Answering Event-Related Questions over Long-term News Article Archives

Jiexin Wang[1], Adam Jatowt[1], Michael Färber[2], and Masatoshi Yoshikawa[1]

[1] Department of Social Informatics, Kyoto University, Japan
wang.jiexin.83m@st.kyoto-u.ac.jp
[2] Karlsruhe Institute of Technology, Germany

Abstract Long-term news article archives are valuable resources about our past, allowing people to know detailed information of events that occurred at specific time points. To make better use of such heritage collections, this work considers the task of large scale question answering on long-term news article archives. Questions on such archives are often event-related. In addition, they usually exhibit strong temporal aspects and can be roughly categorized into two types: (1) ones containing explicit temporal expressions, and (2) ones only implicitly associated with particular time periods. We focus on the latter type as such questions are more difficult to be answered, and we propose a retriever-reader model with an additional module for reranking articles by exploiting temporal information from different angles. Experimental results on carefully constructed test set show that our model outperforms the existing question answering systems, thanks to an additional module that finds more relevant documents.

Keywords: News Archives · Question Answering · Information Retrieval

1 Introduction

With the application of digital preservation techniques, more and more old news articles are being digitized and made accessible online. News article archives help users to learn detailed information on events that occurred at specific time points in the past and constitute part of our heritage [1]. Some professionals, like historians, sociologists, or journalists need to deal with these time-aligned document collections for a variety of purposes [2]. Moreover, average users can verify information about the past using original, primary resources. However, it is difficult for users to efficiently make use of news archives due to their large sizes and complexities. Large scale question answering systems (QA systems) can solve the problem, with the aim to identify the most correct answer from relevant documents for a particular information need, expressed as a natural language question. User questions on these archives are often event-related and include temporal aspects. They can be divided into two types: (1) those with explicit temporal expressions (e.g., "NATO signed a peace treaty with which country in 1999?"), and (2) those only implicitly associated to time periods, hence not

Table 1. Examples of questions in our test set, their answers, and dates of their events

Questions	Answers	Event Dates
Which party, led by Buthelezi, threatened to boycott the South African elections?	Inkatha Freedom Party	1993.08
What bill was signed by Clinton for firearms purchases?	Brady Bill	1993.11
Which federal prosecutor that led the investigation for the leak of identity of Valerie Plame?	Patrick J. Fitzgerald	2003.11
Riot in Los Angeles occurred because of the acquittal of how many officers in police department?	Four	1992.04
Which American professional pitcher died because his small airplane crashed in New York?	Cory Lidle	2006.10

containing any temporal expression (e.g., "How many members of International Olympic Committee were expelled or resigned because of the bribery scandal?"). We focus on the latter type, which is more challenging, as the temporal information cannot be obtained directly. Table 1 shows some examples of the questions that we use.

This paper presents a large scale question answering system called QANA (Question Answering in News Archives) designed specifically for answering event-related questions on news article archives. It exploits the temporal information of a question, of a document content and of its timestamp for reranking candidate documents. In the experiments, we use New York Times (NYT) archive as the underlying knowledge source and a carefully constructed test set of questions which are associated with past events. The questions are selected from existing datasets and history quiz websites, and they lack any temporal expressions which makes them particularly difficult to be answered. Experimental results show that our proposed system improves retrieval effectiveness and outperforms the existing QA systems commonly used for large scale question answering.

We make the following contributions: (a) we propose a new subtask of QA, which uses long-term news archives as the data source, (b) we build effective models for solving this task by exploiting temporal characteristics of both questions and documents, (c) we perform experiments to prove their effectiveness and construct a novel dedicated test set for evaluating QA on news archives.

The remainder of this paper is structured as follows. The next section overviews the related work. In Section 3, we introduce our model. Section 4 describes experimental settings and results. Finally, we conclude the paper in Section 5.

2 Related Work

Question Answering System Current large scale question answering systems usually consist of two modules: (1) IR module (called also a document retriever module) responsible for selecting relevant articles from an underlying corpus and (2) Machine Reading Comprehension (MRC) module (called also a document reader module) used to extract answer spans from relevant articles, typically, by using neural network models.

Latest MRC models, especially those that use Bert [3] can even surpass human-level performance (based on EM (Exact Match) and F1 scores) on both SQuAD 1.1 [4] and SQuAD 2.0 [5], the two most widely-used MRC datasets, where each question is connected with a given reading passage. However, recent

studies [6, 7, 8] indicate that IR module is a bottleneck having a significant impact on the performance of the whole system (degraded performance of MRC component due to noisy input). Hence, few works tried to improve the IR task. Chen et al. [9] propose one of the most well-known large scale question answering system, DrQA whose IR component is based on a TF-IDF retriever that uses bigrams with TF-IDF matching. Wang et al. [7] introduce R^3 model, where IR component and MRC component are trained jointly by reinforcement learning. Ni et al. [10] propose ET-RR model, which improves IR part by identifying essential terms of a question and reformulating the query.

Nonetheless, as the existing question answering systems are essentially designed for synchronic document collections (e.g., Wikipedia), they are incapable of utilizing temporal information like document timestamp when answering questions on long-term news article archives, despite temporal information constituting an important feature of events reported by news articles. The questions and documents are then processed in the same way as on synchronic collections. Even though some temporal question answering systems that can exploit temporal information of question and document content have been proposed in the past [11, 12], they are still designed for synchronic document collections (e.g., Wikipedia or Web) and they do not use document timestamps. Besides, they are based on traditional rule-based methods and their performance is rather poor.

In addition, there are very few resources available for temporal question answering. Jia et al. [13] propose a dataset with 1,271 temporal question-answer pairs where 209 pairs do is without any explicit temporal expression. However, only few pairs can be used in our case, as most are about events which happened long time ago (e.g., Viking Invasion of England) or are not event-related.

Our approach contains an additional module that is used for reranking documents which improves the retrieval of correct documents by exploiting temporal information from different angles. We not only utilize the inferred time scope information from the questions themselves, but also combine it with the document timestamp information and with temporal information embedded inside document content. To the best of our knowledge, no studies, as well as no available datasets that can help to design a question answering system on news article archives have been proposed so far. Building a system that makes full use of the past news articles and satisfies different user information needs is however of great importance due to the continuously growing document archives.

Temporal Information Retrieval In Information Retrieval (IR) domain, several research studies have already been proposed for temporal ranking of documents [14, 15, 16]. Li and Croft [17] introduce a time-based language model that takes into account timestamp information of documents to favor recent documents. Metzler et al. [18] propose a method that analyzes query frequencies over time to infer the implicit temporal information of queries and exploit this information for ranking results. Arikan et al. [19] design a temporal retrieval model that integrates temporal expressions of document content into query-likelihood language modeling. Berberich et al. [20] propose a similar model but also consider uncertainty in temporal expressions. However, in [19] and [20], the temporal

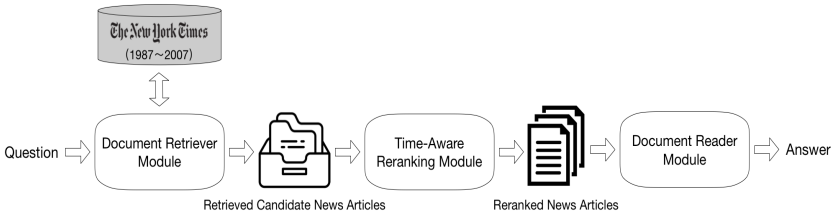

Figure 1. The Architecture of QANA System

scopes of queries are explicitly given in their setting and the proposed methods do not utilize timestamp information. Kanhabua and Nørvåg [21] propose three different methods to determine the implicit temporal scope of queries and exploit this temporal information to improve the retrieval effectiveness by reranking documents. [21] is probably the most related work to ours as it also linearly combines both textual and temporal similarity to rerank documents, however, that work does not use any temporal information embedded in document content and the linear combination is done in a static way. In our experiments, for comparison with [21] we will replace our ranking method in the reranking module with the best one proposed in [21].

All the above-mentioned temporal information retrieval methods are designed for short queries instead of questions, and none of them exploits both timestamps and content temporal information. We are the first to adapt and improve concepts from temporal information retrieval to the QA research domain, showing significant improvement in answering questions on long-term news archives.

3 Methodology

In this section, we present the proposed system that is designed specifically for answering questions over news archives. We focus on questions for which the time periods are not given explicitly, and so further knowledge is required for obtaining or inferring their time periods (e.g. "Who replaced Goss as the director of the Central Intelligence Agency?"). Fig. 1 shows the architecture of QANA system which is composed of three modules: *Document Retriever Module*, *Time-Aware Reranking Module* and *Document Reader Module*. Compared with the architectures of other common large scale question answering systems, we add an additional component: Time-Aware Reranking Module which exploits temporal information from different angles for selecting the best documents.

3.1 Document Retriever Module

This module firstly performs keyword extraction and expansion, then retrieves candidate documents from the underlying document archive. First, single-token nouns, compound nouns, and verbs from each question are extracted based on analyzing part of speech (POS) and dependency information using spaCy[3]. After removing common stop words, the module expands keywords with their synonyms taken from WordNet [22]. The synonyms are further filtered by keeping

[3] https://spacy.io/

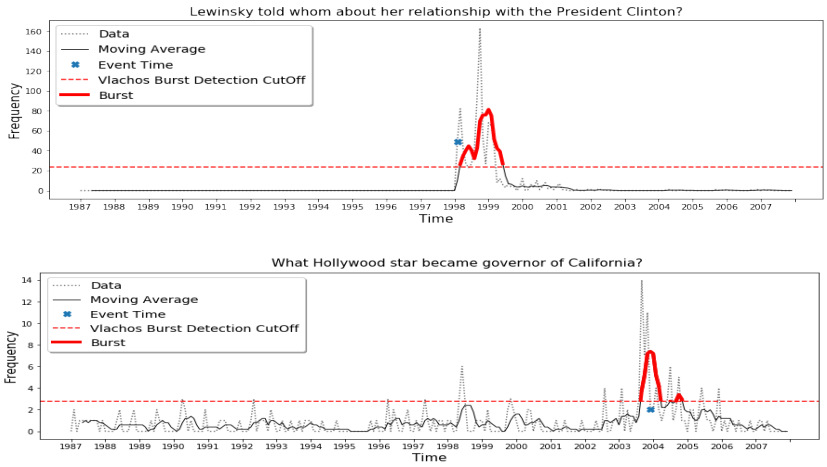

Figure 2. Burst detection results of two questions

those whose POS types match the original term in the question, and whose word embeddings[4] similarity to question terms is over 0.5. Finally, a query is issued to Solr [24] search engine which returns the top 300 documents ranked by BM25.

3.2 Time-Aware Reranking Module

In this module, temporal information is exploited from different angles to rerank retrieved candidate documents. Since the time scope information of questions is not provided explicitly, the module firstly determines candidate periods of the time scope $T(Q)$ of a question Q. These are supposed to represent when an event mentioned in the question could occur. Each inferred candidate period is assigned a weight to indicate its importance. Then, the module contrasts the query time scope against the information derived from the document timestamp $t_{pub}(d)$ and the temporal information embedded inside document content $T_{text}(d)$, in order to compute two temporal scores $S^{temp}_{pub}(d)$ and $S^{temp}_{text}(d)$ for each candidate document d. Finally, both the textual relevance score $S^{rel}(d)$ and the final temporal score $S^{temp}(d)$ are used for document reranking.

Query Time Scope Estimation Although the time scope information of the questions is not given explicitly, the distributions of relevant documents over time should provide information regarding temporal characteristics of the questions. Examining the timeline of a query's result set should allow us to characterize how temporally dependent the topic is. For example, in Fig. 2, the dashed lines of the data show the distribution of relevant documents obtained from the NYT archive per month for two example questions: "Lewinsky told whom about her relationship with the President Clinton?", and "Which Hollywood star became governor of California?". We use a cross mark to indicate the time of each corresponding event, which is also the true time scope of the question.

We can see that the actual time scope (January, 1988) of the first question is reflected relatively well by its distribution of relevant documents as generally

[4] We use Glove [23] embeddings trained on the Common Crawl dataset with 300 dimensions.

these documents are located between 1998 and 1999. However, still most of the relevant documents are published in October rather than January, because another event - the impeachment of Bill Clinton - occurred at that time. On the other hand, the distribution of relevant documents corresponding to the second question is more complex as it contains many peaks, and documents are not located in a specific short time period, and the number of relevant documents published around the actual event time is relatively small when compared to the total number of relevant documents. However, the distribution line near the actual time of the event (November, 2003) still reveals useful features, i.e, the highest peak (maxima) of the dashed line of the data is near the event time. Therefore, the characteristics of the distribution of relevant documents over time can be used for inferring hidden time scopes of questions.

We perform burst detection on the retrieved relevant time-aligned documents, as the time and the duration of bursts are likely to signify the start point and the end point of events underlying the questions. More specifically, we apply burst detection method used by Vlachos et al. [25], which is a simple yet effective approach[5]. Bursts are detected as points with values higher than β standard deviations above the mean value of the moving average (MA). The procedure of calculating the candidate periods of time scope $T(Q)$ of question Q is as follows:

Algorithm 1: Query Time Scope Estimation

INPUT: Timestamp sequence $T_{pub}(Q)$, window size w, cutoff parameter β
OUTPUT: Candidate periods of question time scope $T(Q)$

$T(Q) \leftarrow \emptyset$;
calculate moving average MA_w of w for sequence $T_{pub}(Q)$;
$cutoff \leftarrow mean(MA_w) + \beta \cdot std(MA_w)$;
$T(Bursts) \leftarrow \{t_i | MA_w(t_i) > cutoff\}$, and further represented by $(t(Burst_1), t(Burst_2), ...)$, t_i is a time point;
$C \leftarrow \{t(Burst_0)\}$;
foreach $t(Burst_j) \in T(Bursts)$ **do**
    if $t(Burst_j) == t(Burst_{j+1}) - 1$ *// test if two bursts are adjacent //*
    then
        $C \leftarrow C \cup \{t(Burst_{j+1})\}$ *// add $t(Burst_{j+1})$ to C if true //*
    else
        $t_i^s \leftarrow C.selectFirstElement()$;
        $t_i^e \leftarrow C.selectLastElement()$;
        $T(Q) \leftarrow T(Q) \cup \{(t_i^s, t_i^e)\}$;
    end
end

$T_{pub}(Q)$ can be easily obtained by collecting timestamp information of each retrieved candidate document, $T(Q)$ is a list of tuples of t_i^s and t_i^e, which are two border time points of the ith estimated time period. There are two parameters in our burst detection: w and β. For simplicity, moving Average MA_w of $T_{pub}(Q)$ of each question is calculated using w equal to 4, corresponding to four months. Following [25] that uses typical values of β equal to 1.5-2.0, we use 2.0 in the experiments. In Fig. 2, the red solid lines show the bursts of previously mentioned two example questions. The inferred time scope of the first question is [('1998-03', '1999-05')], while the time scope of the second question contains three periods: [('2003-08', '2004-02'), ('2004-06', '2004-06'), ('2004-09', '2004-10')]. Note that

[5] Note that other techniques could be used to perform burst detection (e.g., [26], [27], [28])

the second time period of the second time scope is actually a single time point (shown as a single small red point in the graph).

After calculating $T(Q)$, each candidate period is assigned a weight indicating its importance, which is obtained by dividing the number of documents published within the period over the total number of documents published in all the candidate periods of time scope $T(Q)$. For example, for the second example question, the number of documents published within the period ('2003-08', '2004-02') is 43, while the total number of documents published within all the periods $T(Q)$ is 55, so the weight assigned to this period is $\frac{43}{55}$. We use $W(T(Q))$ to represent the weight list, such that $W(T(Q)) = [(w(t_1^s, t_1^e)), ...(w(t_m^s, t_m^e))]$, where m is the number of candidate periods of time scope $T(Q)$.

Timestamp-based Temporal Score Calculation After obtaining candidate periods of time scope $T(Q)$, the module computes the timestamp-based temporal score $S_{pub}^{temp}(d)$ of each candidate document d as shown in Eq. 1. We calculate $S_{pub}^{temp}(d)$ based on the intuition that articles published within or soon after time period of the question have high probability of containing detailed information of the event mentioned in the question. The calculation way is as follows:

$$\begin{aligned} S_{pub}^{temp}(d) &= P(T(Q)|t_{pub}(d)) = P(\{(t_1^s, t_1^e), ...(t_m^s, t_m^e)\}|t_{pub}(d)) \\ &= \frac{1}{m} \sum_{(t_i^s, t_i^e) \in T(Q)} P((t_i^s, t_i^e)|t_{pub}(d)) \end{aligned} \tag{1}$$

$S_{pub}^{temp}(d)$ is estimated as $P(T(Q)|t_{pub}(d))$, which is the average probability of generating m candidate periods of time scope $T(Q)$. Then, the probability of generating a period (t_i^s, t_i^e) given document timestamp $t_{pub}(d)$ is defined as:

$$P((t_i^s, t_i^e)|t_{pub}(d)) = \begin{cases} 0.0 & when\ t_i^s > \ t_{pub}(d) \\ w(t_i^s, t_i^e) \cdot (1.0 - \frac{|t_i^s - t_{pub}(d) + t_i^e - t_{pub}(d)|}{2 \cdot TimeSpan(D)}) & elsewhere \end{cases} \tag{2}$$

$TimeSpan(D)$ is the length of time span of news archive D. In the experiments, we use NYT archive with monthly granularity, so $TimeSpan(D)$ equals to 246 units, corresponding to the number of all months in the archive. $w(t_i^s, t_i^e)$ is the weight indicating the importance of (t_i^s, t_i^e) over candidate periods of time scope $T(Q)$ (as explained before). $P((t_i^s, t_i^e)|t_{pub}(d))$ equals to 0.0 when document d is published before t_i^s, as such document usually cannot provide much information on the events that occurred after its publication. Otherwise, $P((t_i^s, t_i^e)|t_{pub}(d))$ can be larger when the timestamp is closer to the time period (t_i^s, t_i^e), and when the importance weight $w(t_i^s, t_i^e)$ of this period is large.

Content-based Temporal Score Calculation Next, we compute another temporal score, $S_{text}^{temp}(d)$, of a candidate document d based on the relation between temporal information embedded in d's content and the candidate periods of time scope $T(Q)$. We compute $S_{text}^{temp}(d)$ because some news articles, even the ones published long time ago after the events mentioned in questions, may retrospectively refer to these events, providing salient information on them, and can thus help to distinguish between similar events. For example, articles published near a certain US presidential election may also discuss previous elections

for comparison or for other purposes. Such references are often in the form of temporal expressions that refer to particular points in the past.

Temporal expressions are detected and normalized by the combination of temporal tagger (we use SUTime [29]) and temporal signals[6] (words that help to identify temporal relations, e.g. "before","after","during"). The normalized result of each temporal expression is mapped to the time interval with the "start" and "end" information. For example, temporal expression "between 1999 and 2002" is normalized to [('1999-01', '2002-12')]. Special cases like "until January 1992" are normalized as [('', '1992-01')], since the "start" temporal information can not be determined. Finally, we can obtain a list of time scopes of temporal expressions contained in a document d, denoted as $T_{text}(d) = \{\tau_1, \tau_2, ..., \tau_{m(d)}\}$ where $m(d)$ is the total number of temporal expressions found in d.

As each time scope τ_i has its "start" information, denoted as τ_i^s, and "end" information, τ_i^e, we create two lists $T_{text}^s(d)$, $T_{text}^e(d)$ containing all τ_i^s and all τ_i^e, respectively. Next, we construct two probability density functions by using kernel density estimation (KDE) based on these two lists. KDE is a technique closely related to histograms, which has characteristics that allow it to asymptotically converge to any density function. The probabilities of $t^s(Q)$ and $t^e(Q)$ denoted as $S_{text}^{temp\_b}(d)$, $S_{text}^{temp\_e}(d)$, respectively, can be then estimated using the probability density functions.

$$S_{text}^{temp\_b}(d) = \hat{f}\left(t^s(Q); h\right) = \frac{1}{m(d)} \sum_{i=1}^{m(d)} K_h\left(t^s(Q) - \tau_i^s\right) \tag{3}$$

where h is a bandwidth (equal to 4) and K is a Guassian Kernel defined by:

$$K_h\left(x\right) = \frac{1}{\sqrt{2\pi} \cdot h} exp\left(-\frac{x^2}{2 \cdot h}\right) \tag{4}$$

$S_{text}^{temp\_e}(d)$ is calculated in the same way but using τ_i^e and $t^e(Q)$, and $S_{text}^{temp}(d)$ is:

$$S_{text}^{temp}(d) = \frac{1}{2} \cdot (S_{text}^{temp\_b}(d) + S_{text}^{temp\_e}(d)) \tag{5}$$

Final Temporal Score Calculation & Document Ranking After computing the two temporal scores, the final temporal score of d is given by:

$$S^{temp}(d) = \frac{1}{2} \cdot (S_{pub}^{temp'}(d) + S_{text}^{temp'}(d)) \tag{6}$$

where $S_{pub}^{temp'}(d)$ and $S_{text}^{temp'}(d)$ are the normalized values computed by dividing by the corresponding maximum scores among all candidate documents.

Additionally, document relevance score $S^{rel}(d)$ is used after normalization:

$$S^{rel}(d) = \frac{BM25(d)}{MAX\_BM25} \tag{7}$$

Finally, we rerank documents by a linear combination of their relevance scores and temporal scores:

$$S(d) = (1 - \alpha(Q)) \cdot S^{rel}(d) + \alpha(Q) \cdot S^{temp}(d) \tag{8}$$

$\alpha(Q)$ is an important parameter, which determines the proportion between document temporal score and its relevance score. For example, when $\alpha(Q)$ equals

[6] We use the list of temporal signals taken from [13].

to 0.0, the relevance of the temporal information is completely ignored. As different questions have different shapes of the distributions of their relevant documents, we propose to dynamically determine $\alpha(Q)$ per each question. The idea is that when a question has many bursts, meaning that the event of the question is frequently mentioned at different times or many similar or related events occurred over time, then time should play lesser role. In this case we want to decrease $\alpha(Q)$ value to pay more attention to document relevance. In contrast, when only few bursts are found, which means that the question has obvious temporal character, time should be considered more. $\alpha(Q)$ is computed as follows:

$$\alpha(Q) = \begin{cases} 0.0 & when\ burst\_num = 0 \\ ce^{-(1-\frac{1}{burst\_num})} & elsewhere \end{cases} \quad (9)$$

c is a constant set to 0.25. $\alpha(Q)$ assumes small values when the number of bursts is high, while it is the highest for the case of a single burst. When the relevant document distribution of the question does not exhibit any bursts, which also means that the list of candidate periods of the question time scope ($T(Q)$) is empty, $\alpha(Q)$ is set to 0 and the reranking is based on document relevance.

3.3 Document Reader Module

For this module, we utilize a commonly used MRC model called BiDAF [30] which achieves Exact Match score 68.0 and F1 score 77.5 on the SQuAD 1.1 dev set. We use BiDAF model to extract answers of the top N reranked documents and we select the most common answer as the final answer. Note that BiDAF could be replaced by other MRC models, for example, the models that combine with Bert [3]. We use BiDAF for easy comparison with DrQA, whose reader component performance is similar although a little better than the one of BiDAF.

4 Experiments

4.1 Experimental Setting

Document Archive and Test Set As we mentioned before, NYT archive [31] is used as the underlying document collection, and is indexed using Solr. The archive contains over 1.8 million articles published from January 1987 to June 2007 and is often used for Temporal Information Retrieval researches [15, 16].

To evaluate the performance of our approach, we first need a set of answerable questions. To the best of our knowledge, there was no previous proposal for answering questions on news archives or available question answering test sets designed for news archives. Thus we have to manually construct the test set making sure that the questions can be answered in NYT archive. We finally construct a test set of 200 questions[7] for NYT archive, that are carefully selected from other existing datasets and history quiz websites, and that (a) fall into the time frame of NYT archive, (b) their answers could be found in NYT archive and (c) they do not contain any temporal expressions[8]. The second condition was

[7] The test set is available at `https://www.dropbox.com/s/ygy7xy4k80wmcfl/TestQuestion.csv?dl=0`

[8] We note that we have also tested QANA on 200 separate questions containing explicit temporal expressions, hence with time scopes directly given, and found that it outperforms the same baselines with even better results.

Table 2. Resources used for constructing the test set

Resources	Number
TempQuestions [13]	15
SQuAD 1.1 [4]	15
history quizzes from funtrivia[9]	50
quizwise[10]	70
Wikipedia pages	50
Total	200

Table 3. Performance of different models using EM and F1

Model	Top 1		Top 5		Top 10		Top 15	
	EM	F1	EM	F1	EM	F1	EM	F1
DrQA-NYT [9]	22.50	27.58	28.00	32.78	29.50	34.11	32.00	36.87
DrQA-Wiki [9]	21.00	26.17	22.50	27.92	26.00	31.49	29.00	34.37
QA-NLM-U [21]	23.50	30.54	33.00	39.71	41.00	48.02	43.00	50.71
QA-Not-Rerank [30]	25.50	32.45	30.00	37.84	40.50	47.32	42.00	48.95
QANA-TempPub	26.00	33.69	36.00	42.75	39.50	47.19	44.00	50.71
QANA-TempCont	22.50	29.70	32.50	40.67	41.50	49.05	44.50	51.09
QANA	**26.50**	**34.27**	**37.00**	**43.76**	**42.00**	**49.20**	**45.50**	**52.71**

verified by manually selecting correct keywords from the questions and checking whether at least one retrieved document can infer the correct answer. Table 2 shows the distribution of resources used for creating the test set while Table 1 gives few examples.

Baselines and Methods We test the following models:

1. DrQA-NYT [9]: DrQA system which uses NYT archive.
2. DrQA-Wiki [9]: DrQA system which uses Wikipedia as its unique knowledge source. We would like to test if Wikipedia could be sufficient for answering questions on events distant in the past.
3. QA-NLM-U [21]: QA system that uses the best reranking method in [21], while the Document Retriever Module and Document Reader Module are the same as the modules of QANA.
4. QA-Not-Reranking [30]: QANA system without reranking module, same as other large scale question answering systems. The Document Retriever Module and Document Reader Module are the same as the modules of QANA.
5. QANA-TempPub: QANA version that uses only temporal information of timestamp for reranking in Time-Aware Reranking Module.
6. QANA-TempCont: QANA version that only uses temporal information embedded in document content for Time-Aware Reranking Module.
7. QANA: QANA with complete Time-Aware Reranking Module.

4.2 Experimental Results

We measure the performance of the models under comparison using exact match (EM) and F1 score - the two standard measures commonly used in QA research. As shown in Table 3, QANA with full components outperforms other systems for all different N, which represent the numbers of reranked documents used in

[9] http://www.funtrivia.com/quizzes/history/index.html
[10] https://www.quizwise.com/history-quiz

Table 4. Performance of the models when answering questions having few relevant documents vs. when answering questions with many relevant documents

		Top 1		Top 5		Top 10		Top 15	
		EM	**F1**	**EM**	**F1**	**EM**	**F1**	**EM**	**F1**
Questions with few relevant documents	QA-Not-Rerank	31.00	40.48	35.00	43.93	46.00	55.79	48.00	55.12
	QANA	31.00	40.52	45.00	54.18	48.00	57.28	52.00	59.22
Questions with many relevant documents	QA-Not-Rerank	20.00	24.41	25.00	31.75	35.00	42.86	36.00	42.84
	QANA	22.00	28.02	29.00	33.33	36.00	41.11	39.00	46.21

the Document Reader Module. The performance improvement is due to the use of temporal information for locating more correct documents which is derived from the question itself, document timestamp and document content. We then compare our model with others by considering the top 1 and top 5 documents. When comparing with the DrQA system, which is often used as QA baseline, the improvement is in the range of 17.77% to 32.14%, and from 24.25% to 33.49% on EM and F1 metrics, respectively.

We have also examined the performance of DrQA when using Wikipedia articles as its knowledge source. In this case, the results are worse than the ones of any other compared method that uses NYT (including DrQA), which implies that Wikipedia cannot successfully answer questions on distant past events, and they need to be answered using primary sources, i.e., news articles from the past.

When comparing with QA-NLM-U [21], the improvement ranges from 12.76% to 12.12% on EM score, and 12.21% to 10.19% on F1 score. In addition, when comparing with QA-Not-Rerank [30] that does not include reranking module, we can also observe an obvious improvement, when considering the top 5 and top 15 documents, ranging from 23.33% to 8.33%, and from 15.64% to 7.11% on EM and F1 metrics, respectively. Moreover, QANA-TempPub performs better than QANA-TempCont when using the top 1 and top 5 documents, but worse when using top 10 and top 15. In addition, we can observe that just using only timestamp information still allows achieving relatively good performance. Nevertheless, QANA with all the proposed components, which make use of the inferred time scope of the questions and the temporal information from both document timestamps and document content, achieves the best results.

We next evaluate the performance of QANA based on the number of relevant documents, and compare it with QA-Not-Rerank. We first rank questions by the number of documents they return, and then group them into two equal parts. As shown in Table 4, we can see that both the tested models achieve better results on questions with few relevant documents, as it is likely easier to locate more relevant documents from small number of documents. We also observe an improvement when comparing our model with QA-Not-Rerank, especially, for the top 5 and top 15 documents, which proves the effectiveness of the reranking method by utilizing temporal information.

Moreover, we also analyze the impact of the number of bursts on performance. About half of the questions (96 questions) have few bursts (less than or equal to 4). Table 5 shows that both QANA and QA-Not-Rerank perform much better when answering such questions. The events in questions with many bursts are likely to be similar to other events that occurred at different times, which causes the difficulty to distinguish between the events. As our system considers the

Table 5. Performance of the models when answering questions with few bursts vs. when answering questions with many bursts

		Top 1		Top 5		Top 10		Top 15	
		EM	**F1**	**EM**	**F1**	**EM**	**F1**	**EM**	**F1**
Questions with few bursts	QA-Not-Rerank	30.20	37.24	38.54	44.32	45.83	52.55	50.00	56.79
	QANA	30.20	38.11	42.70	48.55	46.87	54.98	52.08	58.96
Questions with many bursts	QA-Not-Rerank	21.15	28.10	22.11	31.87	35.57	40.16	34.61	41.74
	QANA	23.07	30.72	31.73	39.33	37.50	43.86	39.42	46.95

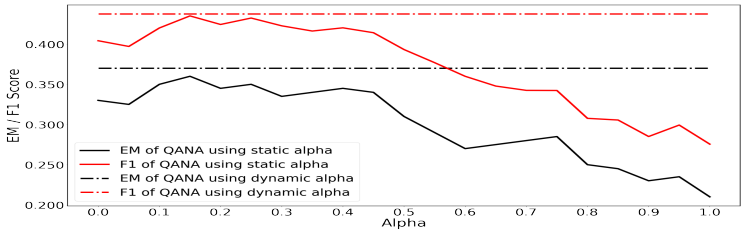

Figure 3. QANA Performance with different static alpha values vs. one with dynamic alpha for the top 5 reranked documents

importance of bursts by assigning weights to them, it significantly outperforms QA-Not-Rerank. Although $\alpha(Q)$ is smaller in this case (according to Eq. 9), it still plays an important part in selecting relevant documents. For example, if the number of bursts of a question is 10, $\alpha(Q)$ approximately equals to 0.1, which means that the final reranking is driven by about 10% of the temporal score.

Finally, we examine the effect of $\alpha(Q)$, which determines the proportion between temporal relevance score and query relevance score. As shown in Fig. 3, the model using dynamic alpha (depicted by dashed lines) performs always better than the model with static alpha, since the dynamic value is dependent on the distributions of relevant documents over time for each question. The dynamic approach flexibly captures the changes in importance of temporal information and relevance information, resulting in better overall performance.

5 Conclusions

In this work we propose a new research task of answering event-related questions on long-term news archives and we show effective solution for it. Unlike other common QA systems designed for synchronic document collections, questions on long-term news archives are usually influenced by temporal aspects, resulting from the interplay between the document timestamps, temporal information embedded in document content and query time scope. Therefore, exploiting temporal information is crucial for this type of QA, as also demonstrated in our experiments. We are also the first to incorporate and adapt temporal information retrieval approaches to QA systems.

Finally, our work makes few general observations. First, to answer event-related questions on long-span news archives one needs to (a) *infer the time scope embedded within a question*, and then (b) *rerank documents based on their closeness and order relation to this time scope*. Moreover, (c) *using temporal expressions in documents* further helps to select best candidates. Lastly, (d) *applying dynamic way to determine the importance between query relevance and temporal relevance is quite helpful.*

Bibliography

[1] Laura Korkeamäki and Sanna Kumpulainen. 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, 2019. ACM. ISBN 978-1-4503-6025-8. https://doi.org/10.1145/3295750.3298931. URL `http://doi.acm.org/10.1145/3295750.3298931`.

[2] Tessel Bogaard, Laura Hollink, Jan Wielemaker, Lynda Hardman, and Jacco Van Ossenbruggen. Searching for old news: User interests and behavior within a national collection. In *Proceedings of the 2019 Conference on Human Information Interaction and Retrieval*, pages 113–121. ACM, 2019.

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

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

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

[6] Christian Buck, Jannis Bulian, Massimiliano Ciaramita, Wojciech Gajewski, Andrea Gesmundo, Neil Houlsby, and Wei Wang. Ask the right questions: Active question reformulation with reinforcement learning. *arXiv preprint arXiv:1705.07830*, 2017.

[7] Shuohang Wang, Mo Yu, Xiaoxiao Guo, Zhiguo Wang, Tim Klinger, Wei Zhang, Shiyu Chang, Gerald Tesauro, Bowen Zhou, and Jing Jiang. R3: Reinforced ranker-reader for open-domain question answering. In *AAAI*, 2018.

[8] Wei Yang, Yuqing Xie, Aileen Lin, Xingyu Li, Luchen Tan, Kun Xiong, Ming Li, and Jimmy Lin. End-to-end open-domain question answering with bertserini. *arXiv preprint arXiv:1902.01718*, 2019.

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

[10] Jianmo Ni, Chenguang Zhu, Weizhu Chen, and Julian McAuley. Learning to attend on essential terms: An enhanced retriever-reader model for scientific question answering. *arXiv preprint arXiv:1808.09492*, 2018.

[11] Marius Pasca. Towards temporal web search. In *Proceedings of the 2008 ACM symposium on Applied computing*, pages 1117–1121. ACM, 2008.

[12] Sanda Harabagiu and Cosmin Adrian Bejan. Question answering based on temporal inference. In *Proceedings of the AAAI-2005 workshop on inference for textual question answering*, pages 27–34, 2005.

[13] Zhen Jia, Abdalghani Abujabal, Rishiraj Saha Roy, Jannik Strötgen, and Gerhard Weikum. Tempquestions: A benchmark for temporal question answering. In *Companion of the The Web Conference 2018 on The Web Conference 2018*, pages 1057–1062. International World Wide Web Conferences Steering Committee, 2018.

[14] Omar Alonso, Michael Gertz, and Ricardo Baeza-Yates. On the value of temporal information in information retrieval. In *ACM SIGIR Forum*, volume 41, pages 35–41. ACM, 2007.

[15] Ricardo Campos, Gaël Dias, Alípio M Jorge, and Adam Jatowt. Survey of temporal information retrieval and related applications. *ACM Computing Surveys (CSUR)*, 47(2):15, 2015.

[16] Nattiya Kanhabua, Roi Blanco, and Kjetil Nørvåg. Temporal information retrieval. *Foundations and Trends in Information Retrieval*, 9(2):91–208, 2015. https://doi.org/10.1561/1500000043. URL `https://doi.org/10.1561/1500000043`.

[17] Xiaoyan Li and W Bruce Croft. Time-based language models. In *Proceedings of the twelfth international conference on Information and knowledge management*, pages 469–475. ACM, 2003.

[18] Donald Metzler, Rosie Jones, Fuchun Peng, and Ruiqiang Zhang. Improving search relevance for implicitly temporal queries. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 700–701. Citeseer, 2009.

[19] Irem Arikan, Srikanta Bedathur, and Klaus Berberich. Time will tell: Leveraging temporal expressions in ir. In *In WSDM*. Citeseer, 2009.

[20] Klaus Berberich, Srikanta Bedathur, Omar Alonso, and Gerhard Weikum. A language modeling approach for temporal information needs. In *European Conference on Information Retrieval*, pages 13–25. Springer, 2010.

[21] Nattiya Kanhabua and Kjetil Nørvåg. Determining time of queries for re-ranking search results. In *International Conference on Theory and Practice of Digital Libraries*, pages 261–272. Springer, 2010.

[22] George A Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.

[23] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

[24] Trey Grainger, Timothy Potter, and Yonik Seeley. *Solr in action*. Manning Cherry Hill, 2014.

[25] Michail Vlachos, Christopher Meek, Zografoula Vagena, and Dimitrios Gunopulos. Identifying similarities, periodicities and bursts for online search queries. In *Proceedings of the 2004 ACM SIGMOD international conference on Management of data*, pages 131–142. ACM, 2004.

[26] Gabriel Pui Cheong Fung, Jeffrey Xu Yu, Philip S Yu, and Hongjun Lu. Parameter free bursty events detection in text streams. In *Proceedings of the 31st international conference on Very large data bases*, pages 181–192. VLDB Endowment, 2005.

[27] Tristan Snowsill, Florent Nicart, Marco Stefani, Tijl De Bie, and Nello Cristianini. Finding surprising patterns in textual data streams. In *2010 2nd International Workshop on Cognitive Information Processing*, pages 405–410. IEEE, 2010.

[28] Jon Kleinberg. Bursty and hierarchical structure in streams. *Data Mining and Knowledge Discovery*, 7(4):373–397, 2003.

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

[30] Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. Bidirectional attention flow for machine comprehension. *arXiv preprint arXiv:1611.01603*, 2016.

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