

# Improving Question Answering with Generation of NQ-like Questions

Anonymous EMNLP submission

## Abstract

Question Answering (QA) systems require a large amount of annotated data which is costly and time-consuming to gather. Converting datasets of existing QA benchmarks are challenging due to different formats and complexities. To address these issues, we propose an algorithm to automatically generate shorter questions resembling day-to-day human communication in the Natural Questions (NQ) dataset from longer trivia questions in Quizbowl (QB) dataset by leveraging conversion in style among the datasets. This provides an automated way to generate more data for our QA systems. To ensure quality as well as quantity of data, we detect and remove ill-formed questions using a neural classifier. We demonstrate that in a low resource setting, using the generated data improves the QA performance over the baseline system on both NQ and QB data. Our algorithm improves the scalability of training data while maintaining quality of data for QA systems.

## 1 Introduction

Large-scale data collection is a challenging process in the domain of Question Answering and Information Retrieval due to the necessity of high-quality annotations which are scarce and expensive to generate. There are several QA datasets (Joshi et al., 2017), (Rajpurkar et al., 2016), (Yang et al., 2018), (Kwiatkowski et al., 2019), (Rodriguez et al., 2021) with significantly different structure and complexity. Large quantities of high quality data are effective in training a more efficacious Machine Learning system.

In this paper, we focus on converting questions normally spoken in a trivia competition to questions resembling day-to-day human communication. Trivia questions have multiple lines, consist of multiple hints as standalone sentences to an answer, and players can buzz on any sentence in the question to give an answer. In contrast, questions

used in daily human communication are shorter (often a single line). We propose an algorithm to generate multiple short natural questions from every long trivia question by converting each of the sentences with multiple hints to several shorter questions. We also add a BERT (Devlin et al., 2019) based quality control method to filter out the ill-formed questions and retain the well-formed questions. We show that our algorithm for generating questions improves the performance of two question answering (QA) systems in a low resource setting. We also demonstrate that concatenation of original natural questions with generated questions improves the QA system performance. Finally, we prove that by using such a method to generate synthetic data, we can achieve higher scores than a system that uses only NQ data.

## 2 Dataset and Data Extraction

We use two popular datasets- the Quizbowl (Rodriguez et al., 2021), henceforth referred to as QB dataset, and NQ-Open dataset (Lee et al., 2019), derived from Natural Questions (Kwiatkowski et al., 2019). QB has a total of 119247 question/answer samples and NQ has 91434 total question/answer samples. For the NQ dataset, we use the same 1800 dev and 1769 test question/answer splits as used in the EfficientQA Competition (Min et al., 2021).

As our task involves transforming QB questions to NQ-like questions, we extract pairs of questions that are semantically similar. We first extract every possible question-question pair with the same answer using string matching resulting in 95651 question-question pairs. From this parallel corpus, we extract the last sentence of the QB questions and pass them through a pre-trained Sentence-BERT (Reimers and Gurevych, 2019) model along with the corresponding NQ question. We take the cosine similarity between the [CLS] embedding to find pairs that are semantically equivalent by setting the threshold to 0.5. From this we extract

19439 question-question pairs having moderate semantic equivalence. We also use the same index from the last sentence of QB questions paired with NQ questions to retrieve corresponding QB full questions in paragraph form. Out of the extracted corpora, we create a smaller dataset of the last sentence of every QB question (and corresponding QB full paragraph) paired with NQ questions for our low resource setting. This paired dataset has a total of 1218 training samples, 93 validation samples and 563 test samples which are semantically similar. We outline the statistics of the baseline and generated datasets in Table 1.

### 3 Methods to generate NQ-like Questions

We use the following NLP techniques to generate NQ-like questions from our QB dataset

- Tokenization
- Coreference resolution
- Parse tree output
- Bag of words based question generation

We outline our methods in Algorithm 1. At first, we tokenize the long paragraph of a QB question into individual sentences with hints to an answer providing sufficient context. Using every sentence annotated by the question id was not sufficient due to annotation errors of sentence delimiters like a ‘.’.

- **Initial:** ... and "k." For 10 points, ...
- **Tokenized:**
  - ... and " k. "
  - For 10 points , ...

However, each sentence contains 2-3 clues about the answer whereas NQ questions generally are in 1 sentence with 1 clue. To resolve this issue, we use Coreference Resolution on every sentence to obtain clusters of nouns and pronouns referring to those nouns (Kirstain et al., 2021), and Parse Trees.

We then split the sentences to get parts of a sentence with similar syntactic structure using Parse Tree based on ADVCL (adverbial clause) and CC (conj) tags. Any pronoun that is present in a split is replaced with the noun from the clusters found using Coreference Resolution. After breaking into splits, we run a check on the number of words in

every split and append sentences with smaller number of words (less than 8) back into the original sentence. We finally clean the split sentences of trailing punctuation marks and words that should not be at the sentence ending like ‘and’, ‘but’ etc. This gives us sentences or phrases that usually have one clue similar to NQ.

To make the outputs question-like, we use a bag of words approach by replacing words like ‘this’ with ‘which’, ‘it’ with ‘what’ for all but the last sentence. The last sentence in QB has a specific syntactic structure where the sentence contains, ‘For x points , name this’ followed by the identifier to the answer. We replace this whole phrase according to whatever comes after. For example,

- ‘name this author’ : ‘who is the author’
- ‘name this 1985 event’ : ‘which is the 1985 event’
- ‘name this phenomenon’ : ‘what is the phenomenon’

---

#### Algorithm 1 NQ-like Question Generation

---

```

1: procedure NQLIKEGEN(sentences)
2:   Individual(sentences) ▷ split full QB
   question into single sentences
3:   Pre_processing(sentences) ▷ remove
   punctuations, conjunctions or any other word
   not expected at the end of the sentence.
4:   Coreference_clusters(sentences) ▷
   forms clusters like ["this country", "its"]
5:   Parse_Tree(sentences) ▷ Split
   the conjoined clues based on similar syntactic
   structure
6:   if Noun, Pronoun not in
   same_parse_split then
7:     Replace ▷ replace the pronoun in the
   other split with the NP
8:   for QB_last_sentences do
9:     String_Replacement(last_sentence)
   ▷ Replace ‘name this’ with the annotated ‘wh’
   question by NOUNS vocabulary formed
10:  for QB_non_last_sentences do
11:    Bag_of_Words ▷ use
   frequency based vocabulary table to replace
   “this” to “which”
12:  return nq_like_questions

```

---

To achieve this, we extract noun phrases following ‘name this’ from 1000 samples and form a

vocabulary to manually annotate them with ‘who is the’, ‘which is the’, ‘what is the’, and their plural counterparts. This gives us questions that resemble NQ questions.

### 3.1 Quality Control

We trained a BERT based (Devlin et al., 2019) classifier using the pretrained bert-base-uncased model for detecting well-formed questions and improving the data quality. We use the dataset from (Faruqui and Das, 2018) which has questions annotated with scores from 0 to 1 signifying how well the questions are structured. With the help of this classifier, we generate the scores for our training set (7266 samples) of generated NQ like questions. We only take the questions with a score greater than 0.5 to maintain a good balance between quality and quantity. This provides us with 1445 synthetic good quality samples. We add these good-quality NQ-like questions to our original NQ training set thereby increasing the number of samples to 2663 QA pairs. In the example shown in Figure 1, the well-formed questions detected by the classification model are:

- Chris Carney represents which state ’s 10th district in congress , which includes Snyder and Wyoming counties
- What includes Raystown Lake ; the Monongahela ends in which state , where what meets the Allegheny river
- With capital at Harrisburg , what northeastern state has Philadelphia as its metropolis , and is named after its Quaker founder ?

## 4 Experiments

We use a smaller subset (1874 samples) from the total paired samples to form our paired dataset containing NQ and QB questions. The train, dev, and test splits are 1218, 93, and 563 respectively. We provide generated samples sizes using our algorithm and the scores of two QA systems; DrQA and RAG which trained on these varying experiments.

### 4.1 Generated Data

We originally possess 1874 total samples [1218 train, 93 dev, 563 test] for both the NQ dataset and QB full dataset consisting of QB questions having multiple sentences. We extract only the

last sentence from the QB questions (QB last sentence) with the same split [1218 train, 93 dev, 563 test]. We generate NQ-like questions from QB questions with all sentences (NQ-like from QB full) [7266 train, 517 dev, 3279 test] and questions in the last sentence of QB questions (NQ-like from QB last sentence) [1385 train, 105 dev, 622 test], as outlined in Algorithm 1. We then extract only the well-formed questions by utilizing a quality control classifier as outlined in Section 3.1. From this we obtain the filtered NQ-like questions from the QB questions with multiple sentences (Quality controlled NQ-like from QB full) dataset [1445 train, 113 dev, 517 test], and the filtered NQ-like questions from the last sentence of the QB questions (Quality controlled NQ-like from QB last sentence) [1093 train, 74 dev, 465 test]. The NQ dataset is concatenated with the 6 different types of datasets outlined above. Assuming a generic [Dataset Name] for the 6 datasets, we name the augmented system like (NQ + [Dataset Name]) shown in Tables 2 and 3.

### 4.2 Question Answering Systems

After reviewing several QA systems (Zhu et al., 2021), we finally used the Retrieval Augmented Generation (RAG) (Lewis et al., 2021) and DrQA (Chen et al., 2017) as our baseline QA systems. We used a small subset of the Wikipedia dump as our retrieval dump for the low resource setting.

**RAG QA System** We retrieve 5 documents using the RAG retriever and use a batch size of 2 for our training. We train our systems for 5 epochs by using AdaGrad (Duchi et al., 2011) as the optimizer with a learning rate of  $1e^{-4}$ .

**DrQA System** We retrieve 10 documents for gold passage computation. We use SGD as the optimizer with 0.1 learning rate, and pre-trained word vector GloVe with 840 billion tokens and 300 dimensional vector embeddings (Pennington et al., 2014). 3-layer bidirectional long short-term memory network (LSTM) with 128 hidden units is used for both paragraph and question encoding. The numbers of document layers and answer layers are both set to 3. We train all our systems for 50 epochs with batch size 64 and max length 30.

## 5 Results

Our research work demonstrates a clear improvement in QA performance for RAG and DrQA sys-

System Description	Data Size	Mean	Median	Mode
QB system with full questions	1874	4.39	4.0	4.0
Generated NQ-like full questions	11062	5.9	6.0	6.0
Filtered generated NQ-like full questions	2075	1.7	1.0	1.0
QB system with full questions	119247	6.2	6.0	6.0
Generated NQ-like full questions	772456	6.5	6.0	6.0

Table 1: Statistics of number of sentences per Quizbowl question

Question	RAG Answer	DrQA Answer
<b>Original Full QB question</b>		
Chris Carney represents this state's 10th district in congress, which includes Snyder and Wyoming counties. It is home to the nation's first zoo, and houses the Harry Houdini museum. It has the eastern hemlock as its state tree, the ruffed grouse as state bird, and Bloomsburg is the only officially incorporated town in this state. Its highest point is at mount Davies, and it includes Raystown Lake; the Monongahela ends in this state, where it meets the Allegheny river. Allentown and Reading are two of the larger cities in this commonwealth, and Bethlehem gave its name to a large steel company here. Scranton is the center of the coal mining industry in this rustbelt state, which is the starting point of the Ohio river. Also known as the Keystone state, and with capital at Harrisburg, FTP what northeastern state has Philadelphia as its metropolis, and is named after its Quaker founder?	maine	tri
<b>Tokenization</b>		
1. Chris Carney represents this state's 10th district in congress, which includes Snyder and Wyoming counties. 2. It is home to the nation's first zoo, and houses the Harry Houdini museum. 3. It has the eastern hemlock as its state tree, the ruffed grouse as state bird, and Bloomsburg is the only officially incorporated town in this state. 4. Allentown and Reading are two of the larger cities in this commonwealth, and Bethlehem gave its name to a large steel company here. 5. Scranton is the center of the coal mining industry in this rustbelt state, which is the starting point of the Ohio river. 6. Also known as the Keystone state, and with capital at Harrisburg, FTP what northeastern state has Philadelphia as its metropolis, and is named after its Quaker founder? 7. It is home to the nation's first zoo, and houses the Harry Houdini museum.		
<b>Coreference Output</b>		
In Tokenization point 4: [Bethlehem, its] In Tokenization point 6: [what northeastern state, its, its]		
<b>Parse-tree Output</b>		
1. Chris Carney represents this state 's 10th district in congress , which includes Snyder and Wyoming counties 2. It is home to the nation 's first zoo , andhouses the Harry Houdini museum 3. It has the eastern hemlock as its state tree , the ruffed grouse as state bird , 4. Bloomsburg is the only officially incorporated town in this state 5. Its highest point is at mount Davies , 6. it includes Raystown Lake ; the Monongahela ends in this state , where it meets the Allegheny river 7. Allentown and Reading are two of the larger cities in this commonwealth , 8. Bethlehem gave Bethlehem's name to a large steel company here 9. Scranton is the center of the coal mining industry in this rustbelt state , which is the starting point of the Ohio river 10. Also known as the Keystone state , 11. with capital at Harrisburg , FTP what northeastern state has Philadelphia as its metropolis , and is named after its Quaker founder ?		
<b>Bag of Words on last sentence with string replacement</b>		
with capital at Harrisburg , what northeastern state has Philadelphia as its metropolis , and is named after its Quaker founder ?	pennsylvania	Pennsylvania
<b>Bag of Words on the remaining sentences with word frequency based dictionary</b>		
Chris Carney represents which state 's 10th district in congress , which includes Snyder and Wyoming counties	florida	New York
What is home to the nation 's first zoo , and .houses the Harry Houdini museum	salvador allende	Hungary
What has the eastern hemlock as what's state tree , the ruffed grouse as state bird ,	california	Virginia
Bloomsburg is the only officially incorporated town in which state	new york	Pennsylvania
It's highest point is at mount Davies ,	appalachian mountains	John Bull
What includes Raystown Lake ; the Monongahela ends in which state , where what meets the Allegheny river	pennsylvania	New York
Allentown and Reading are two of the larger cities in which commonwealth ,	philadelphia	Harare
Bethlehem gave It's name to a large steel company here	andrew carnes	Andrew Carnegie
Scranton is the center of the coal mining industry in which rustbelt state , which is the starting point of the Ohio river	minnesota	New Jersey
Also known as the Keystone state ,	wyoming	New York

Figure 1: Generation of NQ-like Questions from the original QB Questions along with answers of the QA systems (Gold Answer: [pennsylvania](#))

tems as shown in Tables 2 and 3. We see an improvement in QA evaluation scores when trivia questions with multiple sentences in QB dataset

can be used to automatically generate NQ-like questions which are then concatenated with the original NQ questions. For the RAG QA system, the



System	Total samples	Accuracy	Precision	Recall	F1	SacreBLEU
NQ	1874	0.490	0.550	0.540	0.540	29.57
NQ + NQ-like from QB full	9140	0.390	0.420	0.410	0.420	27.95
NQ + Quality controlled NQ-like from QB full	3319	0.467	0.505	0.503	0.502	34.03
NQ + QB full	3092	0.465	0.503	0.502	0.499	27.62
NQ + QB last sentence	3092	0.516	0.558	0.560	0.556	37.37
NQ + NQ-like from QB last sentence	3259	0.500	0.557	0.556	0.553	34.98
NQ + Quality controlled NQ-like from QB last sentence	2967	<b>0.518</b>	<b>0.567</b>	<b>0.564</b>	<b>0.562</b>	<b>37.46</b>
QB full	1874	0.232	0.261	0.259	0.259	12.17
NQ-like from QB full	11062	0.171	0.188	0.188	0.188	12.45
Quality controlled NQ-like from QB full	2075	0.176	0.192	0.191	0.191	14.11
QB last sentence	1874	0.44	<b>0.495</b>	<b>0.488</b>	<b>0.488</b>	22.18
NQ-like from QB last sentence	2112	0.421	0.454	0.456	0.453	34.49
Quality controlled NQ-like from QB last sentence	1632	<b>0.441</b>	<b>0.495</b>	<b>0.488</b>	<b>0.488</b>	<b>38.25</b>

Table 2: Question Answering Metric values with RAG system

System	Total samples	Accuracy	Precision	Recall	F1	SacreBLEU
NQ	1874	0.211	0.243	0.242	0.241	8.74
NQ + NQ-like from QB full	9140	0.243	0.271	0.270	0.269	10.49
NQ + Quality controlled NQ-like from QB full	3319	0.227	0.256	0.254	0.254	<b>11.01</b>
NQ + QB full	3092	0.229	0.257	0.258	0.256	10.97
NQ + QB last sentence	3092	<b>0.257</b>	<b>0.291</b>	<b>0.295</b>	<b>0.290</b>	9.65
NQ + NQ-like from QB last sentence	3259	0.253	0.281	0.276	0.277	10.20
NQ + Quality controlled NQ-like from QB last sentence	2967	0.234	0.268	0.267	0.266	8.91
QB full	1874	0.191	0.251	0.228	0.233	7.17
NQ-like from QB full	11062	0.027	0.085	0.085	0.085	2.23
Quality controlled NQ-like from QB full	2075	0.197	0.224	0.218	0.219	8.71
QB last sentence	1874	0.328	0.394	0.384	0.383	9.66
NQ-like from QB last sentence	2112	0.310	0.363	0.358	0.357	<b>25.40</b>
Quality controlled NQ-like from QB last sentence	1632	<b>0.367</b>	<b>0.428</b>	<b>0.429</b>	<b>0.424</b>	23.16

Table 3: Question Answering Metric values with DrQA system

baseline score was improved by 3 of our proposed systems, out of which generated outputs from the last sentence concatenated with original NQ data has seen the most success with an improvement of 2.8 points in accuracy and an increase of 1.7, 2.4, 2.2 and 7.89 points on Precision, Recall, F1 measure, and BLEU respectively. The performance of the QA system trained on NQ-like filtered data concatenated with the NQ data is also better than the system based on the concatenation of QB dataset with NQ dataset. For DrQA, all of our proposed systems have done better than the baseline with NQ data and the system with concatenated NQ and QB data. The generated outputs from the last sentence of the QB concatenated with NQ have shown an improvement of 4.6 points in accuracy and 4.8, 5.3, 4.9, 0.91 points in Precision, Recall, F1 measure, and BLEU respectively on the DrQA system. Most of our systems exhibit an increase in BLEU score over the baseline. We also observe that the performance of RAG and DrQA systems trained on only NQ like data generated from QB dataset is better than the baseline systems on QB dataset.

This is understandable as the last sentence usu-

ally has the easiest clue. On top of which, it has a regular structure. These characteristics make the last sentence from the QB more semantically aligned with NQ-like data after conversion using our algorithm.

Our proposed algorithm as outlined in Algorithm 1 and quality filtering technique (1093 extra samples) as mentioned in Section 3.1 can be effective in a low resource setting with a lack of sufficiently well annotated QA data. This also helps in scaling the training data by converting multiple datasets automatically.

A broad example of NQ-like question generation and subsequent answer generation has been provided in Figure 1.

## 6 Conclusion and Future Work

We clearly observe from the results that adding filtered NQ-like questions from the QB data has given a boost over using only NQ questions. In finer detail, we observe that questions from the last sentence of the QB are of higher quality than from intermediate sentences and therefore provide a higher boost to performance even with less sam-

ples. Even by simply adding questions generated from last sentence, we increase the exact match accuracy by nearly 2 points. We also observe that the BLEU score of answers generated from quality controlled NQ like system is 16 points more than the BLEU score of the baseline QB system for the RAG system and by 13 points for the DrQA system. This shows that our algorithm to generate NQ-like questions has been effective in improving the quality of the training dataset.

We are working to extend this system from the smaller datasets to the entire 119247 QB and 91494 NQ samples by using the generated 772456 unfiltered NQ-like questions from the entire QB data, retain the well-formed questions and improve the performance over the baseline system trained on NQ dataset.

We observe that QB trivia questions are in passive voice while NQ questions are in active voice. So Neural Machine Translation can help in converting the style of QB questions. Our dataset pairing method automatically filters out adversarial samples and improves the quality of the generated NQ-like corpus. This opens the door towards a possible research area where multiple non-adversarial datasets can be used to filter the NQ dataset from adversarial samples. We plan on improving the filtering process by generating our own annotated dataset regarding well-formed and ill-formed questions in order to detect well formed questions and merge ill formed questions with better quality questions.

Specifically, through our manual transformation for NQ-like questions, we are specifying three different answer types like WHO/WHICH/WHAT, which might be useful to propagate to intermediate parse outputs during NQ-like question generation in order to improve answer generation.

Other information like difficulty and year associated with QB questions can be leveraged to improve quality or filtering and can even be used to predict extra information on NQ questions.

Further work can also be done on augmenting other datasets to NQ-like questions to generate more synthetic samples which may improve question answering performance.

## References

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. [Reading wikipedia to answer open-domain questions](#).

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [Bert: Pre-training of deep bidirectional transformers for language understanding](#).

John Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(Jul):2121–2159.

Manaal Faruqi and Dipanjan Das. 2018. Identifying Well-formed Natural Language Questions. In *Proc. of EMNLP*.

Mandar Joshi, Eunsol Choi, Daniel S. Weld, and Luke Zettlemoyer. 2017. [Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension](#).

Yuval Kirstain, Ori Ram, and Omer Levy. 2021. [Coreference resolution without span representations](#).

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.

Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. 2019. [Latent retrieval for weakly supervised open domain question answering](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6086–6096, Florence, Italy. 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, Sebastian Riedel, and Douwe Kiela. 2021. [Retrieval-augmented generation for knowledge-intensive nlp tasks](#).

Sewon Min, Jordan 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 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 Schlichtkrull, Sonal Gupta, Yashar Mehdad, and Wen tau Yih. 2021. [Neurips 2020 efficientqa competition: Systems, analyses and lessons learned](#).

- Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. [Glove: Global vectors for word representation](#). In *Empirical Methods in Natural Language Processing (EMNLP)*, page 1532–1543.
- Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. [Squad: 100,000+ questions for machine comprehension of text](#).
- Nils Reimers and Iryna Gurevych. 2019. [Sentencebert: Sentence embeddings using siamese bert networks](#).
- Pedro Rodriguez, Shi Feng, Mohit Iyyer, He He, and Jordan Boyd-Graber. 2021. [Quizbowl: The case for incremental question answering](#).
- Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. [Hotpotqa: A dataset for diverse, explainable multi-hop question answering](#).
- 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](#).