ArchivalQA: A Large-scale Benchmark Dataset for Open Domain Question Answering over Archival News Collections

Anonymous ACL submission

Abstract

In the last few years, open-domain question answering (ODQA) has advanced rapidly due to the development of deep learning techniques and the availability of large-scale QA datasets. However, the current datasets are essentially designed for synchronic document collections (e.g., Wikipedia). Temporal news collections such as long-term news archives spanning several decades, are rarely used in training the models despite they are quite valuable for our society. To foster the research in the field of ODQA on such historical collections, we present ArchivalQA, a large question answering dataset consisting of 532,444 question-answer pairs which is designed for temporal news QA. We divide our dataset into four subparts based on the question difficulty levels and the containment of temporal expressions, which we believe are useful for training and testing ODQA systems characterized by different strengths and abilities. The novel QA dataset-constructing framework that we introduce can be also applied to create datasets over other types of collections[1].

1 Introduction

With the application of digital preservation techniques, more and more past news articles are being digitized and made accessible online. This results in the availability of large news archives spanning multiple decades. They offer immense value to our society, contributing to our understanding of different time periods in the history and helping us to learn about the details of the past (Korkeamäki and Kumpulainen, 2019). However, due to their large sizes and complexities, it is difficult for users to effectively utilize such temporal news collections. A reasonable solution is to use open-domain question answering (ODQA), which attempts to answer natural language questions based on large-scale unstructured documents. Yet, the existing QA datasets are essentially constructed from Wikipedia or other synchronic document collections[2]. The lack of large-scale datasets for temporal news collections hinders the development of ODQA on news archives where Temporal IR (Campos et al., 2014; Kanhabua et al., 2015) techniques need to be utilized. Note that ODQA on historical document collections can be useful in many cases such as providing support for journalists who wish to relate their stories to certain past events, historians who investigate the past as well as employees of diverse professions, such as insurance or broad finance sectors, who wish to assess current risks based on historical accounts or support their decision making. As indicated in previous studies (Wang et al., 2020, 2021), synchronic document collections like Wikipedia cannot successfully answer many minor or detailed questions about the past that have temporal character.

To overcome these shortcomings of existing QA datasets, we devise a novel framework that assists in the creation of a diverse, large-scale ODQA dataset over a temporal document collection. The framework utilizes automatic question generation as well as a series of carefully-designed filtering steps to remove poor quality instances. As an underlying archival document collection, we use the New York Times Annotated Corpus (NYT corpus) (Sandhaus, 2008), which contains over 1.8 million news articles published between January 1, 1987 and June 19, 2007. The NYT corpus has been frequently used over the recent years for many researches in temporal IR, temporal news content

[1]The core part of the ArchivalQA dataset is uploaded as a supplementary material, and will be publicly available after the publication, including its four sub-datasets and the code of the entire framework.

[2]Note that existing news datasets such as CNN/Daily Mail (Hermann et al., 2015) and NewsQA (Trischler et al., 2016) are more suited to MRC tasks rather than to ODQA task due to the cloze question type or the ambiguity prevalent in their questions as we will discuss later. In addition, their underlying document collections span relatively short time periods, which are also quite recent (such as after June 2007 or April 2010).

analysis, archival search, historical analysis and in other related tasks (Campos et al., 2014; Kanhabua et al., 2015). The final dataset that we release, ArchivalQA, contains 532,444 data instances and is divided into different sub-parts based on question difficulty and the presence of temporal expressions.

We choose a semi-automatic way to construct our dataset for several reasons. First, manually generating questions would be too costly as it requires knowledge of history from annotators. Second, since question generation (QG) has recently attracted considerable attention, the available models already achieve quite good performance. Third, current "data-hungry" complex neural network models require larger and larger datasets to maintain high performance. Finally, synthetic datasets have been effective in boosting deep learning models' performance and are especially useful in use cases involving distant target domains with highly specialized content and terminology, for which there is only a small amount of labeled data (Walonoski et al., 2020; Li et al., 2020; Feng et al., 2020). We then approach the dataset generation based on a cascade of carefully designed filtering steps that remove low quality questions from a large initial pool of generated questions. We note that our dataset is not only spanning the longest time period compared to other QA datasets, but it also provides detailed questions on the events that occurred from 14 to 34 years ago. It is also one of the largest ODQA datasets available. Our another contribution besides the development of a large-scale dataset for an unexplored domain is the presentation of an approach to generate large datasets in an inexpensive way.

2 Related Work

In the recent years, a large number of QA benchmarks have been introduced (Zeng et al., 2020; Baradaran et al., 2020; Dzendzik et al., 2021; Rogers et al., 2021). The SQuAD 1.1 (Rajpurkar et al., 2016) consists of question-answer pairs that are made from the paragraphs of 536 Wikipedia articles, which was later extended by SQuAD 2.0 (Rajpurkar et al., 2018) that contains also unanswerable questions. NarrativeQA (Kočiskỳ et al., 2018) dataset uses a different resource, the summaries of movie scripts and books, to create its question-answer pairs. MS MARCO (Nguyen et al., 2016) and NaturalQuestions (Kwiatkowski et al., 2019) use the search query logs of Bing and Google search engines as the questions, and the retrieved web documents and Wikipedia pages are collected as the evidence documents.

Most of the existing datasets are designed over synchronic document collections, such as books, Wikipedia articles and web search results. While there are some MRC datasets created based on the news collections, they mostly belong to the cloze style datasets, such as CNN/Daily Mail (Nallapati et al., 2016), WhoDidWhat (Onishi et al., 2016) and ReCoRD (Zhang et al., 2018), with the aim to predict the missing word in a passage rather than to answer proper questions; hence these datasets cannot be actually used in the ODQA task. Although Lelkes et al. (2021) constructed the NewsQuizQA dataset based on news articles, too, its questions belong to the multiple-choice type, which are easier to be answered, and the dataset contains only 20K question-answer pairs. The question-answer pairs were also obtained from only 5K summaries derived from the recent news articles.

To the best of our knowledge, NewsQA (Trischler et al., 2016) is the only MRC dataset in which an answer is a text span which is created based on the temporal document collection, the CNN news articles. However, our dataset has significant differences when compared to NewsQA. First, dataset size of NewsQA is much smaller than ours (119K vs. 532K). Second, its underlying CNN corpus contains less news articles which span shorter and also more recent time period (93k articles from 2007/04 to 2015/04 vs. 1.8M articles from 1987/01 to 2007/06 as in our case). We have also found that NewsQA is essentially appropriate for the MRC task and is not very suitable for the ODQA task. This is because many questions require additional background knowledge about their original paragraphs for understanding and correctly answering. These questions tend to be ambiguous, unclear and generally impossible to be answered over the large news collection, because they are not specific enough and tend to have multiple correct answers (e.g., the questions *"When were the findings published?"*, *"Who drew inspiration from presidents?"* and *"Whose mother is moving to the White House?"*[3]). Note that questions on some QA datasets also have similar characteristics, for example, Min et al. (2020) found that over half of the questions in the NaturalQuestions are ambiguous, with diverse sources of ambiguity such as event and

[3]These questions are shown as examples on the NewsQA website: https://www.microsoft.com/en-us/research/project/newsqa-dataset/stats/

Table 1: Comparison of related datasets. Note that there are more synchronic datasets that are not listed here (Zhu et al., 2021) (roughly about 30 common QA datasets based on our investigation).

Dataset	#Questions	Answer Type	Question Source	Corpus Source	Synch/Diach	Non-ambiguous
MS MARCO	1M	Generative, Boolean	Query logs	Web documents	Synchronic	✗
SQuAD 1.1	108K	Extractive	Crowd-sourced	Wikipedia	Synchronic	✗
SQuAD 2.0	158K	Extractive	Crowd-sourced	Wikipedia	Synchronic	✗
NaturalQuestions	323K	Extractive, Boolean	Query logs	Wikipedia	Synchronic	✗
NewsQuizQA	20K	Multiple-choice	Crowd-sourced	News	Diachronic (2018/06-2020/06)	✗
NewsQA	119K	Extractive	Crowd-sourced	News	Diachronic (2007/04-2015/04)	✗
ArchivalQA	532K	Extractive	Automatically Generated	News	Diachronic (1987/01-2007/06)	✓

entity references. Finally, the questions in NewsQA have been created from 7 times less articles than in our final dataset (12,744 vs. 88,431).

Thus, the goal of this work is to create a large-scale QA dataset over long-term historical document collections, that can promote the development of ODQA systems on historical news archives. We summarize differences between ArchivalQA and the most related datasets in Tab. 1.

3 Methodology

We introduce here the framework that generates and selects questions from temporal collections. Fig. 1 shows its architecture which consists of five modules: Article Selection Module, Question Generation Module, Syntactic & Temporal Filtering Module, General & Temporal Ambiguity Filtering Module and Final Quality Filtering Module. The modules are described below.

3.1 Article Selection Module

This module is responsible for deciding which articles are used to generate the initial set of questions. We use two approaches for selecting the articles.

3.1.1 Selection based on Wikipedia Events

The first one is to use the short descriptions of important events in Wikipedia year pages[4] as the seeds to find related articles. Since we utilize the NYT corpus, we use 2,976 event descriptions which occurred between January 1, 1987 and June 19, 2007. Then, for each event description, we select keywords to be used as search queries for retrieving relevant articles from the news archive. We choose Yake![5] (Campos et al., 2020) as our keyword extraction method, which is a state-of-the-art unsupervised approach that relies on statistical features extracted to select the most important keywords. Next, the query composed of the extracted keywords is sent to the ElasticSearch[6] installation which returns the top 25 relevant documents ranked by BM25. Finally, 53,991 news articles are obtained to be used for generating questions.

3.1.2 Random Selection

The second way is to randomly select long news articles from the corpus, which have at least 100 tokens. Based on this step, additional 55,000 news articles were collected.

We followed these two ways because we wanted the final dataset to contain questions related to important events from the past and also questions on some minor things, especially ones which are likely not recorded in Wikipedia[7].

3.2 Question Generation Module

The second step is to generate questions for the collected articles. We first separate articles into paragraphs and use the neural network model to generate candidate questions from each paragraph. Similar to Lelkes et al. (2021) that use PEGASUS-base model (Zhang et al., 2020) for question generation, we apply a novel large pre-trained Transformer encoder-decoder model, called T5-base (Raffel et al., 2019), as the QG model to generate the questions. We do not choose however PEGASUS-base model since we found that it generates questions that sometimes contain information not found in the document (probably due to its Gap Sentences Generation pre-training task). The work of Lelkes et al. (2021), is probably the most related work to ours as the authors have also applied QG methods to generate questions over news articles. They applied PEGASUS model to generate the questions using NewsQuizQA dataset. However, their ques-

[4] List of year pages: https://en.wikipedia.org/wiki/List_of_years and events for an example year: https://en.wikipedia.org/wiki/1989

[5] Yake! is available in the PKE tookit: https://github.com/boudinfl/pke

[6] https://www.elastic.co/

[7] In the experiments we show that only a small number of our questions can be successfully answered using Wikipedia.

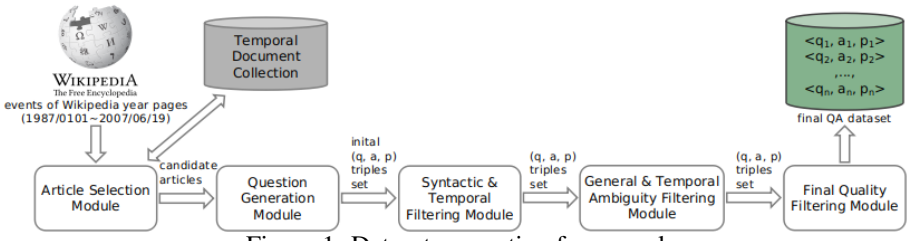

Figure 1: Dataset generation framework

tions belong to the quiz-style multiple-choice type, which is not suitable for ODQA task.

We fine-tune the model using SQuAD 1.1 (Rajpurkar et al., 2016) whose inputs are the answers together with their corresponding paragraphs, and the outputs are the questions. The final model achieves good performance on the SQuAD 1.1 dev set (the scores of BLEU-4, METEOR, ROUGE-L are 21.19, 26.48, 42.79, respectively). After fine-tuning the model, every named entity[8] in a given paragraph of each article is labeled as an answer, and is used along with the paragraph as the input to the model. Note that the answers of many QA datasets, such as CNN/Daily Mail (Nallapati et al., 2016), TriviaQA (Joshi et al., 2017), Quasar-T (Dhingra et al., 2017), SearchQA (Dunn et al., 2017) and XQA (Liu et al., 2019a), also mainly use the entities as answers (e.g., 92.85% of the answers in TriviaQA are Wikipedia entities), as this improves answering accuracy. In addition, we restrict the number of tokens of the paragraphs and of the corresponding sentences that include the answers. More specifically, the paragraphs that have less than 30 tokens are eliminated. Additionally, the answers whose corresponding sentences have less than 10 tokens are discarded, too. Finally, 6,408,036 questions are generated in this way from 1,194,730 paragraphs of 106,197 articles.

3.3 Syntactic & Temporal Filtering Module

This module consists of 8 processing steps that remove or transform the candidate question-answer pairs obtained so far:

1. Remove questions that do not end with a question mark (107,586 such questions removed).
2. Remove questions whose answers are explicitly indicated inside the questions' content (127,212 questions removed).
3. Remove duplicate questions. The same questions generated from different paragraphs are removed (492,257 questions removed).
4. Remove questions that have too few or too many named entities. Questions without any named entity or with more than 7 named entities are eliminated (1,310,621 questions removed).
5. Remove questions that are too short or too long. Questions that contain less than 8 or more than 30 tokens are dropped (463,726 questions removed).
6. Remove questions with unclear pronouns, for example, "*What was the name of the agency that she worked for in the Agriculture Department?*" (63,300 questions removed). The details of this step are described in Appendix A.
7. Transform relative temporal information in questions to absolute temporal information. For example, *"How many votes did President Clinton have in New Jersey last year?"* is transformed to *"How many votes did President Clinton have in New Jersey in 1996?"* (140,658 questions transformed). The details are given in Appendix A.
8. Transform relative temporal information of the answers of generated questions to absolute temporal information. We apply the same approach as in the previous step. For example, the answers to questions *"When did Rabbi Riskin write about protests by West Bank settlers in Israel?"* and *"When were the three teenagers convicted of murdering Patrick Daly?"*, which are "Aug. 7" and "yesterday", respectively, are transformed to "August 07, 1995" and "June 15, 1993", by incorporating the articles' publication dates: '1995-08-12' and '1993/06/16' (279,671 answers transformed in this way).

3.4 General & Temporal Ambiguity Filtering Module

3.4.1 Filtering by Content Specificity

Sentence specificity is often pragmatically defined as the level of detail in the sentence (Louis and Nenkova, 2011; Li and Nenkova, 2015). In contrast to specific sentences that contain informative messages, general sentences are the ones that do not reveal much detailed information (e.g., overview statements). In the examples shown below, the first sentence is general as it is clearly less informative than the second sentence (specific one), and is not suitable to be used to generate questions.

1) *Despite recent declines in yields, investors continue to pour cash into money funds.*

[8] We use the named entity recognizer from the spaCy: https://github.com/explosion/spaCy.

Table 2: Temporal ambiguity of example questions.

No.	Question	Ambiguity
1	Who did President Bush announce he would submit a trade agreement with?	Temporally ambiguous
2	When was the National Playwrights Conference held?	Temporally ambiguous
3	Who won the Serbian presidential election in October, 2002?	Temporally non-ambiguous
4	Where did the Tutsi tribe massacre thousands of Hutu tribesmen?	Temporally non-ambiguous

2) *Assets of the 400 taxable funds grew by $1.5 billion during the last week, to $352.7 billion.*

Thus, in this step, we aim to remove questions that were generated from general sentences. We use the training dataset from Ko et al. (2019), which is composed of three publicly available, labeled datasets (Louis and Nenkova, 2012; Li and Nenkova, 2015; Li et al., 2016). The resulting combined dataset contains 4,342 sentences taken from news articles together with their sentence-level binary labels (general vs. specific). We partition this dataset randomly into the training set (90%), and the test set (10%). We next fine-tune three Transformer-based classifiers: BERT-based model (Devlin et al., 2018), RoBERTa-base model (Liu et al., 2019b) and ALBERT-base model (Lan et al., 2019), such that each classifier consists of the corresponding pre-trained language model followed by a dropout layer and a fully connected layer. We finally choose RoBERTa-base model (Liu et al., 2019b) as the specificity-determining model because it achieves the best results on the test set - 84.49% accuracy. Finally, the questions are discarded if the sentences that include their answers have been classified by the above-described approach as general. This step removed 952,398 questions.

3.4.2 Filtering by Temporally Ambiguity

When manually analyzing the resulting dataset we have observed that some questions are problematic due to their temporal ambiguity, e.g., *"How many people were killed by a car bomb in Baghdad?"*. Such questions can be matched to several distinct events. The first and the second generated example questions in Tab. 2 exhibit such characteristics; the correct answers of such questions should be actually a list of answers rather than a single answer. However, the datasets having multiple correct answers for each question are quite rare in the current ODQA field (Zhu et al., 2021) (we are only aware of AMBIGQA dataset (Min et al., 2020) which contains multiple possible answers to ambiguous

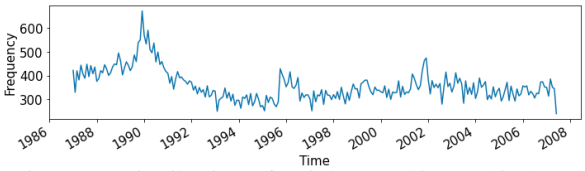

Figure 2: Distribution of articles used in ArchivalQA

questions). This might be because it would not be clear how to rank systems as some of the ground-truth answers might be more preferred than others. In our case, for example, some events related to the ambiguous questions could be more important or more popular than other related events. Also, and perhaps more importantly, finding all the possible answers to such questions is quite difficult if not impossible in a large news collection (especially an archival one that spans two decades such as ours). Hence, we decided to remove temporally ambiguous questions, however we will make them available for the community to download as a separate data, should anyone be interested in studying questions of this type.

We define temporally ambiguous questions as ones that have multiple correct and different answers in different time periods. Note that temporally ambiguous questions are specific to temporal datasets like ours and they have not been studied before. Since there is no readily available dataset for detecting temporally ambiguous questions, we have manually labeled 5,500 questions obtained from the previous filtering steps. Then, we again fine-tuned three Transformer-based classifiers, same as when training the specificity-evaluating model. The BERT-based model (Devlin et al., 2018) has been finally chosen as it performs best on the test set achieving 81.82% accuracy. We then used it to remove 1,823,880 questions classified as temporally ambiguous.

3.5 Final Quality Filtering Module

In the final module, we aim to remove remaining bad data instances by analyzing the entire <question, answer, paragraph> triples, that might be due to several reasons (e.g., questions with incorrect answers, questions containing information not found in paragraphs, or other bad questions that have not been filtered out by the previous filtering stages). Firstly, we created a dedicated dataset for this task by asking 10 annotators to label 10k samples from the results obtained after applying the previous filters as either "Good" or "Bad" given <paragraph, question, answer> triples[9]. The annotators had

[9]This dataset will be also available, as it could be useful for QG research.

to not only consider the problems we discussed before, but also check whether the questions are grounded in their paragraphs and whether they can be answered by their answers, and whether the questions are grammatically correct or not. The dataset was then randomly split into the training set (90%), and the test set (10%). Then, we trained a RoBERTa-base model (Liu et al., 2019b) that takes the triples as the input after adding a special token ([SEP]) to separate question-answer pairs and paragraphs. We set a high threshold that permits only the predicted good triples with probabilities higher than 0.99 be chosen as the final good triples. This last filtering step resulted in the precision of finding good triples to be 86.74% on our test set. Finally, we removed 534,612 questions whose corresponding triples were classified as bad.

4 Dataset Analysis

4.1 Data Statistics

After all the above filtering steps, we have finally obtained the dataset which includes 532,444 question-answer pairs that originate from 313,100 paragraphs of 88,431 news articles. About half of the questions (263,292) come from the randomly selected articles, and the other questions (269,152) are based on articles that were selected based on Wikipedia events. Paragraph IDs are also appended to each question-answer pair to let ODQA systems explicitly train their IR components. We partition the entire dataset randomly into the training set (80%, 425,956 examples), the development set (10%, 53,244 examples), and the test set (10%, 53,244 examples). Tab. 3 shows few examples. More detailed dataset statistics are presented in Tab. 4. Fig. 2 shows also the temporal distribution of documents used for ArchivalQA questions.

We have also analyzed the named entity types[10] of the answers in the dataset. As shown in the left pie chart in Fig. 3, the answers that belong to PERSON, ORG, DATE, GPE and NORP account for a large part of ArchivalQA. The right pie chart in Fig. 3 shows the distribution of 9 event categories of the questions, that are classified by our prepared classifier, which is trained based on the event dataset created by (Sumikawa and Jatowt, 2018) and achieves 85.86% accuracy. Fig. 4 presents also the distribution of frequent trigram prefixes. While nearly half of SQuAD questions are "what" questions (Reddy et al., 2019), the distribution of ArchivalQA is more evenly spread across multiple question types.

[10] 18 entity types used by NE recognizer in spaCy.

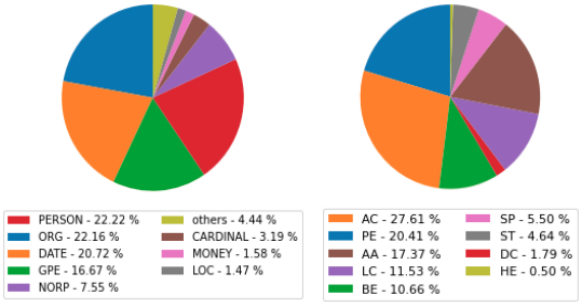

Figure 3: Left: Answers' named entity distribution ("others": named entities that account for a very small part (< 1%)). Right: Questions' category distribution ("AC": "arts & culture", "PE": "politics & elections", "AA": "armed conflicts & attacks", "LC": "law and crime", "BE": "business & economy", "SP": "sport", "ST": "science & technology", "DC": "disasters & accidents", "HE": "health & environment").

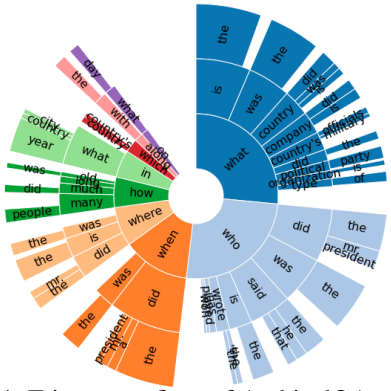

Figure 4: Trigram prefixes of ArchivalQA questions

4.2 Model Performance

We use the following well-established ODQA approaches and show their results on ArchivalQA:

1. DrQA-Wiki (Chen et al., 2017): DrQA model which uses Wikipedia as the knowledge source. With this setting we would like to test if Wikipedia alone could be sufficient for answering questions about the historical events.
2. DrQA-NYT (Chen et al., 2017): DrQA model which uses NYT archive.
3. DrQA-NYT-TempRes (Chen et al., 2017): DrQA model which uses NYT archive and transforms the answers with relative temporal information by a similar approach to the one that we used for the publication date information of the retrieved document or paragraph (see the 7th and 8th filtering steps of Sec. 3.3).
4. BERTserini-Wiki (Yang et al., 2019): BERTserini model which uses Wikipedia.
5. BERTserini-NYT (Yang et al., 2019): BERTserini model which uses NYT archive.
6. BERTserini-NYT-TempRes (Yang et al., 2019): BERTserini model which uses NYT archive and

Table 3: ArchivalQA Dataset Examples. trans_que, trans_ans, and doc_sel represent whether the question is transformed, whether the answer is transformed and the selection method of the utilized document, respectively. Note that para_id contains concatenated information of the document ID (the metadata of each news article in the NYT corpus) and the ith paragraph used to generate the questions.

id	question	answer	para_id	trans_que	trans_ans	doc_sel
train_0	Who claimed responsibility for the bombing of Bab Ezzouar?	Al Qaeda	1839755_20	0	0	wiki
train_4	When did Tenneco announce it was planning to sell its oil and gas operations?	May 26, 1988	148748_0	0	1	rand
val_45	What threat prompted Mr. Paik's family to flee to Hong Kong?	the Korean War	1736040_7	0	0	wiki
test_84	Along with the French Open, what other tournament did Haarhuis win in 1998?	Wimbledon	1043631_15	1	0	rand

Table 4: Basic statistics of ArchivalQA

Number of QA pairs	532,444
Number of transformed questions	29,696
Number of transformed answers	47,972
Avg. question length (words)	12.43
Avg. questions / document	6.02
Avg. questions / paragraph	1.70

Table 5: Models' performance on ArchivalQA

Model	**EM**	**F1**
DrQA-Wiki	7.53	11.64
DrQA-NYT	38.13	46.12
DrQA-NYT-TempRes	44.84	53.06
BERTserini-Wiki	10.19	16.25
BERTserini-NYT	54.84	66.05
BERTserini-NYT-TempRes	56.34	68.93

transforms the relative temporal answers.

We measure the performance of these models using exact match (EM) and F1 score - the two standard measures commonly used in QA research. The results of all the models are given in Tab. 5. Firstly, we can observe that the models that utilize Wikipedia as the knowledge source perform much worse than the models that utilize NYT corpus, which is due to many questions being about minor things or events that Wikipedia does not seem to record. Secondly, the models that resolve implicit temporal answers perform better than the ones without this step. Temporal information resolution is then clearly important.

4.3 Human Evaluation

We finally conduct human evaluation on ArchivalQA to study the quality of the generated questions. We randomly sampled 5K question-answer pairs along with their original paragraphs and publication dates and asked 10 graduate students for evaluation. The volunteers were requested to rate the generated questions from 1 (very bad) to 5 (very good) on four criteria: *Fluency* measures if a question is grammatically correct and is fluent to read. *Answerability* indicates if a question can be answered by the given answer. *Relevance* measures if a question is grounded in the given passage, while *Non-ambiguity* defines if a question is non-ambiguous. The average scores for each evaluation metric are shown in Tab. 6. Our model achieves high performance over all the metrics, especially on *Fluency* and *Relevance*. In addition, the *Non-ambiguity* result is high, indicating that large majority of the questions are non-ambiguous.

Table 6: Human evaluation results of ArchivalQA

Fluency	**Answerability**	**Relevance**	**Non-ambiguity**
4.80	4.57	4.79	4.60

5 Sub-Dataset Creation

We also distinguish subparts of the dataset which we believe could be used for training/testing ODQA systems with diverse strengths and abilities.

5.1 Difficult/Easy Questions Dataset

We first created two sub-datasets (called ArchivalQAEasy and ArchivalQAHard) based on the difficulty levels of questions, such that 100,000 are easy and another 100,000 are difficult questions. We use open-source Anserini[11] IR toolkit with BM25 as the ranking function to create these subsets. The samples are labeled as easy if the paragraphs used to generate the questions appeared within the top 10 retrieved documents; otherwise they are considered difficult. We then partitioned each sub-dataset randomly into the training set (80%, 80,000 examples), the development set (10%, 10,000 examples), and the test set (10%, 10,000 examples).

5.2 Division based on Time Expressions

We created the next two sub-datasets based on the temporal characteristics of the questions. In particular, we constructed two sub-datasets of 75,000 questions with temporal expressions and 75,000 without temporal expressions (called ArchivalQA-Time and ArchivalQANoTime, respectively). We

[11] https://github.com/castorini/anserini

Table 7: Performance of different models over different Sub-Datasets

Model	ArchivalQAEasy		ArchivalQAHard		ArchivalQATime		ArchivalQANoTime	
	EM	F1	EM	F1	EM	F1	EM	F1
DrQA-NYT	42.10	51.97	22.81	31.24	31.32	42.17	39.59	47.18
DrQA-NYT-TempRes	48.41	57.26	27.37	34.02	33.19	44.01	46.39	54.91
BERTserini-NYT	59.15	69.16	25.00	33.73	50.65	63.24	55.36	68.37
BERTserini-NYT-TempRes	61.80	71.56	29.88	38.44	51.12	65.67	58.27	70.19

used SUTime (Chang and Manning, 2012) combined with our handcrafted rules to collect the former questions, while the latter were randomly chosen questions without temporal expressions. Note that questions with temporal expressions should let ODQA systems limit the search time scope from the entire time frame of the news archive to the narrower time periods specified by the temporal expressions contained in these questions. For example, for the question *"Which team won the 1990 World Series?"*, the answers could be just searched in documents that were published either during or at least some time after 1990. Both sub-datasets were randomly split into the training (80%, 60,000 examples), the development (10%, 7,500 examples), and the test set (10%, 7,500 examples).

5.3 Model Performance on Sub-Datasets

Tab. 7 presents the performance of different models over the four sub-datasets. We can see that all the models achieve better results on ArchivalQAEasy than on ArchivalQAHard, indicating that the questions of ArchivalQAHard tend to be difficult. For example, the improvement of BERTserini-NYT-TempRes (Yang et al., 2019) is in the range of 106.83% and 86.16% on EM and F1 metrics, respectively. We expect ArchivalQAHard be difficult for our tested approaches, and that dense retrieval models could exhibit better performance. When considering ArchivalQATime and ArchivalQANoTime, the models perform slightly better on ArchivalQANoTime. A possible reason for that can be that such temporal signals are currently just used as usual textual information (rather than being utilized as time selectors) which can even cause harm, despite the fact that time expressions actually constitute an important feature. Future models should pay special attention to temporal signals.

6 Dataset Use

Our dataset can be used in several ways. First, ODQA models can use the questions, answers and paragraphs[12] for training their IR and MRC modules (Karpukhin et al., 2020; Ding et al., 2020) on a novel kind of data that poses challenges in terms of highly changing contexts of different years, high temporal periodicity of events and rich temporal signals in terms of document timestamps and temporal expressions embedded in document content. As shown in (Wang et al., 2020, 2021) systems that utilize such complex temporal signals (using Temporal IR approaches or others) achieve better results than conventional approaches.

When it comes to the underlying news dataset, most systems would use our QA pairs against the NYT corpus. They might however use also other temporal news collections that temporally align with the NYT collection (i.e., ones that also span 1987-2007), although naturally this would result in a more difficult task. It might be also possible to try to answer questions using synchronic knowledge bases such as Wikipedia, although as we have observed earlier, Wikipedia lacks a lot of detailed information on the past. The questions in our dataset are often detailed and minor and relate to old events, hence they may be different than questions in other popular ODQA datasets. Such questions can be particularly valuable considering that the true utility of QA systems lies in answering hard questions that humans cannot (at least easily) answer by themselves. Finally, the testing can be more fine-grained based on the question difficulty, question specificity and the appearance of temporal components contained in questions.

7 Conclusions

In this work we introduce a novel large-scale ODQA dataset for answering questions over a long-term archival news collection, that the final objective is to foster the research in the field of ODQA on news archives. The dataset is unique since it covers the longest time period among all the ODQA datasets and deals with events that occurred in relatively distant past. It also contains over a million question-answer pairs. An additional contribution is that we consider and mitigate the problem of temporally ambiguous questions for temporal document datasets. While this issue has not been observed in other ODQA datasets and researches, it is of high importance in long-term temporal datasets such as news archives. Finally, we demonstrate a semi-automatic pipeline to generate large datasets via a series of carefully designed filtering steps.

[12]Note that another way to use the dataset is to train models without using the paragraph information (Lee et al., 2019).

References

Razieh Baradaran, Razieh Ghiasi, and Hossein Amirkhani. 2020. A survey on machine reading comprehension systems. *arXiv preprint arXiv:2001.01582*.

Ricardo Campos, Gaël Dias, Alípio M Jorge, and Adam Jatowt. 2014. Survey of temporal information retrieval and related applications. *ACM Computing Surveys (CSUR)*, 47(2):1–41.

Ricardo Campos, Vítor Mangaravite, Arian Pasquali, Alípio Jorge, Célia Nunes, and Adam Jatowt. 2020. Yake! keyword extraction from single documents using multiple local features. *Information Sciences*, 509:257–289.

Angel X Chang and Christopher D Manning. 2012. Sutime: A library for recognizing and normalizing time expressions. In *Lrec*, volume 2012, pages 3735–3740.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading wikipedia to answer open-domain questions. *arXiv preprint arXiv:1704.00051*.

Kevin Clark and Christopher D Manning. 2016. Deep reinforcement learning for mention-ranking coreference models. *arXiv preprint arXiv:1609.08667*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Bhuwan Dhingra, Kathryn Mazaitis, and William W Cohen. 2017. Quasar: Datasets for question answering by search and reading. *arXiv preprint arXiv:1707.03904*.

Yingqi Qu Yuchen Ding, Jing Liu, Kai Liu, Ruiyang Ren, Xin Zhao, Daxiang Dong, Hua Wu, and Haifeng Wang. 2020. Rocketqa: An optimized training approach to dense passage retrieval for open-domain question answering. *arXiv preprint arXiv:2010.08191*.

Matthew Dunn, Levent Sagun, Mike Higgins, V Ugur Guney, Volkan Cirik, and Kyunghyun Cho. 2017. Searchqa: A new q&a dataset augmented with context from a search engine. *arXiv preprint arXiv:1704.05179*.

Daria Dzendzik, Carl Vogel, and Jennifer Foster. 2021. English machine reading comprehension datasets: A survey. *arXiv preprint arXiv:2101.10421*.

Steven Y Feng, Varun Gangal, Dongyeop Kang, Teruko Mitamura, and Eduard Hovy. 2020. Genaug: Data augmentation for finetuning text generators. *arXiv preprint arXiv:2010.01794*.

Karl Moritz Hermann, Tomáš Kočiskỳ, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. *arXiv preprint arXiv:1506.03340*.

Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. *arXiv preprint arXiv:1705.03551*.

Nattiya Kanhabua, Roi Blanco, and Kjetil Nørvåg. 2015. Temporal information retrieval. *Foundations and Trends® in Information Retrieval*, 9(2):91–208.

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

Wei-Jen Ko, Greg Durrett, and Junyi Jessy Li. 2019. Domain agnostic real-valued specificity prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6610–6617.

Tomáš Kočiskỳ, Jonathan Schwarz, Phil Blunsom, Chris Dyer, Karl Moritz Hermann, Gábor Melis, and Edward Grefenstette. 2018. The narrativeqa reading comprehension challenge. *Transactions of the Association for Computational Linguistics*, 6:317–328.

Laura Korkeamäki and Sanna Kumpulainen. 2019. Interacting with digital documents: A real life study of historians' task processes, actions and goals. In *Proceedings of the 2019 Conference on Human Information Interaction and Retrieval*, CHIIR '19, pages 35–43, New York, NY, USA. ACM.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*.

Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. 2019. Latent retrieval for weakly supervised open domain question answering. *arXiv preprint arXiv:1906.00300*.

Adam D Lelkes, Vinh Q Tran, and Cong Yu. 2021. Quiz-style question generation for news stories. *arXiv preprint arXiv:2102.09094*.

Junyi Li and Ani Nenkova. 2015. Fast and accurate prediction of sentence specificity. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 29.

Junyi Jessy Li, Bridget O'Daniel, Yi Wu, Wenli Zhao, and Ani Nenkova. 2016. Improving the annotation of sentence specificity. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 3921–3927.

Yu Li, Xiao Li, Yating Yang, and Rui Dong. 2020. A diverse data augmentation strategy for low-resource neural machine translation. *Information*, 11(5):255.

Jiahua Liu, Yankai Lin, Zhiyuan Liu, and Maosong Sun. 2019a. Xqa: A cross-lingual open-domain question answering dataset. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2358–2368.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019b. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Annie Louis and Ani Nenkova. 2011. Automatic identification of general and specific sentences by leveraging discourse annotations. In *Proceedings of 5th international joint conference on natural language processing*, pages 605–613.

Annie Louis and Ani Nenkova. 2012. A corpus of general and specific sentences from news. In *LREC*, pages 1818–1821.

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

Ramesh Nallapati, Bowen Zhou, Caglar Gulcehre, Bing Xiang, et al. 2016. Abstractive text summarization using sequence-to-sequence rnns and beyond. *arXiv preprint arXiv:1602.06023*.

Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. Ms marco: A human generated machine reading comprehension dataset. In *CoCo@ NIPS*.

Takeshi Onishi, Hai Wang, Mohit Bansal, Kevin Gimpel, and David McAllester. 2016. Who did what: A large-scale person-centered cloze dataset. *arXiv preprint arXiv:1608.05457*.

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, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for squad. *arXiv preprint arXiv:1806.03822*.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*.

Siva Reddy, Danqi Chen, and Christopher D Manning. 2019. Coqa: A conversational question answering challenge. *Transactions of the Association for Computational Linguistics*, 7:249–266.

Anna Rogers, Matt Gardner, and Isabelle Augenstein. 2021. Qa dataset explosion: A taxonomy of nlp resources for question answering and reading comprehension. *arXiv preprint arXiv:2107.12708*.

Evan Sandhaus. 2008. The new york times annotated corpus. *Linguistic Data Consortium, Philadelphia*, 6(12):e26752.

Yasunobu Sumikawa and Adam Jatowt. 2018. System for category-driven retrieval of historical events. In *Proceedings of the 18th ACM/IEEE on Joint Conference on Digital Libraries*, pages 413–414.

Adam Trischler, Tong Wang, Xingdi Yuan, Justin Harris, Alessandro Sordoni, Philip Bachman, and Kaheer Suleman. 2016. Newsqa: A machine comprehension dataset. *arXiv preprint arXiv:1611.09830*.

Jason Walonoski, Sybil Klaus, Eldesia Granger, Dylan Hall, Andrew Gregorowicz, George Neyarapally, Abigail Watson, and Jeff Eastman. 2020. Synthea™ novel coronavirus (covid-19) model and synthetic data set. *Intelligence-based medicine*, 1:100007.

Jiexin Wang, Adam Jatowt, Michael Färber, and Masatoshi Yoshikawa. 2020. Answering event-related questions over long-term news article archives. In *European conference on information retrieval*, pages 774–789. Springer.

Jiexin Wang, Adam Jatowt, Michael Färber, and Masatoshi Yoshikawa. 2021. Improving question answering for event-focused questions in temporal collections of news articles. *Information Retrieval Journal*, 24(1):29–54.

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. *arXiv preprint arXiv:1902.01718*.

Changchang Zeng, Shaobo Li, Qin Li, Jie Hu, and Jianjun Hu. 2020. A survey on machine reading comprehension—tasks, evaluation metrics and benchmark datasets. *Applied Sciences*, 10(21):7640.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter Liu. 2020. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In *International Conference on Machine Learning*, pages 11328–11339. PMLR.

Sheng Zhang, Xiaodong Liu, Jingjing Liu, Jianfeng Gao, Kevin Duh, and Benjamin Van Durme. 2018. Record: Bridging the gap between human and machine commonsense reading comprehension. *arXiv preprint arXiv:1810.12885*.

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 Appendix

A.1 Unclear Pronouns Questions Removal

The questions with unclear pronouns are removed in the 6th step of the Syntactic & Temporal Filtering Module. We first utilize part-of-speech tagger in spaCy to obtain the fine-grained POS information of each token in the generated questions. The questions whose tokens are classified as "PRP" or "PRP$" are collected as the initial set of unclear-pronoun questions. Then we utilize the novel coreference resolution tool (NeuralCoref (Clark and Manning, 2016)) to obtain the coreference results of each sentence in the question set, e.g., for the question "*When did Sampras win his first Grand Slam?*", the information that 'his' points to 'Sampras' can be obtained. Then we apply several heuristic rules to collect clear-pronoun questions, and the set of final unclear-pronouns questions is then obtained. A sentence is considered correct if its pronoun points to named entities inside the question content (e.g., 'Sampras' in the previous example), or if the question asks about the actual resolution of the pronouns (e.g., "*Who dived into rough waters near her home in Maui to save a Japanese woman?*"), etc.

A.2 Relative Temporal Information Transformation

The relative temporal information in questions and answers is transformed in the 7th and 8th step of the Syntactic & Temporal Filtering Module. We use SUTime (Chang and Manning, 2012) along with the publication date information of the articles, that include the paragraphs used to generate the question, as the reference date to transform the relative temporal information. Note that we do not transform all the temporal expressions in the entire corpus, since this is time-consuming. Additionally, this would change the original contents of the articles in the corpus, the situation which we try to avoid. Any systems that will use our dataset should see only the original, unchanged NYT content for answering the dataset questions. We expect that models which need to use temporal signals in form of expressions should utilize article timestamps to resolve the temporal information of content.