xiaoguang Li[1], Minghui Huang[2], Meizhen Ding[2]

[1] Huawei Noah's Ark Lab

[2] AI Application Research Center (AARC)

Huawei Technologies Co., Ltd

{liushuang30, wangdong153}@huawei.com

Abstract

Open-domain question answering is a challenging task with a wide variety of practical applications. Existing modern approaches mostly follow a standard two-stage paradigm: retriever then reader. In this article, we focus on improving the effectiveness of the reader module and propose a novel copy-augmented generative approach that integrates the merits of both extractive and generative readers. In particular, our model is built upon the powerful generative model FiD (Izacard and Grave, 2021b). We enhance the original generative reader by incorporating a pointer network to encourage the model to directly copy words from the retrieved passages. We conduct experiments on the two benchmark datasets, NaturalQuestions and TriviaQA, and the empirical results demonstrate the performance gains of our proposed approach.

Question: where was a hologram for the king filmed?
Passages (Truncated): title: A Hologram for the King (film) context: Production was set to begin in first quarter of 2014. Principal photography commenced on March 6, 2014 in Morocco. *Filming also took place in Hurghada in Egypt, as well as in Berlin and Düsseldorf in Germany*. Shooting wrapped in June 2014.
Answer: Hurghada in Egypt, Berlin and Düsseldorf in Germany
FiD: Dubai in Germany
FiD-PGN: Hurghada in Egypt
Question: who has the most trophies in la liga?
Passages (Truncated): title: La Liga context: A total of 62 teams have competed in La Liga since its inception. *Nine teams have been crowned champions, with Real Madrid winning the title a record 33 times and Barcelona 25 times.*
Answer: Real Madrid
FiD: 33
FiD-PGN: Real Madrid

Table 1: Comparisons of answers generated by FiD and our approach. The orange text represents supportive sentences.

1 Introduction

Open-domain question answering (ODQA) focuses on providing highly precise answers to natural language questions from a large collection of unstructured text data (Voorhees, 1999). With the pioneering work of DrQA (Chen et al., 2017), modern approaches to ODQA commonly adopt a simple two-stage *retriever-reader* pipeline, that firstly retrieve a relatively small number of support passages (Karpukhin et al., 2020; Min et al., 2021b; Yamada et al., 2021), followed by the reader identifying the answer.

The reader models can be broadly categorized into two classes: extractive (Chen et al., 2017; Asai et al., 2020; Karpukhin et al., 2020) and generative (Izacard and Grave, 2021b; Lewis et al., 2020b; Wu et al., 2021). Recently, benefiting from the powerful ability of large-scale pre-trained encoder-decoder language models (Lewis et al., 2020a; Raffel et al., 2019) and the capability of aggregating information from multiple passages (Izacard

___
*This work was done when she was at AARC.

and Grave, 2021b), generative approaches have achieved in general better performance than extractive methods.

Compared to extractive models, generative models generate text more freely, which makes it often suffer from the problem of producing hallucinated text that is factual inaccuracy or inconsistent to the input. This problem has been addressed in tasks like text summarization (Maynez et al., 2020) and machine translation (Zhou et al., 2021). We found that the phenomenon also happens in ODQA. As shown in Table 1, the answer "Dubai in Germany" produced by the generative model FiD (Izacard and Grave, 2021b) is factual incorrect and the answer "33" in the second example is not coherent to the question. While in both cases, the ground-truth answers are present in the retrieved passages. Thus, we hypothesize that if we could put a constraint on the produced words to the input text, the generated answer will be more faithful.

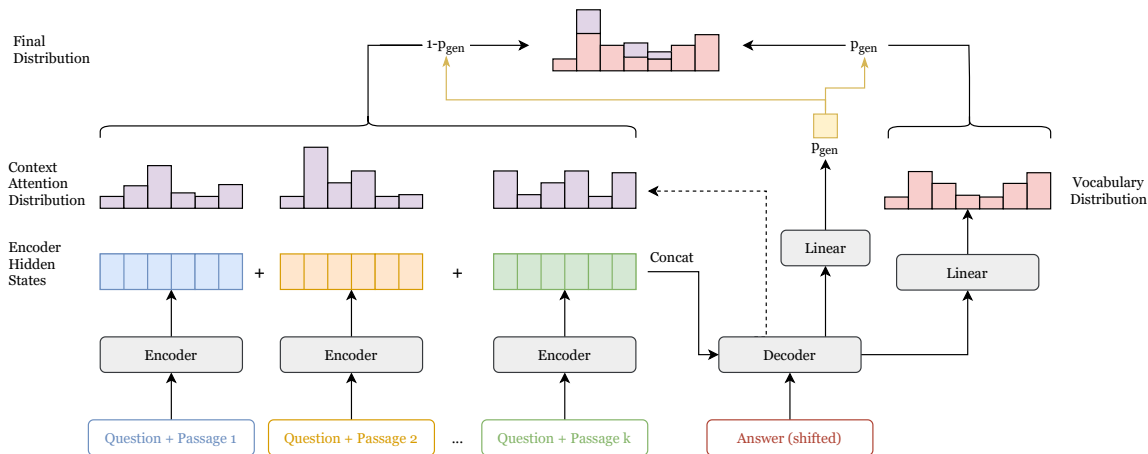Inspired by the work of See et al. (2017), we enhance the generative model with a pointer net-

Figure 1: The overall architecture of our proposed model. We add a linear layer to calculate the generation probability, which decides the weights of generating words from vocabulary or copying from source passages.

work (Vinyals et al., 2015), that enables the model to directly copy text from the retrieved passages while retains the ability of generating new words when the true answers are not explicitly present in the input. To be more specific, our model fusion-in-decoder pointer-generator network (FiD-PGN) is built upon the state-of-the-art model FiD. We reuse the encoder-decoder attention scores as the copy distribution to reduce the computational cost. Compared to FiD, we achieve comparative or even better accuracy on the NaturalQuestions (NQ) (Kwiatkowski et al., 2019) and TriviaQA (Joshi et al., 2017) benchmarks, with less passages used in training. Our experiments results show the effectiveness and efficiency of our model.

2 Related Work

2.1 Open-Domain Question Answering

In this era of data explosion, ODQA offers a way to rapidly and accurately fulfill user's information needs, and hence has recently received significant attention from both industry and academia (Min et al., 2021a). Following the work of DrQA (Chen et al., 2017), most recent works build a two-stage *retriever-reader* system to tackle the problem. The retriever aims at retrieving supportive passages to the given question from a large document corpus. The reader intends to find answer of the question from the first stage retrieved passages. Early work of Chen et al. (2017) adapts a BiLSTM architecture with various lexical and semantic features from the question and passages as inputs. Later, with the emergence of large-scale pre-trained language models, readers based on pre-trained models such

as BERT (Devlin et al., 2019) and T5 (Raffel et al., 2019) have become a common approach (Yang et al., 2019; Izacard and Grave, 2021b; Karpukhin et al., 2020).

2.2 Generative Readers

Compared to extractive models which extract spans from the retrieved passages, generative models are able to produce new words out of the retrieved passages, and thus provide a more flexible modeling framework. Min et al. (2020) and Lewis et al. (2020b) concatenate the given question with top retrieved passages and feed the concatenation to the BART model (Lewis et al., 2020a). Izacard and Grave (2021b) separately encodes the question with each top retrieved passage, then takes the concatenation of the encoder outputs as input to the decoder. Their method provide a way to better aggregate evidence from multiple passages and improve the performance significantly. FiD-KD (Izacard and Grave, 2021a) is an extension of FiD model that increases the accuracy of passage retrieval by training the dense retriever with the guidance of the FiD reader iteratively.

2.3 Pointer-Generator Network

Pointer-Generator Network (See et al., 2017) is an extension of the sequence-to-sequence model by integrating a copy mechanism (Vinyals et al., 2015) into the generator. At each decoding stage, the model is able to either directly copy a word from the input or generate one with certain probability, and thus can be viewed as a combination of extractive and generative approaches. It has been frequently used in natural language tasks like

summarization (Gu et al., 2016; See et al., 2017; Gehrmann et al., 2018) and neural machine translation (Luong et al., 2015; Gu et al., 2018), but its application to ODQA has been less explored.

3 Method

Our model follows the standard two-stage *retriever-reader* framework with a focus on the enhancement of the reader module built upon the FiD reader. We adopt the retriever results of FiD-KD, where a dense retriever similar to DPR (Karpukhin et al., 2020) is used. A pointer network is integrated into the FiD reader to facilitate copying words from the retrieved passages. The overall reader architecture is depicted in Figure 1.

Reader Encoder. The reader encoder of our model is identical to the one of FiD reader. We firstly concatenate the given question q with each retrieved passage p_i as $x_i = [q; p_i]$. Next, we pass each x_i individually to the reader encoder, i.e., the encoder of T5 or BART model, and obtain the hidden representations $h_i = (h_{i,1}, h_{i,2}, \ldots, h_{i,n})$ of the question-passage pair where $h_{i,j} \in \mathbb{R}^d$ and d is the model dimension. Finally, we concatenate all the hidden representations of top-k passages $\{h_1, \ldots, h_k\}$ as input to the decoder.

Reader Decoder. Our approach mainly differs from FiD reader in the decoder module by adding a pointer network. Specifically, at each decoding step t, let $e_t \in \mathbb{R}^d$ be the embedding vector of the input token at this step, and denote $s_t^L \in \mathbb{R}^d$ as the output representation of the last layer L of transformer decoder, then the probability of generation is given as follows,

$$p_{\text{gen}} = \sigma(w_e^T e_t + w_s^T s_t^L + b) \quad (1)$$

where $w_e \in \mathbb{R}^d$, $w_s \in \mathbb{R}^d$ and $b \in \mathbb{R}$ are all learnable parameters and $\sigma(\cdot)$ represents the sigmoid function. In addition, the probability of copying is $1 - p_{\text{gen}}$.

Next, let $\mathcal{V}$ denote the vocabulary containing words for the generative model and $|\mathcal{V}|$ be the size of the vocabulary. Then at step t, the probability distribution of words generation over the vocabulary is computed as,

$$P_{\text{vocab}} = \text{softmax}(W_E s_t^L) \quad (2)$$

where $W_E \in \mathbb{R}^{|V| \times d}$ is a learnable weight matrix.

Benefiting from the encoder-decoder attention layer in transformer architecture, we directly utilize the cross-attention score α_t^L of the last decoder layer L over the source tokens for the target token y_t as copy distribution. Then the probability of selecting y_t in source sequence is calculated as,

$$P_{\text{ctx}}(y_t) = \sum\nolimits_{j:x_{1:k,j}=y_t} \alpha_{t,j}^L \quad (3)$$

where $x_{1:k}$ denotes the concatenation of the top-k retrieved passages, $x_{1:k,j}$ is the j-th token of $x_{1:k}$, and $\alpha_{t,j}^L$ is the j-th element of α_t^L. If y_t is not present in the top-k retrieved passages, $P_{\text{ctx}}(y_t)$ will be zero.

Finally, put all the above together, the target token y_t could both be generated from vocabulary with probability p_{gen}, and copy from the source passages. The final prediction probability is defined as

$$P(y_t) = p_{\text{gen}} P_{\text{vocab}}(y_t) + (1 - p_{\text{gen}}) P_{\text{ctx}}(y_t). \quad (4)$$

4 Experiments

4.1 Datasets

We evaluate the performance of our approach on two standard ODQA datasets, NQ and TriviaQA. The NQ dataset comprises real queries that user issued on Google search engine along with answers. The TriviaQA dataset consists of question-answer pairs collected from trivia and quiz-league websites. The details of data statistics are listed in Table 2. It can be seen that TriviaQA has on average longer question length than NQ, indicating that questions in TriviaQA are relatively more complex. We use the data released on the repository of FiD[1], containing question-answer pairs and top-100 passages retrieved by FiD-KD.

Statistics	NQ	TriviaQA
Train	79,168	78,785
Validation	8,757	8,837
Test	3,610	11,313
Avg. Qlen	9.3	16.9
Avg. Alen	2.4	2.2

Table 2: Summary statistics of the two datasets. Avg. Qlen and Avg. Alen denote the average number of tokens per question and answer, respectively.

---

[1] https://github.com/facebookresearch/FiD

Model	Reader Size	Top-k	NQ	TriviaQA
DPR (BERT-base) (Karpukhin et al., 2020)	110M	24	41.5	57.9
RAG-Seq (BART-large) (Lewis et al., 2020b)	406M	50	44.5	56.8
FiD (T5-base) (Izacard and Grave, 2021b)	220M	100	48.2	65.0
FiD-KD (T5-base) (Izacard and Grave, 2021a)	220M	100	<u>49.6</u>	**68.8**
FiD-KD (Our implementation)	220M	25	48.5	67.5
FiD-PGN	220M	25	**51.4**	<u>68.4</u>

Table 3: Exact match (EM) scores on NQ and TriviaQA test sets. Top-k indicates the number of retrieved passages used during reader training. The performance of SOTA model is in **bold** and the second best model is in <u>underline</u>.

4.2 Implementation Details

We follow the experimental settings as in FiD. Our model is initialized with a pre-trained T5-base model, and trained using AdamW (Loshchilov and Hutter, 2017) algorithm with a learning rate of 10^{-4}, linear scheduling with 15k total steps and 1k warm-up steps. Moreover, we train our model using the top-25 retrieved passages for each question and set the batch size as 64 due to computational limitation. All experiments are run on eight Nvidia V100 32GB GPUs.

4.3 Results

Table 3 shows the experimental results of our model and other approaches on the test sets, evaluated with the standard exact match (EM) score (Rajpurkar et al., 2016). For a fair comparison, we retrained the FiD reader on the top-25 retrieved passages to match our experimental settings.

As shown in Table 3, our model outperforms FiD-KD on both NQ and TriviaQA datasets under the same setting. This demonstrates that the pointer network could help to generate answers more accurately. It is worth noting that, compared with FiD-KD trained with the top-100 retrieved passages, our model achieves comparative or even better results with only 1/4 of the input data and without introducing many parameters (only 1537 extra parameters are added), indicating the efficiency of our model.

5 Analysis

Generation Probability. We explore the probability of generation during training to further investigate the effects of the pointer module. As shown in Figure 2, the generation probability p_{gen} in TriviaQA is always higher than the one in NQ. Note that a higher generation probability means that more tokens are produced from the vocabulary instead of copying from the input. We conjecture that this phenomenon is caused by the different
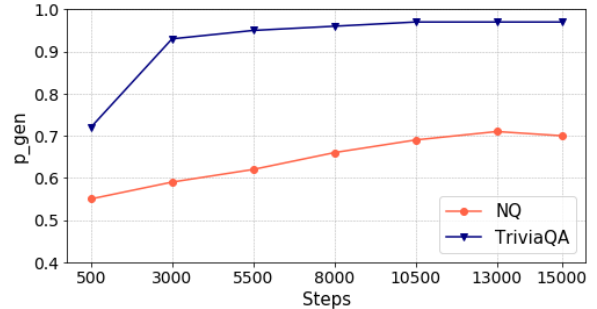

Figure 2: Generation probability p_{gen} over training steps on NQ and TriviaQA.

question types. As stated in Rogers et al. (2021), Trivia questions are more like probing questions. Compared to the information-seeking questions in NQ, probing questions tend to need more complex reasoning, and thus it is difficult to directly extract relevant tokens from input texts. Moreover, this observation is also consistent with the results that the improvements of our model over FiD reader is smaller in TriviaQA than the one in NQ (0.9 vs. 2.9 EM for TriviaQA and NQ, respectively).

Test-Train Overlap Evaluation. The study of test-train overlap (Lewis et al., 2021) provides valuable insights into the model's question answering behavior. We evaluate our model on the same test data splits as in Lewis et al. (2021). Table 4 reports the results with respect to three kinds of test-train overlaps. It can be seen that our approach improves most over FiD reader on "No Overlap" category, the most challenging setting, indicating a better generalization ability to question answering.

Training with Varying Number of Passages. Figure 3 shows the performance of our model and FiD reader with regard to different number of retrieved training passages. We train both models with top-k passages ($k \in \{1, 5, 10, 25\}$) and evaluate on the development sets with the same number of pas-

Dataset	Overlap Type	FiD	FiD-PGN	Δ
NQ	Total	48.5	**51.4**	2.9
	Question Overlap	73.5	**75.9**	2.4
	Answer Overlap Only	41.0	**45.1**	4.1
	No Overlap	28.8	**38.4**	9.6
TriviaQA	Total	67.5	**68.4**	0.9
	Question Overlap	88.4	**89.6**	1.2
	Answer Overlap Only	66.9	**68.4**	1.5
	No Overlap	41.5	**43.4**	1.9

Table 4: Test-train overlap evaluation on NQ and TriviaQA test sets. Exact match (EM) scores are reported.
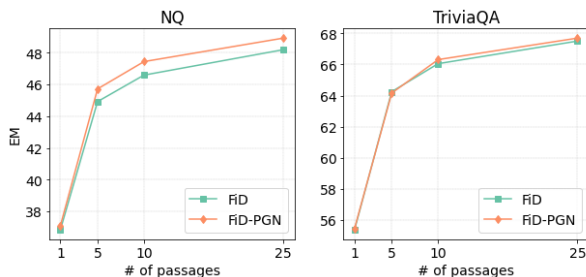

Figure 3: The variation of performance with different number of retrieved passages used in reader training. Exact match (EM) scores are measured on the development sets of NQ and TriviaQA.

sages. We can observe that the matching scores of both models increase with respect to the number of passages used in training, consistent with the findings in Izacard and Grave (2021b) that sequence-to-sequence model is capable of gathering information across multiple retrieved passages. Moreover, the two models show comparative performance when the number of training passages is small, but when more passages are included, our model outperforms FiD, especially on the NQ dataset.

6 Conclusion

In this article, we propose a novel FiD-PGN approach for the reader module of ODQA under the standard *retriever-reader* framework. Specifically, we integrate a pointer network into the FiD reader to allow the model to directly select words from the retrieved passages. Experimental results show that our model outperforms FiD-KD on two benchmark datasets under the same setting, demonstrating the advantages of our method.

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 *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

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.

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.

Sebastian Gehrmann, Yuntian Deng, and Alexander Rush. 2018. Bottom-up abstractive summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4098–4109, Brussels, Belgium. Association for Computational Linguistics.

Jiatao Gu, James Bradbury, Caiming Xiong, Victor O. K. Li, and Richard Socher. 2018. Non-autoregressive neural machine translation. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Jiatao Gu, Zhengdong Lu, Hang Li, and Victor O.K. Li. 2016. Incorporating copying mechanism in sequence-to-sequence learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1631–1640, Berlin, Germany. Association for Computational Linguistics.

Gautier Izacard and Edouard Grave. 2021a. Distilling knowledge from reader to retriever for question answering. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Gautier Izacard and Edouard Grave. 2021b. 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.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611, Vancouver, Canada. Association for Computational Linguistics.

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.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466.

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, Pontus Stenetorp, and Sebastian Riedel. 2021. Question and answer test-train overlap in open-domain question answering datasets. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1000–1008, Online. Association for Computational Linguistics.

Patrick S. H. Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020b. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Ilya Loshchilov and Frank Hutter. 2017. Fixing weight decay regularization in adam. *ArXiv preprint*, abs/1711.05101.

Thang Luong, Ilya Sutskever, Quoc Le, Oriol Vinyals, and Wojciech Zaremba. 2015. Addressing the rare word problem in neural machine translation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 11–19, Beijing, China. Association for Computational Linguistics.

Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. 2020. On faithfulness and factuality in abstractive summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1906–1919, Online. Association for Computational Linguistics.

Sewon Min, Jordan L. Boyd-Graber, Chris Alberti, Danqi Chen, Eunsol Choi, Michael Collins, Kelvin Guu, Hannaneh Hajishirzi, Kenton Lee, Jennimaria Palomaki, Colin Raffel, Adam Roberts, Tom Kwiatkowski, Patrick S. H. Lewis, Yuxiang Wu, Heinrich Küttler, Linqing Liu, Pasquale Minervini, Pontus Stenetorp, Sebastian Riedel, Sohee Yang, Minjoon Seo, Gautier Izacard, Fabio Petroni, Lucas Hosseini, Nicola De Cao, Edouard Grave, Ikuya Yamada, Sonse Shimaoka, Masatoshi Suzuki, Shumpei Miyawaki, Shun Sato, Ryo Takahashi, Jun Suzuki, Martin Fajcik, Martin Docekal, Karel Ondrej, Pavel Smrz, Hao Cheng, Yelong Shen, Xiaodong Liu, Pengcheng He, Weizhu Chen, Jianfeng Gao, Barlas Oguz, Xilun Chen, Vladimir Karpukhin, Stan Peshterliev, Dmytro Okhonko, Michael Sejr Schlichtkrull, Sonal Gupta, Yashar Mehdad, and Wen-tau Yih. 2021a. Neurips 2020 efficientqa competition: Systems, analyses and lessons learned. *ArXiv preprint*, abs/2101.00133.

Sewon Min, Kenton Lee, Ming-Wei Chang, Kristina Toutanova, and Hannaneh Hajishirzi. 2021b. Joint passage ranking for diverse multi-answer retrieval. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6997–7008, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Sewon Min, Julian Michael, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2020. AmbigQA: Answering ambiguous open-domain questions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5783–5797, Online. Association for Computational Linguistics.

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*, abs/1910.10683.

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.

Anna Rogers, Matt Gardner, and Isabelle Augenstein. 2021. QA dataset explosion: A taxonomy of NLP resources for question answering and reading comprehension. *ArXiv preprint*, abs/2107.12708.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083, Vancouver, Canada. Association for Computational Linguistics.

Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. 2015. Pointer networks. In *Advances in Neural*

Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada, pages 2692–2700.

Ellen M. Voorhees. 1999. The TREC-8 question answering track report. In *Proceedings of The Eighth Text REtrieval Conference, TREC 1999, Gaithersburg, Maryland, USA, November 17-19, 1999*, volume 500-246 of *NIST Special Publication*. National Institute of Standards and Technology (NIST).

Yuxiang Wu, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. 2021. Training adaptive computation for open-domain question answering with computational constraints. 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 447–453, Online. Association for Computational Linguistics.

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.

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 *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 72–77, Minneapolis, Minnesota. Association for Computational Linguistics.

Chunting Zhou, Graham Neubig, Jiatao Gu, Mona Diab, Francisco Guzmán, Luke Zettlemoyer, and Marjan Ghazvininejad. 2021. Detecting hallucinated content in conditional neural sequence generation. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1393–1404, Online. Association for Computational Linguistics.