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

Jiexin Wang
Kyoto University
Japan
wang.jiexin.83m@st.kyoto-u.ac.jp

Adam Jatowt
University of Innsbruck
Austria
adam.jatowt@uibk.ac.at

Masatoshi Yoshikawa
Kyoto University
Japan
yoshikawa@i.kyoto-u.ac.jp

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 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 generate high-quality, non-ambiguous questions over other types of temporal document collections.[1]

CCS CONCEPTS

• **Information systems** → **Content analysis and feature selection**; **Digital libraries and archives**.

KEYWORDS

datasets, question answering, question generation

---

[1]The core part of the ArchivalQA dataset, its four sub-datasets, filtered ambiguous questions, other related training datasets and the code of the entire framework are available at https://github.com/WangJiexin/ArchivalQA.

---

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, helping us to learn about the details of the past, and offering valuable lessons for future generations [26]. For example, sociologists have used news archives to examine vital questions like how different jurisdictions slowed the spread of the 1918 flu [40], which can also offer valuable lessons for the COVID-19 pandemic we are facing today. 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 hinders the development of ODQA on document archives such as news article archives where Temporal IR [3, 21] 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 in order to support their decision making. As indicated in previous studies [61, 62], synchronic document collections like Wikipedia cannot successfully answer many minor or detailed questions about the events from the past since the relevant data for answering those questions is only available in primary sources preserved in the form of large archival document collections.

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) [54], 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 analysis, archival search, historical analysis

---

[2]Note that existing news datasets such as CNN/Daily Mail [17] and NewsQA [59] are more suited to machine reading comprehension (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).

and in other related tasks [3, 21]. 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 [15, 35, 60]. We then approach the dataset generation techniques based on a cascade of carefully designed filtering steps that remove low quality questions from a large initial pool of generated questions.

To sum up, we make the following contributions in this work:

- We propose one of the largest ODQA datasets for news collections,[3] which 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.
- We propose an approach to generate large datasets in an inexpensive way, whose resulting questions tend to be non-ambiguous and of good quality, thus having only a single potential answer. Compared with other QG methods, most questions generated by our approach are clear and non-ambiguous, and thus they can be especially useful in improving computational approaches to education, e.g., to support generating questions for exams.
- We undertake comprehensive analysis of the generated dataset, which does not only show the quality and utility of the resulting data, but also proves the effectiveness of our QG framework.

2 RELATED WORK

2.1 QA Benchmarks

In the recent years, a large number of QA benchmarks have been introduced [2, 14, 52, 68]. The SQuAD 1.1 dataset [49] consists of question-answer pairs that are made from the paragraphs of 536 Wikipedia articles. This dataset was later extended by SQuAD 2.0 [48] that contains also unanswerable questions. NarrativeQA [25] uses a different resource, the summaries of movie scripts and books, to create its question-answer pairs. SearchQA [13] and TriviaQA [19] create a more challenging setting by utilizing web search to collect multiple documents to form the context given existing question-answer pairs from Jeopardy! quiz show and quiz websites, which may be useful for inferring the correct answers. MS MARCO [44] and NaturalQuestions [28] 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 (machine reading comprehension) datasets created based on the news articles, they mostly belong to the cloze style datasets, such as CNN/Daily Mail [43], WhoDidWhat [45] and ReCoRD [70], with the aim to predict the missing word in a passage rather than to answer proper questions; hence they cannot be used in the ODQA task. Although Lelkes et al. [31] 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 pairs were also obtained from only 5K summaries derived from the recent news articles. In addition, it has been designed as a dataset for generating the quiz-style question-answer pairs.

To the best of our knowledge, NewsQA [59] 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 them. 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?"*[4]). Note that questions on some QA datasets also have similar characteristics, for example, Min et al. [42] found that over half of the questions in the NaturalQuestions are ambiguous, with diverse sources of ambiguity such as event and 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, non-ambiguous QA dataset over a long-term historical document collection that can promote the development of ODQA systems on historical news archives. In Table 1 we summarize differences between ArchivalQA and the most related datasets.

2.2 Automatic Question Generation

In the recent years, automatic question generation (AQG) has greatly advanced thanks to deep learning techniques, and it has received increasing attention due to its wide applications in education [27], dialogue systems [64], and question answering [12]. Diverse types of neural sequence-to-sequence models have been proposed for the AQG task. Zhao et al. [71] introduce a model that incorporates paragraph-level inputs - the first QG model that achieved large improvement over sentence-level inputs. Sun et al. [58] and Kim et al. [23] improved the performance by encoding answer positions, which can help to generate better-quality answer-focused questions.

---

[3]The largest existing dataset that uses news articles, CNN/Daily Mail dataset [17], has been created based on a straightforward cloze test and thus cannot be considered as a proper ODQA dataset.

[4]These questions are actually 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 [72] (roughly about 30 common QA datasets based on our investigation).

Dataset	#Questions	Answer Type	Question Source	Corpus Source	Synch/Diach	Non-ambiguous
MS MARCO [44]	1M	Generative, Boolean	Query logs	Web documents	Synchronic	✗
SQuAD 1.1 [49]	108K	Extractive	Crowd-sourced	Wikipedia	Synchronic	✗
SQuAD 2.0 [48]	158K	Extractive	Crowd-sourced	Wikipedia	Synchronic	✗
NaturalQuestions [28]	323K	Extractive, Boolean	Query logs	Wikipedia	Synchronic	✗
CNN/Daily Mail [43]	1M	Cloze	Automatically Generated	News	Diachronic (2007/04 - 2015/04)	✗
NewsQuizQA [31]	20K	Multiple-choice	Crowd-sourced	News	Diachronic (2018/06-2020/06)	✗
NewsQA [59]	119K	Extractive	Crowd-sourced	News	Diachronic (2007/04-2015/04)	✗
ArchivalQA	532K	Extractive	Automatically Generated	News	Diachronic (1987/01-2007/06)	✓

Some works also propose QG models under particular constraints, e.g., controlling the difficulty [16] and topic [18] of the generated questions. In addition, models that can jointly learn to ask (QG) as well as answer questions (QA) have been also proposed [53, 63]. Moreover, it has been shown that having a large, even synthetic dataset, is useful for training QA models with different objectives. For example, Puri et al. [46] train their model using only the synthetic data and obtain state-of-the-art performance on SQuAD dev set. Shakeri et al. [56] improve the performance of models in target domains by utilizing the synthetic dataset. Saxena et al. [55] demonstrate that the large size model-generated dataset can help in training temporal reasoning models. Lewis et al. [32] propose to use unsupervised question generation (e.g., template/rule-based methods) to tackle unsupervised QA task, a setting in which no aligned question, neither context no answer data are available. They demonstrate that their method can outperform early supervised models on SQuAD 1.1 dataset without using the SQuAD training data, and modern QA models can learn to answer human questions surprisingly well using only synthetic training data. In addition, some existing Visual Question Answering (VQA) datasets, such as COCO-QA [51] and Visual Madlibs [67], have also had AQG techniques applied to generate their questions.

However, we argue that most of the questions automatically generated by the above QG models can be applied only to machine reading comprehension setting when a relevant paragraph is given. When used for ODQA task, some questions turn to be ambiguous and result in several potential answers (the same problem we observed in the NewsQA dataset as discussed above). Therefore, we propose a semi-automatic method that combines AQG with a cascade of customized filtering steps to generate the final ODQA dataset, whose resulting questions are non-ambiguous and of good quality. We believe that this approach could be also applied to other types of temporal document collections. Such framework would be also useful in education field, where forming good and clear questions is crucial for evaluating students knowledge and for stimulating self-learning.

3 METHODOLOGY

We introduce here the framework that generates and selects questions from temporal document collections. Fig. 1 shows its architecture which consists of five modules: *Article Selection Module, Question Generation Module, Syntactic & Temporal Filtering/Transforming*

Module, General & Temporal Ambiguity Filtering Module and *Triple-based Filtering Module*. All these 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 selection strategies.

3.1.1 Selection based on Wikipedia Events. The first one relies on the short descriptions of important events available in Wikipedia year pages[5] 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 such event description, we select keywords to be used as search queries for retrieving articles related to this description from the news archive. We choose Yake![6] [5] as our keyword extraction method, which is a state-of-the-art unsupervised approach that relies on statistical features to select the most important keywords. Next, the query composed of the extracted keywords is sent to the ElasticSearch[7] installation which returns the top 25 relevant documents ranked by BM25. Finally, 53,991 news articles are obtained in this way 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 are collected.

We applied these two selection strategies because we wanted the final dataset to contain questions related to important past events as well as also questions on minor issues, especially ones which are likely not recorded in Wikipedia, and thus more challenging and unique.[8]

3.2 Question Generation Module

The second step is to generate questions from the collected articles. We first separate articles into paragraphs and then use a neural network model to generate candidate questions from each paragraph. We apply T5-base [47] - a recent, large, pre-trained Transformer encoder-decoder model. We note that, same as us, Lelkes et al. [31]

---

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

[6]https://yake.inesctec.pt

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

[8]In the experiments we actually show that only a small number of our questions can be successfully answered when using Wikipedia.
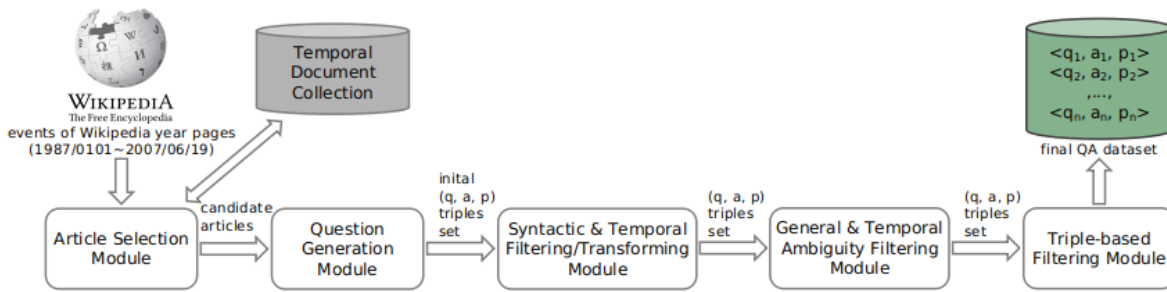
Figure 1: Dataset generation framework

have used QG methods to generate questions from news articles in an automatic way, although in their case PEGASUS model [69] was utilized to generate the questions using the NewsQuizQA dataset. However, we did not choose PEGASUS-base model since we found that it generates questions which sometimes contain information not found in the underlying documents (probably due to the Gap Sentences Generation pre-training task that PEGASUS model applies). Furthermore, the questions generated by Lelkes et al. [31] belong to the quiz-style multiple-choice type which is not suitable for ODQA.

The QG model is fine-tuned using SQuAD 1.1[9] [49] whose inputs are the answers together with their corresponding paragraphs, and the questions form the outputs. 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[10] 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 [43], TriviaQA [19], Quasar-T [10], SearchQA [13] and XQA [36], are also mainly in the form of entities (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 which 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. Finally, we generated 6,408,036 questions in this way from 1,194,730 paragraphs of 106,197 news articles.

3.3 Syntactic & Temporal Filtering/Transforming Module

This module consists of 8 basic processing steps that further 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). For example, question like *"Where did Mr. Roche serve in Vietnam?"* that has gold answer "Vietnam" is 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.1.
(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.2.
(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 of the information contained in the sentence [33, 38]. In contrast to specific sentences that contain informative messages, general sentences do not reveal much specific 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 for question generation.

1) *"Despite recent declines in yields, investors continue to pour cash into money funds."*
2) *"Assets of the 400 taxable funds grew by $1.5 billion during the last week, to $352.7 billion."*

---

[9]We decided not to use NewsQA for training as it contains too many ambiguous questions.
[10]We use the named entity recognizer from 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

Thus, in this step, we aim to remove questions that have been generated from general sentences. We use the training dataset from Ko et al. [24], which is composed of three publicly available, labeled datasets [33, 34, 39]. The 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 [9], RoBERTa-base model [37] and ALBERT-base model [29], 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 [37] as our specificity-determining model because it achieves the best results on the test set - 84.49% accuracy. Finally, we discard all questions whose underlying sentences from which they were generated have been classified by the above-described approach as general. This filtering step removed 952,398 questions. Few examples of the removed general questions are given in Table 3.

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 Table 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 [72] (we are only aware of AMBIGQA dataset [42] which contains multiple possible answers to ambiguous 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 correct answers to such questions is quite difficult, if not impossible, within 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 over time. Note that temporally ambiguous questions are specific to temporal datasets like ours, and consequently 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
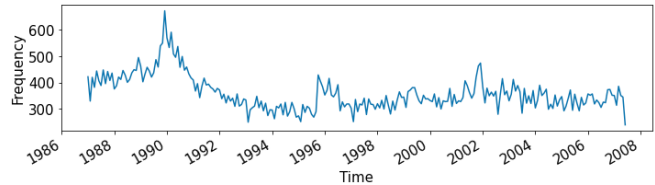
Figure 2: Distribution of articles used in ArchivalQA

from the previous filtering steps.[11] Then, we again fine-tuned three Transformer-based classifiers, same as when training the specificity-evaluating model. The BERT-based model [9] 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.[12] Similarly, in Table 3, we also give few examples of the removed ambiguous questions.

Table 3: Examples of Questions Removed by the General & Temporal Ambiguity Filtering Module.

No.	Question	Answer	Type
1	Who goes to Central Park to walk, touch grass, play?	New Yorkers	General
2	The Italian economy has been deteriorating compared to what other country?	Germany	General
3	Who is a nice, sweet Southern boy that people underestimate?	Bobby	General
4	How many countries are in the World Trade Organization?	142	Temporally ambiguous
5	What country agreed to normalize relations with the United States?	North Korea	Temporally ambiguous
6	What was the unemployment rate in Jordan?	20 percent	Temporally ambiguous

3.5 Triple-based Filtering Module

In the final module, we aim to remove remaining poor quality data instances by analyzing the entire <question, answer, paragraph> triples. Some instances are still problematic due to several reasons (e.g., questions with incorrect answers, questions containing information not found in paragraphs, or other wrong questions that have not been filtered out by the previous filtering stages). To construct the last filter we first created a dedicated dataset by asking 10 annotators to label 10k samples selected from the results obtained after applying the previously-introduced filtering stages. The labels were either "Good" or "Bad" based on <paragraph, question, answer> triples.[13] The annotators had to not only consider the particular 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, which contains 5,699 "Good" questions and 4,301 "Bad" questions, was then randomly split into the training set (90%), and the test set (10%). Then, we trained a RoBERTa-base model [37] that takes the triples as the input after adding a special token ([SEP]) to the question-answer pair

---

[11]This dataset is also made freely available, as it could be useful for improving QG research.

[12]As mentioned before, the data of temporally ambiguous questions is also released, which could be useful for developing systems that can provide multiple possible answers.

[13]This dataset is also available.

and paragraph of each sample. 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 were derived 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. This provenance information is recorded for each question. 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). Table 4 shows few examples. More detailed dataset statistics are presented in Table 5. Fig. 2 shows also the temporal distribution of documents used for producing ArchivalQA questions.

We have also analyzed the named entity types[14] 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[15] account for a large part of ArchivalQA. Further, the right hand side's pie chart in Fig. 3 shows the distribution of 9 event categories of the questions that are classified by another dedicated classifier prepared by us, which has been trained based on the event dataset created by Sumikawa and Jatowt [57] achieving 85.86% accuracy. We can see that ArchivalQA contains questions related to diverse event categories, while the "arts & culture", "politics & elections", "armed conflicts & attacks", "law and crime" and "business & economy" events account for a large portion of questions. Fig. 4 presents also the distribution of frequent trigram prefixes. While nearly half of SQuAD questions are "what" questions [50], the distribution of ArchivalQA is more evenly spread across multiple question types.

4.2 Model Performance

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

(1) DrQA-Wiki [7]: DrQA combines a search component based on bigram hashing and TF-IDF matching with a multi-layer recurrent neural network model trained to extract answers from articles. We first test the DrQA model which uses Wikipedia as the knowledge source (DrQA's default 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 [7]: DrQA model which uses NYT.

(3) DrQA-NYT-TempRes [7]: DrQA model which uses NYT and transforms the answers with relative temporal information by an approach similar to the one we used for transforming

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

relative temporal information in Syntactic & Temporal Filtering/Transforming Module (see 7th and 8th steps of Sec. 3.3).

(4) BERTserini-Wiki [66]: BERTserini tackles end-to-end question answering by combining BERT [9] with the Anserini [65] IR toolkit, with BM25 as the ranking function. We also first test BERTserini model using Wikipedia (BERTserini's default knowledge source).

(5) BERTserini-NYT [66]: BERTserini model which uses NYT.

(6) BERTserini-NYT-TempRes [66]: BERTserini model which uses NYT archive and transforms the relative temporal answers.

(7) DPR-NYT [22][16]: Unlike previous ODQA approaches, this end-to-end QA model incorporates BERT [9] reader module[17] with dense retriever module that has been trained for 15 epochs using ArchivalQA dataset and NYT corpus. In the retriever

---

[14]18 entity types used by NE recognizer in spaCy.

[15]NORP denotes nationality or religious or political groups; for example, "Catholic".

[16]We have not decided to test DPR using Wikipedia as the knowledge source, due to considerable time cost required.

[17]The same reader module that is used in BERTserini model.

Table 4: ArchivalQA Dataset Examples. org_answer, answer_start, trans_que, trans_ans, and source represent the original answer text, its start index in the document, flag indicating whether the question has been transformed, flag showing whether the answer has been transformed and the selection method of the document used for producing the question, respectively. para_id contains concatenated information of the document ID (the metadata of each article in the NYT corpus) and the ith paragraph used to generate the question.

id	question	answer	org_answer	answer_start	para_id	trans_que	trans_ans	source
train_0	Who claimed responsibility for the bombing of Bab Ezzouar?	Al Qaeda	Al Qaeda	184	1839755_20	0	0	wiki
train_4	When did Tenneco announce it was planning to sell its oil and gas operations?	May 26, 1988	today	103	148748_0	0	1	rand
val_45	What threat prompted Mr. Paik's family to flee to Hong Kong?	the Korean War	the Korean War	327	1736040_7	0	0	wiki
test_84	Along with the French Open, what other tournament did Haarhuis win in 1998?	Wimbledon	Wimbledon	527	1043631_15	1	0	rand

Table 5: 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 6: Models' performance on ArchivalQA

Model	EM	F1
DrQA-Wiki [7]	7.53	11.64
DrQA-NYT [7]	38.13	46.12
DrQA-NYT-TempRes [7]	44.84	53.06
BERTserini-Wiki [66]	10.19	16.25
BERTserini-NYT [66]	54.30	66.05
BERTserini-NYT-TempRes [66]	**56.34**	**68.93**
DPR-NYT [22]	44.30	56.64
DPR-NYT-TempRes [22]	49.27	60.72

module, the paragraphs and questions are represented by dense vector representations, computed using two BERT networks. The ranking function is given by the dot product between the query and passage representations.

(8) DPR-NYT-TempRes [22]: DPR model which uses NYT archive and transforms the relative temporal answers.

We measure the performance of the above-listed 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 Table 6. Firstly, we can observe that the models that utilize Wikipedia as the knowledge source perform much worse than the models that use NYT corpus, which is due to many questions being about minor things or events that Wikipedia does not seem to record (or it describes them only shallowly). Secondly, the models that resolve implicit temporal answers perform better than the ones without this step. Temporal information resolution is then clearly important. Thirdly, we notice that BERTserini models outperform DrQA models by large margins. There are two possible reasons, one is that DrQA models retrieve the entire long articles containing many non-relevant sentences rather than short paragraphs; the other is that DrQA uses RNN-base reader component rather than a better choice which would be the BERT-base reader component. Finally, DPR models which use dense vector representations for retrieval also achieve relatively good results on both metrics. Future work

Table 7: Human evaluation results of ArchivalQA

Fluency	Answerability	Relevance	Non-ambiguity
4.80	4.57	4.79	4.60

on combining dense retrieval with sparse retrieval could be studied to further improve the performance.

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 their evaluation. The evaluators 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 whether 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 Table 7. 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.

We then examine the effectiveness of the General & Temporal Ambiguity Filtering Module by analyzing reasons as for why 10 annotators labelled 10k data samples as "Bad" for the Triple-based Filtering Module. As shown in Table 8, among 10k questions, there are 390 (3.90%) questions labelled as "Bad" due to specificity problems, and 806 (8.06%) questions have temporal ambiguity problems.[18] These relatively small numbers suggest that the General & Temporal Ambiguity Filtering Module should have removed most of the questions with specificity or ambiguity issues. The final filtering step using the Triple-based Filtering Module is supposed to remove the remaining "Bad" questions by analyzing <question, answer, paragraph> at the same time.

5 SUB-DATASET CREATION

We also distinguish subparts of the dataset based on the question difficulty levels and the containment of temporal expressions, which we believe could be used for training/testing ODQA systems with

---

[18]Other "Bad" questions are the questions with incorrect answers, questions containing information not found in paragraphs, or questions with bad grammar, etc.)

Table 8: Statistics of the dataset used in Triple-based Filtering

Questions generated from general sentences	390
Temporally ambiguous questions	806
Other "Bad" questions	3,105
"Good" questions	5,699
Total questions	10,000

Table 9: ArchivalQA Sub-Dataset Examples

id	question	answer	sub-dataset
train_134512	What political party was Larry Rockefeller a candidate for?	Republican	Easy
val_45168	What country did President Bush send 30,000 troops to?	Somalia	Difficult
train_123981	What company was formed in 1986 by the merger of Burroughs and Sperry?	Unisys	Exp-Temp
test_26021	What Prince was overthrown by Lon Nol?	Sihanouk	Imp-Temp

diverse strengths and abilities. Table 9 presents few randomly sampled examples for each of the four subdivisions of our dataset which we describe below.

5.1 Difficult/Easy Questions Dataset

We created two sub-datasets (called ArchivalQAEasy and ArchivalQA-Hard) based on the difficulty levels of their questions, such that 100,000 are easy and another 100,000 are difficult questions. We use the open-source Anserini 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 both these two sub-datasets 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 their questions. In particular, we constructed two sub-datasets containing 75,000 questions with temporal expressions and 75,000 without temporal expressions (called ArchivalQATime and ArchivalQANoTime, respectively). We used SUTime [6] 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 accurate answers could be just searched within news articles published during (or perhaps also some time after) 1990. Same as with ArchivalQAEasy and ArchivalQAHard, both ArchivalQATime and ArchivalQANoTime were randomly split into the training (80%, 60,000 examples), development (10%, 7,500 examples), and test sets (10%, 7,500 examples).

5.3 Model Performance on Sub-Datasets

Table 10 presents the performance of different ODQA models over the four sub-datasets discussed above. We can see that all the models achieve better results on ArchivalQAEasy than on ArchivalQAHard, indicating that the questions of ArchivalQAHard tend to be indeed

harder to answer. For example, the improvement of BERTserini-NYT-TempRes is in the range of 106.83% and 86.16% on EM and F1 metrics, respectively. However, DPR models using dense vector representations for retrieving relevant paragraphs are subject to a small performance drop on two sub-datasets (ArchivalQAEasy and ArchivalQAHard) and they manage to surpass the other ODQA approaches that use sparse retrievers by large margins on ArchivalQA-Hard. For example, when considering DPR-NYT-TempRes model on ArchivalQAHard and ArchivalQAEasy, the improvements are only 25.09% and 31.76% on EM and F1, respectively. When comparing DPR-NYT-TempRes with BERTserini-NYT-TempRes on ArchivalQA-Hard, the improvements are 39.39% and 27.37% on EM and F1 metrics, respectively. This is likely because questions in ArchivalQA-Hard contain less lexical overlap with the NYT articles while DPR excels at semantic representation and handles lexical variations well. 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 such important temporal signals to find more relevant documents, which has been widely leveraged in temporal information retrieval [1, 4, 20]. Wang et al. [61, 62] have already used such temporal signals to answer temporally-scoped questions about the past.

6 DATASET USE

Our dataset can be used in several ways. First, ODQA models can use the questions, answers and paragraphs[19] for training their IR and MRC modules [11, 22] 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, for example, training ODQA models with time-aware dense retriever components. As shown in [61, 62], 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 potentially use other temporal news collections that temporally align with the NYT collection (i.e., ones that also span 1987-2007), although naturally this would make the task more challenging. It might be even feasible to consider answering our questions using synchronic knowledge bases such as Wikipedia, although as we have observed earlier, Wikipedia seems to lack a lot of detailed information on the past. The questions in our dataset are often specific and minor, and relate to relatively 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, system testing and comparison can be made to be more fine-grained based on the question difficulty and the occurrence of temporal

---

[19]Note that another way to use the dataset is to train models without using the paragraph information [30].

Table 10: Performance of different models over different Sub-Datasets

Model	ArchivalQAEasy		ArchivalQAHard		ArchivalQATime		ArchivalQANoTime	
	EM	F1	EM	F1	EM	F1	EM	F1
DrQA-NYT [7]	42.10	51.97	22.81	31.24	31.32	42.17	39.59	47.18
DrQA-NYT-TempRes [7]	48.41	57.26	27.37	34.02	33.19	44.01	46.39	54.91
BERTserini-NYT [66]	59.15	69.16	25.00	33.73	50.65	63.24	55.36	68.37
BERTserini-NYT-TempRes [66]	**61.80**	**71.56**	29.88	38.44	**51.12**	**65.67**	**58.27**	**70.19**
DPR-NYT [22]	46.24	59.63	39.99	48.03	42.29	53.73	45.28	57.92
DPR-NYT-TempRes [22]	52.10	64.51	**41.65**	**48.96**	42.91	54.27	51.13	62.75

components contained in questions. Also, another practical application could be to use our generated questions for education, e.g., for evaluating students knowledge and stimulating self-learning in history courses.

7 CONCLUSIONS

We introduce in this paper a novel large-scale ODQA dataset for answering questions over a long-term archival news collection, with the objective to foster the research in the field of ODQA on news archives. Our dataset is unique since it covers the the longest time period among all the ODQA datasets and deals with events that occurred in a relatively distant past. 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, which could also be used to generate high-quality questions over other document collections.

In the future we plan to extend our dataset by incorporating also multi-hop questions in order to foster multi-hop question answering research [41] on archival news article collections.

REFERENCES

[1] Omar Alonso, Jannik Strötgen, Ricardo Baeza-Yates, and Michael Gertz. 2011. Temporal Information Retrieval: Challenges and Opportunities. *Twaw* 11 (2011), 1–8.
[2] Razieh Baradaran, Razieh Ghiasi, and Hossein Amirkhani. 2020. A survey on machine reading comprehension systems. *arXiv preprint arXiv:2001.01582* (2020).
[3] 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 (2014), 1–41.
[4] 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 (2014), 1–41.
[5] 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 (2020), 257–289.
[6] Angel X Chang and Christopher D Manning. 2012. Sutime: A library for recognizing and normalizing time expressions.. In *Lrec*, Vol. 2012. 3735–3740.
[7] Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading wikipedia to answer open-domain questions. *arXiv preprint arXiv:1704.00051* (2017).
[8] Kevin Clark and Christopher D Manning. 2016. Deep reinforcement learning for mention-ranking coreference models. *arXiv preprint arXiv:1609.08667* (2016).
[9] 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* (2018).
[10] Bhuwan Dhingra, Kathryn Mazaitis, and William W Cohen. 2017. Quasar: Datasets for question answering by search and reading. *arXiv preprint arXiv:1707.03904* (2017).
[11] 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* (2020).
[12] Nan Duan, Duyu Tang, Peng Chen, and Ming Zhou. 2017. Question generation for question answering. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. 866–874.
[13] 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* (2017).
[14] Daria Dzendzik, Carl Vogel, and Jennifer Foster. 2021. English Machine Reading Comprehension Datasets: A Survey. *arXiv preprint arXiv:2101.10421* (2021).
[15] 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* (2020).
[16] Yifan Gao, Lidong Bing, Wang Chen, Michael R Lyu, and Irwin King. 2018. Difficulty controllable generation of reading comprehension questions. *arXiv preprint arXiv:1807.03586* (2018).
[17] Karl Moritz Hermann, Tomáš Kočiský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. *arXiv preprint arXiv:1506.03340* (2015).
[18] Wenpeng Hu, Bing Liu, Jinwen Ma, Dongyan Zhao, and Rui Yan. 2018. Aspect-based question generation. (2018).
[19] 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* (2017).
[20] Nattiya Kanhabua and Avishek Anand. 2016. Temporal information retrieval. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*. 1235–1238.
[21] Nattiya Kanhabua, Roi Blanco, and Kjetil Nørvåg. 2015. Temporal Information Retrieval. *Foundations and Trends® in Information Retrieval* 9, 2 (2015), 91–208. https://doi.org/10.1561/1500000043
[22] 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* (2020).
[23] Yanghoon Kim, Hwanhee Lee, Joongbo Shin, and Kyomin Jung. 2019. Improving neural question generation using answer separation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 6602–6609.
[24] 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*, Vol. 33. 6610–6617.
[25] Tomáš Kočiský, 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 (2018), 317–328.
[26] 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* (Glasgow, Scotland UK) (*CHIIR '19*). ACM, New York, NY, USA, 35–43.
[27] Ghader Kurdi, Jared Leo, Bijan Parsia, Uli Sattler, and Salam Al-Emari. 2020. A systematic review of automatic question generation for educational purposes. *International Journal of Artificial Intelligence in Education* 30, 1 (2020), 121–204.
[28] 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 (2019), 453–466.
[29] 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* (2019).
[30] Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. 2019. Latent retrieval for weakly supervised open domain question answering. *arXiv preprint arXiv:1906.00300* (2019).
[31] Adam D Lelkes, Vinh Q Tran, and Cong Yu. 2021. Quiz-Style Question Generation for News Stories. *arXiv preprint arXiv:2102.09094* (2021).

[32] Patrick Lewis, Ludovic Denoyer, and Sebastian Riedel. 2019. Unsupervised question answering by cloze translation. *arXiv preprint arXiv:1906.04980* (2019).

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

[34] 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)*. 3921–3927.

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

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

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

[38] 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*. 605–613.

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

[40] Howard Markel, Harvey B Lipman, J Alexander Navarro, Alexandra Sloan, Joseph R Michalsen, Alexandra Minna Stern, and Martin S Cetron. 2007. Non-pharmaceutical interventions implemented by US cities during the 1918-1919 influenza pandemic. *Jama* 298, 6 (2007), 644–654.

[41] Vaibhac Mavi, Anubhav Jangra, and Adam Jatowt. 2022. A Survey on Multi-hop Question Answering and Generation. *arXiv preprint arXiv:2204.09140* (2022).

[42] Sewon Min, Julian Michael, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2020. AmbigQA: Answering ambiguous open-domain questions. *arXiv preprint arXiv:2004.10645* (2020).

[43] 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* (2016).

[44] 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*.

[45] 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* (2016).

[46] Raul Puri, Ryan Spring, Mostofa Patwary, Mohammad Shoeybi, and Bryan Catanzaro. 2020. Training question answering models from synthetic data. *arXiv preprint arXiv:2002.09599* (2020).

[47] 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* (2019).

[48] Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for SQuAD. *arXiv preprint arXiv:1806.03822* (2018).

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

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

[51] Mengye Ren, Ryan Kiros, and Richard Zemel. 2015. Exploring models and data for image question answering. *Advances in neural information processing systems* 28 (2015).

[52] 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* (2021).

[53] Mrinmaya Sachan and Eric Xing. 2018. Self-training for jointly learning to ask and answer questions. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. 629–640.

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

[55] Apoorv Saxena, Soumen Chakrabarti, and Partha Talukdar. 2021. Question Answering Over Temporal Knowledge Graphs. *arXiv preprint arXiv:2106.01515* (2021).

[56] Siamak Shakeri, Cicero Nogueira dos Santos, Henry Zhu, Patrick Ng, Feng Nan, Zhiguo Wang, Ramesh Nallapati, and Bing Xiang. 2020. End-to-end synthetic data generation for domain adaptation of question answering systems. *arXiv preprint arXiv:2010.06028* (2020).

[57] Yasunobu Sumikawa and Adam Jatowt. 2018. Classifying short descriptions of past events. In *European Conference on Information Retrieval*. Springer, 729–736.

[58] Xingwu Sun, Jing Liu, Yajuan Lyu, Wei He, Yanjun Ma, and Shi Wang. 2018. Answer-focused and position-aware neural question generation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. 3930–3939.

[59] 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* (2016).

[60] 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 (2020), 100007.

[61] 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*. Springer, 774–789.

[62] 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 (2021), 29–54.

[63] Tong Wang, Xingdi Yuan, and Adam Trischler. 2017. A joint model for question answering and question generation. *arXiv preprint arXiv:1706.01450* (2017).

[64] Yansen Wang, Chenyi Liu, Minlie Huang, and Liqiang Nie. 2018. Learning to ask questions in open-domain conversational systems with typed decoders. *arXiv preprint arXiv:1805.04843* (2018).

[65] Peilin Yang, Hui Fang, and Jimmy Lin. 2017. Anserini: Enabling the use of Lucene for information retrieval research. In *Proceedings of the 40th international ACM SIGIR conference on research and development in information retrieval*. 1253–1256.

[66] 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* (2019).

[67] Licheng Yu, Eunbyung Park, Alexander C Berg, and Tamara L Berg. 2015. Visual madlibs: Fill in the blank description generation and question answering. In *Proceedings of the ieee international conference on computer vision*. 2461–2469.

[68] 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 (2020), 7640.

[69] 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*. PMLR, 11328–11339.

[70] 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* (2018).

[71] Yao Zhao, Xiaochuan Ni, Yuanyuan Ding, and Qifa Ke. 2018. Paragraph-level neural question generation with maxout pointer and gated self-attention networks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. 3901–3910.

[72] 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* (2021).

A APPENDIX

A.1 Removal of Questions with Unclear Pronouns

The questions with unclear pronouns are removed in the 6th step of the Syntactic & Temporal Filtering/Transforming 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 [8]) to obtain the coreference results of each sentence in the question set. For example, for the question "*When did Sampras win his first Grand Slam?*", the information that 'his' points to 'Sampras' is derived. Then we apply several heuristic rules to collect only clear-pronoun questions. A sentence is considered correct if its pronoun points to named entities appearing inside the question's content (e.g., 'Sampras' in the previous example), or if the question asks about the actual resolution of the pronoun (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/Transforming Module. We apply SUTime [6] to recognize temporal expressions, and we use the publication date information of the articles, which 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 would be too 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 content of NYT's news articles for answering our dataset's questions. We expect that models which need to use temporal expressions should utilize article timestamps to resolve temporal expressions.