Information Retrieval Journal
https://doi.org/10.1007/s10791-020-09387-9

SPECIAL ISSUE ON ECIR 2020

Improving question answering for event-focused questions in temporal collections of news articles

Jiexin Wang[1] · **Adam Jatowt**[1] · **Michael Färber**[2] · **Masatoshi Yoshikawa**[1]

Received: 16 July 2020 / Accepted: 8 December 2020

Abstract
Temporal collections of news articles (or news archives) contain numerous accurate and time-aligned articles, which offer immense value to our society, helping users to know details of events that occurred at specific time points in the past. Currently, the access to such collections is rather difficult for average users due to their large sizes and complexities. For better use of these valuable resources on our heritage, this study considers the task of machine reading at scale on long-term news article archives. We make use of the observation that questions on news archives are usually related to particular events and show strong temporal aspects. We propose a large scale question answering model designed specifically for long-term news article collections, with an additional module for re-ranking articles by using temporal information from different perspectives. The experimental results show that our model is superior to the existing question answering systems, thanks to dedicated module that allows finding more relevant documents.

Keywords News article collections · Question answering · Information retrieval · Document archives

1 Introduction

In recent years, many old news articles have been digitized and made accessible to wide public. They serve valuable purpose in building our understanding of particular time periods in history and they preserve data about the past including information about key people, places, events, situations and etc. (Korkeamäki and Kumpulainen 2019). Different kinds of professionals (e.g., journalists, historians, sociologists) often need to deal with these collections for a variety of reasons and needs. In addition, it is expected that ordinary users could utilize them to verify information about the past, to understand the evolution or the impact of the events or just to enjoy reading information from the past times. Especially elderly tend to find enjoyment in reminiscing and analyzing the

✉ Jiexin Wang
wang.jiexin.83m@st.kyoto-u.ac.jp

[1] Graduate School of Informatics, Kyoto University, Kyoto, Japan

[2] Karlsruhe Institute of Technology, Karlsruhe, Germany

Published online: 02 January 2021

past (Bryant et al. 2005); however, the history offers valuable lessons for young users as well.

Yet, due to their large sizes, complexities and different context, it is difficult for users to use news archives effectively. Searching, for example, requires knowledge of correct and effective queries which may not be trivial for users with limited knowledge of history. On the other hand, effective browsing is difficult or impossible considering typically large size of data, lack of explicit links and the complex order of documents discussing different news events.

An effective solution would be to use large scale question answering systems (QA systems), which attempt to find out correct answers to questions posed in natural language. We think that questions about the past and also questions that could be issued to news archives tend to be usually related to particular events and exhibit certain temporal aspects. We categorize such questions into two crude types: (1) *explicitly time-scoped questions*: ones containing explicit temporal expressions (e.g., "Which unarmed man was mistaken as a suspect and was shot by police in New York in 1999 ?"), and (2) *implicitly time-scoped questions*: ones without any explicit temporal expression in their content yet being implicitly related to specific time periods (e.g., "Slovenia and Croatia became the first republics to declare independence from which country?"). Table 1 shows some examples of the temporal questions.

This article describes a large-scale question answering system which we call QANA (Question Answering in News Archives). Its objective is answering the two above-mentioned types of event-related questions asked against news article archives. We note that existing QA models are mainly designed for answering questions over synchronic document collections (e.g., Wikipedia). As these systems lack the ability of utilizing temporal information, they process event-related questions and documents of the news archives in the same way as questions and documents in generic, synchronic document corpora. In contrast, QANA does not only utilize the temporal information associated with a question, but also exploits timestamp metadata of documents and the temporal information embedded in document content. Based on the combination of these kinds of temporal information it re-ranks candidate documents so as the probability of finding the correct answer in the top results is increased.

In the experimental evaluation, we tested our approach using the New York Times (NYT) Annotated corpus as a an underlying temporal document collection, based on carefully constructed test set of questions related to past events. These datasets are composed of two types of questions (explicitly and implicitly time-scoped) which have been selected from existing data sets and also from test sites focused on historical content, which makes them particularly difficult to answer. The experimental results show that our proposed approach can improve retrieval effectiveness and surpasses the existing QA systems that are commonly used for large-scale automatic question answering.

This paper is an extension of our prior publication (Wang et al. 2020). In comparison with the previous work that mainly focus on answering the implicitly time-scoped questions, in the current work, we improve the capability of QANA to better utilize the temporal information, and we also introduce an additional method to answer the first type of event-focused questions - the explicitly time-scoped questions, as well as we provide more detailed experimentation. Moreover, we carefully construct a larger test set of questions belonging to the two question types. Thus, this paper provides a more comprehensive and complete view of question answering over the temporal collections of news articles. To sum up, we make the following contributions in this paper:

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

Questions	Time scoped	Answers	Event dates
The USSR flag was lowered and the Russian flag raised over in which building on 25 December 1991?	Explicitly	Kremlin	1991.12
Which country signed an economic accord with Palestinian Liberation Organization in April 1994?	Explicitly	Israel	1994.04
Who famously described his experiences to the media as “a near death experience” during November 2003?	Explicitly	Iain Duncan Smith	2003.11
Democratic U.S. presidential Gary Hart bowed out of the race due to his extra-marital affair with whom?	Implicitly	Donna Rice	1987.05
The dissolution of the Soviet Union occurred after whose resignation?	Implicitly	Mikhail S. Gorbachev	1991.12
Which famous painting by Norwegian Edvard Munch was stolen from the National Gallery in Oslo?	Implicitly	The Scream	2004.08

(1) We describe a novel subtask of QA, which uses long-term temporal news collections as the data source.
(2) We provide effective models for answering questions against temporal document collections by exploiting diverse temporal characteristics of both questions and documents.
(3) We create and provide the test sets for automatically answering questions about the history.
(4) We conduct extensive experimental evaluation of our proposed solution using extended, dedicated test sets and a document collection spanning 20 years.

The remainder of this paper is structured as follows. The next section surveys the related work in question answering and temporal information retrieval and extraction. In Sect. 3, we describe our approach. Section 4 explains experimental settings and shows experimental results. Finally, we conclude the paper in Sect. 5.

2 Related work

2.1 Question answering systems

Large scale question answering systems must be able to effectively retrieve and comprehend relevant documents in order to infer correct answers. This is typically realized by two modules: (1) IR module (or a document retriever module) (2) Machine Reading Comprehension (MRC) module (or a document reader module).

Considerable efforts have been made to develop models for the task of machine reading comprehension, which aims to identify answer within a single passage. Thanks to the advance of deep learning and the availability of high-quality datasets, much progress has been achieved in MRC. Latest MRC models, especially those that integrate BERT (Devlin et al. 2018) or versions derived on the basis of BERT (Lan et al. 2019; Sanh et al. 2019), can even go beyond human performance (as quantified based on EM (Exact Match) and F1 scores) on both SQuAD 1.1 (Rajpurkar et al. 2016) and SQuAD 2.0 (Rajpurkar et al. 2018), which are currently the two most widely-used MRC datasets. However, most proposed MRC models eschew retrieval entirely, as there is only a single document from which to infer answers, which also ignores the difficulty of retrieving question-related documents from large document collections. Recent researches (Wang et al. 2018; Lee et al. 2018; Yang et al. 2019; Ni et al. 2018) have examined the role of IR process and reveal that IR module is a bottleneck that can greatly influence the performance of the whole large scale question answering systems. Hence, there has recently been growing interest in building better IR modules for QA. Chen et al. (2017) introduce DrQA model, one of the most well-known question answering systems, whose IR component is based on a TF-IDF weighting scheme combined with bigrams. Wang et al. (2018) propose R^3 model, whose IR component is trained jointly with MRC component by reinforcement learning based method. Lee et al. (2018) propose a Paragraph Ranker, which uses dot product of representations between the passages and questions to score each passage. Yang et al. (2019) propose BERTserini that integrates IR component using Anserini IR toolkit (Yang et al. 2017) with BERT-based MRC model. Ni et al. (2018) improve IR component by detecting essential terms within a question and reformulating the query.

However, as most QA systems use synchronic document collections (e.g., Wikipedia) as their knowledge source, when answering event-related questions based on temporal

collections of news articles, these systems have no ability to utilize temporal information like timestamp metadata of news articles. Because of the omission of the temporal information, the systems process the questions and the news articles in essentially the same way as in the case of synchronic document collections. Temporal information however constitutes significant feature of news articles and is crucial for event-oriented questions. Although some temporal QA systems that can exploit temporal information have been proposed (Harabagiu and Bejan 2005; Moldovan et al. 2005; Saquete Boró et al. 2004; Saquete et al. 2009; Pasca 2008), they are nevertheless designed for synchronic document collections and thus they do not utilize timestamp information of the temporal collections. The temporal information is utilized mainly for content temporal reasoning (Harabagiu and Bejan 2005; Moldovan et al. 2005), complex question decomposition (Saquete Boró et al. 2004; Saquete et al. 2009) or answering "when" type of questions (Pasca 2008). Besides, these works represent primarily traditional rule-based models and their performance is quite poor.

Furthermore, few resources are available for answering event-related questions over news archives. Jia et al. (2018) release a benchmark for temporal question answering with 1271 QA pairs. Since we use NYT corpus which contains news articles published between 1987 and 2007, only few of the questions whose corresponding events occurred within that time interval could be used.

An important feature of QANA is an additional component which increases the retrieval effectiveness by utilizing diverse temporal information to re-rank retrieved documents. More specifically, not only we exploit the estimated question time scope information, but we also integrate this temporal information with the timestamp information and with the content temporal information extracted from each retrieved document. To the best of our knowledge, besides our previous work (Wang et al. 2020), no other studies, as well as no available datasets which can help in designing a QA system to effectively work on long-term temporal collections of news articles, have been proposed so far. We believe that building QA systems over temporal document collections and historical document archives is necessary to make better use of the valuable data stored in past news articles and to fulfill different information needs (both of professionals working with such collections and average users), especially nowadays when the sizes of the news archives grow quite fast.

2.2 Temporal information retrieval

In the area of temporal information retrieval, several works for temporal ranking of documents have been proposed (Alonso et al. 2007; Campos et al. 2015; Kanhabua et al. 2015). For example, Li and Croft (2003) introduce a time-based language model considering the timestamp metadata of documents to give preference to more recent documents. Similar research studies (Dai and Davison 2010; Dong et al. 2010; Elsas and Dumais 2010) also focus on promoting documents that were recently created or updated. Other works propose approaches for ranking documents by taking the relevant time periods of a temporal query into account, in which temporal expressions may or may not be explicitly given. Arikan et al. (2009) propose a language model based retrieval framework which exploits temporal expressions of document content. Berberich et al. (2010) apply the similar idea but take also uncertainty in temporal expressions into account. These two methods are based on language models that do not exploit timestamp information, and their queries are assumed to contain explicit temporal expressions. For the queries that do not contain explicit temporal expressions, Metzler et al. (2009) introduce an approach to infer the implicit temporal

information by analyzing the frequency information of the query logs over time and then to utilize it for re-ranking the results. This approach can be applied when query logs are available, and typically, for web search scenarios.

Probably the most related work to our research is Kanhabua and Nørvåg (2010). Kanhabua and Nørvåg (2010) introduce three different ways to estimate the implicit time scopes of queries and also to exploit this information for re-ranking the retrieved results. More specifically, their proposal linearly combines the similarity of textual and temporal information for re-ranking. Nonetheless, it does not use any temporal information embedded in document content and the linear combination is done in a static way, unlike in our case. In the experiment, we also compare QANA with the QA system that utilizes the best method proposed in Kanhabua and Nørvåg (2010) to re-rank documents. That method uses the timestamps of top-k retrieved documents as the query time, and integrates them with timestamp of each document to calculate its temporal score, which is then linearly combined with the textual relevance score for re-ranking.

All the temporal ranking approaches mentioned above are applied on short queries rather than on natural language questions, and none of them jointly utilizes query time scope, document timestamp information and content temporal information at the same time. Our research is the first study to borrow concepts from temporal information retrieval area for the QA research, in order to achieve improvement in answering event-focused questions on the temporal collections of news articles.

2.3 Temporal expressions annotation & normalization

Temporal Markup Language (TimeML, Pustejovsky et al. (2005)) is frequently used for annotating temporal information. A temporal expression can be annotated to one of four types: Date, Time, Duration, and Set. Strötgen and Gertz (2012) show that in news domain, temporal expressions of Date type and Duration type account for more than 95% of all temporal expressions. Temporal expression of Date type directly refers to a particular point in time (e.g. “in 1998”) and one of Duration type describes the length of an interval (e.g. “from 1990 to 1998”). In addition, implicit temporal expressions (e.g. “D-Day”) are relatively rare in news documents (Strötgen and Gertz 2012), which means that most temporal expressions can be well annotated.

Temporal taggers, such as HeidelTime(Strötgen and Gertz 2010), SUTime(Chang and Manning 2012), are commonly used tools for recognizing temporal expressions and normalizing them according to the TimeML annotation standard.

3 Proposed method

In the following we describe our proposed system, which is designed for answering two types of event-focused questions over temporal collections of news articles. For the questions of the first type, i.e., the explicitly time-scoped questions, the time scopes of these questions can be obtained directly by extracting and normalizing temporal expressions (e.g., “Which New Mexico Governor announces plans to run for President in January 2007?”). As for the implicitly time-scoped questions that do not contain any temporal expressions, further knowledge is necessary for estimating time periods they refer to (e.g., “Which Welsh singer was knighted by Queen Elizabeth II for service to music?”). We use the underlying document collection for this purpose.

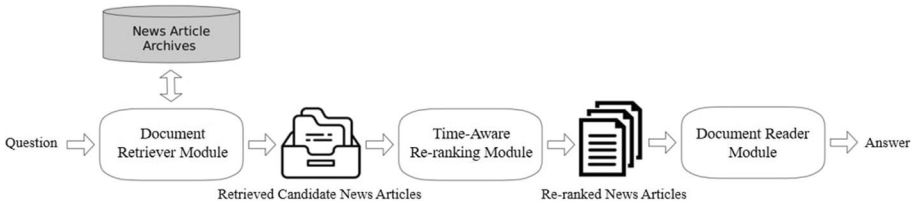

Fig. 1 The architecture of the proposed system

The system architecture is shown in Fig. 1 and is comprised of three modules which are *Document Retriever Module*, *Time-Aware Re-ranking Module* and *Document Reader Module*. In comparison with other question answering systems, we add an additional component called Time-Aware Re-ranking Module which utilizes temporal information (both publication dates as well as content dates) from different perspectives for selecting the best documents. The Time-Aware Re-ranking Module works differently when answering questions of the above-mentioned two types of questions. The remaining two modules work exactly same for both types of the questions.

3.1 Document retriever module

In this module, candidate documents are retrieved from the temporal document collection. Firstly, the module performs keywords extraction by selecting words that are tagged as single-token nouns, compound nouns, adjectives and verbs, based on part-of-speech and dependency information generated using spaCy[1]. Then the module carries out also a stop words removal (the stop words list is taken from spaCy, too) and synonym-based keywords expansion. The synonyms are first derived from WordNet (Miller 1995) and are further filtered by leaving those whose POS types match the original question terms, and whose cosine similarity[2] to question terms is above 0.5. Finally, a query is sent to the Elastic-Search[3] installation which returns the top 100 candidate documents that are ranked by BM25.

3.2 Time-aware re-ranking module

In this module, candidate documents are re-ranked by exploiting temporal information from different aspects. Firstly, the module estimates candidate periods of the time scope $T(Q)$ of a question Q, which are supposed to denote when an event mentioned in the question could have occurred. Then, for each retrieved document d, the module calculates two temporal scores $S^{temp}_{pub}(d)$ and $S^{temp}_{text}(d)$ by contrasting the question time scope against the temporal information derived from the document's timestamp $t_{pub}(d)$ and the temporal information embedded in the document's content $T_{text}(d)$. Finally, the module re-ranks candidate documents by integrating the final temporal score $S^{temp}(d)$ with textual relevance score $S^{rel}(d)$. However, due to the differences in temporal characteristics of the two types of

[1] https://spacy.io/.

[2] We use Glove (Pennington et al. 2014) word vectors trained on the Common Crawl dataset with 300 dimensions.

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

event-focused questions, Time-Aware Re-ranking Module works differently for explicitly time-scoped questions in some details.

3.2.1 Question time scope estimation

The procedures of estimating question time scope $T(Q)$ for the two different types of questions are different, hence we discuss them one by one.

Explicitly Time-scoped Questions As we mentioned before, the time scope of the explicitly time-scoped question can be obtained directly. We use SUTime (Chang and Manning 2012) to recognize and normalize the temporal expression of the question Q.[4] The time scope $T(Q)$ is mapped to the time interval with the "start" and "end" information, which is represented by $(t^s(Q), t^e(Q))$ and denotes the start time and the end time of the mentioned event[5]. For example, the time scope of the question "Which country officially opens its border to Austria in September 1989?" is ('198909','198909'), and the time scope of the question "Radovan Karadzic is associated with genocide between 1992 and 1995 in which country?" is ('199201','199512'). Note that in case when the question contains several temporal expressions, we take only the first one.[6]

Implicitly Time-scoped Questions Further knowledge is required to estimate the time scope information of the implicitly time-scoped questions, which cannot be obtained directly from question content. The distribution of relevant candidate documents over time can be utilized for this purpose as it can reflect useful information regarding temporal characteristics of questions (Amodeo et al. 2011; Peetz et al. 2014; Zahedi et al. 2017). First, the question time scope can be inferred and, second, examining the timeline of a query's result set should allow us to characterize how temporally dependent the topic is. For example, the black dashed lines in Figure 2 depict the distributions of retrieved relevant documents from the New York Times Annotated Corpus per month for four example questions: "Which province had a referendum to ask voters whether it should secede from Canada?", "Which TV network retracted an unsubstantiated report about the use of nerve gas?", "Who was convicted of the crime of Lockerbie Bombing?" and "Which English football team had nine players arrested in Spain for alleged sexual assault?". The blue cross mark indicates the actual occurrence time of the associated events (i.e., the correct time scope of the question).

The distribution of retrieved documents of the first question reflects well its corresponding event occurrence time (October 1995) as most news articles are published near that time. However, in the second question, whose event occurrence time is April 1998, the distribution graph has two relatively high peaks. We found that the reason why the first peak, which does not locate within the question time scope, appears, is due to another nerve gas related event that happened in March 1995-Tokyo Subway Sarin Attack. The third example question in Fig. 2 is even more complicated as it has several peaks which are caused by the evolution of the related event - the analysis of the Lockerbie Bombing, and the repeated discussions in the news. Nevertheless, the distributions of the second and the third

[4] We tested SUTime on 346 temporal questions selected from TREC question classification dataset (Li and Roth 2002), and we added rules to normalize specific temporal expressions that SUTime cannot work well with (e.g., "between 1999 and 2002" should be a Duration type instead of two Date types).

[5] In the experiments, we use monthly granularity.

[6] In our test set, there are actually no such explicitly time-scoped questions. The system can however be extended by considering a set of time periods as the representation of the time scope of a question.

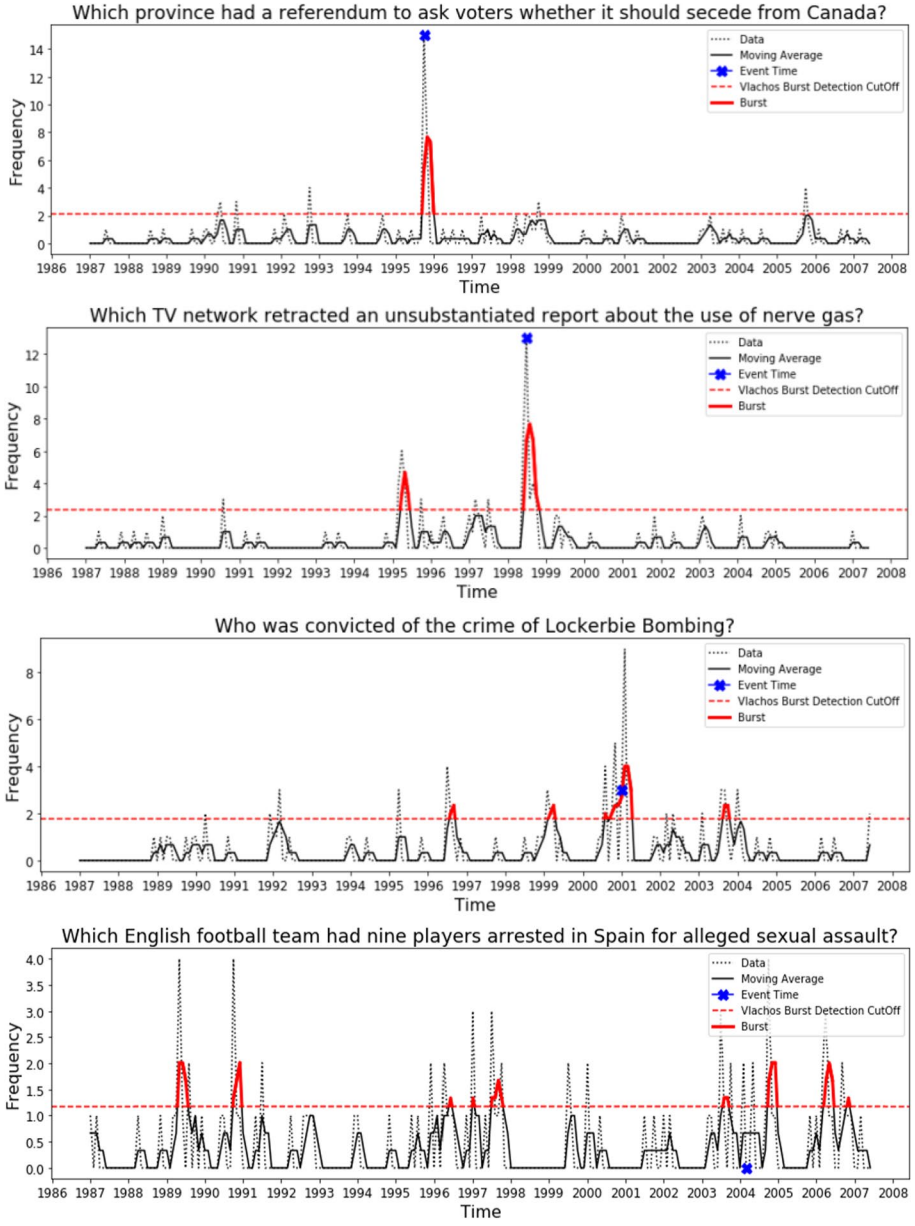

Fig. 2 Burst detection results of four questions using the New York Times Annotated collection. The questions were converted to their corresponding queries as described in Sect. 3.1, and the top 100 ranked results by BM25 were used. Best viewed in color

questions still exhibit useful information, i.e, the highest peak (maxima) of the dashed line is located near the correct time scope. Thus, as we can notice, the distribution of retrieved documents over time could be utilized for estimating the implicit time scope of questions. However, there are some questions whose event occurrence time is not located within or

near the relatively high peaks (e.g., the forth question). We can see that in the plot for the last question, there are nine relatively high peaks, but none of them includes the month in which the event occurred (March 2003). Furthermore, no article was published during the month of the event while most retrieved articles were published even before the event date. After manual check of the retrieved documents, we found that no news articles published before or after the event date refer to the event of the fourth question. Most of these documents report other similar events. By analyzing other questions that exhibit similar characteristics, we found the main reasons for such situations are: (1) the Document Retriever Module does not work well so that few truly relevant documents are retrieved, while the retrieved articles tend to report other similar events, and (2) the event was not reported at all or was mentioned as an event of minor importance so that there are few articles about it, which also means the question cannot be answered or answering it is quite difficult, based on the used document collection. In addition, we also found that this type of questions often has multiple high peaks. For this kind of questions, it is thus better to rely more on the document content relevance.

Based on the relationship between that relevant document distribution over time and the implicit question time scope, we apply the burst detection on the returned documents obtained from the underlying temporal collection. Burst detection method used by Vlachos et al. (2004) is chosen, which provides a simple yet effective way to identify bursts[7]. The assumption is that the correct time period (i.e., the occurrence time of the event referred to in the question) is likely to be covered by the time scopes during which bursts are observed. Naturally, multiple bursts can be detected for a question, due to the occurrence of similar events or the development of different stages of the target event. Thus the estimated time scope of an implicitly time-scoped question needs to be represented by a list of candidate periods. The burst detection method that we apply is based on the computation of the moving average (MA) that annotates bursts as points with values higher than β standard deviations above the mean value of the MA. More specifically, the process of the estimation of the candidate periods of the time scope $T(Q)$ is given in Algorithm 1.

[7] There are many alternative burst detection techniques that could be potentially used (e.g., Fung et al. 2005; Snowsill et al. 2010; Kleinberg 2003).

Algorithm 1: Question 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 and $t_i < t_{i+1}$;
$C \leftarrow \{t(Burst_0)\}$;
foreach $t(Burst_j) \in T(Bursts)$ **do**
    instructions;
    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
        $t_i^s(Q) \leftarrow C.selectFirstElement()$;
        $t_i^e(Q) \leftarrow C.selectLastElement()$;
        $T(Q) \leftarrow T(Q) \cup \{(t_i^s(Q), t_i^e(Q))\}$;
    end
end

Timestamp sequence $T_{pub}(Q)$ is obtained by collecting timestamp information of retrieved candidate documents. The question time scope $T(Q)$ is represented by a list of $(t_i^s(Q), t_i^e(Q))$ pairs, each of which denotes the border time points of the ith estimated time period representing the ith burst. w and β are the two parameters in the above algorithm, which affect the results of burst detection. For simplicity, when calculating the moving Average MA_w of timestamp sequence $T_{pub}(Q)$, we use in the experiments the window size w equal to 3, representing 3 months. β affects the cutoff value. We use β equal to 2.0 which is a suggested value by Vlachos et al. (2004). In Fig. 2, the red solid lines depict the burst detection results. The estimated time scope of the first question is [('1995–10', '1996–01')], while the time scope of the second question is [('1995–04', '1995–06'), ('1998–07', '1998–10')] and the result of the third question is [('1996–08', '1996–09'), ('1999–03', '1999–04'), ('2000–08', '2001–04'), ('2003–09', '2003–10')].

Furthermore, a weight corresponding to each candidate period is calculated when estimating $T(Q)$, indicating the importance of each period. The weight is computed by dividing the number of retrieved documents published within the period over the total number of retrieved documents published in all the derived candidate periods of $T(Q)$. For example, for the second question, the weight assigned to the candidate period ('1998–07', '1998–10') is $\frac{23}{33}$, as the number of retrieved documents published within this period is 23, while the total number of retrieved documents within all total candidate periods is 33. Finally, $W(T(Q))$ is used to signify the weight list: $W(T(Q)) = [(w(t_1^s(Q), t_1^e(Q))), ...(w(t_m^s(Q), t_m^e(Q)))]$, where m is the number of periods in $T(Q)$.

3.2.2 Timestamp-based temporal score calculation

After obtaining the question time scope $T(Q)$, the module calculates the timestamp-based temporal score $S_{pub}^{temp}(d)$ for each candidate document d. We compute this temporal score based on the intuition that news articles published within or soon after the actual time

period associated to the question have high probability of containing detailed information of the event. Below, we introduce the calculation of this score for the two types of the event-focused questions.

Explicitly Time-scoped Questions For explicitly time-scoped questions, the time scope $T(Q)$ is represented by $(t^s(Q), t^e(Q))$, which is a pair of start time point and end time point. The timestamp-based temporal score $S^{temp}_{pub}(d)$ is calculated as follows:

$$\begin{aligned} S^{temp}_{pub}(d) &= P(T(Q)|t_{pub}(d)) \\ &= \lambda^{Dis(T(Q),t_{pub}(d))} = \lambda^{Dis((t^s(Q),t^e(Q)),t_{pub}(d))} \quad (0 < \lambda < 1) \end{aligned} \tag{1}$$

$S^{temp}_{pub}(d)$ is estimated as $P(T(Q)|t_{pub}(d))$, which means the probability of generating time scope $T(Q)$ (following Kanhabua and Nørvåg 2010), and is defined as an exponential decay function of the distance between the document's publication date and question time scope. The general function of calculating the distance between publication date and the pair of two border time points is defined by:

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

To calculate the distance $Dis((t^s(Q), t^e(Q)), t_{pub}(d))$ for explicitly time-scoped questions, (t^s, t^e) in Eq. 2 is replaced by $(t^s(Q), t^e(Q))$. $TimeSpan(D)$ denotes the total length of time frame of the temporal document collection D. In the experiments, we use NYT corpus with monthly granularity, so $TimeSpan(D)$ equals to 246 units, corresponding to the number of all months in the corpus. The decay rate λ is set to 0.0625, such that when the distance equals 0.5, the timestamp-based temporal score is 0.25. When document d is published before $t^s(Q)$ of the time scope, the distance $Dis((t^s(Q), t^e(Q)), t_{pub}(d))$ equals to positive infinity, making $P((t^s(Q), t^e(Q))|t_{pub}(d))$ equal to 0.0, as such a document usually cannot provide much information on the events that occurred after its publication[8]. Otherwise, the timestamp-based temporal score is larger when the timestamp is closer to the question time period $(t^s(Q), t^e(Q))$.

Implicitly Time-scoped Questions Unlike the explicitly time-scoped question type, the estimated time scope $T(Q)$ of the implicitly time-scoped questions is a list of the candidate periods, along with the corresponding weights $W(T(Q))$ indicating their importance. The calculation of $S^{temp}_{pub}(d)$ is then different, and is as follows:

$$\begin{aligned} S^{temp}_{pub}(d) &= P(T(Q)|t_{pub}(d)) \\ &= P(\{(t^s_1(Q), t^e_1(Q)), ...(t^s_m(Q), t^e_m(Q))\}|t_{pub}(d)) \\ &= \frac{1}{m}\sum_{i=1}^{m} P((t^s_i(Q), t^e_i(Q))|t_{pub}(d)) \end{aligned} \tag{3}$$

$S^{temp}_{pub}(d)$ is also estimated as $P(T(Q)|t_{pub}(d))$ same as in the case of the explicitly time-scoped questions, however, the score is equal now to the average probability of generating

[8] We neglect through this setting the possibility of providing "future" information on the event as seen from the document's publication date. We have decided not to use such future-pointing information in our research because we think that predictions are basically only useful for scheduled events, and still they carry risk of providing incorrect information. They could however be investigated in the future.

m candidate periods of time scope $T(Q)$. Then, by considering the importance weight $w(t_i^s(Q), t_i^e(Q))$, the probability of generating the period $(t_i^s(Q), t_i^e(Q))$ given the document timestamp $t_{pub}(d)$ is:

$$P((t_i^s(Q), t_i^e(Q))|t_{pub}(d)) = w(t_i^s(Q), t_i^e(Q)) \cdot \lambda^{Dis((t_i^s(Q),t_i^e(Q)),t_{pub}(d))} \quad (4)$$

$Dis((t_i^s(Q), t_i^e(Q)), t_{pub}(d))$ is the distance between the publication date $t_{pub}(d)$ and a candidate period $(t_i^s(Q), t_i^e(Q))$, and is also calculated by Eq. 2. Similarly, $P((t_i^s(Q), t_i^e(Q))|t_{pub}(d))$ equals to 0.0 when document d is published before $t_i^s(Q)$ and is larger when the timestamp is closer to the time period $(t_i^s(Q), t_i^e(Q))$, and when the importance weight $w(t_i^s(Q), t_i^e(Q))$ of this period is large.

3.2.3 Content-based temporal score calculation

For each candidate document d, the module computes also content-based temporal score, $S_{text}^{temp}(d)$. $S_{text}^{temp}(d)$ is the temporal score calculated based on the relation between temporal information embedded in the content of document d and the estimated question time scope $T(Q)$. We compute this score as some news articles, which may not be published near or during the event time, may still retrospectively relate to the event, giving salient or additional information. Such news articles may be even published long time after the target

Question: How many people were killed in Concorde crash in 2000?
Answer: **113**
Event Occurred Date: 2000/07/25

Relavent news articles from The New York Times Annotated Corpus
Relevant news article 1:
Title: Brian Trubshaw, 77, Dies; Tested Concorde
Published Time: 2001/03/28
Content:
Brian Trubshaw, a pilot who tested the British-French Concorde supersonic airliner and became its staunchest champion, died on March 24 at his home near Tetbury, …
…
British Airways and Air France, the only airlines to buy the Concorde, are still struggling to return their fleets to service after grounding them **last year** for safety improvements following an Air France Concorde crash near Paris that killed **113** people.
…
Relevant news article 2:
Title: French Report on Concorde Crash Blames Debris and Structural Flaw
Published Time: 2004/12/15
Content:
A metal strip that fell off a Continental Airlines plane was a major element in the crash of an Air France Concorde jet near Paris **in July 2000** that killed **113** people, …
…
The Concorde crashed into a hotel soon after it took off from Charles de Gaulle airport **on July 25, 2000**, when one of its tires exploded after hitting the titanium strip that had fallen from a Continental DC-10 that had taken off minutes before.
…
Relevant news article 3:
Title: World Briefing | Europe: France: Ex-Concorde Head In Crash Inquiry
Published Time: 2005/09/28
Content:
Henri Perrier, the former director of the French Concorde program, was questioned for more than 11 hours by a judge in the crash of an Air France Concorde just after takeoff from Paris **in 2000** that killed **113** people, and he was placed under formal investigation -- a step short of formal charges. …

Fig. 3 The examples of news articles that retrospectively refer to the target event mentioned in the question. Best viewed in color

event; for example, they may be focusing on other similar events or on the subsequent development or effect of the target event. For example, in Fig. 3, the second and the third top-relevant news articles retrieved from the NYT collection, provide important and extra details on the target event and contain the correct answers of the question even though they were published four and five years after the event, respectively. Thus, as we can see temporal information embedded in document content can be useful.

Furthermore, as we mentioned in Sect. 2.3, implicit temporal expressions are relatively rare in news articles and most of the temporal expressions can be successfully normalized (Strötgen and Gertz 2012). To calculate the content-based temporal score, temporal expressions embedded in the content of retrieved documents need to be first recognized and normalized, which is the shared step for both the two types of event-focused questions. Just like the normalization of the temporal expression of the explicitly time-scoped questions, temporal tagger SUTime (Chang and Manning 2012) is used and each detected temporal expression is also mapped to the time interval with the "start" and "end" information. For example, "from 1995 to 2000" is normalized to [('1995–01', '2000–12')]. Moreover, temporal signals[9] (words that help to identify temporal relations, e.g. "prior to","after","following") are used to normalize special temporal expressions, of which one time point of the interval can not be determined. For example, "after March 2000" is normalized as [('2000–03', 'null')], since the "end" temporal information is not clear. Finally, we get 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 recognized in d. For each interval τ_i, we denote its "start" information as τ_i^s, and its "end" information as τ_i^e. Then, two lists $T_{text}^s(d)$, $T_{text}^e(d)$ are constructed by collecting all τ_i^s and all τ_i^e, respectively.

Next, we describe the calculation of the content-based temporal score, which varies between the two question types.

Explicitly Time-scoped Questions As we mentioned before, the time scope $T(Q)$ of explicitly time-scoped questions is a pair of a start time point and end time point, $(t^s(Q), t^e(Q))$. We integrate the question time scope with the content temporal information by constructing two probability density functions using kernel density estimation (KDE), corresponding to two lists $T_{text}^s(d)$, $T_{text}^e(d)$. KDE is a technique related to histograms, and is a statistically efficient non-parametric method commonly used for probability density estimation. After obtaining two probability density functions, the module calculates two scores, $S_{text}^{temp\_s}(d)$ and $S_{text}^{temp\_e}(d)$, which are then combined to compute the final content-based temporal score $S_{text}^{temp}(d)$ of the document d. Similar to the idea in computing the timestamp-based temporal score, $S_{text}^{temp\_s}(d)$ and $S_{text}^{temp\_e}(d)$ are estimated as $P(t^s(Q)|T_{text}^s(d))$ and $P(t^e(Q)|T_{text}^e(d))$, which means the probabilities of generating $t^s(Q)$ and $t^e(Q)$ based on $T_{text}^s(d)$ and $T_{text}^e(d)$, respectively. Then, the probability of a "start" information t^s of the time period using the kernel density function of $T_{text}^s(d)$ is:

$$P(t^s|T_{text}^s(d)) = \hat{f}(t^s;h) = \frac{1}{m(d)} \sum_{i=1}^{m(d)} K_h\left(t^s - \tau_i^s\right) \tag{5}$$

where h is a bandwidth (equals to 0.75) and K is a Guassian Kernel defined by:

[9] The temporal signals' list is taken from Jia et al. (2018).

$$K_h(x) = \frac{1}{\sqrt{2\pi} \cdot h} exp\left(-\frac{x^2}{2 \cdot h}\right) \tag{6}$$

Then, $S^{temp\_s}_{text}(d)$, which is estimated as $P(t^s(Q)|T^s_{text}(d))$, can be calculated by replacing t^s with $t^s(Q)$ in Eq. 5. $S^{temp\_e}_{text}(d)$ can also be calculated in a similar way by replacing t^s with $t^e(Q)$, and $T^s_{text}(d)$ with $T^e_{text}(d)$. Finally, $S^{temp}_{text}(d)$ is defined by:

$$S^{temp}_{text}(d) = P(T(Q)|T_{text}(d)) = \frac{1}{2} \cdot (S^{temp\_s}_{text}(d) + S^{temp\_e}_{text}(d)) \tag{7}$$

where $S^{temp\_s}_{text}(d) = P(t^s(Q)|T^s_{text}(d))$, and $S^{temp\_e}_{text}(d) = P(t^e(Q)|T^e_{text}(d))$.

Implicitly Time-scoped Questions For implicitly time-scoped questions, we also construct two probability density functions by using KDE based on two lists $T^s_{text}(d)$, $T^e_{text}(d)$ for each candidate document d. In addition, the probabilities of generating $t^s_i(Q)$ and $t^e_i(Q)$ of the ith candidate time period of $T(Q)$ based on the two lists, represented by $P(t^s_i(Q)|T^s_{text}(d))$ and $P(t^e_i(Q)|T^e_{text}(d))$, are also calculated in the same way as in Eq. 5. The probability of the ith candidate time period, denoted by $P((t^s_i(Q), t^e_i(Q))|T_{text}(d))$, which also equals to the score of the time period, is computed similarly as in Eq. 7 but considering its weight which indicates the importance:

$$P((t^s_i(Q), t^e_i(Q))|T_{text}(d)) = \frac{1}{2} \cdot (P(t^s_i(Q)|T^s_{text}(d)) + P(t^e_i(Q)|T^e_{text}(d))) \cdot w(t^s_i(Q), t^e_i(Q)) \tag{8}$$

Finally, the score $S^{temp}_{text}(d)$, which is estimated as the overall probability $P(T(Q)|T_{text}(d))$, is computed as follows:

$$S^{temp}_{text}(d) = P(T(Q)|T_{text}(d)) = \frac{1}{m} \sum_{i=1}^{m} P((t^s_i(Q), t^e_i(Q))|T_{text}(d)) \tag{9}$$

3.2.4 Final temporal score calculation & document ranking

The last step works only a bit differently for the two different types of event-focused questions, so we discuss them together.

The final temporal score of a document d is firstly calculated by averaging the two calculated temporal scores:

$$S^{temp}(d) = \frac{1}{2} \cdot (S^{temp'}_{pub}(d) + S^{temp'}_{text}(d)) \tag{10}$$

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

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

$$S^{rel}(d) = \frac{BM25(d)}{MAX\_BM25} \tag{11}$$

Finally, we re-rank documents by a linear combination of their relevance scores and temporal scores:

$$S(d) = (1 - \alpha(Q)) \cdot S^{rel}(d) + \alpha(Q) \cdot S^{temp}(d) \tag{12}$$

$\alpha(Q)$ is a crucial parameter, which determines the proportion between using the document temporal score and its document relevance score. For example, when $\alpha(Q)$ equals to 0.0, the temporal information is ignored. As different questions have different shapes of the temporal distributions of their relevant documents, we propose to dynamically determine $\alpha(Q)$ per each question. The idea is that when the temporal distribution of relevant documents for a question is characterized by many bursts, meaning that either the event of the question was frequently mentioned at different times, or many similar or related events occurred over time (e.g., see the fourth question in Fig. 2), then time should play a lesser role. We then want to decrease $\alpha(Q)$ value to pay more attention to document relevance because the answers based on temporal analysis can be noisy or misleading in this case. In contrast, when only few bursts are found, which could be interpreted in a way that the question has an obvious temporal character (e.g., see the first two questions in Fig. 2) and there is one or a small number of underlying events, time should be considered more. Note that in order to calculate $\alpha(Q)$ the burst detection needs to be also performed for the explicitly time-scoped questions. $\alpha(Q)$ is computed as follows:

$$\alpha(Q) = \begin{cases} 0.0 & when\ burst\_num = 0 \\ ce^{-(1-\frac{1}{burst\_num})} & elsewhere \end{cases} \tag{13}$$

$\alpha(Q)$ assumes small values when the number of bursts is high, while it has the highest value 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 re-ranking is based on document relevance. c is a parameter that influences $\alpha(Q)$. The smaller the value of c is, the smaller the $\alpha(Q)$ will be. When the question belongs to the explicitly time-scoped question type, we set c to a high value of 0.5, since the question's time scope can be correctly obtained. On the other hand, c is set to a small value (i.e., 0.25) When the question belongs to the implicitly time-scoped type of questions, whose time scope may be composed of multiple time periods, or might sometimes be incorrectly determined.

3.3 Document reader module

The last module infers answer from the candidate documents delivered from the previous module. We utilize here a commonly used MRC model called BiDAF (Seo et al. 2016) which achieves Exact Match score of 68.0 and F1 score of 77.5 on the SQuAD 1.1 dev set. BiDAF model is applied to extract answers of the top N re-ranked documents and to select the most common answer as the final answer. Note that BiDAF could be replaced by other MRC models, for example, ones that are combined with BERT (Devlin et al. 2018) or with versions derived on the basis of BERT (Lan et al. 2019; Sanh et al. 2019). We use here BiDAF for easy comparison with DrQA, whose reader component performance is similar although a little better than the one of BiDAF.

4 Experiments

In this section, we first introduce the construction of our test set, and then we discuss the experimental results comparing with other models.

Table 2 Resources used for constructing the test set

Resources	Number of explicitly time-scoped questions	Number of explicitly time-scoped questions
History quizzes from funtrivia[a]	235	204
History quizzes from quizwise[b]	67	75
Wikipedia pages	140	143
Questions from datasets Rajpurkar et al. (2016), Jia et al. (2018)	58	78

[a] http://www.funtrivia.com/quizzes/history/index.html

[b] https://www.quizwise.com/history-quiz

4.1 Experimental setting

4.1.1 Document archive and test set

As previously mentioned, NYT corpus (Sandhaus 2008) is used as the underlying temporal news collection, and is indexed by Elasticsearch. Over 1.8 million articles published between January 1, 1987 and June 19, 2007 with their metadata are contained in the corpus. NYT has been often used for Temporal Information Retrieval researches (Campos et al. 2015; Kanhabua et al. 2015). Note that NYT is especially challenging for our method as it is a single-source dataset (i.e., news articles were published by a single particular newspaper company), hence the redundancy on which the burst detection, as well as to some degree the content-based temporal score computation, are based, is rather small in such data. We expect that using a temporal document collection composed of articles originating from multiple news sources would result in a better performance of the proposed model. At the same time, the choice of a single source can be regarded as more realistic, since for many past time periods (especially distant ones), and also for less common languages, gathering documents from many sources is rather difficult.

To the best of our knowledge, besides our previous work (Wang et al. 2020), there was no previous proposal for QA system nor there were any test sets designed specifically for the temporal document collections of news articles. Hence, we have to manually construct the test set for the two types of questions and make sure that the occurrence time of the events mentioned in the questions fall into the time frame of the NYT corpus. We create a test set containing 1000 questions (500 of explicitly time-scoped questions, 500 of implicitly time-scoped questions), paired with their answers.[10] Unlike in the case of the test set that we used in our previous work (Wang et al. 2020), we have not checked if at least one retrieved document in NYT can infer the correct answer of the question. This choice helps to learn the ability of the tested systems to answer event-related questions in real scenarios. Furthermore, we did not want to bind the test set to any particular dataset. Hence, the test set can be used for answering questions based on other underlying temporal news article collections or even it could be utilized for testing approaches that just work with

[10] The test set is available at https://www.dropbox.com/sh/fdepuisdce268za/AACtiPDaO_RwLCwhIwaET4Iba?dl=0.

synchronic document collections such as Wikipedia, as a domain-specific (i.e., history-focused) question-answer pairs' set.

The questions in the test set were carefully selected from several history quiz websites or from other existing datasets. The distribution of resources used for creating the test set is shown in Table 2. Table 1 gives a few example questions.

4.1.2 Tested approaches

For evaluating our proposal we have compared it with several methods that are representative of different approaches (e.g., information retrieval, question answering). The following models are tested in our experiments using the NYT document collection:

(1) DrQA-NYT (Chen et al. 2017): DrQA, a robust system for automatic question answering which is composed of a Document Retriever module and a Document Reader module.
(2) QA-NLM-U (Kanhabua and Nørvåg 2010): QA system for answering implicitly time-scoped questions that uses the best re-ranking method in Kanhabua and Nørvåg (2010) (as described in Sec. 2.2), while the Document Retriever Module and Document Reader Module are the same as the modules of QANA.
(3) QA-No-Re-ranking (Seo et al. 2016): QANA system without re-ranking module, same as other QA systems that consist of only two modules. Same as for QA-NLM-U, the Document Retriever Module and Document Reader Module are also the same as the modules of QANA.
(4) QANA-TempPub: QANA version that uses only temporal information related to timestamps for re-ranking in Time-Aware Re-ranking Module (i.e., Eqs. 1 and 3).
(5) QANA-TempCont: QANA version that only uses temporal information embedded in document content for Time-Aware Re-ranking Module (i.e., Eqs. 7 and 9).
(6) QANA: QANA with complete Time-Aware Re-ranking Module.

4.2 Experimental results

4.2.1 Results of explicitly time-scoped questions

We use exact match (EM) and F1 score as our evaluation metrics. Table 3 shows the performance of the tested models in answering explicitly time-scoped questions. We can see that QANA with complete Time-Aware Re-ranking Module surpasses other models for all different N (the number of re-ranked documents used in the Document Reader Module). The performance improvement is due to the utilization of temporal information, and because more relevant candidate documents are assigned higher scores. The temporal information, which constitutes an important feature of events, is obtained from the question itself, document timestamp metadata and document content.

We next compare QANA with other models using the top 1 and top 5 results. We can see that the performance of QANA far exceeds the one of DrQA-NYT, which is one of the most notable QA systems and is often used as a baseline in QA researches. The improvement ranges from 40.90 to 35.55% on EM score, and from 43.86 to 35.22% on F1 score. Additionally, we can also notice a clear improvement when comparing with QA-No-Re-ranking, which does not contain the re-reranking module to utilize the temporal

Table 3 Performance of different models on explicitly time-scoped questions

Model	Top 1		Top 5		Top 10		Top 15	
	EM	F1	EM	F1	EM	F1	EM	F1
DrQA-NYT Chen et al. (2017)	13.20	17.60	18.00	23.73	21.20	26.51	21.00	26.85
QA-No-Re-ranking Seo et al. (2016)	13.60	19.86	18.20	24.97	23.80	31.92	26.20	34.45
QANA-TempPub	17.20	23.31	23.60	30.81	27.20	36.60	30.20	38.91
QANA-TempCont	16.80	23.30	24.00	31.68	27.60	36.19	29.60	38.51
QANA	**18.60**	**25.32**	**24.40**	**32.09**	**30.02**	**39.01**	**31.20**	**40.50**

Best results are shown in bold

Table 4 Performance of DrQA using different knowledge source vs. QANA in answering explicitly time-scoped questions

Model	Top 1		Top 5		Top 10		Top 15	
	EM	F1	EM	F1	EM	F1	EM	F1
DrQA-Wiki Chen et al. (2017)	**18.80**	22.92	22.60	27.49	24.60	29.35	25.40	30.25
DrQA-NYT Chen et al. (2017)	13.20	17.60	18.00	23.73	21.20	26.51	21.00	26.85
QANA	18.60	**25.32**	**24.40**	**32.09**	**30.02**	**39.01**	**31.20**	**40.50**

Best results are shown in bold

information, and in this case the improvement ranges from 36.76 to 34.06%, and from 27.49 to 28.51% on EM and F1 metrics, respectively. In addition, the performance of QANA-TempPub and QANA-TempCont is similar in answering explicitly time-scoped questions for different top N, and thus using only timestamp information or only content temporal information can still bring comparatively good results. However, QANA with the complete components utilizing the temporal information from different angles to re-rank the candidate documents, achieves the best performance.

We also test the DrQA when using Wikipedia articles as the underlying knowledge source, and the results are shown in Table 4. We can clearly observe that when answering explicitly time-scoped questions, DrQA-Wiki is always better than DrQA-NYT, especially

Table 5 Performance of the models on explicitly time-scoped questions having few bursts vs. ones having many bursts

		Top 1		Top 5		Top 10		Top 15	
		EM	F1	EM	F1	EM	F1	EM	F1
Questions with few bursts	QA-no-re-ranking Seo et al. (2016)	13.91	22.04	20.61	27.79	25.77	34.23	28.35	36.72
	QANA	20.61	27.96	25.77	34.88	34.53	43.42	36.59	45.28
Questions with many bursts	QA-no-re-ranking Seo et al. (2016)	13.39	18.48	16.66	23.18	22.54	30.45	24.83	33.02
	QANA	17.32	23.64	23.52	30.32	27.45	36.21	27.77	37.46

using the top 1 document. The improvement is 42.42% on EM score and 30.22% on F1 score. In addition, DrQA-Wiki performs a bit better than QANA on EM score when considering the top 1 documents (the improvement is about 1.07%), but QANA performs much better in other cases. For example, when considering the top 10 and the top 15, the improvement ranges from 22.03 to 22.83%, and from 32.91 to 33.88% on EM and F1 metrics, respectively. This means that answering history-related questions on primary sources (at least using our test set) tends to be better than on Wikipedia which represents a type of a secondary source. It also suggests that the combination of both the source types could be promising.

Furthermore, we also analyze the performance of the QANA and QA-No-Re-ranking based on the number of detected bursts. We regard the questions with bursts number smaller than 4 as questions with few bursts. The results are shown in Table 5. We can clearly observe that both QANA and QA-No-Re-ranking always perform better when answering questions with few bursts. As mentioned before, when the temporal distribution of relevant documents returned for a question exhibits many bursts, either the target event is frequently mentioned at different time points, or the event is a long lasting event, or multiple other similar events are found. Nonetheless, our system still outperforms QA-No-Re-ranking in both the cases, as it takes both the importance and the number of bursts into account.

Finally, we test the effect of $\alpha(Q)$, which plays an important role in calculating the final re-ranking score, by determining the proportion between document temporal score and query relevance score. In Fig. 4, the performance of QANA using dynamic alpha is depicted by the straight dashed line. For all different top N values, the performance of QANA using dynamic alpha is always better than the one of the system which uses a static alpha (depicted by the solid lines in Fig. 4). Therefore, the dynamic alpha, which is dependent on the analysis of the temporal distribution of retrieved documents, is able to flexibly capture the variations in the importance of temporal information and relevance information related to queries, resulting thus in better overall performance.

4.2.2 Results of implicitly time-scoped questions

Table 6 shows the performance of the tested models in answering implicitly time-scoped questions. Firstly, we can observe that QANA with complete Time-Aware Re-ranking component also outperforms other models for all different N, which is the same as answering explicitly time-scoped type of questions. Although the improvement is not as great as in answering the explicitly time-scoped question type, we can still see a large improvement when using the top 5, top 10 and top 15 results. When comparing with DrQA-NYT using the top 5 and top 10 results, the improvement ranges from 11.02 to 30.53% on EM score, and from 14.65 to 28.94% on F1 score. In comparison with QA-No-Re-ranking, the improvement ranges from 12.80 to 12.50%, and from 10.00 to 14.07% on EM and F1 metrics, respectively. When comparing with the system without Time-Aware Re-ranking Module, the improvement is in the range of 14.63 to 17.93% on EM score, and from 12.31 to 14.25% on F1 score. Furthermore, we also can see comparatively good results of QANA version that either utilizes only timestamp information or only content temporal information; yet still the complete model that exploits both two temporal information types obtains the best performance.

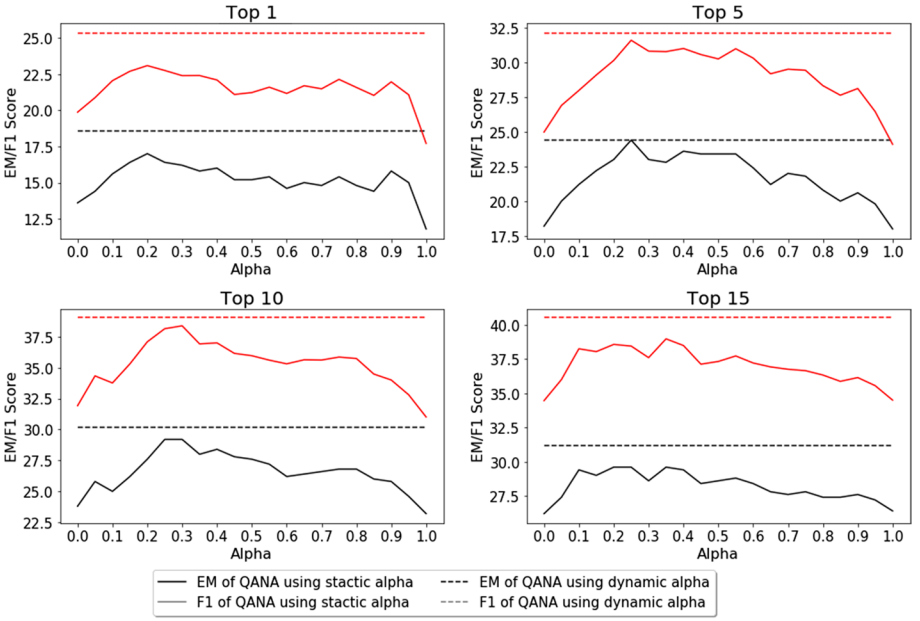

Fig. 4 QANA Performance with different static alpha values vs. one with dynamic alpha for different top-N results over explicitly time-scoped questions

Table 6 Performance of different models answering implicitly time-scoped questions

Model	Top 1		Top 5		Top 10		Top 15	
	EM	F1	EM	F1	EM	F1	EM	F1
DrQA-NYT (Chen et al. 2017)	19.40	25.65	25.40	32.14	26.20	34.13	27.00	35.86
QA-NLM-U (Kanhabua and Nørvåg 2010)	20.40	28.34	25.00	33.50	30.40	38.58	31.40	39.95
QA-No-Re-ranking (Seo et al. 2016)	19.00	27.19	24.60	32.81	29.00	38.52	31.00	40.17
QANA-TempPub	20.40	28.27	26.20	34.27	32.80	42.88	35.60	45.06
QANA-TempCont	20.00	28.03	26.00	33.76	32.20	42.17	33.80	43.71
QANA	**21.00**	**28.90**	**28.20**	**36.85**	**34.20**	**44.01**	**36.20**	**45.63**

Best results are shown in bold

Table 7 Performance of DrQA using different knowledge source vs. QANA in answering implicitly time-scoped questions

Model	Top 1		Top 5		Top 10		Top 15	
	EM	F1	EM	F1	EM	F1	EM	F1
DrQA-Wiki (Chen et al. 2017)	**21.20**	25.76	22.00	26.30	23.00	26.97	24.40	28.70
DrQA-NYT (Chen et al. 2017)	19.40	25.65	25.40	32.14	26.20	34.13	27.00	35.86
QANA	21.00	**28.90**	**28.20**	**36.85**	**34.20**	**44.01**	**36.20**	**45.63**

Best results are shown in bold

We next examine the performance of DrQA when using Wikipedia articles as its knowledge source, whose result is shown in Table 7. DrQA-Wiki also performs the best on EM score using the top 1 document, but when considering the top 5, top 10 and top 15 documents, it performs worse than DrQA-NYT. We guess that this might be due to the fact that more articles about the events mentioned in the implicitly time-scoped questions can be found in NYT corpus. In addition, QANA outperforms DrQA-Wiki greatly using except using the top 1 result. For example, the improvement is 28.18% on EM score, and is 40.11% on F1 score using top 5 results.

Next, we evaluate the performance of QANA based on the number of bursts. As shown in Table 8, we can get the same observation as in the explicitly time-scoped questions: questions with few bursts (less than 4) are likely to be answered more easily. When

Table 8 Performance of the models answering implicitly time-scoped questions having few bursts vs. having many bursts

	Top 1		Top 5		Top 10		Top 15	
	EM	F1	EM	F1	EM	F1	EM	F1
Questions with few bursts								
QA-No-Re-ranking (Seo et al. 2016)	20.94	29.81	28.63	37.41	35.89	46.30	39.74	49.49
QANA	22.64	31.54	30.76	40.63	38.03	49.08	41.02	52.17
Questions with many bursts								
QA-No-Re-ranking (Seo et al. 2016)	17.29	24.88	21.05	28.77	22.93	30.90	23.30	31.21
QANA	19.54	26.59	25.93	33.54	30.82	39.56	31.95	39.87

Fig. 5 QANA Performance with different static alpha values vs. one with dynamic alpha for different top-N results over implicitly time-scoped questions

Table 9 Results of the experiment on treating explicitly time-scoped questions as implicitly time-scoped type

Model	Top 1		Top 5		Top 10		Top 15	
	EM	F1	EM	F1	EM	F1	EM	F1
QA-NLM-U (Kanhabua and Nørvåg 2010)	12.80	18.67	16.40	23.02	19.40	27.46	22.20	30.33
Imp-QANA	14.80	21.65	20.60	27.48	25.40	33.77	28.40	36.48
QA-No-Re-ranking (Seo et al. 2016)	13.60	19.86	18.20	24.97	23.80	31.92	26.20	34.45
QANA	**18.60**	**25.32**	**24.40**	**32.09**	**30.02**	**39.01**	**31.20**	**40.50**

Best results are shown in bold

comparing the results of questions with many bursts using the top 10 and top 15 results, QANA surpasses QA-No-Re-ranking with the improvement ranging from 34.40 to 37.12% on EM score, and from 28.02 to 27.74% on F1 score.

In the end, we examine the effect of $\alpha(Q)$. As shown in Fig. 5, we can get the same conclusion that using dynamic alpha can help to better determine the proportion between document temporal score and query relevance score.

4.2.3 Additional experiment by answering explicitly time-scoped questions as implicitly time-scoped questions

We also conduct an additional experiment by treating each explicitly time-scoped question as an implicitly time-scoped one. We test two models in this setting: (1) QA-NLM-U, which is designed for answering questions of implicitly time-scoped type, and (2) QANA version which always requires to estimate the time scope of any question (both the explicitly or implicitly time-scoped one) by utilizing the distribution of retrieved documents, denoted as Imp-QANA. The result is shown in Table 9, and we compare these two models with QA-No-Re-ranking and QANA in answering the questions of explicitly time-scoped type. As we can see, QA-NLM-U performs quite poor and the performance is even worse than the model without re-ranking. Imp-QANA can surpass QA-No-Re-ranking for all different top N, but it still shows a gap compared to QANA, which probably is caused by incorrectly estimating the time scope. The result shows the importance of correctly estimating the correct question's time scope, which can greatly improve the performance of the re-ranking.

5 Conclusions

In this work we investigate a novel research task focused on answering event-related questions that are issued against primary collections of documents, in particular, news article archives. There are many potential benefits from such a task such as automatically assisting professionals (e.g., historians, journalists) in their works or fact checking by investigating original accounts of events as they were provided in the past. We design an effective

solution for answering questions on long-term news article collections. Unlike questions issued against 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 question's 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 diverse types of temporal information within IR component for QA systems.

Finally, this work leads to few useful observations. First, to answer event-related questions on long-span news archives one should (a) *infer the time scope embedded within a question*. This step may involve analyzing temporal distribution of relevant documents in case there are no temporal signals coming directly from the question. Next, (b) *re-ranking documents based on their closeness and order relation to this time scope* helps to locate correct answer. Moreover, (c) *using temporal expressions embedded in documents* further supports the selection of best candidate documents. Lastly, (d) *joining the two temporal scores (i.e., (b) and (c)) and applying dynamic way to determine the importance between query relevance and temporal relevance are helpful to answer questions*.

In the future, we plan to extend the test set and to conduct more detailed evaluation on the longer temporal collections of the news articles. We will also enhance the system by improving the question's time scope estimation method for the implicitly time-scoped questions. Here several approaches can be applied, for example, ones based on investigating the relevant document distribution in more detail to consider the distance between detected bursts, combining time series and text information in a more effective way, or by utilizing external knowledge bases.

References

Alonso, O., Gertz, M., & Baeza-Yates, R. (2007). On the value of temporal information in information retrieval. In: *ACM SIGIR forum*, (vol. 41, pp. 35–41). ACM.

Amodeo, G., Amati, G., & Gambosi, G. (2011). On relevance, time and query expansion. In: *Proceedings of the 20th ACM international conference on Information and knowledge management*, pp. 1973–1976.

Arikan, I, Bedathur, S., & Berberich, K. (2009). Time will tell: Leveraging temporal expressions in IR. In: *WSDM*, Citeseer.

Berberich, K., Bedathur, S., Alonso, O., & Weikum, G. (2010). A language modeling approach for temporal information needs. In: *European conference on information retrieval*, pp. 13–25. Springer.

Bryant, F. B., Smart, C. M., & King, S. P. (2005). Using the past to enhance the present: Boosting happiness through positive reminiscence. *Journal of Happiness Studies*, *6*(3), 227–260.

Campos, R., Dias, G., Jorge, A. M., & Jatowt, A. (2015). Survey of temporal information retrieval and related applications. *ACM Computing Surveys (CSUR)*, *47*(2), 15.

Chang, A. X., Manning, C. D. (2012). Sutime: A library for recognizing and normalizing time expressions. In: *LREC*, vol. 2012, pp. 3735–3740.

Chen, D., Fisch, A., Weston, J., & Bordes, A. (2017). Reading wikipedia to answer open-domain questions. arXiv preprint arXiv:1704.00051.

Dai, N., & Davison, B. D. (2010). Freshness matters: In flowers, food, and web authority. In: *Proceedings of the 33rd international ACM SIGIR conference on research and development in information retrieval*, pp. 114–121.

Devlin, J., Chang, M.W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

Dong, A., Chang, Y., Zheng, Z., Mishne, G., Bai, J., Zhang, R., Buchner, K., Liao, C. & Diaz, F. (2010). Towards recency ranking in web search. In: *Proceedings of the third ACM international conference on web search and data mining*, pp. 11–20.

Elsas, J. L., & Dumais, S. T. (2010). Leveraging temporal dynamics of document content in relevance ranking. In: *Proceedings of the third ACM international conference on web search and data mining*, pp. 1–10.

Estela Saquete, J., Vicedo, L., Martínez-Barco, P., Munoz, R., & Llorens, H. (2009). Enhancing QA systems with complex temporal question processing capabilities. *Journal of Artificial Intelligence Research*, *35*, 775–811.

Fung, G.P.C., Yu, J.X., Yu, P.S. & Lu, H. (2005). Parameter free bursty events detection in text streams. In: *Proceedings of the 31st international conference on very large data bases*, pp. 181–192. VLDB Endowment.

Harabagiu, S., & Bejan, C. A. (2005). Question answering based on temporal inference. In: *Proceedings of the AAAI-2005 workshop on inference for textual question answering*, pp. 27–34.

Jia, Z., Abujabal, A., Saha Roy, R., Strötgen, J., & Weikum, G. (2018). Tempquestions: A benchmark for temporal question answering. In: *Companion of the the web conference 2018 on the web conference 2018*, pp. 1057–1062. International World Wide Web Conferences Steering Committee.

Kanhabua, N., Blanco, R., & Nørvåg, K. (2015). Temporal information retrieval. *Foundations and Trends in Information Retrieval*, *9*(2), 91–208. https://doi.org/10.1561/1500000043..

Kanhabua, N., & Nørvåg, K. (2010). Determining time of queries for re-ranking search results. In: *International conference on theory and practice of digital libraries*, pp. 261–272. Springer.

Kleinberg, J. (2003). Bursty and hierarchical structure in streams. *Data Mining and Knowledge Discovery*, *7*(4), 373–397.

Korkeamäki, L., & Kumpulainen, S. (2019). Interacting with digital documents: A real life study of historians' task processes, actions and goals. In: *Proceedings of the 2019 conference on human information interaction and retrieval, CHIIR '19*, pp. 35–43, New York, NY, USA. ACM. ISBN 978-1-4503-6025-8. https://doi.org/10.1145/3295750.3298931.

Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., & Soricut, R. (2019). Albert: A lite bert for self-supervised learning of language representations. arXiv preprint arXiv:1909.11942.

Lee, J., Yun, S., Kim, H., Ko, M., & Kang, J. (2018). Ranking paragraphs for improving answer recall in open-domain question answering. arXiv preprint arXiv:1810.00494.

Li, X., & Bruce C. W. (2003). Time-based language models. In: *Proceedings of the twelfth international conference on information and knowledge management*, pp. 469–475. ACM.

Li, X., & Roth, D. (2002). Learning question classifiers. In: *Proceedings of the 19th international conference on computational linguistics*(Vol. 1, pp. 1–7). Association for Computational Linguistics.

Metzler, D., Jones, R., Peng, F., & Zhang, R. (2009). Improving search relevance for implicitly temporal queries. In: Proceedings of the 32nd international ACM SIGIR conference on research and development in information retrieval, pp. 700–701. Citeseer.

Miller, G. A. (1995). Wordnet: A lexical database for english. *Communications of the ACM*, *38*(11), 39–41.

Moldovan, D., Clark, C., & Harabagiu, S. (2005). Temporal context representation and reasoning. In: *International joint conference on artificial intelligence*, vol. 19, p. 1099. Citeseer.

Ni, J., Zhu, C., Chen, W., & McAuley, J. (2018). Learning to attend on essential terms: An enhanced retriever-reader model for scientific question answering. arXiv preprint arXiv:1808.09492.

Pasca, M. (2008). Towards temporal web search. In: *Proceedings of the 2008 ACM symposium on applied computing*, pp. 1117–1121. ACM.

Peetz, M.-H., Meij, E., & de Rijke, M. (2014). Using temporal bursts for query modeling. *Information Retrieval*, *17*(1), 74–108.

Pennington, J., Socher, R., Manning, C. (2014). Glove: Global vectors for word representation. In: *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pp. 1532–1543.

Pustejovsky, J., Knippen, R., Littman, J., & Saurí, R. (2005). Temporal and event information in natural language text. *Language Resources and Evaluation*, *39*(2–3), 123–164.

Rajpurkar, P., Zhang, J., Lopyrev, K., & Liang, P. (2016). Squad: 100,000+ questions for machine comprehension of text. arXiv preprint arXiv:1606.05250.

Rajpurkar, P., Jia, R., & Liang, P. (2018). Know what you don't know: Unanswerable questions for squad. arXiv preprint arXiv:1806.03822.

Sandhaus, E. (2008). The New York times annotated corpus. *Linguistic Data Consortium, Philadelphia*, *6*(12), e26752.

Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2019). Distilbert, a distilled version of bert: Smaller, faster, cheaper and lighter. arXiv preprint arXiv:1910.01108.

Saquete Boró, E., Martinez-Barco, P., Munoz, R., & Vicedo, J. L. et al. (2004). Splitting complex temporal questions for question answering systems. Association for Computational Linguistics (ACL).

Seo, M., Kembhavi, A., Farhadi, A., & Hajishirzi, H. (2016). Bidirectional attention flow for machine comprehension. arXiv preprint arXiv:1611.01603.

Snowsill, T., Nicart, F., Stefani, M., De Bie, T., & Cristianini, N. (2010). Finding surprising patterns in textual data streams. In: 2010 2nd International workshop on cognitive information processing, pp. 405–410. IEEE.

Strötgen, J., & Gertz, M. (2012). Temporal tagging on different domains: Challenges, strategies, and gold standards. *LREC*, *12*, 3746–3753.

Strötgen, J., & Gertz, M. (2010). Heideltime: High quality rule-based extraction and normalization of temporal expressions. In: Proceedings of the 5th international workshop on semantic evaluation, pp. 321–324. Association for Computational Linguistics.

Vlachos, M., Meek, C., Vagena, Z., & Gunopulos, D. (2004). Identifying similarities, periodicities and bursts for online search queries. In: *Proceedings of the 2004 ACM SIGMOD international conference on management of data*, pp. 131–142. ACM.

Wang, J., Jatowt, A., Färber, M., Yoshikawa, M. (2020). Answering event-related questions over long-term news article archives. In: *European conference on information retrieval*, pp. 774–789. Berlin: Springer.

Wang, S., Yu, M., Guo, X., Wang, Z., Klinger, T., Zhang, W., Chang, S., Tesauro, G., Zhou, B., & Jiang, J. (2018). R3: Reinforced ranker-reader for open-domain question answering. In: AAAI.

Yang, W., Xie, Y., Lin, A., Li, X., Tan, L., Xiong, K., Li, M., & Lin, J. (2019). End-to-end open-domain question answering with bertserini. arXiv preprint arXiv:1902.01718.

Yang, P., Fang, H., & Lin, J. (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*, pp. 1253–1256.

Zahedi, M. S., Aleahmad, A., Rahgozar, M., Oroumchian, F., & Bozorgi, A. (2017). Time sensitive blog retrieval using temporal properties of queries. *Journal of Information Science*, *43*(1), 103–121.