Doctoral Thesis

A Study on Effective Approaches for Exploiting Temporal Information in News Archives

Jiexin Wang

June 2022

Department of Social Informatics
Graduate School of Informatics
Kyoto University

Doctoral Thesis
submitted to Department of Social Informatics,
Graduate School of Informatics,
Kyoto University
in partial fulfillment of the requirements for the degree of
DOCTOR of INFORMATICS

Thesis Committee: Masatoshi Yoshikawa, Professor
Keishi Tajima, Professor
Sadao Kurohashi, Professor
Donghui Lin, Associate Professor

A Study on Effective Approaches for Exploiting Temporal Information in News Archives*

Jiexin Wang

Abstract

With the application of digital preservation techniques, more and more past news articles are being digitized and made accessible online. This results in the availability of large news archives spanning multiple decades. They offer immense value to our society, contributing to our understanding of different time periods in the history and helping us to learn about the details of the past. Some professionals, like historians, sociologists, or journalists need to deal with these temporal news collections for a variety of purposes. Moreover, average users can verify information about the past using original, primary resources. However, the large sizes and complexities of news archives have gone far beyond user ability to utilize them efficiently. The need on how to quickly find the important, useful, precise or interesting information among an overwhelmingly large amount of news articles has rapidly arisen.

Additionally, in the news domain especially, time has long been an integral part of search engine ranking with most major search engine giving a ranking boost for recently published news articles. There are two distinct temporal aspects of a news article: timestamp (i.e., publication date) and content time (i.e., temporal expressions embedded in the document content). In the recent years, exploiting these two kinds of temporal information has been gaining increased importance in various tasks or applications, such as temporal web search, temporal question answering, search results diversification and so on.

In this dissertation, we first introduce three different methods of exploiting two distinct temporal information over temporal news collections. We demonstrate that injecting temporal information can not only improve the models' performance, but also benefit better utilization of news archives. Moreover, we construct

*Doctoral Thesis, Department of Social Informatics, Graduate School of Informatics, Kyoto University, KU-I-DT6960-31-7852, June 2022.

a large open-domain question answering (ODQA) dataset over news archives, which could further foster the research in exploiting temporal information. More specifically, we address the following four research topics:

- **Topic 1: Exploiting temporal information in question answering.** To quickly find the relevant information among an overwhelmingly large amount of news articles, we propose a question answering (QA) system called QANA. QANA is designed specifically for answering event-related questions over news archives, with an additional module which increases the retrieval effectiveness by utilizing diverse temporal information.

- **Topic 2: Exploiting temporal information in event occurrence time estimation.** Estimating the event occurrence time has many applications in IR, QA (e.g., QANA model in Topic 1), general document understanding and downstream NLP tasks. We propose TEP-Trans, which is a Transformer-based model to approach this task, by exploiting both temporal and textual information from different angles, represented by multivariate time series. TEP-Trans is able to estimate the event occurrence time and achieves state-of-the-art results at different temporal granularities.

- **Topic 3: Exploiting temporal information in constructing time-aware language representation.** A novel language model called TimeBERT is introduced, which is trained on news archives via two new pre-training tasks, harnessing the two kinds of temporal information to construct time-aware language representation. TimeBERT consistently outperforms BERT and other existing pre-trained models, with substantial gains on different different time-related downstream tasks.

- **Topic 4: Creating a large ODQA dataset over temporal news collections.** To foster the research in the field of ODQA on news archives, we propose one of the largest ODQA datasets called ArchivalQA over news collections. With the large-scale ArchivalQA dataset, more powerful temporal QA models with dense retriever modules that make use of both kinds of temporal information, can thus be well trained.

Keywords: Temporal News Collections, Temporal Information, Question Answering, Event Time Estimation, Text Representation, Question Generation

Contents

List of Figures

List of Tables

CHAPTER 1

INTRODUCTION

1.1 Background

In recent years, the global news industry has witnessed a drastic shift of its focus from traditional paper medium to publishing digital news articles. News archives (i.e., temporal news collections), constituting a large amount of fairly reliable, accurate digital news articles, play an essential role in preserving our heritage about the past and contributing to our understanding of different time periods in the history [76]. In addition, with full-text searching, users can do more than simply browse the pages, that they can delve into the pages of time and pluck out long-forgotten articles on any topic they choose over the news archives. These temporal news collections are fundamental resources for journalists, historians, librarians, and sociologists as they offer a detailed primary source record of how social processes evolve across time [13]. For example, sociologists have used news archives to examine vital questions such as the way United States abolished slavery [40] and how different jurisdictions slowed the spread of the 1918 flu [101], which can also offer valuable lessons for the racism problems and COVID-19 pandemic, the two major global issues we are facing today. Moreover, ordinary users could also use them for a variety of purposes, such as 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. Despite the great importance of temporal news

collections, users still feel difficult to efficiently make use of them due to their large sizes and complexities.

A news article contains multiple dimensions, according to which it can be analyzed, such as time, location, people, topical dimension and so on. These are also the essential elements that reporters should keep in mind during writing and publishing a news story, and can be used by average users to gather and organize important information. For example, if we look into the people characteristics of an article, we can know who and how many people are involved in the events and also their relationship. Similarly, if we regard the topic as a key dimension, we can know where the events happen. In this thesis, we particularly focus on the time dimension, which is one of the most significant dimensions especially in the temporal historical collections. Time could be leveraged to organize and search relevant information in news texts, aiding in exploration of the causalities, developments, and effects of the events, etc. For example, many current news search engines use time to boost the relevance of the most recent stories. According to Campos et al. [17], there are two distinct temporal aspects of a document: timestamp and content time. In news domain, the timestamp refers to the time when the news article has been published, while the content time refers to the temporal expressions embedded in the document content. Both temporal signals constitute important features of events or topics reported by news articles. On one hand, timestamp information can help readers locate the news reports published in specific periods quickly as well as let them assess the degree of document uptodateness. On the other hand, the content time can help to strengthen our understanding of document content, for instance, events developments and their causal relations can be understood by analyzing the relations between different content temporal information. A great deal research studies have already been proposed for using temporal information for exploration and search purposes [4, 17, 61], such as temporal web search [146], summarization [9], temporal question answering[161, 162] and so on. However, most of the existing work either utilizes only one of the two important temporal signals, or can only obtain poor performance.

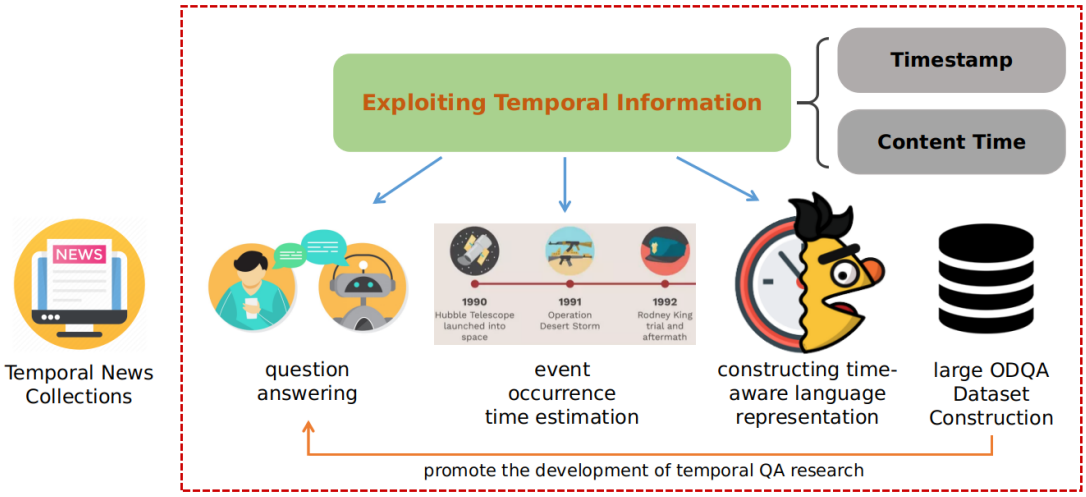

Figure 1.1. Framework of Doctoral Thesis.

1.2 Overview of the Research

In this dissertation, we also focus on the temporal historical collections and four research topics are addressed, with the objective to benefit better utilization of such valuable resources. Figure 1.1 shows the overall framework of the thesis, including four topics. We first propose three novel approaches by exploiting two previously introduced distinct temporal information (i.e., timestamp and content time) in different ways. The experimental results in these three work demonstrate that injecting such temporal information can result in better performances or even achieve new state-of-the-art results in various tasks. In the last research topic, we additionally construct a large-scale open-domain question answering (ODQA) dataset over temporal news collections, which aims to promote the development of QA research over news archives. As detailed motivations will be respectively described in the following chapters, in this section, we briefly introduce the motivation, target and approach of each research topic.

1.2.1 Exploiting Temporal Information in Question Answering

As we discussed earlier, the efficient utilization of temporal news collections is rather difficult for average users due to their large sizes and complexities. An effective solution would be to use open-domain question answering systems (ODQA

systems), which attempt to identify the most correct answer from a large document collection for a particular information need, expressed as a natural language question. However, as existing QA systems are essentially designed for synchronic document collections (e.g., Wikipedia), they are incapable of using important temporal information like timestamp or content time when answering questions on temporal news collections.

In this chapter, we present a large-scale question answering system called QANA (Question Answering in News Archives) designed specifically for answering two types of event-related questions on news archives, with an additional module called Time-Aware Re-ranking Module for re-ranking articles by using temporal information from different angles. More specifically, QANA system exploits the temporal information of a question, of a document content and of its timestamp for re-ranking candidate documents. The experimental results show that our proposed approach can improve retrieval effectiveness on two types of questions and surpasses the existing QA systems that are commonly used for large-scale automatic question answering.

1.2.2 Exploiting Temporal Information in Event Occurrence Time Estimation

Estimating the event occurrence time has many potential applications, such as search results diversification [11, 46, 144], timeline construction [43, 86, 147], and historical event ordering [50], etc. It can also be applied in temporal information retrieval or temporal question answering system, for example, an important step of QANA system is the question time scope estimation, which requires to gauge the possible time periods of the event mentioned in the question. Nonetheless, the performance of the existing methods is unsatisfactory for the temporal event profiling task, especially at fine-grained temporal granularities (e.g., day, week).

In this chapter, we address the problem of event occurrence time estimation task that defined as follows: *given a short event description and a chosen temporal granularity, the task is to estimate event's occurrence time at the specified granularity using a temporal document collection as the underlying knowledge source*. To approach the task, we propose a model called TEP-Trans (Temporal Event Profiling Transformer-based model) over temporal document collections, by exploiting both temporal and textual information, represented by multivari-

ate time series. The TEP-Trans model is capable of modeling useful features of the input multivariate time series and achieves new state-of-the-art results at all the temporal granularities. In addition, we show that QANA system, the model we introduced in the first research topic, achieves better performance by using TEP-Trans model in the question time scope estimation step.

1.2.3 Exploiting Temporal Information in Constructing Time-aware Language Representation

Pre-trained language models such as BERT [28], RoBERTa [98], XLNet [28], which capture contextual information from large-scale corpora via pre-training tasks, have achieved promising results in various NLP tasks. However, existing language models are pre-trained either on general-purpose large-scale text corpora (e.g., Wikipedia) or without utilizing important temporal signals such as document timestamps, which limits their applications to specific domains or particular tasks.

Inspired by the development of language models, we aim to construct effective time-aware language representation, which could be easily applied in various time-related downstream tasks. We introduce TimeBERT, a novel pre-trained language model trained on a temporal news collection by exploiting both timestamp and content time, via two specially designed time-oriented pre-training tasks. The experimental results show that our proposed TimeBERT language model could simultaneously utilize both distinct temporal aspects in an effective way, as it outperforms other pre-trained language models by a large margin on several time-related tasks.

1.2.4 Creating a Large-scale ODQA Dataset over Temporal News Collections

Over the past few years, a large number of QA benchmarks have been introduced, which contributed to the success and development of question answering research. Nonetheless, most of the existing datasets are designed over synchronic document collections, such as Wikipedia and web search results. The lack of large-scale datasets hinders the development of ODQA models over temporal news collections. For example, dense passage retrieval model, that needs large amounts of

training data, has become a new paradigm to retrieve relevant passages for finding answers. The QANA system, which we introduce in the first research topic, can only use sparse dense retrieval rather than dense passage retrieval, resulting from the small amount of labeled data.

Thus, to foster the research in the field of ODQA on temporal historical collections, we present ArchivalQA, a large-scale question answering dataset consisting of 532,444 question-answer pairs which is designed for temporal news QA. The dataset is constructed through a semi-automatic pipeline, which uses automatic question generation techniques based on a cascade of carefully designed filtering steps that remove low quality questions. In the experiments, we undertake comprehensive analysis of ArchivalQA, showing that the resulting dataset is of high quality. The novel QA dataset-constructing framework can be also applied to generate high-quality, non-ambiguous questions over other types of temporal document collections.

1.3 Thesis Structure

The structure of this thesis is as follows. In Chapter 2, we discuss and overview previous studies related to the four research topics presented in this thesis. Chapter 3 to Chapter 6 in this thesis correspond to the above four introduced research topics. Chapter 3 presents QANA system, which is designed specifically for answering two types of event-related questions on news archives. In Chapter 4, we propose TEP-Trans model to approach the task of event occurrence time estimation, which achieves new state-of-the-art results. Chapter 5 introduces TimeBERT, a novel language representation model trained on a temporal collection of news articles via two new pre-training tasks, with substantial gains on two different time-related downstream tasks. In Chapter 6, we propose a large-scale question answering dataset called ArchivalQA over temporal news collections. Finally, Chapter 7 summarizes the thesis and addresses several directions to be explored as future work.

CHAPTER 2

Related Work

2.1 Two Temporal Dimensions of Text

According to Campos et al. [17], there are two distinct temporal dimensions of a document or a query: timestamp (or creation time) and focus time (sometimes called content time). The document timestamp refers to the time when the document has been created, while the focus time is the time mentioned or implicitly referred to in the content of document. For example, one could write an article about 9/11 terrorist attacks in 2022 in which case the timestamp would be 2022 while the main content time would be September 11, 2001. Similarly, the query timestamp refers to the time when the query was issued, while the focus time [54] is the content time of the query. For example, "2004 Summer Olympics" query issued in year 2022 would have its timestamp of 2022 and would refer to the time period when the summer sports took place in Athens, Greece in 2004. For readers, timestamp information can help them locate the news reports published in specific periods quickly as well as let them assess the degree of document uptodateness. On the other hand, the content time can help to strengthen our understanding of document content, for instance, events developments and their causal relations can be understood by analyzing the relations between different content temporal information.

In this thesis, we investigate different approaches of incorporating the two

distinct temporal signals, and demonstrate that the utilization of the temporal information can result in better performance or even new state-of-the-art results in various tasks. Every following section in this chapter corresponds to the related work of a particular topic among the four previously introduced research topics.

2.2 Temporal Information Retrieval and Open-Domain Question Answering

QANA model is proposed in the first research topic, which is the first study to adapt and improve concepts from temporal information retrieval to the QA research domain, showing significant improvement in answering event-related questions on temporal news collections. Therefore, we introduce the studies related to temporal information retrieval and question answering models in this section.

2.2.1 Temporal Information Retrieval

In recent years, exploiting the temporal information in documents and queries has been gaining increased importance, leading to the formation of a subset of information retrieval area called temporal information retrieval (TIR) in which both query and document temporal aspects are of key concern. In the area of temporal information retrieval, several works for temporal ranking of documents have been proposed [2, 17, 65]. For example, Li and Croft [91] introduce a time-based language model considering the timestamp metadata of documents to give preference to more recent documents. Similar research studies [24, 33, 38] 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. [7] propose a language model based retrieval framework which exploits temporal expressions of document content. Berberich et al. [12] 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. [105] 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.

The most related work to our first research topic is [64]. Kanhabua and Nørvåg [64] 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 [64] 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. Note also that all the related 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.

2.2.2 Open-Domain Question Answering

Open-domain question answering systems (ODQA 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 [28] or versions derived on the basis of BERT [81, 137], can even go beyond human performance (as quantified based on EM (Exact Match) and F1 scores) on both SQuAD 1.1 [126] and SQuAD 2.0 [127], 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 [47, 67, 83, 85, 114, 164, 171] 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. Some models which use term-based sparse passage retrievers (e.g. TF-IDF and BM25) have been proposed first. Chen et al. [22] 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. [164] propose R^3 model, whose IR component is trained jointly with MRC component by reinforcement learning based method. Yang et al. [171] propose BERTserini that integrates IR component using Anserini IR toolkit [170] with BERT-based MRC model. Different from the term-based sparse retriever approaches, over the past few years, models of dense passage retrieval have been proposed, which represent both questions and documents as dense vectors [47, 67, 85, 123]. These advance ODQA models incorporate BERT-based reader module with BERT-based dense retriever module, yield substantial improvements over the traditional methods. However, large amounts of training data is required for training an effective retriever module.

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. Although some temporal QA systems that can exploit temporal information have been proposed [48, 109, 118, 138, 139], 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 [48, 109], complex question decomposition [138, 139] or answering "when" type of questions [118]. Besides, these works represent primarily traditional rule-based models and their performance is quite poor.

Furthermore, there are very few resources available for answering event-related questions over news archives. Jia et al. [56] release a benchmark with 1,271 temporal question-answer pairs. Since we use NYT corpus which contains news articles published between 1987 and 2007, only few of the questions whose cor-

responding events occurred within that time interval could be used.* Note that also because of the lack of large-scale datasets over temporal news collections, the advanced QA models that use dense retriever module (e.g., DPR model [67], RocketQA model [123]) cannot be well trained. Thus, our proposed QANA model uses traditional sparse retriever module. However, the final research topic tackles this problem by introducing a large ODQA dataset.

QANA model contains an additional module that is used for reranking documents which improves the retrieval of correct documents by exploiting temporal information from different angles. 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, no studies, as well as no available datasets that can help in designing a QA system to effectively work on long-term temporal collections of news articles, have been proposed so far. Building a QA system that can make better use of the past news articles and fulfill different information needs (both of professionals working with such collections and average users), is however of great importance especially nowadays due to the continuously growing document archives.

2.3 Temporal Information Estimation

In the second research topic, we address the problem of event occurrence time estimation. As this task requires predicting the time of a given short event description, it is similar to the query focus time estimation, which aims to identify the time of the interest of short queries. Thus, in this section, we discuss some work related to the estimation of different types of temporal information.

2.3.1 Document Timestamp Estimation

Document timestamp estimation, or document dating, is a challenging problem which requires extensive reasoning over the temporal structure of the document. One of the first automatic document dating studies is the work of Jong et al. [58]. They use unigram language models for specific time periods and score articles

*Most questions are about events which happened long time ago (e.g., Viking Invasion of England) or are not event-related.

with log-likelihood ratio scores. Kanhabua and Nørvåg [62] further extend this work by expanding its unigrams with POS tags, collocations, and tf-idf scores. Chambers [20] propose a discriminative model, which is based on the temporal expressions by leveraging the a Maximum Entropy classifier and additional time constraints. Kotsakos et al. [77] introduce a purely statistical method which considers lexical similarity alongside burstiness [82] of terms. Vashishth et al. [157] propose NeuralDater model, the first method to utilize deep learning techniques for predicting the document timestamp information, which is based on Graph Convolutional Networks (GCN) that jointly exploits syntactic and temporal graph structures of document.

2.3.2 Document Focus Time Estimation

Document focus time estimation [55], which aims to determine the temporal distribution reflecting the time periods the content of a given document treats about. As a document usually contains sentences related to different events that take place in different time points, document focus time is often represented by a set of time intervals [54]. In order to estimate the correct focus time of a document, approaches need to evaluate the time to which individual sentences refer. The authors of [55] propose a graph-based approach that constructs a date-term association graph based on the co-occurrence of words and temporal expressions, and identify discriminative associations which are then used to estimate the focus time. Shrivastava et al. [143] also introduce a graph-based method but treat documents and years as nodes which are connected by intermediate related Wikipedia concepts. They leverage the temporal relations between the concepts present in the text to estimate the document focus time. The shortcoming here is that documents may not always contain temporal expressions.

Unlike the two above-mentioned tasks, the event occurrence time estimation does not aim to predict the publication date of text, it focuses strictly on events (rather than states), as well as it has different input which is not a document but a short event mention.

2.3.3 Query Focus Time Estimation

Another relevant research problem is the task of query focus time estimation, or query temporal profiling, which aims to temporally disambiguate queries (e.g.,

queries about past, future, present or queries that are temporally neutral) as well as identify the time of their interest. Note that this task is similar to the task of event occurrence time estimation, and plays a significant role in temporal information retrieval so that time of queries and time of documents can be matched. Nonetheless, this task focuses on short queries rather than event descriptions (e.g. "Hurricane Katrina"). Kanhabua and Nørvåg [64] introduce three different methods to identify the time of interest of queries and exploit this information for re-ranking the retrieved results. Their best-performing method uses only the timestamps of the top k retrieved documents as the query time. Thus the query time contains more than one time point when the timestamps of top k documents are different and the approach cannot determine which one is correct. Methods proposed by Dakka et al. [25], Jones and Diaz [57] also utilize timestamp information and identify query time by analyzing distribution of retrieved documents over time. Unlike these methods, Gupta and Berberich [45] take both timestamp information and temporal expressions from the content into account, and employ a probabilistic approach for the selection of suitable documents for a given query to subsequently generate a time interval from the temporal information. Differently to our approach, the authors mainly focus on the temporal expressions in the content and utilize the timestamp information only as additional temporal information of the content.

2.3.4 Event Occurrence Time Estimation

Other related works propose different ways to estimate the occurrence time of a given short event description [26, 50, 110]. Das et al. [26] introduce event-based time vector by integrating word vectors and global time vector, and estimate the occurrence time by calculating the cosine similarity between event-describing sentences and event-based time vectors corresponding to temporal expression. Morbidoni et al. [110] utilize Wikipedia as well as the external knowledge base - DBpedia, and estimate the occurrence time by leveraging linked entities' centered representation of sentences and temporal information. Honovich et al. [50] propose two methods to tackle the task where the best one is realized by first extracting relevant sentences from the Wikipedia, and then using LSTM with attention mechanism to compute the encodings of event text and extracted sentences, and finally using an MLP to estimate the occurrence time that takes the concaten-

ated encodings as input. Nonetheless, neither of these three methods is designed to work over primary document collections such as news archives, making them incapable of utilizing temporal information such as document timestamps. Although knowledge bases and Wikipedia contain abundant information on the major things from the past, they cannot provide information on numerous minor events that took place in the history. Finally, those methods work on rather coarse level granularity predicting only year information of the event time.

In comparison to the existing methods, our proposed TEP-Trans model is designed over news archives. We leverage the novel Transformer architecture [158] and we let it utilize both temporal information and textual information embedded in documents. Our model can infer the event occurrence time at different temporal granularities. We also construct a large dataset for training the proposed model and release it to the research community. Event occurrence time estimation constitutes a significant building block for many downstream tasks (e.g. temporal information retrieval [2, 17], search result diversification [11, 46, 144], etc.), and might even serve as a fallback of question answering when the answer of the question about event date is not explicitly given in the text.

2.4 Pre-trained Language Models

In the third topic, we present TimeBERT, a novel language model trained on a temporal news collection via two new pre-training tasks, which harness two distinct temporal signals to construct time-aware language representation. Thus, in this section, we discuss some research related to pre-trained language models.

2.4.1 General Pre-trained Language Models

BERT [28] has emerged as one of key breakthroughs that contributed to the recent success and development of pre-trained language models, which capture contextual information from large-scale corpora via pre-training tasks. In particular, BERT relies on two well-designed pre-training tasks: *masked language modeling* (MLM) and *next sentence prediction* (NSP). Loosely speaking, MLM first masks out some tokens from the input sequence and the model is trained to predict the masked tokens, while NSP is a binary classification task that aims to predict whether two segments follow each other in the original text. Thanks to its success

in various NLP tasks, many variants of BERT based on transformer [158] have been proposed. RoBERTa [98], for instance, is an improved version of BERT obtained in result of a careful analysis of the impact of many key hyperparameters, and of removing NSP objective as well as increasing the size of training data, etc. ALBERT [81] replaces NSP with the sentence order prediction (SOP) objective and consistently outperforms BERT on various downstream tasks even with a smaller parameter size than BERT. Moreover, many recent works also suggest adaptations and update of MLM in order to further improve BERT. For example, XLM [80] replaces MLM with translation language modeling (TLM), which improves crosslingual language model pre-training by leveraging parallel data. XLNet [60] replaces MLM with permuted language modeling (PLM), which randomly permutes a sequence and predicts the tokens in an autoregressive way. ERNIE [152] uses phrase and named entity masking and shows improvements on Chinese NLP tasks. SpanBERT [60] extends BERT by masking contiguous random spans and utilizes a span boundary objective. However, the problem with the above-listed language models is that they are pre-trained on general-purpose large-scale text corpora (e.g., Wikipedia), which limits their applications to specific domains or particular tasks.

2.4.2 Language Models for Specific Domains

Some studies adapt pre-trained models to specific domains by directly applying the two pre-training tasks of BERT on domain-constrained datasets. The well-known examples are SciBERT [10] trained on scientific corpus, BioBERT [84] obtained using a biomedical document corpus, and ClinicalBERT [52] derived from a clinical corpus. Another line of work attempts to adapt available pre-trained models to target applications or tasks. For example, Ke et al. [68] propose SentiLARE for sentiment analysis task, which replaces MLM with label-aware masked language model, introducing word-level linguistic knowledge into pre-trained models. Xiong et al. [167] design WKLM for entity-related tasks, which is trained using the entity replacement objective. This objective requires the model to make a binary prediction indicating whether an entity has been replaced or not. The experimental results with WKLM suggest that this kind of adaptation can better capture knowledge about real-world entities. Similarly, Yang et al. [169] propose KT-NET for machine reading comprehension task

that improves the models with additional knowledge obtained from knowledge bases. Althammer et al. [5] introduce linguistically informed masking (LIM), a domain adaptive pre-training method for the patent and legal domain. LIM shifts the masking probabilities in domain-adaptive pre-training towards the highly informative noun chunks in patent language. Its effectiveness is proved on two patent-related downstream tasks.

2.4.3 Incorporating Time with Language Models

In recent years, incorporating time with language models has also been gaining increased importance [30, 44, 133, 134]. Giulianelli et al. [44] propose the first unsupervised approach to use contextualized embeddings from BERT to model lexical semantic change. Dhingra et al. [30] propose a simple modification to pre-training that parametrizes MLM objective with timestamp information using temporally-scoped knowledge, and tested the proposed language model on question answering task. [133, 134] address mainly the tasks of semantic change detection,[†] that needs to identify which words undergo semantic changes and to what extent. More specifically, Rosin and Radinsky [133] extend the self-attention mechanism of the transformer architecture [158] by incorporating timestamp information, which is used to compute attention scores. Rosin et al. [134] further train BERT by using the concatenation of timestamp and text sequences as input, which helps to achieve the SOTA performance on semantic change detection. As we can see, these models mainly focus on the problem of lexical semantic change and utilize the timestamp at only coarse granularity (i.e., year or even decade). They also do not utilize the content time, despite the fact that the content time actually constitutes an important temporal signal and is relatively common.

Similar to the above pre-trained models, TimeBERT is also a transformer-based [158] language representation model. However, unlike all the aforementioned approaches, it exploits both timestamp and content time during pre-training on a temporal news collection, and achieve high performance on two downstream time-related tasks. Building such a language model that can further help to make better use of the temporal information in various applications is of great

[†]Although Rosin et al. [134] additionally experiment with sentence time prediction task, they test on two datasets that are of rather coarse granularity, such that the number of classes in the harder setting of year granularity is 40, while it is 4 in the easier setting of decade granularity. In addition, they only achieved a small improvement compared with other baselines.

importance, especially in temporal information retrieval field, question answering over temporal collections, and in other NLP tasks that rely on temporal signals.

2.5 Question Answering Datasets

We finally discuss the work related to the forth research topic, whose goal is to create a large-scale QA dataset over news archives that can promote the development of ODQA research on such valuable collections. As creating questions by hand requires much time and cost, and manual answer assessment, as well as demands knowledge of history in our particular case, automatic question generation techniques are utilized to solve this problem.

2.5.1 QA Benchmarks

In recent years, a large number of QA benchmarks have been introduced [8, 37, 132, 176]. The SQuAD 1.1 dataset [126] consists of question-answer pairs that are made from the paragraphs of 536 Wikipedia articles. This dataset was later extended by SQuAD 2.0 [127] that contains also unanswerable questions. NarrativeQA [75] uses a different resource, the summaries of movie scripts and books, to create its question-answer pairs. SearchQA [36] and TriviaQA [59] create a more challenging setting by utilizing web search to collect multiple documents to form the context given existing question-answer pairs from Jeopardy! quiz show and quiz websites, which may be useful for inferring the correct answers. MS MARCO [113] and NaturalQuestions [79] use the search query logs of Bing and Google search engines as the questions, and the retrieved web documents and Wikipedia pages are collected as the evidence documents. XQA dataset [96] is constructed for cross-lingual OpenQA research that consists of a training set in English as well as of the development and test sets in eight other languages.

Most of the existing datasets are designed over synchronic document collections, such as books, Wikipedia articles and web search results. While there are some MRC (machine reading comprehension) datasets created based on the news articles, they mostly belong to the cloze style datasets, such as CNN/Daily Mail [112], WhoDidWhat [116] and ReCoRD [178], with the aim to predict the missing word in a passage rather than to answer proper questions; hence they cannot be used in the ODQA task. Although Lelkes et al. [87] constructed the

NewsQuizQA dataset based on news articles, too, its questions belong to the multiple-choice type, which are easier to be answered, and the dataset contains only 20K question-answer pairs. The pairs were also obtained from only 5K summaries derived from the recent news articles. In addition, it has been designed as a dataset for generating the quiz-style question-answer pairs.

To the best of our knowledge, NewsQA [156] is the only MRC dataset in which an answer is a text span which is created based on the temporal document collection, the CNN news articles. However, our dataset has significant differences when compared to NewsQA. First, dataset size of NewsQA is much smaller than ours (119K vs. 532K). Second, its underlying CNN corpus contains less news articles which span shorter and also more recent time period (93k articles from 2007/04 to 2015/04 vs. 1.8M articles from 1987/01 to 2007/06 as in our case). We have also found that NewsQA is essentially appropriate for the MRC task and is not very suitable for the ODQA task. This is because many questions require additional background knowledge about their original paragraphs for understanding and correctly answering them. These questions tend to be ambiguous, unclear and generally impossible to be answered over the large news collection, because they are not specific enough and tend to have multiple correct answers (e.g., the questions *"When were the findings published?"*, *"Who drew inspiration from presidents?"* and *"Whose mother is moving to the White House?"*.[‡]) Note that questions on some QA datasets also have similar characteristics, for example, Min et al. [107] found that over half of the questions in the NaturalQuestions are ambiguous, with diverse sources of ambiguity such as event and entity references. Finally, the questions in NewsQA have been created from 7 times less articles than in our final dataset (12,744 vs. 88,431). In Table 2.1 we summarize differences between ArchivalQA and the most related datasets.

2.5.2 Automatic Question Generation

In recent years, automatic question generation (AQG) has greatly advanced thanks to deep learning techniques, and it has received increasing attention due to its wide applications in education [78], dialogue systems [166], and question answering [35]. Diverse types of neural sequence-to-sequence models have been

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

Table 2.1. Comparison of related QA datasets.

Dataset	#Que	Answer Type	Question Source	Corpus	Synch/Diach	Non-ambiguous
SQuAD 1.1 [126]	108K	Extractive	Crowd-sourced	Wikipedia	Synchronic	✗
SQuAD 2.0 [127]	158K	Extractive	Crowd-sourced	Wikipedia	Synchronic	✗
NaturalQuestions [79]	323K	Extractive, Boolean	Query logs	Wikipedia	Synchronic	✗
CNN/Daily Mail [112]	1M	Cloze	Automatically Generated	News	Diachronic 2007/04-2015/04	✗
NewsQuizQA [87]	20K	Multiple-choice	Crowd-sourced	News	Diachronic 2018/06-2020/06	✗
NewsQA [156]	119K	Extractive	Crowd-sourced	News	Diachronic 2007/04-2015/04	✗
ArchivalQA	532K	Extractive	Automatically Generated	News	Diachronic 1987/01-2007/06	✓

proposed for the AQG task. Zhao et al. [180] introduce a model that incorporates paragraph-level inputs - the first QG model that achieved large improvement over sentence-level inputs. Sun et al. [151] and Kim et al. [71] improved the performance by encoding answer positions, which can help to generate better-quality answer-focused questions. Some works also propose QG models under particular constraints, e.g., controlling the difficulty [42] and topic [51] of the generated questions. In addition, models that can jointly learn to ask (QG) as well as answer questions (QA) have been also proposed [135, 165]. Moreover, it has been shown that having a large, even synthetic dataset, is useful for training QA models with different objectives. For example, Puri et al. [121] train their model using only the synthetic data and obtain state-of-the-art performance on SQuAD dev set. Shakeri et al. [142] improve the performance of models in target domains by utilizing the synthetic dataset. Saxena et al. [140] demonstrate that the large size model-generated dataset can help in training temporal reasoning models. Lewis et al. [88] propose to use unsupervised question generation (e.g., template/rule-based methods) to tackle unsupervised QA task, a setting in which no aligned question, neither context no answer data are available. They demonstrate that their method can outperform early supervised models on SQuAD 1.1 dataset without using the SQuAD training data, and modern QA models can learn to answer human questions surprisingly well using only synthetic training data. In addition, some existing Visual Question Answering (VQA) datasets, such as COCO-QA [130] and Visual Madlibs [173], have also had AQG techniques applied to generate their questions.

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

CHAPTER 3

Exploiting Temporal Information in Question Answering

Temporal news collections (or news archives) are valuable resources about our past, allowing people to know detailed information of events that occurred at specific time points. 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, the first research topic considers the task of open-domain question answering over news archives. Questions on such archives are usually related to particular events and show strong temporal aspects. We propose an ODQA system called QANA to answer event-related questions, which is designed specifically for news archives, with an additional module for re-ranking articles by using temporal information from different perspectives.

3.1 Introduction

With the application of digital preservation techniques, more and more old news articles are being digitized and made accessible online. News archives help users to learn detailed information on events that occurred at specific time points in

the past and serve valuable purpose in building our understanding of particular time periods in history [76]. Some professionals, like historians, sociologists, or journalists need to deal with these temporal news collections for a variety of purposes [13]. In addition, average users could also use them for a variety of purposes, such as 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. Yet, it is difficult for users to efficiently make use of news archives due to their large sizes and complexities. 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 open-domain question answering systems (ODQA systems), with the aim to identify the most correct answers to questions from a large document collection. 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 3.1 shows some examples of the temporal questions.

This chapter presents 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 the temporal news collections. 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

Table 3.1. 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

temporal information it re-ranks candidate documents so as the probability of finding the correct answer in the top results is increased.

In the experiments, we tested our approach using the New York Times Annotated corpus (NYT corpus) as the underlying knowledge source, based on carefully constructed test set of questions related to past events. The test set is composed of two types of questions (explicitly and implicitly time-scoped) which have been selected from existing data sets and history quiz websites. 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 question answering.

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

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 dedicated test sets and a document collection spanning 20 years.

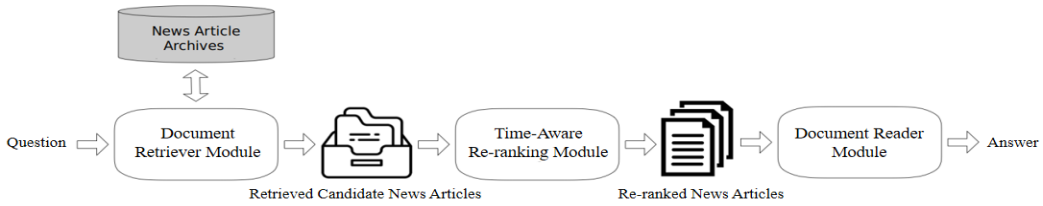

Figure 3.1. The architecture of the proposed system

The remainder of this chapter is structured as follows.* In Section 3.2, we describe our approach. Section 3.3 explains experimental settings and shows experimental results. Finally, we conclude the chapter in Section 3.4.

3.2 Approach

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.

The system architecture is shown in Figure 3.1 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.

*Note also that the related work of temporal information retrieval and open-domain question answering is discussed in Section 2.2.

3.2.1 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.[†] 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 [106] and are further filtered by leaving those whose POS types match the original question terms, and whose cosine similarity[‡] to question terms is above 0.5. Finally, a query is sent to the ElasticSearch[§] installation which returns the top 100 candidate articles ranked by BM25.[¶]

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

I. 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.

[†] `https://spacy.io/`

[‡] We use Glove [120] word vectors trained on the Common Crawl dataset with 300 dimensions.

[§] `https://www.elastic.co/`

[¶] Note that as we discussed in Section 2.2.2, due to the lack of large amounts of training data over news archives that is required for train an effective retriever module, QANA can only use sparse retriever approach rather than the advanced dense retrieval methods. However, we solve this problem by constructing a large-scale ODQA dataset in Chapter 6.

Explicitly Time-scoped Questions As we mentioned before, the time scope of the explicitly time-scoped question can be obtained directly. SUTime [21], a tool for recognizing temporal expressions and normalizing them according to the TimeML annotation standard [122], is used to recognize and normalize the temporal expression of the question Q.* 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.† 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.‡

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 [6, 119, 175]. 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 3.2 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).

*A temporal expression is annotated to one of four types: Date, Time, Duration, and Set. We tested SUTime on 346 temporal questions selected from TREC question classification dataset [92], 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).

†In the experiments, we use monthly granularity.

‡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.

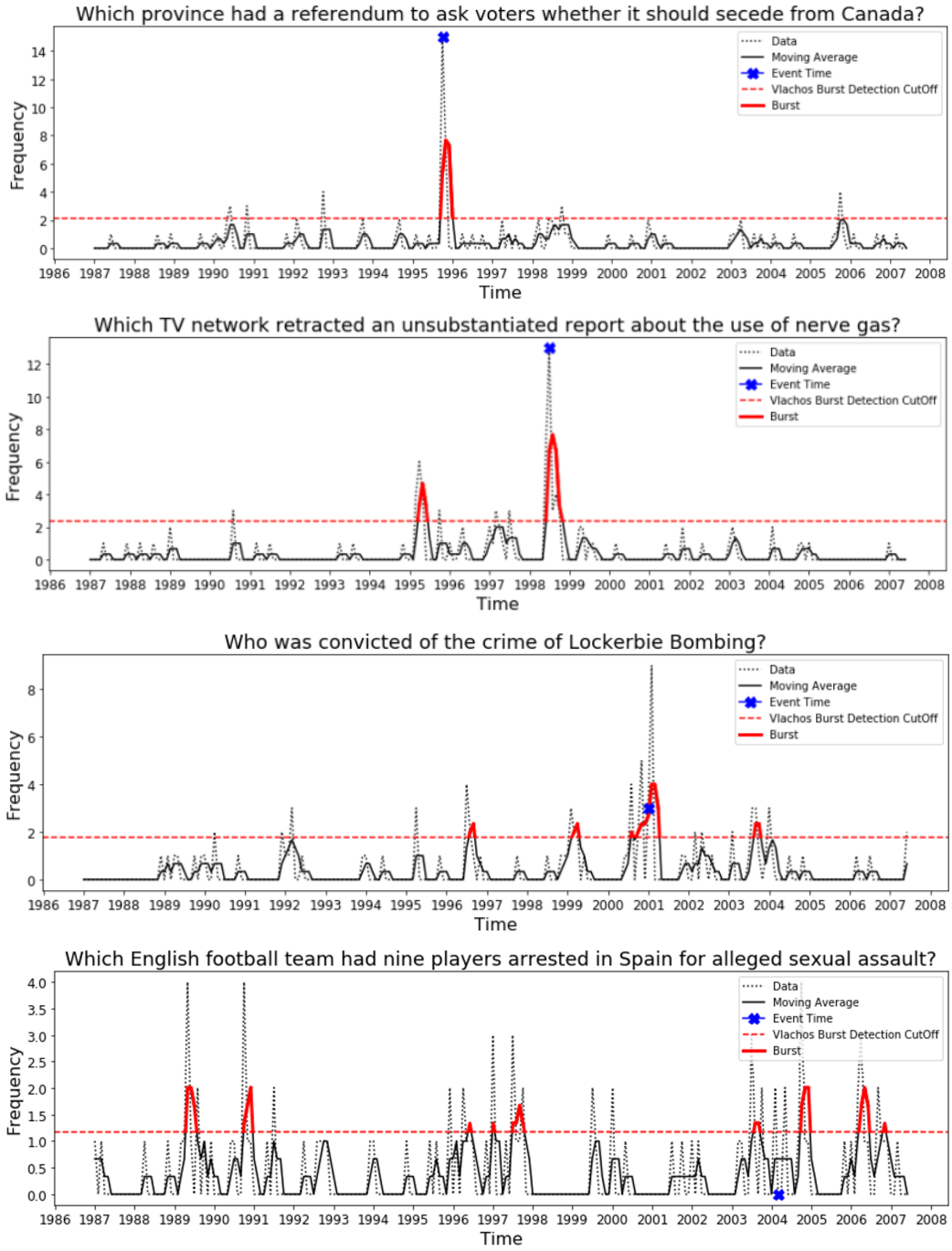

Figure 3.2. Burst detection results of four questions using the New York Times Annotated collection. The questions were converted to their corresponding queries as described in Section 3.2.1, and the top 100 ranked results by BM25 were used. Best viewed in color.

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 Figure 3.2 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 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. [159] is chosen, which provides a simple yet

effective way to identify bursts.[§] 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.

Algorithm 1: Question Time Scope Estimation

Data: Timestamp sequence $T_{pub}(Q)$, window size w, cutoff parameter β
Result: 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.

[§]There are many alternative burst detection techniques that could be potentially used (e.g., [41], [145], [73])

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 three months. β affects the cutoff value. We use β equal to 2.0 which is a suggested value by [159]. In Figure 3.2, 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: $W(T(Q)) = [(w(t^s_1(Q), t^e_1(Q))), ...(w(t^s_m(Q), t^e_m(Q)))]$, where m is the number of periods in $T(Q)$.

II. Timestamp-based Temporal Score Calculation. After obtaining the question time scope $T(Q)$, the module calculates the timestamp-based temporal score $S^{temp}_{pub}(d)$ 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:

$$\begin{aligned} S^{temp}_{pub}(d) &= P(T(Q)|t_{pub}(d)) \\ &= \lambda^{Dis(T(Q),t_{pub}(d))} = \lambda^{Dis((t^s(Q),t^e(Q)),t_{pub}(d))} \quad (0 < \lambda < 1) \end{aligned} \tag{3.1}$$

$S^{temp}_{pub}(d)$ is estimated as $P(T(Q)|t_{pub}(d))$, which means the probability of generating time scope $T(Q)$ (following [64]), 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:

$$Dis((t^s, t^e), t_{pub}(d)) = \begin{cases} +\infty & when\ t^s > t_{pub}(d) \\ 1.0 - \frac{|t^s - t_{pub}(d)| + |t^e - t_{pub}(d)|}{2 \cdot TimeSpan(D)} & elsewhere \end{cases} \tag{3.2}$$

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. 3.2 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.* 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 explicitly time-scoped questions, 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:

$$\begin{aligned} S^{temp}_{pub}(d) &= P(T(Q)|t_{pub}(d)) \\ &= P(\{(t^s_1(Q), t^e_1(Q)), ...(t^s_m(Q), t^e_m(Q))\}|t_{pub}(d)) \\ &= \frac{1}{m}\sum_{i=1}^{m} P((t^s_i(Q), t^e_i(Q))|t_{pub}(d)) \end{aligned} \tag{3.3}$$

$S^{temp}_{pub}(d)$ 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 m candidate periods of time scope $T(Q)$. Then, by

*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.

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. ...

Figure 3.3. The examples of news articles that retrospectively refer to the target event mentioned in the question. Best viewed in color.

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:

$$P((t_i^s(Q), t_i^e(Q))|t_{pub}(d)) = w(t_i^s(Q), t_i^e(Q)) \cdot \lambda^{Dis((t_i^s(Q), t_i^e(Q)), t_{pub}(d))} \quad (3.4)$$

$Dis((t_i^s(Q), t_i^e(Q)), t_{pub}(d))$ 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. 3.2. 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.

III. Content-based Temporal Score Calculation. For each candidate document d, the module computes also content-based temporal score, $S_{text}^{temp}(d)$.

$S^{temp}_{text}(d)$ 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 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 Figure 3.3, 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, according to Strötgen and Gertz [149], implicit temporal expressions (e.g. "D-Day") are relatively rare in news articles , which means that most temporal expressions can be well annotated. 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 [21] 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* (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 $T^s_{text}(d)$, $T^e_{text}(d)$ 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

*The temporal signals' list is taken from [56].

$T(Q)$ 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 $T^s_{text}(d)$, $T^e_{text}(d)$. 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, $S^{temp\_s}_{text}(d)$ and $S^{temp\_e}_{text}(d)$, which are then combined to compute the final content-based temporal score $S^{temp}_{text}(d)$ of the document d. Similar to the idea in computing the timestamp-based temporal score, $S^{temp\_s}_{text}(d)$ and $S^{temp\_e}_{text}(d)$ are estimated as $P(t^s(Q)|T^s_{text}(d))$ and $P(t^e(Q)|T^e_{text}(d))$, which means the probabilities of generating $t^s(Q)$ and $t^e(Q)$ based on $T^s_{text}(d)$ and $T^e_{text}(d)$, respectively. Then, the probability of a "start" information t^s of the time period using the kernel density function of $T^s_{text}(d)$ is:

$$P(t^s|T^s_{text}(d)) = \hat{f}\left(t^s; h\right) = \frac{1}{m(d)} \sum_{i=1}^{m(d)} K_h\left(t^s - \tau^s_i\right) \tag{3.5}$$

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

$$K_h\left(x\right) = \frac{1}{\sqrt{2\pi}\cdot h} exp\left(-\frac{x^2}{2\cdot h}\right) \tag{3.6}$$

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. 3.5. $S^{temp\_e}_{text}(d)$ 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:

$$S^{temp}_{text}(d) = P(T(Q)|T_{text}(d)) = \frac{1}{2} \cdot (S^{temp\_s}_{text}(d) + S^{temp\_e}_{text}(d)) \tag{3.7}$$

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. 3.5. 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. 3.7 but considering its weight which indicates the importance:

$$\begin{aligned} &P((t^s_i(Q), t^e_i(Q))|T_{text}(d)) \\ &= \frac{1}{2} \cdot (P(t^s_i(Q)|T^s_{text}(d)) + P(t^e_i(Q)|T^e_{text}(d))) \cdot w(t^s_i(Q), t^e_i(Q)) \end{aligned} \tag{3.8}$$

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

$$S^{temp}_{text}(d) = P(T(Q)|T_{text}(d)) = \frac{1}{m}\sum_{i=1}^{m} P((t^s_i(Q), t^e_i(Q))|T_{text}(d)) \tag{3.9}$$

IV. 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:

$$S^{temp}(d) = \tfrac{1}{2} \cdot (S^{temp'}_{pub}(d) + S^{temp'}_{text}(d)) \tag{3.10}$$

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:

$$S^{rel}(d) = \tfrac{BM25(d)}{MAX\_BM25} \tag{3.11}$$

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

$$S(d) = (1 - \alpha(Q)) \cdot S^{rel}(d) + \alpha(Q) \cdot S^{temp}(d) \tag{3.12}$$

$\alpha(Q)$ 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 Figure 3.2), 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 Figure 3.2) 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:

$$\alpha(Q) = \begin{cases} 0.0 & when\ burst\_num = 0 \\ ce^{-(1-\frac{1}{burst\_num})} & elsewhere \end{cases} \tag{3.13}$$

$\alpha(Q)$ 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.

3.2.3 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 [141] 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 [28] or with versions derived on the basis of BERT [81, 137]. We use here BiDAF for easy comparison with DrQA, whose reader component performance is similar although a little better than the one of BiDAF.

3.3 Experiments

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

3.3.1 Experimental Setting

Document Archive and Test Set. As previously mentioned, NYT corpus [136] is used as the underlying news archive, 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 [17, 65]. 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, no studies, as well as no available datasets that can help to design a question answering system over temporal news collections have been proposed so far. 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 1,000 questions (500 of explicitly time-scoped questions, 500 of implicitly time-scoped questions), paired with their answers.* Note that 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 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 used resources is

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

Table 3.2. Resources used for constructing the test set

Resources	Number of explicitly time-scoped questions	Number of explicitly time-scoped questions
history quizzes from funtrivia[†]	235	204
history quizzes from quizwise[‡]	67	75
Wikipedia pages	140	143
Questions from datasets ([126],[56])	58	78
Total	500	500

shown in Table 3.2. Table 3.1 gives a few example questions.

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 [22]: DrQA, a robust ODQA system which is composed of a Document Retriever module and a Document Reader module.

2. QA-NLM-U [64]: QA system for answering implicitly time-scoped questions that uses the best re-ranking method in [64] (as described in Section 2.2.1), while the Document Retriever Module and Document Reader Module are the same as the modules of QANA.

3. QA-No-Re-ranking [141]: 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., Eq. 3.1 and Eq. 3.3).

5. QANA-TempCont: QANA version that only uses temporal information embedded in document content for Time-Aware Re-ranking Module (i.e., Eq. 3.7 and Eq. 3.9).

6. QANA: QANA with complete Time-Aware Re-ranking Module.

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

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

Table 3.3. 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 [22]	13.20	17.60	18.00	23.73	21.20	26.51	21.00	26.85
QA-No-Re-ranking [141]	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**

3.3.2 Experimental Results

Results of Explicitly Time-scoped Questions. We use exact match (EM) and F1 score as our evaluation metrics. Table 3.3 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, that 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 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-ranking module to utilize the temporal 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 know-

ledge source, and the results are shown in Table 3.4. We can clearly observe that when answering explicitly time-scoped questions, DrQA-Wiki is always better than DrQA-NYT, especially 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 3.5. 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 cases, as it takes both the importance and bursts' number 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 Figure 3.4, 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 Figure 3.4). 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.

Results of Implicitly Time-scoped Questions. Table 3.6 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 com-

Table 3.4. 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 [22]	**18.80**	22.92	22.60	27.49	24.60	29.35	25.40	30.25
DrQA-NYT [22]	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**

Table 3.5. 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 [141]	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 [141]	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

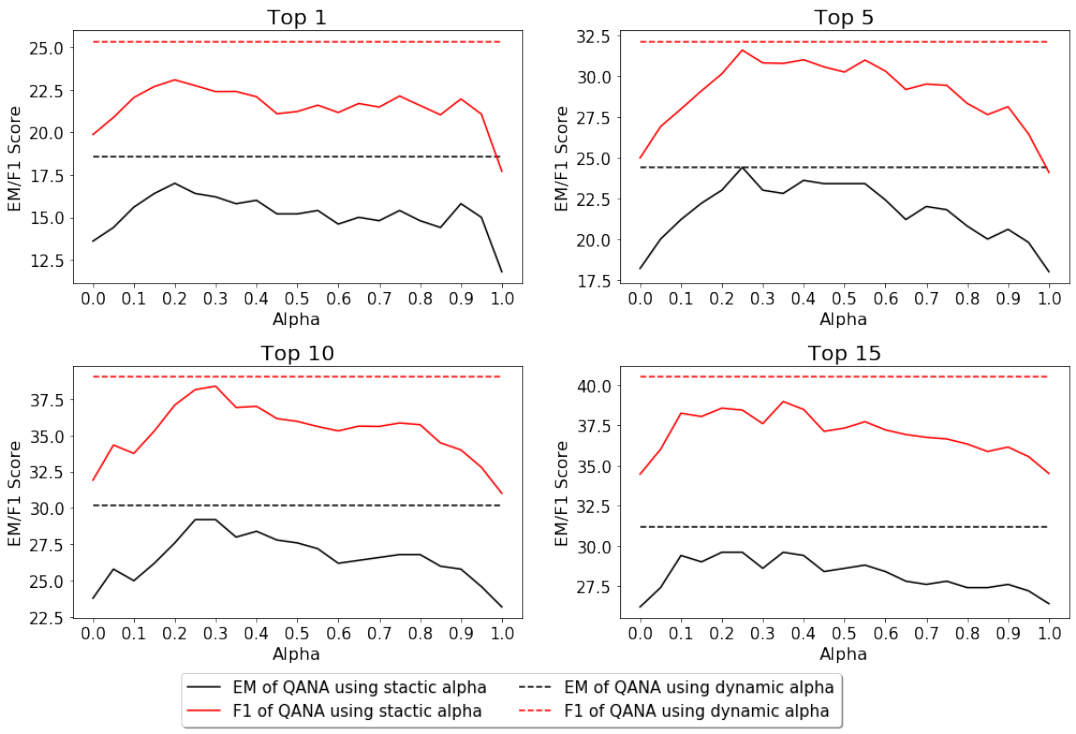

Figure 3.4. QANA Performance with different static alpha values vs. one with dynamic alpha for different top-N results over explicitly time-scoped questions.

Table 3.6. 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 [22]	19.40	25.65	25.40	32.14	26.20	34.13	27.00	35.86
QA-NLM-U [64]	20.40	28.34	25.00	33.50	30.40	38.58	31.40	39.95
QA-No-Re-ranking [141]	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**

Table 3.7. 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 [22]	**21.20**	25.76	22.00	26.30	23.00	26.97	24.40	28.70
DrQA-NYT [22]	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**

ponent 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.

We next examine the performance of DrQA when using Wikipedia articles as its

Table 3.8. 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 [141]	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 [141]	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

knowledge source, whose result is shown in Table 3.7. 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 3.8, 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 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 Figure 3.5, 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.

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 3.9, and we compare these two models with QA-No-Re-ranking and QANA in answering

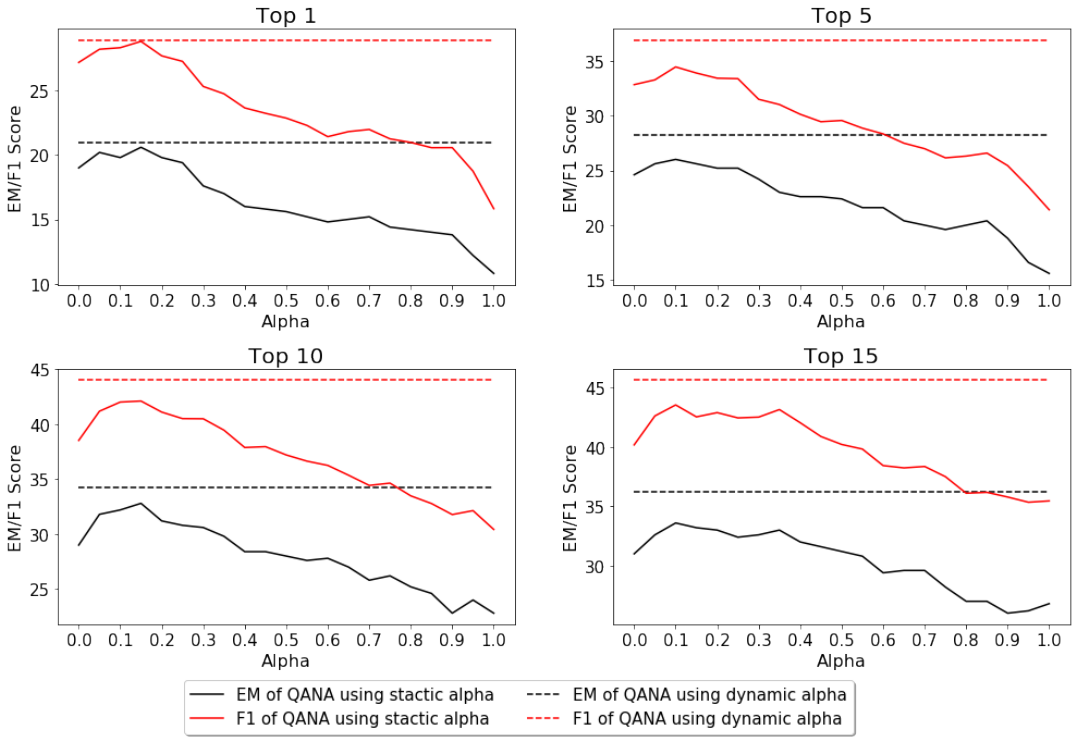

Figure 3.5. QANA Performance with different static alpha values vs. one with dynamic alpha for different top-N results over implicitly time-scoped questions.

Table 3.9. 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 [64]	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 [141]	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**

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.

3.4 Summary

In this chapter, we investigate a novel research task focused on answering event-related questions over temporal news collections. We introduce an effective ODQA system called ODQA for answering questions on news archives. 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 chapter 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. In addition, in Chapter 4 and Chapter 5, we propose different effective models, that can enhance the QANA system by improving the question's time scope estimation method for the implicitly time-scoped questions.

CHAPTER 4

Exploiting Temporal Information in Event Occurrence Time Estimation

In this chapter, we focus on the event occurrence time estimation task, which has many applications in IR, QA, general document understanding and downstream NLP tasks. The proposed TEP-Trans model is able to estimate the time at different temporal granularities (e.g., day, week, month, or year). As evidenced through extensive experiments, TEP-Trans model outperforms the existing methods by a large margin at all granularities. In addition, we demonstrate that it can further improve the performance of QANA model, the work introduced in Chapter 3, in answering implicitly time-scoped questions about the past events.

4.1 Introduction

Time can be leveraged to organize and search relevant information in news texts, aiding in exploration of the causalities, developments, and effects of the events, etc. Event occurrence time, indicating when an event took place, constitutes then one of the most significant type of information about the event. In recent years, utilizing event-related information in IR and NLP tasks has attracted in-

creasing attention. Event time information in particular has been exploited in various diverse tasks, such as search results diversification [11, 46, 144], multi-document summarization [108], timeline construction [43, 86, 147, 174], named entity disambiguation [1] and historical event ordering [50].

This research topic addresses the problem of event occurrence time estimation defined as follows: *given a short description of an event and a chosen temporal granularity, the task is to estimate event's occurrence time at the specified granularity using a temporal document collection as the underlying knowledge source.* For example, given the event-describing sentence "A bombing of a Superferry by Abu Sayyaf in the Philippines killed 116" and month granularity, an effective model should infer its occurrence time, which is "2004-02" based on querying a relevant news archive. Note that the task could be also regarded as a variant of question answering with a particular objective to answer questions about when the events occurred. Though we emphasize that a successful model should infer the correct time even if it is not explicitly mentioned in any available document.

In this chapter, we propose a model called TEP-Trans (Temporal Event Profiling Transformer-based model) which is a Transformer-based neural network to approach the task, by exploiting both temporal and textual information from different angles, represented by multivariate time series. We are the first to address the time estimation task by applying the ideas of multivariate time series analysis and the Transformer approach [158], which is a deep learning architecture that leverages attention mechanism and has been proved to be especially effective in natural language processing. We note that the performance of the existing methods is unsatisfactory for the temporal event profiling task, especially at fine-grained granularities (e.g., day, week), as they are either statistical approaches [45, 57, 64], or are designed over synchronic document collections (e.g., Wikipedia) [26, 50] that are incapable of utilizing document timestamp information in contrast to methods based on temporal collections of news articles. We then utilize data directly from temporal document collection and propose a neural network based solution for extracting correct temporal signals.

In the experiments, we use the New York Times Annotated Corpus (NYT corpus) [136] as the underlying data source, which contains over 1.8 million news articles published between January 1, 1987 and June 19, 2007. We construct a large dataset containing 22,398 short event descriptions, paired with their occur-

Table 4.1. Examples of event descriptions and their occurrence time in our dataset

No.	Description	Time
1	An official news agency in the Soviet Union reports the landing of a UFO in Voronezh.	1989-10-09
2	Antonov-26 plane crashes at Gyumri,Armenia, 36 killed.	1993-12-26
3	FBI agent Earl Pitts pleads guilty to selling secrets to Russia.	1997-02-28
4	President of Pakistan Pervez Musharaf narrowly escaped an assassination attempt.	2003-12-14
5	George Bell is 1st Blue Jay ever to win the AL MVP.	1987-11-17
6	Toru Takemitsu's "Archipelago" premieres in Aldeburgh England.	1993-06-18
7	Will Clark, National League's Most Valuable Player signs a $15 million four-year contract with San Francisco Giants.	1990-01-22

rence dates which fall into the time frame of the NYT corpus.* Table 4.1 presents example records in our dataset. Note that some events in our dataset, especially the less well-known ones, are not mentioned in Wikipedia or are only reported with temporal information of crude granularity (e.g., year). For example, Wikipedia does not contain any information about event #6 and event #7 in Table 4.1, and it records only year information of event #5. This necessitates using other resources such as large-scale news archives in order to enable temporal event profiling of lesser-known or minor events, as well as to assure providing fine granularity temporal information. The experimental results show that our proposed model outperforms other models by a large margin at all temporal granularities.

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

- We propose a novel TEP-Trans model based on Transformer architecture and multivariate time series analysis which is able to estimate the event occurrence time at different temporal granularities based on a long-term news archive as the underlying knowledge source.†
- We construct a large dataset of past events and perform extensive experiments to prove the effectiveness of our model.

*In the third research topic, we also use this dataset for testing the models, and is named as EventTime in Section 5.3.2

†The code and the dataset are available at `https://github.com/WangJiexin/Temporal-Event-Profiling`.

- We show that our model can be successfully applied on the downstream IR/NLP tasks such as ODQA task to further improve their performance.

The remainder of this chapter is structured as follows.[‡] In Section 4.2, we introduce our method. Section 4.3 describes experimental settings, while Section 4.4 provides experimental results. In Section 4.5 we demonstrate how the proposed approach can improve other tasks on the example of QANA, the model we introduced in Chapter 3. Finally, we conclude the chapter in Section 4.6.

4.2 Approach

As already mentioned, the task is to estimate the event occurrence time based on an underlying news archive. For each event description, our approach first retrieves the relevant news articles, and then uses both their temporal and textual information. The temporal and textual signals are represented by four univariate time series, the lengths of which are equal to the length of the time frame of the used temporal document collection. These four time series are then aggregated to form a multivariate time series to be utilized as an input by the proposed TEP-Trans model for predicting the occurrence time. The notations used to explain our approach are listed in Table 4.2. Below we describe the steps of our method.

4.2.1 Retrieving Relevant News Articles

The first step is to identify keywords for each event description e and use them to retrieve relevant news articles from the news article archive D. We choose Yake![§] [18] as our keyword extraction method, which is a state-of-the-art unsupervised approach that rests on text statistical features extracted from single documents to select the most important keywords. Next, the query, which is composed of the extracted keywords, is sent to the ElasticSearch[¶] installation which finally returns the top k relevant documents ranked by BM25.

[‡]Note that the related work of temporal information estimation is discussed in Section 2.3.
[§]Yake! is available in the PKE tookit: `https://github.com/boudinfl/pke`
[¶]`https://www.elastic.co/`

Table 4.2. List of notations

Notations	Descriptions
e	A given event description
d, D	A news article and the underlying news archive
l	Length of the time series
$t_{pub}(d)$	Timestamp, i.e., publication date of d
$BM25(d), Rel(d)$	The BM25 and relevance score of d
$top(k)$	The set of top k relevant articles
$T(top(k))$	The set of extracted time intervals of top k articles
$S(top(k))$	The set of extracted sentences which contain extracted time intervals of top k articles
$\boldsymbol{e}, \boldsymbol{s}$	The encodings of e and a sentence s
$Sim(e, s)$	The similarity between e and a sentence s
$\mathbf{X}_{temp}^{pub}, \mathbf{X}_{temp}^{cont}$	Time series from temporal signals (publication and content)
$\mathbf{X}_{text}^{doc}, \mathbf{X}_{text}^{sent}$	Time series from textual information (document and sentence)
$\boldsymbol{\chi}$	The multivariate time series

4.2.2 Obtaining Time Series from Temporal Information

The second step is to extract the temporal information from timestamp and from the content of each retrieved document d, which are then aggregated and utilized to construct two univariate time series of length l:

$$\mathbf{X}_{temp}^{pub} = \left\{ X_{temp,1}^{pub}, X_{temp,2}^{pub}, ..., X_{temp,l}^{pub} \right\} \tag{4.1}$$

$$\mathbf{X}_{temp}^{cont} = \left\{ X_{temp,1}^{cont}, X_{temp,2}^{cont}, ..., X_{temp,l}^{cont} \right\} \tag{4.2}$$

$\mathbf{X}_{temp}^{pub}$ denotes the publication date time series and $\mathbf{X}_{temp}^{cont}$ denotes the content date time series. As previously mentioned, l equals to the length of the time frame of the news archive D, and its value naturally depends on the specified temporal granularity. In the experiments, we use the NYT corpus, which contains news articles published from January 1, 1987 to June 19, 2007. When setting the month granularity, l equals to 246 time units, corresponding to the number of all months in the NYT corpus. For the case of the week granularity, l amounts to 1,069 units (weeks). Similarly, at year and day granularities, l equals to 21 units (years) and 7,475 units (days), respectively. For ease of exposition, we will

introduce our approach using month granularity in the remainder of this section.

Hence, the time unit i of the time series refers to the i-th month of the time frame of D. For example, $X^{pub}_{temp,1}$ represents the value of time series $\mathbf{X}^{pub}_{temp}$ at "January 1987", which is the first month of the NYT corpus. Below we discuss how to generate the above-mentioned two univariate time series.

Publication date time series. Based on the timestamps of the top k retrieved articles, the publication date time series X^{pub}_{temp} is created by counting the number of relevant documents published at each time unit i, denoted as $X^{pub}_{temp,i}$:

$$X^{pub}_{temp,i} = \sum_{\substack{d \in top(k) \\ s.t.\ t_{pub}(d) = time\ i}} 1 \tag{4.3}$$

X^{pub}_{temp} indicates the distribution of the top k relevant news articles over time. Previous studies of query temporal profiling [25, 57, 64], which focus on identifying time of interest of queries, show that this distribution can reflect useful information regarding temporal characteristics of events.

Content date time series. The extraction of content temporal information and the calculation of content date time series $\mathbf{X}^{cont}_{temp}$ are slightly more complex. We utilize this information as some news articles, like ones published after the event time, may still retrospectively relate to the event, providing useful information. Such news articles may be even published long time after the target event, and focus on other similar events or on the subsequent development or effect of the target event. For example, as we can see in Figure 4.1, the two top-relevant news articles retrieved from the NYT collection provide important extra details on the target event (the event is described at the top of the figure). More importantly, they also mention the correct event occurrence time despite having been published six and nine years after the event, respectively. Thus, as we can see based on these examples, the temporal information embedded in document content can be useful for our task.

To utilize the content temporal information, we first use SUTime [21], a popular tool for recognizing temporal expressions, to identify and extract sentences containing temporal expressions from the top k relevant documents. Then, we collect all the extracted temporal expressions and map them to the time interval with the "start" and "end" information. For example, at month granularity, "in May 1990" is mapped to ('1990-05', '1990-05'), and "from 1998 to 2002" is

Event Description: Sarin gas attack on the Tokyo subway: members of the Aum Shinrikyo religious cult release sarin gas on 5 subway trains in Tokyo, killing 13 and injuring 5,510.
Event Occurrence Time: 1995/03/20

Relevant news article 1:
Title: Seeing a Clash of Social Networks; A Japanese Writer Analyzes Terrorists and Their Victims
Publication Date: 2001/10/15
Content:
...they come from a non-Westerner, one, moreover, whose society experienced a terrifying chemical weapons attack by the Aum Shinrikyo religious sect on March 20, 1995...
For all of the pivotal qualities of the events of 1995 in Japan, from the Kobe earthquake to the sarin gas attack...

Relevant news article 2:
Title: After 8-Year Trial in Japan, Cultist Is Sentenced to Death
Publication Date: 2004/02/08
Content:
It took eight years to try Shoko Asahara, the former leader of the religious cult Aum Shinrikyo, on charges of masterminding the sarin gas attack in the Tokyo subway in 1995 that killed 12 people, injured 5,500 and shattered Japan's cherished self-image as one of the world's safest nations...
During the morning rush hour on March 20, 1995, Aum members released sarin into five crowded trains on three subway lines...

Figure 4.1. The examples of news articles (middle and bottom cell) that retrospectively refer to the target event (the description of this event is shown in the top cell).

mapped to ('1998-01', '2002-12').[‖] More fine-grained time expressions such as "March 5, 2005" and "June 14, 2001 to October 10, 2001" are mapped to ('2005-05', '2005-05') and to ('2001-06', '2001-10'), respectively, when assuming monthly granularity of time series to be constructed. For a temporal expression whose one boundary of the interval cannot be determined, we use the start or end date of the document collection to replace the missing "start" or "end" information. For example, "after March 2000" is normalized to ('2000-03', '2007-06') and "before October 1999" is converted to ('1987-01', '1999-10'). Finally, we retain those time

[‖]Similarly, for the case of day, week and year granularities, "from 1998 to 2002" is mapped to ('1998-01-01', '2002-12-31'), ('1998-W01', '2002-W53') and ('1998', '2002'), respectively.

intervals that fall into the time frame of the news archive.** We represent the set of such time expressions as $T(top(k))$ and the set of their corresponding sentences as $S(top(k))$, to be used later.

The calculation of the content date time series at time unit i, denoted as $X^{cont}_{temp,i}$, is as follows:

$$X^{cont}_{temp,i} = \sum_{\substack{t \in T(top(k)) \\ s.t.\ time\ i \in t}} \frac{1}{|t|} \tag{4.4}$$

We first loop over every collected time interval t, and then estimate the probability of generating each time point within that time interval. If the "start" and "end" information are the same (e.g., ('1999-03', '1999-03')), i.e., the temporal expression refers to one particular month i, the length of the time interval $|t|$ is 1, and the probability of generating this time unit i is 100%. Then the corresponding $X^{cont}_{temp,i}$ is incremented by 1. However, if the "start" and "end" date are not the same (i.e., the temporal expression covers multiple months), each corresponding $X^{cont}_{temp,i}$ is increased by the value equal to 1 divided by the length of the time interval $|t|$, which also denotes the probability of generating each time unit i of the time interval. For example, including the time expression that covers ('2000-01', '2000-05') results in $X^{cont}_{temp,i}$ of any i within the time interval ('2000-01', '2000-05') being incremented by $\frac{1}{5}$.

4.2.3 Obtaining Time Series from Textual Information

The third step is to utilize the textual information from the retrieved documents and from the sentences containing temporal expressions obtained in the previous step, which respectively reflect the relevance between event description and documents' content, and the relevance between event description and the extracted sentences containing temporal expressions. We thus introduce two other univariate time series of length l:

$$\mathbf{X}^{doc}_{text} = \left\{X^{doc}_{text,1}, X^{doc}_{text,2}, ..., X^{doc}_{text,l}\right\} \tag{4.5}$$

$$\mathbf{X}^{sent}_{text} = \left\{X^{sent}_{text,1}, X^{sent}_{text,2}, ..., X^{sent}_{text,l}\right\} \tag{4.6}$$

**Time expressions that refer to periods outside of the time frame of the used news collection are for simplicity discarded, although they could be utilized in the future extensions of the method.

$\mathbf{X}_{text}^{doc}$ denotes the document-to-event relevance time series and $\mathbf{X}_{text}^{sent}$ denotes the sentence-to-event similarity time series. We next introduce the computation of these two univariate time series.

Document-to-event relevance time series. As previously mentioned, the top k relevant documents are ranked by BM25. Their relevance scores are computed by dividing the BM25 scores by the maximum value:

$$Rel(d) = \frac{BM25(d)}{MAX\_BM25(top(k))} \tag{4.7}$$

The computation of the document-to-event relevance time series at time unit i, i.e., $X_{text,i}^{doc}$, is as follows:

$$X_{text,i}^{doc} = \sum_{\substack{d \in top(k) \\ s.t.\ t_{pub}(d) = time\ i}} Rel(d) \tag{4.8}$$

The calculation of $X_{text,i}^{doc}$ is similar to Eq. 4.3, but here we take the relevance between an event description and a document into account, so the timestamps of documents that are less relevant would play a lesser role.

Sentence-to-event similarity time series. Among the sentences in $S(top(k))$ that contain the extracted temporal expressions, those that are relevant to the events should be considered more important, (e.g., the sentences that contain temporal expressions in the two relevant news articles shown in Figure 4.1). Thus, for obtaining the last time series, we first calculate the relevance score between the event description and each sentence in $S(top(k))$, which indicates sentence importance and is measured by the cosine similarity between the event description encoding $\boldsymbol{e}$ and the sentence encoding $\boldsymbol{s}$. We utilize Sentence-BERT [129], a state-of-the-art neural network that can derive semantically meaningful sentence embeddings to encode the text. Then, the functions to compute the similarity score $Sim(e, s)$ and the sentence-to-event similarity time series at time unit i, denoted as $X_{text,i}^{sent}$, are as follows:

$$Sim(e, s) = cosine(\boldsymbol{e}, \boldsymbol{s}) \tag{4.9}$$

$$X_{text,i}^{sent} = \sum_{\substack{(t,s) \in (T(top(k)), S(top(k))) \\ s.t.\ time\ i \in t}} \frac{Sim(s, e)}{|t|} \tag{4.10}$$

The calculation is similar to the calculation of $X_{temp,i}^{cont}$. However, the corresponding sentence's relevance is taken into consideration, and the temporal information of the sentences that are less relevant to the event would be considered to a lesser extent.

4.2.4 Constructing Multivariate Time Series

The above-described four univariate time series of each event description are next standardized with mean of 0 and standard deviation of 1, and are aggregated to obtain a multivariate time series $\boldsymbol{\chi}$. The length of $\boldsymbol{\chi}$ equals to l and a slice of $\boldsymbol{\chi}$ at a time unit i is indicated as $\left\{X^{pub}_{temp,i}, X^{cont}_{temp,i}, X^{doc}_{text,i}, X^{sent}_{text,i}\right\}$. Therefore, with a batch size N, the input to the neural network has dimensions (N, M, l), where M equals to 4, and l is the length of the time series, that equals to the length of the time frame covered by the used news archive under the specified granularity.

4.2.5 TEP-Trans Model

In the proposed TEP-Trans network, the Transformer architecture [158], which has excellent expressive ability for representing sequence information, is introduced to model the features of the input multivariate time series. Transformer is a neural network architecture that leverages self-attention mechanism to process a sequence of data, and is mainly used in NLP tasks. We adopt this architecture to approach the occurrence time estimation problem. Equipped with the self-attention mechanism, Transformer can access any part of the history regardless of distance, making it potentially more suitable for focusing on significant time steps in the past and grasping the temporal features of the time series. Figure 4.2 shows the overall architecture of the proposed TEP-Trans for estimating the event occurrence time.

TEP-Trans model is comprised of two convolutional blocks, a multilayer Transformer encoder block, followed by an embedding averaging layer and a softmax layer. Each convolutional block consists of a 1-D convolutional layer with the same padding, followed by a batch normalization layer and a ReLU activation layer. The multilayer Transformer encoder block takes the tensor that combines the results obtained from the last CNN block and positional encodings[††] as input, and derives important features of the input time series. Note that the input tensor or output tensor of convolutional blocks with the same padding as well as the Transformer encoder block always have a dimension size equal to length l. We use C_2 to denote the output channels of the second convolutional layers, and the result of the Transformer encoder block has dimensions (l, N, C_2). Then, the embedding averaging layer transforms the dimensions to $(l, N, 1)$, by performing

[††]The functions to compute the positional encodings are derived from [158].

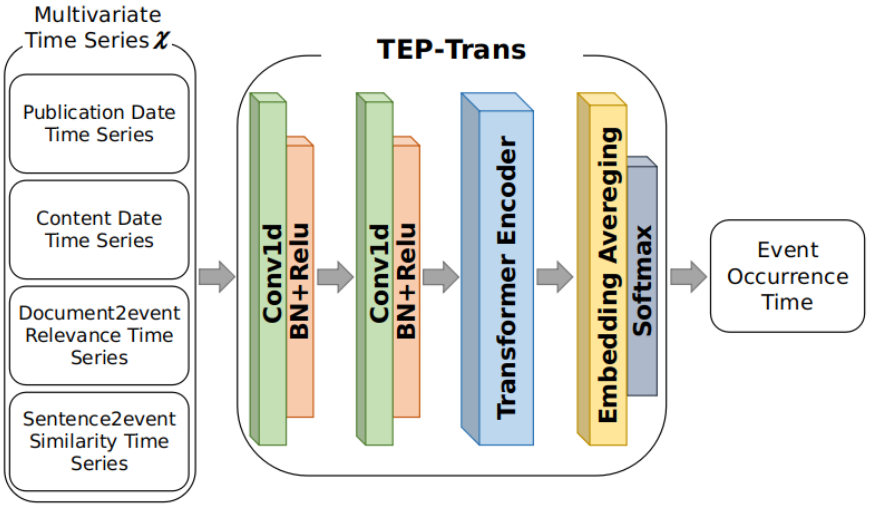

Figure 4.2. The TEP-Trans Model

the averaging across the last dimension's values. Finally, the result is transformed with dimension (N, l), and the estimated time is generated by the softmax layer. Note that we retain the tensor with length l and in the end, the features obtained from the Transformer block are fed into an embedding averaging layer instead of a fully connected layer, playing a similar role as global averaging pooling [94], which minimizes overfitting by largely reducing the number of parameters in the model. TEP-Trans model estimates the event occurrence time by exploiting the capability of convolutional layers for extracting useful knowledge and patterns, and then applying the Transformer for learning the internal representation of multivariate time series.

4.3 Experimental Setting

4.3.1 Document Archive and Event Dataset

As previously mentioned, the NYT corpus [136] 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 publication dates are contained in the corpus. We note that NYT has been often used for Temporal Information Retrieval researches [17, 65].

To the best of our knowledge, there is no available large dataset designed

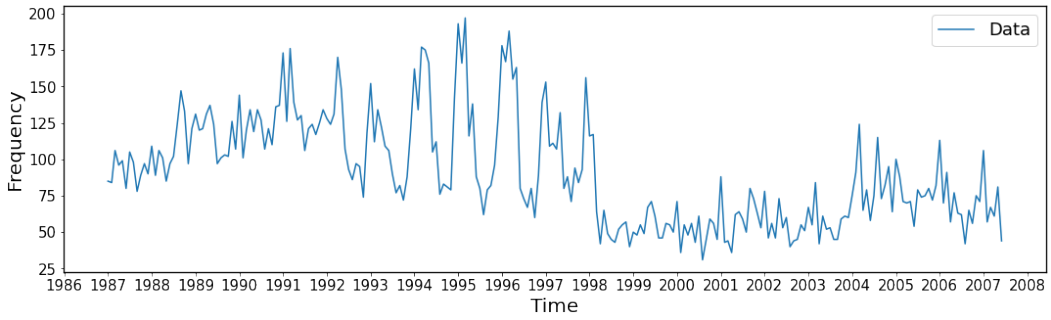

Figure 4.3. Distribution of event's occurrence time in the event dataset (month granularity)

specifically for estimating event occurrence time within the time frame of the NYT corpus.* Hence, we construct the dataset† and make sure that the occurrence times of the included events fall into the time frame of the NYT corpus. We create a dataset containing 22,398 event descriptions, paired with their event occurrence times, and we partition the whole dataset randomly into a training set (80%), a development set (10%), and a test set (10%).

The dataset has been constructed by crawling the descriptions and occurrence time of the events (ones between Jan 1, 1987 and Jun 19, 2007) from two resources: Wikipedia year pages.* and On This Day web pages† As the data extracted from these two resources sometimes contain records of the same events, we manually checked all the records that have the same event occurrence time and removed duplicates from the records that are on the same event. Figure 4.3 shows the monthly distribution of events in our dataset.‡

*Note that event extraction datasets such as ACE2005 or others are not applicable to our task as they require extracting event-related information from documents (actors, locations, dates) which is a different task than the event occurrence time prediction. Also, in their case, if the date information is to be delivered, it is always the one explicitly mentioned in text which does not require any prediction.

†The dataset is available at `https://github.com/WangJiexin/Temporal-Event-Profiling/tree/main/data/dataset`. Note also that this dataset is also used in the third research topic and is named as EventTime in Section 5.3.2

*`https://en.wikipedia.org/wiki/List_of_years`

†`https://www.onthisday.com/dates-by-year.php`

‡Note that our dataset contains a subset of events of the dataset used by [50], however their events are annotated with only the yearly granularity dates.

4.3.2 Hyperparameters of the Model

For each event description, up to 15 keywords are extracted using Yake! with 2-grams as the maximum n-gram size and other parameters set as default. The top 50 ($k = 50$) relevant news articles are then retrieved from the NYT corpus. In the training phase, we run 100 epochs with a batch size of 64, and we apply Adam optimizer with learning rate $1e - 3$. The hyperparameters of the TEP-Trans model that are used in the experiments are as follows: the kernel sizes and the strides of two 1-D convolutional layers with the same padding are set to 3 and 1, and the numbers of filters are set to 16 and 32, respectively. For the Transformer encoder layer, the number of layers, the number of heads, head dimension, and Transformer dropout are 3, 4, 200 and 0.2, respectively.

4.3.3 Evaluation Metrics

For the performance evaluation, we use: accuracy (ACC) and mean absolute error (MAE). The models are evaluated under these two metrics at day, week, month and year temporal granularities.

1) Accuracy (ACC): The percentage of the events whose occurrence time is correctly predicted.

2) Mean absolute error (MAE): The average of the absolute differences between the predicted time and the correct occurrence time, based on the specified granularity.[§]

4.3.4 Compared Methods

We test the following models:

1. **RG**: Random Guess. The event occurrence time is estimated by random guess, and the average of 1,000 random selections is used as the result.

2. **DPD**: Data Peak Date. This naive baseline is used as another lower-bound reference besides the random guess. It always returns the date of the peak of the data's distribution (i.e., peak occurrence time of the aggregated events of the

[§]For example, at day granularity and month granularity, if MAE is 1, the average temporal distance is 1 day and 1 month, respectively.

entire dataset) as the estimated result (e.g., under month granularity, DPD gives '1995-03', as can also be seen in Figure 4.3).

3. **BD** [159]: The burst detection based method which works such that given the temporal granularity, the occurrence time is estimated as the temporal value of the highest-scored peak within the largest burst of the publication date time series. The two parameters of BD, the window size and the cutoff factor, are set to 3 and 1.0, respectively.

4. **NLM** [63]: The best proposed method in [63], that directly uses the timestamps of the top 15 retrieved documents as the predicted time. When there is more than one predicted time point, we use the time point that contains the largest number of retrieved documents.

5. **MSSD**: The most similar sentence date method which works such that the event occurrence time is estimated as the time of the extracted sentence that has the largest similarity score with the event among sentences in $S(top(k))$.

6. **AA** [45]: The best proposed model in [45]. It mainly focuses on the temporal expressions extracted from the document content and regards the publication date as an additional content temporal information. k is set to 50.

7. **CNN-LSTM** [70]: The CNN-LSTM model has been often used to solve the multivariate time series prediction problems. We borrow this model to tackle our task which takes $\boldsymbol{\chi}$ as input.

8. **HEO-LSTM** [50]: The recently proposed variant of a method by [50] that was found by the authors to perform best and that estimates the event occurrence time by extracting relevant sentences from the Wikipedia, and applying a combination of task-specific and general-purpose feature embeddings for classification. As it is designed specifically to estimate the time at the year granularity, we compare this approach only at the year granularity. Note that HEO-LSTM is based on Wikipedia[¶] and cannot work on other collections.

9. **TEP-CNN**: Our proposed model without Transformer block, such that the CNN blocks are followed by embedding averaging layer and a softmax layer.

[¶]It needs to identify key entities of event descriptions, which are linked to the topics (i.e., titles) of the corresponding Wikipedia articles. For example, for the event description "The Sky Bridge is opened", the Wikipedia article "Sky Bridge" is used.

10. **TEP-Trans**: The proposed Transformer-based model.

For fair comparison, all the above methods (except for HEO-LSTM, which uses entities and actions identified by pre-defined rules to extract relevant Wikipedia sentences, and the first two naive methods, RG and DPD) use the same document retrieval approach (as described in Section 4.2.1) to retrieve their top k articles. Note also that RG and DPD are added only for determining the lower bound of the task to set a reference for better understanding of its difficulty.

4.4 Experimental Results

4.4.1 Main Results

Table 4.3 shows the performance of the tested models in estimating event occurrence time. We can see that the proposed TEP-Trans model, that takes $\boldsymbol{\chi}$ as input, surpasses other models in ACC and MAE at all temporal granularities. We first note the results of the two straightforward, naive methods, RG and DPD, which both exhibit very poor performance, indicating that the task is not easy to be solved. Among the next four non-deep learning models that do not use $\boldsymbol{\chi}$, MSSD achieves the best performance on ACC and MAE at all granularities. When comparing TEP-Trans with MSSD using ACC and MAE, at the granularity of month, the improvements are 38.39% and 18.34% and at the fine-grained granularity of day, the improvements are 72.84% and 2.58%, respectively. MSSD performs actually best among all the baseline models on day granularity, which reveals that the temporal sentences that have large similarities with event descriptions are helpful for estimating the occurrence time.

The remaining approaches are based on neural networks, and except for HEO-LSTM, all take $\boldsymbol{\chi}$ as input. The first model, CNN-LSTM, which is one of the most common neural network architectures applied in time series forecasting and prediction [69, 70, 95, 153], achieves relatively good performance on both metrics at year granularity. However, if the granularity turns to be finer, the performance of CNN-LSTM drops dramatically. The reason is that the output size of the last fully-connected layer, whose value equals to l (length of the time frame of the corpus at the chosen granularity) will also increase (e.g., l equals to 7,475 if day granularity is chosen). Thus, CNN-LSTM will overfit the training dataset and more data would be required to solve the problem. We next compare our

Table 4.3. Main results: Performance of different models at different granularities. Note that HEO-LSTM is designed specifically to estimate the time only at the year granularity

Model	**Day**		**Week**		**Month**		**Year**	
	ACC	**MAE**	**ACC**	**MAE**	**ACC**	**MAE**	**ACC**	**MAE**
RG	0.01	2482.12	0.08	355.25	0.40	81.57	4.77	6.91
DPD	0.04	2690.47	0.17	252.34	0.93	56.71	7.90	5.51
BD [159]	1.42	1418.26	14.01	215.80	18.75	49.70	27.09	4.37
NLM [63]	1.38	1300.34	15.53	194.16	21.87	45.85	33.52	3.80
AA [45]	6.02	1508.73	16.96	216.02	21.65	48.39	32.54	3.99
MSSD	9.50	1268.47	17.05	181.22	22.32	44.32	34.82	3.67
CNN-LSTM [70]	1.38	1382.38	7.49	174.26	23.30	37.04	37.54	3.21
HEO-LSTM [50]	-	-	-	-	-	-	15.58	4.81
TEP-CNN	8.39	1518.93	19.41	194.86	25.35	44.17	34.01	3.87
TEP-Trans	**16.42**	**1235.67**	**23.66**	**166.64**	**30.89**	**36.19**	**40.93**	**3.01**

proposed method with HEO-LSTM at the year granularity. Under the ACC and MAE measure, our method surpasses HEO-LSTM by a large margin since the improvements are 162.70% and 37.42%, indicating that using their method that relies on Wikipedia is less effective for estimating the event occurrence time. Moreover, except RG and DPD, the other baseline methods also perform much better than HEO-LSTM, revealing that news archives could be used as another useful knowledge source to infer the event times.

Finally, we compare TEP-Trans with TEP-CNN - the model without the Transformer block. We can see that TEP-CNN achieves the second best performance on ACC measure at week and month granularities. Therefore, CNN block can effectively extract important features of multivariate time series. Yet, by combining the Transformer block with powerful sequence pattern extraction capability, followed by the embedding averaging layer that helps to reduce overfitting problem, important features useful for the event time estimation can be identified. Interestingly, we can still see quite a large improvement at day granularity. Under the ACC and MAE metrics, the improvements are 95.70% and 18.64%, respectively.

Table 4.4. Performance of TEP-Trans model based on different input time series

Features	**Day**		**Week**		**Month**		**Year**	
	ACC	**MAE**	**ACC**	**MAE**	**ACC**	**MAE**	**ACC**	**MAE**
X_{temp}^{pub}	7.76	1563.20	13.25	216.92	17.63	48.04	30.26	3.80
X_{temp}^{cont}	6.60	1623.37	12.32	213.85	16.96	48.60	29.10	3.85
X_{text}^{doc}	8.52	1358.91	16.42	197.48	21.29	44.78	33.48	3.59
X_{text}^{sent}	9.86	1480.49	16.24	194.46	20.66	43.75	31.91	3.62
X_{temp}^{pub}, X_{temp}^{cont}	7.41	1578.50	15.31	211.88	19.28	46.16	30.53	3.73
X_{text}^{doc}, X_{text}^{sent}	13.34	1301.54	18.39	183.91	24.06	41.34	34.46	3.43
X_{temp}^{pub}, X_{text}^{doc}	11.02	1217.29	18.92	174.43	25.93	40.14	38.12	3.27
X_{temp}^{cont}, X_{text}^{sent}	12.18	1435.37	18.43	182.97	23.70	41.30	33.12	3.58
$\boldsymbol{\chi}$	**16.42**	**1235.67**	**23.66**	**166.64**	**30.89**	**36.19**	**40.93**	**3.01**

4.4.2 Input Ablation Study

We next conduct an ablation analysis on the input of the proposed TEP-Trans model. As shown in Table 4.4, the model using $\boldsymbol{\chi}$ as an input achieves the best result, indicating that all the features contribute to the performance of our model. When considering only univariate time series, the models using X_{text}^{doc} or X_{text}^{sent} always perform better than the ones using X_{temp}^{pub} or X_{temp}^{cont}. This suggests that it is useful to combine the relevance of documents or sentences with embedded temporal information to the event descriptions.

We then show the results of aggregating two univariate time series. We can see that in Table 4.4, except for $\left\{X_{temp}^{pub},\ X_{temp}^{cont}\right\}$ at day granularity, the models using one univariate time series achieve worse results on both metrics than models which aggregate the univariate time series with another one. For example, the model whose input is the multivariate time series obtained by aggregating time series of two types of textual information (indicated as $\left\{X_{text}^{doc}, X_{text}^{sent}\right\}$) performs better than the model using X_{text}^{doc} or X_{text}^{sent} only. In addition, we also note that our model achieves relatively good performance by taking $\left\{X_{temp}^{cont}, X_{text}^{sent}\right\}$ as input, which does not utilize the timestamp information. This suggests that our approach can also be applied over document collections without available timestamps.

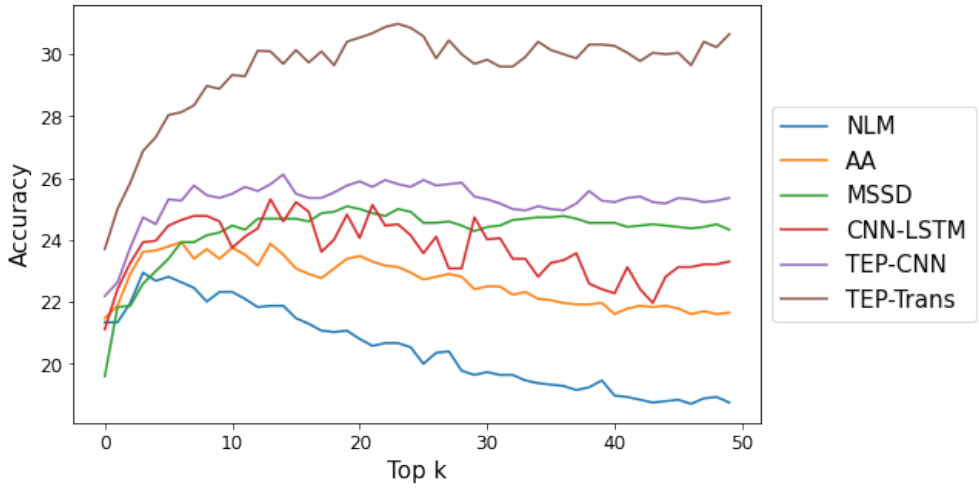

Figure 4.4. Performance of models with different top k at month granularity. Best viewed in color

4.4.3 Performance with Different Top k

Next, we investigate the effect of top k, that is the number of retrieved relevant documents used for constructing $\boldsymbol{\chi}$. Figure 4.4 plots the accuracy of different models with respect to k, which ranges from 1 to 50. First of all, TEP-Trans achieves the best result for all different top k and we can observe an initially growing trend of accuracy with larger k. The accuracy stabilizes around $k = 13$ and the TEP-Trans obtains its best accuracy level of 30.98% when $k = 24$. The TEP-CNN model whose last component comprises of an embedding averaging layer and a softmax layer exhibits similar tendency, and its best accuracy is 26.11% at $k = 15$. MSSD performance also reveals a similar trend along with the larger top k, which is reasonable since the event occurrence time is estimated as the time of the extracted sentence with the largest similarity score to the target event, so with the larger number k of top-relevant documents, a more similar and relevant sentence might be found. Unlike the above three methods, downward trends of accuracy of NLM, AA and CNN-LSTM can be observed when k is greater than a certain value (about 4, 14, 22, respectively), indicating that these models are incapable of filtering the noisy data well. Overall, we conclude that for the larger values of k, TEP-Trans can most effectively extract and filter information useful for event time estimation.

4.4.4 Analysis based on Event Characteristics

We next analyze the performance of our approach with respect to the event characteristics. In particular, we investigate how our model works based on the event description length and the shape of the temporal distribution of relevant documents. The former is represented by the number of words and the latter by the number of bursts in the publication date distribution over time,[‖] respectively. To test the effect of description length, the original test set of 2,240 event descriptions is first divided into two parts: 1,123 descriptions that have few words (less than or equal to 17) and 1,117 descriptions which are longer than 17 words. Note that when testing the effect of burstiness of the publication date time series, the number of bursts in the publication date distribution of events depends also on the specified granularity (coarser granularity results in less bursts in the distribution). Thus, for analyzing the impact of burstiness we divide the test set into two parts (few bursts and many bursts) that contain a similar number of records for each granularity.

Table 4.5 and Table 4.6 show the performance of our method based on the above-described data partitions. When considering the description length, we can see that TEP-Trans achieves better results on the event descriptions that have many words. The events that have longer descriptions are likely to retrieve documents that are more relevant to these events, which causes the obtained temporal or textual information to be more correct and precise. When considering the temporal distribution of the retrieved documents, the proposed model performs much better with the events that have only few bursts. It is more difficult to correctly estimate event time when the temporal distribution of relevant documents exhibits many bursts, since likely many other similar or related events occurred over time, which increases the difficulty of event date prediction.

4.4.5 Comparison with QA Systems

Recently, several works proposed to employ Question Answering (QA) [181] for a variety of NLP problems [34, 104]. For example, McCann et al. [104] transform 10 different NLP tasks including natural language inference, sentiment analysis and relation extraction, into a QA paradigm and propose MQAN model to tackle all

[‖] We use again the burst detection algorithm of [159] with the same parameters to detect and count the bursts.

Table 4.5. TEP-Trans results for events with few/many words

	Day		**Week**		**Month**		**Year**	
	ACC	**MAE**	**ACC**	**MAE**	**ACC**	**MAE**	**ACC**	**MAE**
few words	12.55	1312.46	18.96	174.02	25.73	38.16	36.50	3.01
many words	20.21	1158.48	28.22	159.23	35.88	34.20	45.14	3.00

Table 4.6. TEP-Trans results for events with few/many bursts

	Day		**Week**		**Month**		**Year**	
	ACC	**MAE**	**ACC**	**MAE**	**ACC**	**MAE**	**ACC**	**MAE**
few bursts	19.36	892.96	28.38	121.85	36.15	28.09	44.69	2.75
many bursts	10.85	1644.11	17.62	214.08	20.38	45.94	36.59	3.23

these tasks. In another work, Du and Cardie [34] have proposed a new paradigm for event extraction by formulating it as a QA task. Inspired by them, we test whether the event date estimation can be successfully solved using QA solutions.

We examine the performance of QA systems in the task of event occurrence time estimation, by first transferring the event descriptions to "when" questions, based on rule-based pattern matching (e.g., "Sarah Balabagan returns to the Philippines." is transferred to "When did Sarah Balabagan return to the Philippines?"). We then choose DrQA [22] for comparison, which is one of the most popular QA systems and is often used as a baseline in QA researches [85, 97, 164, 171]. Moreover, we examine DrQA models not only using the NYT corpus but also we investigate its performance when utilizing Wikipedia as the knowledge base. They are indicated as DrQA-NYT and DrQA-Wiki, respectively. Note that some answers returned by QA systems do not contain any temporal information and can not be compared with the ground truth (e.g., some numerical values "207", "100" which are not related to time, or other types of unrelated answers). Thus for ease of evaluation we only evaluate the models using accuracy metric. In addition, we test the models without week granularity since Wikipedia usually does not record week information of events.

As shown in Table 4.7, DrQA-Wiki performs much better than DrQA-NYT at the three granularites. We first found that the main reason is that the news articles often contain implicit temporal expressions, such as, "last month" or "yes-

Table 4.7. Comparison with QA Models

Models	**Day**	**Month**	**Year**
	ACC	**ACC**	**ACC**
DrQA-Wiki [22]	7.90	11.56	26.65
DrQA-NYT [22]	0.62	1.74	11.47
DrQA-NYT-TempRes [22]	3.97	7.41	19.28
TEP-Trans	**16.42**	**30.89**	**40.93**

Table 4.8. Examples of event descriptions that are wrongly estimated by TEP-Trans, based on month granularity

No.	**Description**	**Occurrence Time**	**Estimated Time**
1	William Anthony Odom, North Carolina 15-year-old, accidentally hangs himself staging a gallows scene at a Halloween party.	1990-10	1996-10
2	The flu outbreak in Britain puts pressure on NHS.	2000-01	2005-11
3	Turin, Italy, is awarded the 2006 Winter Olympics.	1999-06	2006-02

terday", which might be returned as answers by DrQA-NYT. We then decided to resolve such implicit temporal expressions by using the inferred time, which is the timestamp information of the corresponding documents in order to improve the performance. We indicate this new system as DrQA-NYT-TempRes, and, as we can see, its performance is now closer to the one of DrQA-Wiki. However, a significant improvement on accuracy can be observed when comparing TEP-Trans with DrQA-Wiki and DrQA-NYT-TempRes at three granularites, indicating that common QA systems are incapable of answering "when" questions well. It also suggests that our method could serve as a fallback of a QA system when the answer is not explicitly given in the text or the answer is of coarse granularity.

4.4.6 Error Analysis

We also analyzed events for which our method has not produced correct results and we show some examples in Table 4.8. We found out that such events are usually not reported in the NYT archive, are periodical or recurring events, or

are ones that include information about other popular events. For example, TEP-Trans model was not able to infer the occurrence time of event #1 in Table 4.8 since it is not reported in the NYT archive (although we found that it was actually reported in the LA Times archive.). The model could not correctly estimate the time of event #2 because similar events recurred multiple times and the description of event #2 is not precise enough. For the event #3, TEP-Trans model wrongly estimated the time as Feb. 2006 because most relevant articles are about the Winter Olympics held at that time.

4.5 Applications

Finally, we look at how the proposed approach can be utilized in downstream tasks and we demonstrate its usefulness on one such task. There are quite many potential applications for temporal profiling of event mentions. Improving relevance estimation to enhance search within news archives or temporal diversification of search results [11, 46, 144], supporting entity extraction [1, 131], improving event mention extraction** [148], enhancing timeline generation†† [43, 86, 147, 174] or question answering in long-term temporal news collections like QANA model [161, 162] that introduced in Chapter 3, is an immediate example.

4.5.1 Application for Question Answering

In this sub-section, we test our approach to see if it can improve effectiveness of answering diverse user questions in news archive. In particular, we use QANA [162] to answer implicitly time-scoped event-related questions, the questions without any temporal expressions. An important step in the system pipeline is the question time scope estimation aimed to gauge the possible time periods of the mentioned events based on analyzing the distribution of the retrieved documents. For example, since there is no temporal expression in the question: "Which party, led by Buthelezi, threatened to boycott the South African elections?", this step

**Judging if two text spans are about the same event can be improved since not only text similarity can be considered but also overlap of their estimated temporal profiles.

††For generating timelines some approaches use explicit temporal expressions mentioned in news [147]. With our method one could find implicit references to news events as there is no need for any explicit date to be present in such references.

Table 4.9. Performance of different models in QA task

Model	Top 1		Top 5		Top 10		Top 15	
	EM	**F1**	**EM**	**F1**	**EM**	**F1**	**EM**	**F1**
QANA [162]	21.00	28.90	28.20	36.85	34.20	44.01	36.20	45.63
QANA + TEP-Trans	**23.00**	**30.89**	**29.60**	**38.17**	**35.40**	**45.49**	**38.00**	**48.35**

requires QANA to estimate the implicit date of the event mentioned in this question (which is "1993-08" under monthly granularity). We replace this step with our proposed approach, and the new system is indicated as QANA + TEP-Trans. We test both the systems on the dataset [162] composed of 500 questions that do not contain any temporal expressions using the NYT collection. This dataset has been created by merging data from various kinds of resources such as TempQuestions [56], SQuAD 1.1 [126] and questions from several history quiz websites. The results of the two systems are presented in Table 4.9. We can see that QANA + TEP-Trans system equipped with our proposed event time estimation approach outperforms the original QANA system [162] for all the different ranges of the top N search results used. When considering the top 1 and top 15 documents, the improvement is in the range of 9.50% to 4.97%, and from 6.88% to 5.96% on Exact Match (EM) and F1 metrics, respectively. As demonstrated in this example, the proposed approach can be utilized as a building block for downstream tasks to further improve their performance.

4.6 Summary

In this chapter we present an effective TEP-Trans model for estimating the event occurrence time. We are the first to address this task by applying the ideas of multivariate time series analysis and the Transformer architecture, which altogether result in promising performance. The proposed approach is capable of modeling useful features of the input multivariate time series and achieves state-of-the-art results at all the temporal granularities. In addition, unlike most of the existing methods which estimate the occurrence time based on temporal information from timestamp or content signals, or which are designed over synchronic document collections (e.g., Wikipedia), our approach addresses the problem by jointly

utilizing two types of temporal information and two types of textual information. Through the experiments we learn that these four types of information contribute altogether to the performance of our model, as demonstrated in the experiments.

In the future, we will explore the inter-relations between the retrieved documents that were published at different time units in order to capture the features reflecting the temporal development of events, as such data could be another useful signal for event date prediction. We will also apply the proposed approach to other IR and NLP tasks besides open-domain question answering.

CHAPTER 5

Exploiting Temporal Information in Constructing Time-aware Language Representation

In the previous Chapter 4, we introduce TEP-Trans, which achieves SOTA results at all the temporal granularities on event occurrence time estimation task. However, we argue that it also has some limitations like can only estimate the time within the time frame of the underlying knowledge source, which is between 1987 and 2007 in our case, or the input construction, that the multivariate time series is created through several complicated steps, such as sentence similarity computation, which requires rather considerable time or effort. In this chapter, we propose a novel language representation model called TimeBERT, which can be easily and effectively applied in various time-related tasks. In particular, TimeBERT is trained on a temporal collection of news articles via two new pre-training tasks, which harness two distinct temporal signals to construct time-aware language representation. Furthermore, we show that TimeBERT can surpass TEP-Trans in estimating event occurrence time at some temporal granularities and can also improve the performance of QANA system in answering event-related questions.

5.1 Introduction

Temporal signals constitute one of the most significant features for many types of text documents, for example, news articles or biographies. They can be leveraged to organize and search for relevant information, aiding in exploration of the causalities, developments, and ramifications of the events, as well as can be helpful for a range of NLP tasks. Indeed, utilizing temporal signals in information retrieval has received considerable attention lately. For example, researchers have addressed time sensitive queries in information retrieval [17, 61] leading to the formation of a subset of Information Retrieval area called Temporal Information Retrieval in which both query and document temporal aspects are of key concern. Event detection and ordering [27, 148], timeline summarization [3, 19, 102, 147, 154], event occurrence time prediction [163], temporal clustering and information retrieval [2, 15, 17], question answering [117, 161] and named entity recognition [1, 131] are other example tasks where utilizing temporal information proved beneficial.

Pre-trained transformer-based [158] language models such as BERT [28], XL-Net [172], GPT [14, 124] have recently achieved impressive performance on a variety of downstream natural language processing tasks, and have been commonly utilized for representing, evaluating or generating text. Despite their huge success, they still however suffer from difficulty in capturing important information in domain specific scenarios, since in general, their training is not adapted to the specificities of documents in particular domains, as well as they are typically carried on large-scale general corpora (e.g., English Wikipedia). For example, such models are incapable of utilizing temporal signals like document timestamp, despite temporal information being of key importance for many tasks such as ones that involve processing content of news articles.

In this chapter, we introduce a novel, pre-trained language model called TimeBERT, which is trained on a temporal news collection by exploiting their two key temporal aspects. The experimental results show that TimeBERT could simultaneously utilize both distinct temporal aspects in an effective way, as it consistently outperforms BERT and other existing pre-trained models, with substantial gains on different downstream NLP tasks or applications for which time is of importance.

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

1. We investigate the effectiveness of incorporating temporal information into pre-trained language models using different pre-training tasks, and we demonstrate that injecting such information via specially designed time-oriented pre-training tasks can benefit various downstream time-related tasks.

2. We propose a novel pre-trained language representation model called TimeBERT, which is trained through two new pre-training tasks that involve two kinds of temporal aspects. To our best knowledge, this is the first work to investigate both types of temporal information (timestamp and content time signals in news articles) when constructing pre-trained language models.

3. We conduct extensive experiments on diverse time-related tasks that involve the two temporal dimensions of documents or queries. The results demonstrate that TimeBERT achieves a new SOTA performance, and thus has capability to be successfully applied in many applications for which time is crucial.

The remainder of chapter is structured as follows.* In Section 5.2, we introduce the details of our method. Section 5.3 describes experimental settings, while Section 5.4 provides experimental results. In Section 5.5, we demonstrate how the proposed language model can improve other downstream tasks or applications on the example of QANA system. Finally, Section 5.6 concludes the chapter.

5.2 Approach

In this section, we present TimeBERT, the proposed pre-trained language representation model based on transformer encoder [158]. As mentioned before, the model is trained on a temporal collection of news articles via two new pre-training tasks, which involve document timestamp and content time (i.e., the temporal expressions embedded in the content) to construct time-aware language representation. Our approach is inspired by BERT model [28], but distinguishes itself from it in three ways. Firstly, it is trained on a news document collection rather than on synchronic document collections (e.g., English Wikipedia or static collection of news), and thus the timestamp information which is of key importance in our collection can be readily obtained and used. Note that even if some pre-trained language models use news datasets for training (e.g., CC-NEWS [111], which is

*Note also that the related work of pre-trained language models is discussed in Section 2.4.

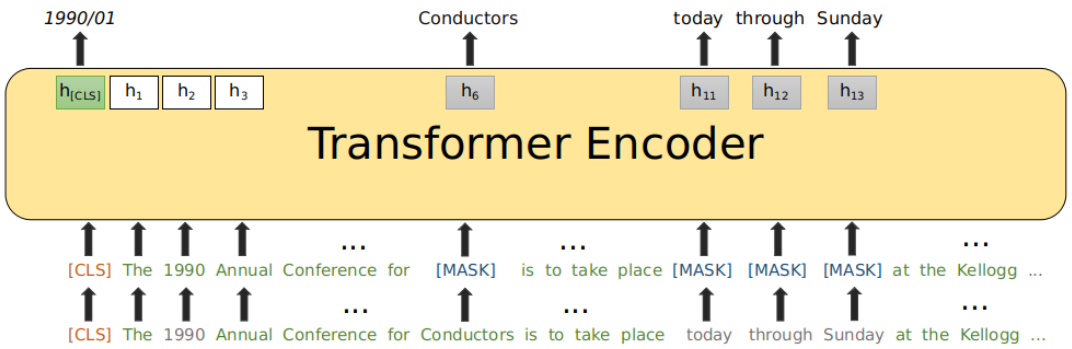

Figure 5.1. An illustration of TimeBERT training, which includes the TAMLM and DTP tasks.

used by RoBERTa [98]), they still utilize the same training technique as on the synchronic document collections, which essentially ignores the temporal aspects of documents. Secondly, we use a different masking scheme, *time-aware masked language modeling* (TAMLM) to randomly mask spans of temporal information first rather than just randomly sample tokens from the input. This explicitly forces the model to incorporate temporal information embedded in the document content. Finally, we replace the next sentence prediction with an auxiliary objective, *document timestamp prediction* (DTP), which also lets the model incorporate timestamp temporal information while training. As a document timestamp prediction is a sub-task of time prediction, this task is also demonstrated to aid in improving the performance of other time-related tasks. Figure 5.1[†] illustrates the two proposed pre-training objectives of TimeBERT. TimeBERT is jointly trained on the two proposed tasks of TAMLM and DTP, with two different additional layers based on the output of its transformer network. In addition, we also propose another third pre-training task that makes use of content temporal information, same as TAMLM task, *temporal information replacement* (TIR), which is found to achieve relatively good performance in some downstream tasks. Figure 5.2[‡] gives a simple example of the replacement procedure in TIR task. All these pre-training objectives use cross entropy as the loss function, and are described in detail in the following sub-sections.

[†]The selected example is the news article published in The New York Times in 1990/01/05, with title "Conductors' Conference To Include Szell Tribute".

[‡]Note that in TIR task, 50% of the time expressions will not be replaced, hence in Figure 5.2, "today through Sunday" is left as it is in our example.

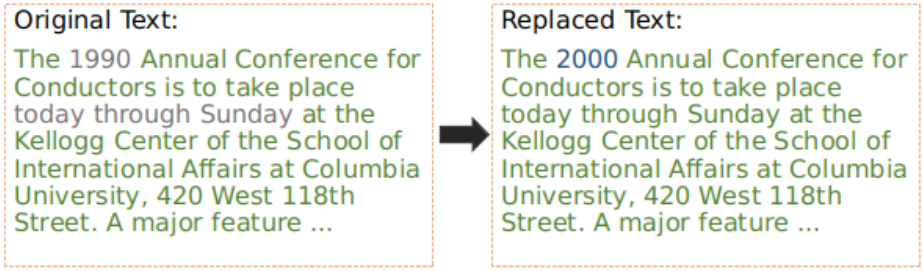

Figure 5.2. Example of the replacement procedure in TIR task

5.2.1 Time-aware masked language modeling (TAMLM)

As mentioned before, the first pre-training objective, time-aware masked language modeling (TAMLM), explicitly introduces content time (the temporal information embedded in the document content) during pre-training. This kind of temporal information could be used in exploring the developments of events and the causal relations between events can be understood by analyzing the relations between different content temporal information. For example, temporal expressions in news content have been used for constructing timeline summaries over temporal news collections [174].

Suppose there is a token sequence $X = (x_1, x_2, ..., x_n)$, where x_i $(1 \leq i \leq n)$ indicates a token in the vocabulary. Firstly, the temporal expressions in the entire document content are recognized using spaCy[§] (as indicated by the gray font in the bottom in Figure 5.1). The recognized temporal expressions' set is denoted by $T = (t_1, t_2, ..., t_m)$, where t_i $(1 \leq i \leq m)$ indicates a particular temporal expression found in a document. Secondly, unlike in the case of BERT where 15% of the tokens are randomly sampled in direct way, we first focus on the extracted temporal expressions. Certain percentage (denoted by α, where $(0.0 \leq \alpha \leq 1.0)$) of the temporal expressions in T are then randomly sampled first (e.g., "today through Sunday" in Figure 5.1). Thirdly, we continuously randomly sample other tokens which are not the tokens in T, until 15% of the tokens in total are sampled and masked (like "Conductors" is masked and "1990" is not allowed to be masked in the same example). Finally, same as in BERT, 80% of the sampled tokens are replaced with [MASK], 10% with random tokens, and 10% with the original tokens.

[§]https://spacy.io/

Through this masking scheme, we encourage the model to be more focused on the content temporal information and the relations between different temporal expressions. Actually, the model is trained to predict the tokens of masked temporal expressions not only from the text, but also from the temporal expressions that are not masked. The effect of different temporal masking ratio α is analyzed in Section 5.4.3.

5.2.2 Document Timestamp Prediction (DTP)

The second pre-training objective, document timestamp prediction (DTP), incorporates document timestamp information during pre-training. In news article collections, each article is usually annotated with a timestamp, corresponding to the date when it was published. Timestamp temporal information, which can help users locate the news reports published in specific periods quickly, has been widely utilized in temporal information retrieval for estimating document relevance scores [64, 91, 162].

Similar to BERT, the [CLS] token is inserted at the beginning of the input and its representation, $h_{[CLS]}$, is utilized to provide the contextual representation of the entire token sequence. However, rather than performing binary classification of the next sentence prediction, we utilize this token to predict the temporal information of document timestamp, as shown in Figure 5.1. Temporal granularity of timestamp,[¶] denoted by g, is an important hyperparameter in this task since timestamp information can be represented at year, month or day temporal granularity. The example shown in Figure 5.1 uses month granularity.

Jatowt and Au Yeung [53] investigate the usage of temporal expressions at different granularities in news showing that it is relatively rare for authors to use day granularity expressions for future or past time points that are further than 3 months from the the publication date of their news articles.[‖] Wang et al. [163] also test their proposed model trained at different granularities for the even time estimation task, and the time is estimated using the same granularity as in the training step. Thus, the choice of g in TimeBERT should also have effect on the results of downstream tasks. Loosely speaking, the coarser the granularity, the easier is for the model to predict the timestamp during pre-training, however, the

[¶]For example, the timestamp of the document published in May 20th 2022, is "2022/05/20" under day granularity, "2022/05" under month granularity, or just "2022" with year granularity.

[‖]The finer granularity expressions are used more often to refer to the nearer past and future.

model trained on coarse granularity (e.g., year granularity) might not perform well on difficult time-related tasks. In Section 5.4.4, we analyze the effect of different choices of temporal granularity g . Note that the temporal information embedded in the document content may sometimes reveal the document timestamp information; for example, in Figure 5.1, the year information of the timestamp, "1990", is repeated in the first sentence of the document. Thus, this objective can be affected by the temporal ratio of time-aware masked language modeling. In other words, the larger the temporal masking ratio α, the more difficult it should be for the model to correctly predict the timestamp information.

5.2.3 Temporal Information Replacement (TIR)

We also experiment with other ways in which temporal information of documents could be utilized while pre-training. The last pretaining task we investigate has been inspired by WKLM [167]. The authors prove that entity replacement objective can help to capture knowledge about real-world entities. We then devise a similar objective called temporal information replacement (TIR) that aims at training the model to capture temporal information of the document content. Similar to WKLM that replaces entities of the same type (e.g., the entities of PERSON type can only be replaced with other entities of PERSON type), we enforce the replaced temporal expressions to be of the same temporal granularity. We first collect temporal expressions of the news articles in NYT corpus. SUTime [21], a popular tool for recognizing and normalizing temporal expressions, is utilized to detect and then group temporal expressions at year, month and day granularities.** Then, at 50% of the time, the temporal expressions of the input sequence are replaced by other temporal expressions, which can be randomly sampled from the collected temporal expressions' set of the same granularity, while not being replaced for the other 50%. For example, in Figure 5.2, "1990" is replaced by "2000" (note that both are of the same granularity), while "today through Sunday" is not replaced. Then, similar to WKLM, for each temporal expression, the final representations of its boundary words (words before and after the temporal expression) are concatenated and used to make binary prediction

**E.g., "1999 May" maps to "1999/05" under month granularity, and implicit temporal expressions like "yesterday" with the corresponding article's timestamp information "1999/05/19" is resolved and converted to "1999/05/18" under day granularity, etc.

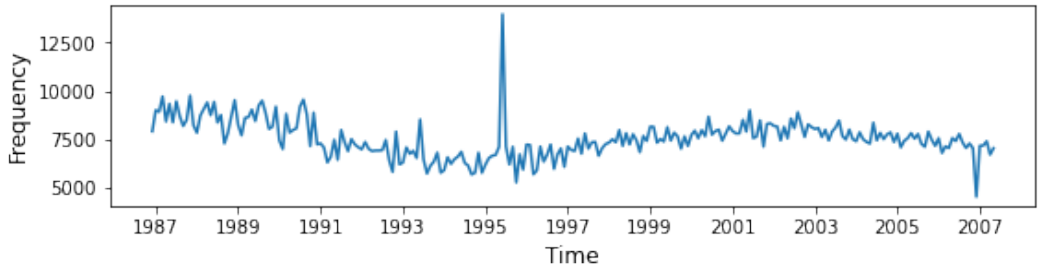

Figure 5.3. Distribution of news articles in the NYT corpus (month granularity)

("replaced" or "not replaced").

Note that this task is an alternative task of time-aware masked language modeling which also utilizes the content temporal information. Besides, our experiments demonstrate that it can even decrease the performance in some downstream tasks. Thus it is not used in the final model TimeBERT.

5.3 Experimental Settings

5.3.1 Pre-training Dataset and Implementation

For the experiments, we use the New York Times Annotated Corpus (NYT corpus) [136] as the underlying dataset for pre-training. The NYT corpus contains over 1.8 million news articles published between January 1987 and June 2007, and it has been frequently used in Temporal Information Retrieval researches [17, 65]. Figure 5.3 shows the monthly distribution of articles in the NYT corpus. As document timestamp estimation is used as a downstream task in our evaluation, 50,000 articles from NYT corpus that were not used in training the model are randomly sampled and kept for the downstream task.

As our method can adapt to all the BERT-style pre-trained language models, we use BERT [28] as the base framework to construct transformer encoder blocks. Considering the high cost of training from scratch, we utilize the parameters of pre-trained $BERT_{BASE}$ (cased) to initialize our model. TimeBERT is trained on the NYT corpus for 10 epochs with the time-aware masked language modeling and document timestamp prediction task.[††] The maximum sequence length was 512, while the batch size was 8. We took AdamW [72] as the optimizer and set

[††]The experiments took about 80 hours on 1 NVIDIA A100 GPU.

the learning rate to be 3e-5, with gradient accumulation equal to 8. In addition, the temporal masking ratio was set to 0.3 in TAMLM task, and the monthly granularity was used in DTP task.[‡‡]

5.3.2 Downstream Tasks

We test our proposal on four datasets of two time-related downstream tasks. These tasks require predicting the following temporal information: *event occurrence time* (with the EventTime dataset [163], WOTD dataset [50]) and *document timestamp* (NYT-Timestamp dataset, TDA-Timestamp dataset).

Note that as event occurrence time estimation requires predicting the time of a given short event description, it is similar to the temporal query analysis (or temporal query profiling) [17, 57, 61], which aims to identify the time of the interest of short queries, and plays a significant role in temporal information retrieval so that time of queries and time of documents can be matched. Another example of how event occurrence time can be used is in question answering, for example, QANA system [161, 162] that introduced in Chapter 3. In Question Answering over temporal document collections like QANA model, a generic type of question that does not contain any temporal expression can be first mapped to its corresponding time period (i.e., time period when the event underlying the question occurred) and then documents from that period can be further processed by a document reader module. Table 5.1 presents examples of the four datasets. The details of these datasets are:

1. **EventTime** [163]: The dataset we constructed and introduced in Chapter 4, consisting of descriptions and occurrence times of 22,398 events (between January 1987 and June 2007). Our previously-introduced TEP-Trans model [163], which achieves SOTA results in this dataset, will also be used to compare with TimeBERT model. As TEP-Trans approach conducts search on the entire NYT corpus and utilizes both kinds of temporal information to estimate events occurrence date, we create an additional dataset called EventTime-WithTop1Doc, with the objective to simulate the similar input setting. The

[‡‡]These two hyperparameters' values of the pre-training tasks are also used in the released TimeBERT version. In Section 5.4.3 and 5.4.4 we will study the effect of temporal masking ratio and temporal granularity of TimeBERT.

Table 5.1. Examples of data instance sampled from four datasets of time-related tasks

Dataset	Text (Event Description or Document Content)	Time
EventTime	Cold War: Soviet Union leader Mikhail Gorbachev is awarded the Nobel Peace Prize for his efforts to lessen Cold War tensions and reform his nation.	1990-10-15
WOTD	American Revolution: British troops occupy Philadelphia.	1777
NYT-Timestamp	IT was a message of support and encouragement that Secretary of State Warren Christopher delivered to President Boris N. Yeltsin in Moscow last week...	1989-10-09
TDA-Timestamp	The Comnaissioners appointed to inquire into the alleged corrupt pratctices at Norwich havo made, their report. It cnmmences with a tribute...	1876-03-20

top-1 relevant document of each event in the NYT corpus is firstly extracted using the same retrieval method (BM25) as in the work of TEP-Trans, and the new model input is provided containing the target event description together with appended timestamp and text content of the top-1 document.

2. **WOTD (Wikipedia On This Day)** [50]: This dataset was scraped from Wikipedia's On this day webpages,* and includes 6,809 short descriptions of events and their occurrence year information. WOTD dataset consists of 635 classes, corresponding to 635 different occurrence years. The earliest year in this dataset is 1302, while the latest is 2018. The median year is 1855.0 whereas the mean is 1818.7. Moreover, the authors additionally provide several sentences about the given event, which they call contextual information (CI).† The contextual information are the relevant sentences extracted from Wikipedia, using a series of carefully designed filtering steps, like key entities extraction, sentence filtering, etc. Thus, we test two versions of this dataset, with contextual information (CI) and without it (No_CI). Note that only year information is given as gold labels, hence the tested models can only predict time at year granularity. Note also that the time span of this dataset (1302-2018) is quite different (and in fact much older) than the one of the NYT

*`https://en.wikipedia.org/wiki/Wikipedia:On_this_day/Today`, accessed 05/2022.

†For example, the contexual information of the WOTD example in Table 5.1 is "The Loyalists never controlled territory unless the British Army occupied it."

corpus (1987-2007) which we use for pre-training. Hence, the generalization ability of the models can be evaluated well.

3. **NYT-Timestamp**: To evaluate the models on the document timestamp estimation task, we use the 50,000 separate news articles of the NYT corpus [136] mentioned in Section 5.3.1.

4. **TDA-Timestamp**[‡]: We also test the timestamp estimation task on another news corpus, the Times Digital Archive (TDA). Times Digital Archive contains over 12 million news articles across more than 200 years (1785-2012),[§] and the time frame of timestamp information of the sampled articles in this dataset are between "1785/01/10" to "2009-12-31". We think that such a long time span can help in evaluating the generalization performance of the models. Same as for NYT-Timestamp we randomly sample 50,000 articles.

As shown in Table 5.1, the examples of EventTime, NYT-Timestamp, and TDA-Timestamp consist of either detailed occurrence date information or timestamp information. Therefore, the models tested on these three datasets can be fine-tuned to estimate the time with different temporal granularities. On the other hand, models fine-tuned on WOTD dataset can only predict the time under year granularity. Note that the dataset difficulty will be greatly increased when the time meeds to be estimated at finer granularities (e.g., month or day), as the number of labels will also greatly increase. For example, for TDA dataset under day granularity, the label count equals to 29,551 which corresponds to the number of days in the dataset. In addition, as [163] and [50] use 80:10:10 split ratio to divide EventTime and WOTD, we also divide the construted NYT-Timestamp, and TDA-Timestamp using the same ratio.

[‡]`https://www.gale.com/binaries/content/assets/gale-us-en/primary-sources/intl-gps/ghn-factsheets-fy18/ghn_factsheet_fy18_website_tda.pdf`

[§]Note that TDA contains more articles spanning longer time period than NYT. Another difference from the NYT is that ORC errors are rather common in TDA (see for example, the last raw in Table 5.1. [115] also shows that TDA has a high OCR error rate especially in early years, that the average OCR error rate from 1785 to 1932 is above 30% while the highest rate can even reach about 60%.). This makes it more challenging to estimate document timestamps. Despite its large size the high number of OCR errors in TDA was the reason why we decided not to use it for pre-training but only for testing.

5.3.3 Evaluation Metrics

Since all the above downstream tasks aim to predict the time, we use: accuracy (ACC) and mean absolute error (MAE) for the performance evaluation. The models are then evaluated under these two metrics at day, month and year temporal granularities on all tested datasets except WOTD dataset, which contains only year information.

1) **Accuracy (ACC)**: The percentage of the events whose occurrence time is correctly predicted.

2) **Mean absolute error (MAE)**: The average of the absolute differences between the predicted time and the correct occurrence time, based on the specified granularity.[¶]

5.3.4 Tested Models

We test the following models:

1. **RG**: Random Guess. The result is estimated by random guess, and the average of 1,000 random selections is used as the result.

2. **BERT**: The pre-trained $BERT_{BASE}$ (cased) model released by [28], which has been trained on BooksCorpus [182] and the English Wikipedia.

3. **BERT-NYT**: The $BERT_{BASE}$ (cased) model that is subsequently trained on the NYT corpus for 10 epochs with BERT's MLM and NSP tasks.

4. **SOTA**: SOTA results of EventTime and WOTD, which are taken from [163] and [50], respectively. Note that SOTA [163] refers to TEP-Trans model introduced in Chapter 4. In addition, the two SOTA methods are not based on language models, both consisting of complicated rules or steps of searching and filtering to obtain the features for estimating the event time, thus cannot be easily and quickly applied in different similar tasks.

5. **BERT_TIR**: The $BERT_{BASE}$ (cased) model trained on the NYT corpus for 10 epochs using MLM and TIR tasks.

[¶]For example, when MAE is 1 at day granularity and month granularity, the average temporal distance is 1 day and 1 month, respectively.

6. **TimeBERT**: Our final language model TimeBERT trained on the NYT corpus for 10 epochs using TAMLM and DTP tasks.

Note that we also study degenerated versions of our proposed model in the ablation studies which will be reported in Section 5.4.2.

5.3.5 Fine-tuning Setting

We fine-tuned the above language models to the downstream tasks of the four datasets that we consider. In all settings, we apply a dropout of 0.1 and optimize cross entropy loss using Adam optimizer, with the learning rate equal to 2e-05 and batch size of 16. The maximum sequence length of the models' fine-tuning on EventTime and WOTD is set to 128 as each input is a short event description, while the maximum sequence length for the models' fine-tuning on EventTime-WithTop1Doc, NYT-Timestamp, TDA-Timestamp is 512, since their input sequence could be very long.

5.4 Experimental Results

5.4.1 Main Results

Event Occurrence Time Estimation. Table 5.2 and Table 5.3 present the results of the tested models on estimating the event occurrence time using EventTime and WOTD, respectively. We first note that the proposed TimeBERT outperforms other language models[‖] in ACC and MAE on the two datasets over different settings (i.e., different granularities, or with/without the top1 document information, or with/without contextual information). In addition, we notice that on both the datasets the task is not easy to be solved as the RG results exhibit very poor performance on both datasets.[**]

When looking at the results obtained for EventTime dataset under two different settings at day granularity, we can see that the performance of all the language models at day granularity is also rather poor; however, still, TimeBERT achieves the best results. We then compare TimeBERT with other models by considering

[‖] The SOTA methods [163] and [50] are not based on language models.

[**] Especially on EventTime dataset for the case when the time is need to be predicted under month/day granularity and on WOTD dataset.

the year and month granularity. When comparing TimeBERT with BERT on original EventTime dataset using ACC and MAE, the improvements are 47.39%, 10.09% at year granularity, and 155.21%, 20.59% at month granularity, respectively. When comparing TimeBERT with BERT on EventTime-WithTop1Doc, the improvements are 16.62%, 38.30% at year granularity, and 330.77%, 23.95% at month granularity, respectively. Our model also performs much better than BERT-NYT, which achieves similar results as BERT. Moreover, a significant improvement of TimeBERT can be observed when top-1 document is provided, for example, at month granularity, the improvement is 98.31% and 17.05% on ACC and MAE, respectively.

In addition, BERT_TIR, the model trained using MLM and our proposed TIR task, shows relatively good performance in most of the cases, too; for example, when comparing with BERT-NYT at year granularity using ACC and MAE, the improvements are 19.53%, 9.27% on EventTime, are 5.83%, 20.45% on EventTime-WithTop1Doc, respectively. When compared with SOTA [163], the TEP-Trans model we introduced in previous chapter, TimeBERT achieves similar or even better results on EventTime-WithTop1Doc under year and month granularities, while the performance at day granularity is rather poor. Although TEP-Trans utilizes both temporal signals and obtains comparative results, it can only estimate the time within the time frame of the underlying knowledge source that is being used.* In addition, TEP-Trans uses multivariate time series as the model's input, which is constructed by analyzing the temporal information of the top-50 retrieved documents through several complicated steps, such as sentence similarity computation, which requires rather considerable time or effort. Therefore, we believe that the results of TimeBERT model could also be further improved by combining with more relevant information derived from more relevant sentences or documents.

When considering WOTD dataset, TimeBERT outperforms SOTA [50] using accuracy as an evaluation metric. Especially when the contextual information† is provided, the improvement is 75.95%. We also observe that BERT-NYT and BERT_TIR can surpass SOTA [50] and BERT when using contextual information.

*Since TEP-Trans use NYT corpus as the knowledge source, the model can only estimate the time of events happened between 1987 and 2007.

†Contextual information contains the relevant sentences extracted from Wikipedia as the external knowledge, as explained in Section 5.3.2.

Table 5.2. Performance of different models on EventTime datasets of event occurrence time estimation with two different settings.

Model	EventTime						EventTime-WithTop1Doc					
	Year		Month		Day		Year		Month		Day	
	ACC	MAE	ACC	MAE	ACC	MAE	ACC	MAE	ACC	MAE	ACC	MAE
RG	4.77	6.92	0.41	81.60	0.01	2484.48	4.77	6.92	0.40	81.70	0.01	2482.83
BERT	21.65	3.47	5.09	43.81	0.36	2055.71	35.98	3.89	5.98	37.95	0.04	2690.48
BERT-NYT	21.25	3.56	5.18	43.50	0.36	2013.87	34.46	4.45	8.21	34.14	0.13	1544.36
SOTA [163]	-	-	-	-	-	-	40.93	3.01	**30.89**	36.19	**16.42**	**1235.67**
BERT_TIR	25.40	3.23	6.83	40.45	0.98	1751.92	36.47	3.54	17.01	31.72	0.09	1654.05
TimeBERT	**31.91**	**3.12**	**12.99**	**34.79**	**1.88**	**1650.46**	**41.96**	**2.40**	25.76	**28.86**	2.07	1404.56

Table 5.3. Performance of different models on WOTD dataset with/without contextual information.

Model	NO_CI		CI	
	ACC	MAE	ACC	MAE
RG	0.16	217.72	0.15	217.57
BERT	7.20	52.58	9.69	41.16
BERT-NYT	8.08	53.75	19.97	36.47
SOTA [50]	11.40	-	13.10	-
BERT_TIR	10.13	54.92	18.36	35.99
TimeBERT	**11.60**	**48.51**	**23.05**	**33.70**

The two latter methods do not utilize news archives, indicating that the news article archives might be more effective to be used in such a task rather than synchronic document collections (e.g., Wikipedia). As our model obtains a good performance on this challenging dataset, whose time is quite different than the training NYT corpus, we conclude that our model has good generalization ability.

Document Timestamp Estimation. Table 5.4 presents the results of document timestamp estimation on NYT-Timestamp and TDA-Timestamp. The RG results are again very poor at both datasets of different granularities. In addition, all language models achieve bad results under day granularity of both datasets and under month granularity at TDA-Timestamp, as the number of time labels at these settings is quite large. For example, NYT-Timestamp dataset has 7,438 day labels, TDA-Timestamp has 2,627 month labels and 29,551 day labels. In addition, the timestamp in the 50,000 articles of TDA-Timestamp dataset ranges from 1785 to 2009, which further increases the difficulty. We then

Table 5.4. Performance of different models for document timestamp estimation on two datasets: NYT-Timestamp and TDA-Timestamp.

Model	NYT-Timestamp						TDA-Timestamp					
	Year		Month		Day		Year		Month		Day	
	ACC	MAE	ACC	MAE	ACC	MAE	ACC	MAE	ACC	MAE	ACC	MAE
RG	4.77	7.06	0.41	81.79	0.01	2488.53	0.45	75.39	0.04	873.88	0.00	11253.72
BERT	35.00	1.64	2.56	22.74	0.10	1813.89	15.84	44.87	0.80	632.66	**0.02**	14404.31
BERT-NYT	38.74	1.41	8.24	18.35	0.02	2961.92	15.04	45.16	0.66	669.02	0.00	16817.59
BERT_TIR	48.06	1.09	20.30	13.54	0.56	486.05	17.72	43.53	1.26	589.69	0.00	17806.36
TimeBERT	**58.72**	**0.80**	**31.10**	**9.54**	**1.28**	**348.87**	**19.00**	**40.11**	**2.38**	**580.25**	0.00	**10780.44**

mainly compare the models on NYT-Timestamp of year and month granularities, and on TDA-Timestamp of year granularity. TimeBERT still outperforms other language models with substantial gains. When considering the year and month granularities of NYT-Timestamp, the improvement comparing TimeBERT with BERT-NYT is in the range of 51.57% to 277.43%, and from 43.26% to 48.01% on ACC and MAE metrics, respectively. When considering TDA-Timestamp under year granularity, the improvement is 26.33% and 11.18% on ACC and MAE, respectively. In addition, BERT_TIR also obtains relatively good results on both timestamp datasets, suggesting that the TIR objective is also effective.

5.4.2 Ablation Study

To study the effect of the two pre-training objectives of TimeBERT, we next conduct an ablation analysis and present the results in Table 5.5 and Table 5.6. We test five models that use different pre-training tasks and test them on the four datasets with specific settings (i.e, we remove some settings that show bad performance on all models described in Section 5.4.1, for example, the test of TDA-Timestamp at month or day granularities is removed). DTP, TAMLM, MLM indicate the corresponding models trained using only DTP, TAMLM or MLM tasks, respectively. MLM+DTP means the model is trained using both BERT's MLM task and our proposed DTP task. For fair and effective comparison, all the models are trained using their specific pre-training tasks for 3 epochs.

As shown in Table 5.5 and Table 5.6, TimeBERT, which uses TAMLM and DTP as the pre-training tasks, achieves the best results across all the datasets, suggesting that the two proposed objectives contribute to the performance of

Table 5.5. Ablation test on event occurrence time estimation. All models are trained using their specific pre-training tasks for 3 epochs.

Model	**EventTime**				**WOTD**			
	Year		**Month**		**NO_CI**		**CI**	
	ACC	**MAE**	**ACC**	**MAE**	**ACC**	**MAE**	**ACC**	**MAE**
DTP	26.95	3.32	9.38	38.56	10.01	65.73	18.50	46.86
TAMLM	23.05	3.37	6.87	41.16	9.43	53.48	19.82	38.74
MLM	21.52	3.45	5.71	44.47	8.66	55.66	18.80	40.85
MLM+DTP	26.13	3.28	8.84	39.68	**11.16**	58.40	19.32	40.24
TimeBERT	**29.51**	**3.17**	**10.80**	**36.11**	**11.16**	**51.09**	**22.47**	**36.80**

Table 5.6. Ablation test on document timestamp estimation. All models are trained using their specific pre-training tasks for 3 epochs.

Model	**NYT-Timestamp**				**TDA-Timestamp**	
	Year		**Month**		**Year**	
	ACC	**MAE**	**ACC**	**MAE**	**ACC**	**MAE**
DTP	51.06	0.92	24.43	13.85	15.84	45.23
TAMLM	39.92	1.46	8.80	16.74	14.96	45.80
MLM	36.98	1.51	3.46	19.17	14.44	46.08
MLM+DTP	53.12	0.96	24.41	14.46	16.14	45.93
TimeBERT	**56.08**	**0.81**	**27.42**	**10.56**	**18.54**	**43.00**

our model. When considering the models that use only one of the pre-training objectives of TimeBERT, DTP and TAMLM, the performance is better than MLM in most cases. This confirms that the two proposed pre-training tasks of TimeBERT are both helpful in obtaining the effective time-aware language representation of text. When considering the models that use DTP objectives, DTP and MLM+DTP, we can also observe that these models achieve relatively good results. This suggests that DTP is very useful in time-related downstream tasks. Yet, incorporating at the same time the two proposed objectives of TimeBERT that make use of different temporal aspects produces the best results.

5.4.3 Effect of Different Temporal Masking Ratios in TAMLM

Temporal masking ratio α ($0.0 \leq \alpha \leq 1.0$) is an important hyperparameter of TAMLM task, which determines how many temporal expressions in the document are sampled during masking. For example, when α equals to 0.0, no tokens of all temporal expressions are sampled, and this could make it easier for a model to predict the document timestamp in DTP task, especially when the contained temporal expressions reveal some part of the predicted timestamp (e.g., in Figure 5.1, the year information of the timestamp, "1990", is repeated in the first sentence of the document.). On the other hand, when α equals to 1.0, the tokens of all temporal expressions are sampled, which will increase the difficulty of DTP task. To examine the effect of α, we pre-train TimeBERT using different α values using TAMLM and DTP tasks for 3 epochs. Figure 5.4 shows the results of different TimeBERT instances fine-tuned on four datasets, which are EventTime under month granularity, WOTD with contextual information, NYT-Timestamp and TDA-Timestamp at year granularity. We can see that smaller α values (e.g., $0.0 \leq \alpha \leq 0.5$) tend to produce better results than larger values. When considering the accuracy metric, TimeBERT achieves the best results on EventTime and NYT-Timestamp when α equals to 0.3, and it produces the best results on WOTD and TDA-Timestamp when α equals to 0.2, 0.1 respectively.[‡]

5.4.4 Effect of Different Temporal Granularities in DTP

We finally examine TimeBERT instances training using different settings for the temporal granularity g in DTP task. Similarly, we first pre-train different TimeBERT with three different temporal granularities for 3 epochs, and then fine-tune the models on four datasets. The models of different granularities are denoted by TimeBERT-Year, TimeBERT-Month and TimeBERT-Day. As shown in Table 5.7 and Table 5.8, we can observe that TimeBERT pre-trained using month granularities achieves most of the best results,[§] while the model pre-trained using day granularities performs poor in some "easy" tests, e.g., for the EventTime and NYT-Timestamp of year granularity, as well as WOTD with CI. We also observe that none of the models can produce relatively good performance on the hard tasks (e.g., EventTime of day granularity). This might be mainly due to: (1)

[‡]The released TimeBERT version uses α value equal to 0.3.

[§]The released TimeBERT version uses g set to month granularity.

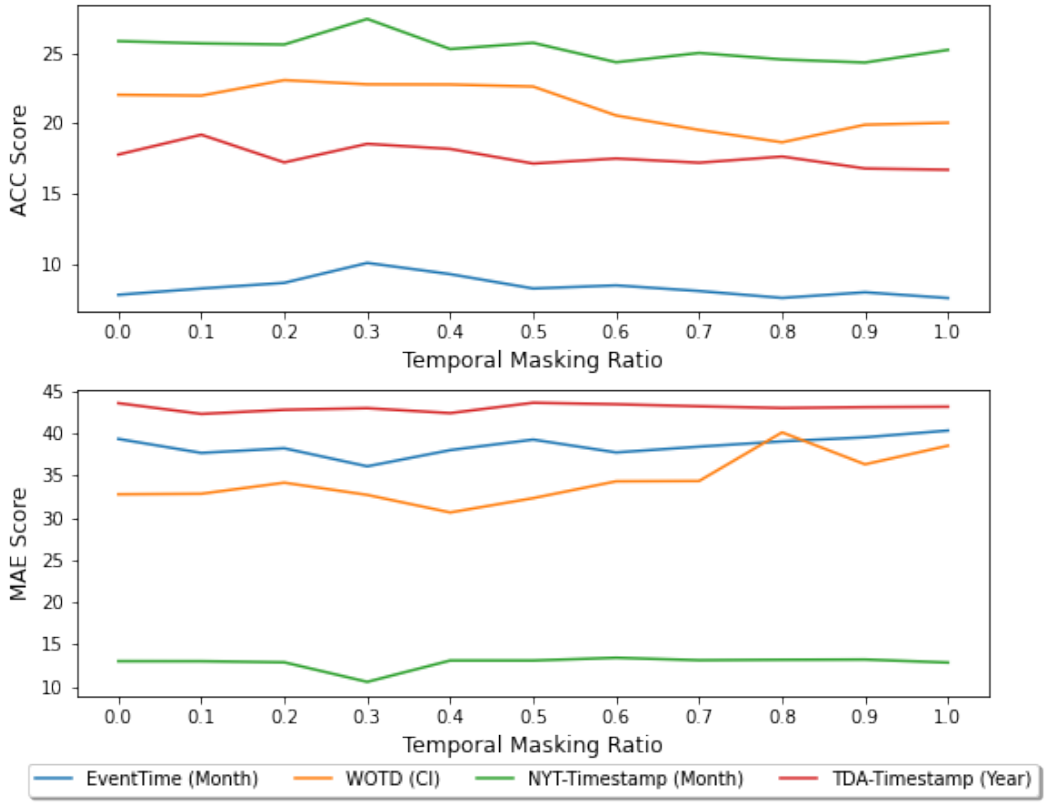

Figure 5.4. TimeBERT performance (accuracy in the top plot and MAE in the bottom plot) with different temporal masking ratios on four datasets.

the models are still underfitting and may need to be trained with more epochs, especially when using day granularity in DTP task and (2) more data is needed for pre-training which includes more historical knowledge.[¶]

5.5 Applications

TimeBERT can be used in several ways and supports different applications for which time is of importance. It can be easily applied in temporal information retrieval domain, for example, aiding in time-based exploration of textual archives by estimating the time of interest of queries, so that the computed query temporal information can then be utilized for time-aware document ranking. Other

[¶]Note that the quality of the news collection may also matter here and might need to be considered; for example, the OCR errors, are quite a serious problem in TDA corpus.

Table 5.7. TimeBERT with different temporal granularities on event occurrence time estimation task. All models are pre-trained at their specific temporal granularity for 3 epochs.

Model	EventTime						WOTD			
	Year		Month		Day		NO_CI		CI	
	ACC	MAE	ACC	MAE	ACC	MAE	ACC	MAE	ACC	MAE
TimeBERT-Year	**30.71**	**3.06**	8.62	38.35	0.76	1772.48	9.84	59.76	20.56	**35.67**
TimeBERT-Month	29.51	3.17	**10.80**	**36.11**	**1.83**	1743.75	**11.16**	**51.09**	**22.47**	32.92
TimeBERT-Day	26.43	3.18	7.99	38.42	1.27	**1647.64**	10.72	53.36	17.47	40.22

Table 5.8. TimeBERT with different temporal granularities on timestamp estimation task. All models are pre-trained at their specific temporal granularity for 3 epochs.

Model	NYT-Timestamp						TDA-Timestamp					
	Year		Month		Day		Year		Month		Day	
	ACC	MAE	ACC	MAE	ACC	MAE	ACC	MAE	ACC	MAE	ACC	MAE
TimeBERT-Year	**57.48**	**0.78**	19.46	11.30	0.34	401.88	17.88	43.93	1.02	**575.04**	0.00	14168.61
TimeBERT-Month	56.08	0.81	**27.42**	**10.56**	**0.72**	406.52	**18.54**	**43.00**	**1.30**	643.38	**0.02**	**12083.72**
TimeBERT-Day	54.06	0.91	19.46	11.02	0.64	**398.77**	18.08	43.41	1.14	603.71	0.00	13794.74

potential application can be: generating a timeline summary for a specific news story [168, 179] or for a given entity [155], temporal image retrieval [31] that helps users to find relevant images which satisfy the temporal intent behind their queries (e.g., user query "iPhone13" should returned images showing the right device model released in recent years), or event detection and ordering [27, 148], temporal clustering and information retrieval [2, 15, 17], question answering [117, 161], etc.

We next demonstrate how the proposed TimeBERT model could be utilized in one such application. In particular, we test QANA model [162], that is introduced in our first research topic in Chapter 3, which achieves good results on answering event-related questions that are implicitly time-scoped (e.g., "Which famous painting by Norwegian Edvard Munch was stolen from the National Gallery in Oslo?" is implicitly time-scoped question as it does not contain any temporal expression, but is implicitly related to specific event temporal information, which is "1994/05"). To answer such questions for which temporal information cannot be extracted directly from the question's content, QANA needs to first

Table 5.9. Performance of different models in QA task

Model	Top 1		Top 5		Top 10		Top 15	
	EM	**F1**	**EM**	**F1**	**EM**	**F1**	**EM**	**F1**
QANA [162]	21.00	28.90	28.20	36.85	34.20	44.01	36.20	45.63
QANA + TimeBERT	**22.40**	**29.31**	**29.20**	**37.14**	**34.80**	**44.34**	**36.40**	**46.01**

estimate the time scope of the event behind the question at month granularity, which is mapped to the time interval with the "start" and "end" information (e.g., one good estimated time scope of the above-mentioned question example is ("1994/03", "1994/08")). Instead of analyzing the temporal distribution of retrieved documents to estimate the time scope as is in original implementation of QANA, we adapt QANA by using the TimeBERT model fine-tuned on EventTime-WithTop1Doc dataset of month granularity. Similar to the way of making EventTime-WithTop1Doc dataset, the top-1 relevant document of each question is first selected using BM25, and then its timestamp and text content are appended to the corresponding questions, which is further sent to the TimeBERT as the input. We then keep two time points of the top 2 probabilities predicted by TimeBERT, which are then ordered and used as "start" and "end" information of the estimated question time scope. The estimated time scope is then utilized for re-ranking documents, and finally the answers are returned by Document Reader Module of QANA. In our adaptation of QANA, we just replace the step of the time scope estimation, and we denote the resulting system as QANA+TimeBERT. We test this system on manually constructed 500 implicitly time-scoped questions introduced in Chapter 3. As the number of the top N re-ranked documents which are used by the Document Reader Module affects the final results, we also test different top N values. As shown in Table 5.9, QANA+TimeBERT outperforms QANA for all the different N values. When considering the top 1 document, the new extended model achieves 6.67% improvement on EM and 1.42% on F1.

5.6 Summary

Time is an important aspect of text documents, which has been widely exploited in natural language processing and has strong influence. For example, it was used

in temporal information retrieval, where the temporal information of queries or documents need to be identified for relevance estimation. Event-related tasks like event ordering, which aim to order events by their occurrence time, needs to determine the temporal information of events, too. In this chapter, we have presented a novel language representation model called TimeBERT which is especially designed for time-related tasks. TimeBERT is trained over a temporal news collection through two new pre-training tasks that involve two kinds of temporal aspects (document timestamp and document content time). We have next conducted experiments to investigate the effectiveness of different pre-training tasks that incorporate temporal information. The results reveal that the proposed pre-training objectives can effectively utilize two distinct temporal aspects and could help to achieve improved performance on two different time-related downstream tasks. In addition, it can be easily applied on applications that consider time, for example, temporal question answering system like QANA.

In the future, we will test TimeBERT model on other time-related tasks and applications, for instance, semantic change detection and timeline summarization. In addition, we will try other ways to incorporate TIP with TAMLM, since both pre-training tasks utilize the same temporal information extracted from content. During pre-training, we will also utilize the temporal relations associated with the temporal expressions, for example, extended temporal expressions like "before 1999", "until Sunday", etc. Such temporal relations are important since they can denote explicit temporal relations held between two abstract entities (time and event, time and time, or event and event).

CHAPTER 6

Creating a Large-scale ODQA Dataset over Temporal News Collections

In Chapter 3, we propose QANA, which is an ODQA system designed specifically for answering event-related questions over news archives. However, the Document Retriever Module of QANA can only use sparse retrieval methods rather than the advanced dense retrievers due to the lack of large-scale datasets over temporal news collections for training, that hinders the development of QA research over such valuable resources. In this chapter, we introduce a large-scale ODQA dataset called ArchivalQA to solve these problems, and test both sparse and dense retrieval methods on the dataset. In addition, the novel QA dataset-constructing framework can be also applied to generate high-quality questions over other document collections.

6.1 Introduction

With the application of digital preservation techniques, more and more past news articles are being digitized and made accessible online. This results in the availability of large news archives spanning multiple decades. They offer immense value

to our society, contributing to our understanding of different time periods in the history, helping us to learn about the details of the past, and offering valuable lessons for future generations [76]. For example, sociologists have used news archives to examine vital questions like how different jurisdictions slowed the spread of the 1918 flu [101], which can also offer valuable lessons for the COVID-19 pandemic we are facing today. However, due to their large sizes and complexities, it is difficult for users to effectively utilize such temporal news collections. A reasonable solution is to use open-domain question answering (ODQA), which attempts to answer natural language questions based on large-scale unstructured documents. Yet, the existing QA datasets are essentially constructed from Wikipedia or other synchronic document collections.* The lack of large-scale datasets hinders the development of ODQA on document archives such as news article archives where Temporal IR [16, 66] techniques need to be utilized. Note that ODQA on historical document collections can be useful in many cases such as providing support for journalists who wish to relate their stories to certain past events, historians who investigate the past as well as employees of diverse professions, such as insurance or broad finance sectors, who wish to assess current risks based on historical accounts in order to support their decision making. As indicated in previous studies [161, 162], synchronic document collections like Wikipedia cannot successfully answer many minor or detailed questions about the events from the past since the relevant data for answering those questions is only available in primary sources preserved in the form of large archival document collections.

To overcome these shortcomings of existing QA datasets, we devise a novel framework that assists in the creation of a diverse, large-scale ODQA dataset over a temporal document collection. The framework utilizes automatic question generation as well as a series of carefully-designed filtering steps to remove poor quality instances. As an underlying archival document collection, we use the New York Times Annotated Corpus (NYT corpus) [136], which contains over 1.8 million news articles published between January 1, 1987 and June 19, 2007. The NYT corpus has been frequently used over the recent years for many researches in

*Note that existing news datasets such as CNN/Daily Mail [49] and NewsQA [156] are more suited to machine reading comprehension (MRC) tasks rather than to ODQA task due to the cloze question type or the ambiguity prevalent in their questions as we will discuss later. In addition, their underlying document collections span relatively short time periods, which are also quite recent (such as after June 2007 or April 2010).

temporal IR, temporal news content analysis, archival search, historical analysis and in other related tasks [16, 66]. The final dataset that we release, ArchivalQA, contains 532,444 data instances and is divided into different sub-parts based on question difficulty and the presence of temporal expressions.

We choose a semi-automatic way to construct our dataset for several reasons. First, manually generating questions would be too costly as it requires knowledge of history from annotators. Second, since question generation (QG) has recently attracted considerable attention, the available models already achieve quite good performance. Third, current "data-hungry" complex neural network models require larger and larger datasets to maintain high performance. Finally, synthetic datasets have been effective in boosting deep learning models' performance and are especially useful in use cases involving distant target domains with highly specialized content and terminology, for which there is only a small amount of labeled data [39, 93, 160]. We then approach the dataset generation techniques based on a cascade of carefully designed filtering steps that remove low quality questions from a large initial pool of generated questions. Note that in Section 2.5.1, we describe the drawbacks of existing QA datasets, which make them cannot be used to train QA models well over news archives.[†]

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

- We propose one of the largest ODQA datasets for news collections,[‡] which is not only spanning the longest time period compared to other QA datasets, but it also provides detailed questions on the events that occurred from 14 to 34 years ago.

- We propose an approach to generate large datasets in an inexpensive way, whose resulting questions tend to be non-ambiguous and of good quality, thus having only a single potential answer. Compared with other QG methods, most of our generated questions are clear and non-ambiguous, and thus they can be especially useful in improving computational approaches to education, e.g., to support generating questions for exams.

[†]Note also that the related studies of QA benchmarks and automatic question generation techniques are discussed in Section 2.5.2. In addition, Table 2.1 presents differences between ArchivalQA and the most related datasets.

[‡]The largest existing dataset that uses news articles, CNN/Daily Mail dataset [49], has been created based on a straightforward cloze test and thus cannot be considered as a proper ODQA dataset.

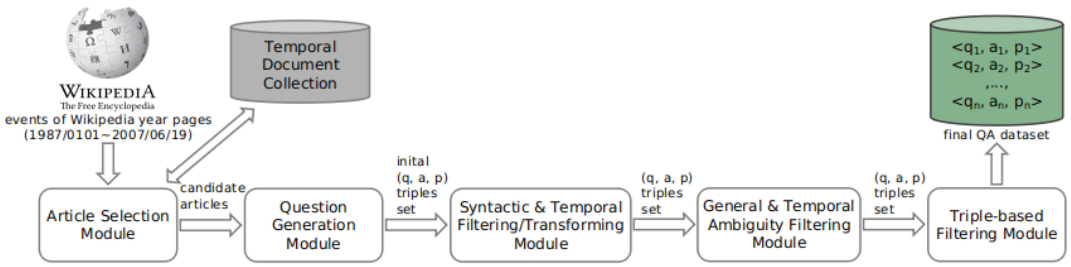

Figure 6.1. Dataset generation framework

- We undertake comprehensive analysis of the generated dataset, which does not only show the quality and utility of the resulting data, but also proves the effectiveness of our QG framework.

6.2 Dataset Generation Framework

We introduce here the framework that generates and selects questions from news archives. Figure 6.1 shows its architecture which consists of five modules: *Article Selection Module*, *Question Generation Module*, *Syntactic & Temporal Filtering/Transforming Module*, *General & Temporal Ambiguity Filtering Module* and *Triple-based Filtering Module*. All these modules are described below.

6.2.1 Article Selection Module

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

Selection based on Wikipedia Events. The first one relies on the short descriptions of important events available in Wikipedia year pages[§] as the seeds to find related articles. Since we utilize the NYT corpus, we use 2,976 event descriptions which occurred between January 1, 1987 and June 19, 2007. Then, for each such event description, we select keywords to be used as search queries for retrieving articles related to this description from the news archive. We choose Yake![¶] [18] as our keyword extraction method, which is a state-of-the-art unsupervised approach that relies on statistical features to select the most important

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

[¶]`https://yake.inesctec.pt`

keywords. Next, the query composed of the extracted keywords is sent to the ElasticSearch[‖] installation which returns the top 25 relevant documents ranked by BM25. Finally, 53,991 news articles are obtained in this way to be used for generating questions.

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

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

6.2.2 Question Generation Module

The second step is to generate questions from the collected articles. We first separate articles into paragraphs and then use a neural network model to generate candidate questions from each paragraph. We apply T5-base [125] - a recent, large, pre-trained Transformer encoder-decoder model. We note that, same as us, Lelkes et al. [87] have used QG methods to generate questions from news articles in an automatic way, although in their case PEGASUS model [177] was utilized to generate the questions using the NewsQuizQA dataset. However, we did not choose PEGASUS-base model since we found that it generates questions which sometimes contain information not found in the underlying documents (probably due to the Gap Sentences Generation pre-training task that PEGASUS model applies). Furthermore, the questions generated by Lelkes et al. [87] belong to the quiz-style multiple-choice type which is not suitable for ODQA.

The QG model is fine-tuned using SQuAD 1.1[††] [126] whose inputs are the answers together with their corresponding paragraphs, and the questions form the outputs. The final model achieves good performance on the SQuAD 1.1 dev set (the scores of BLEU-4, METEOR, ROUGE-L are 21.19, 26.48, 42.79, respectively). After fine-tuning the model, every named entity[‡‡] in a given paragraph

[‖] `https://www.elastic.co/`

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

[††] We decided not to use NewsQA for training as it contains too many ambiguous questions.

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

of each article is labeled as an answer, and is used along with the paragraph as the input to the model. Note that the answers of many QA datasets, such as CNN/Daily Mail [112], TriviaQA [59], Quasar-T [29], SearchQA [36] and XQA [96], are also mainly in the form of entities (e.g., 92.85% of the answers in TriviaQA are Wikipedia entities), as this improves answering accuracy. In addition, we restrict the number of tokens of the paragraphs and of the corresponding sentences which include the answers. More specifically, the paragraphs that have less than 30 tokens are eliminated. Additionally, the answers whose corresponding sentences have less than 10 tokens are discarded. Finally, we generated 6,408,036 questions in this way from 1,194,730 paragraphs of 106,197 news articles.

6.2.3 Syntactic & Temporal Filtering/Transforming Module

This module consists of 8 basic processing steps that further filter or transform the candidate question-answer pairs obtained so far. It first removes the pairs of bad-quality by following six filtering steps:

1. Remove questions that do not end with a question mark (107,586 such questions removed).

2. Remove questions whose answers are explicitly indicated inside the questions' content (127,212 questions removed). For example, question like *"Where did Mr. Roche serve in Vietnam?"* that has gold answer "Vietnam" is removed.

3. Remove duplicate questions. The same questions generated from different paragraphs are removed (492,257 questions removed).

4. Remove questions that have too few or too many named entities. Questions without any named entity or with more than 7 named entities are eliminated (1,310,621 questions removed).

5. Remove questions that are too short or too long. Questions that contain less than 8 or more than 30 tokens are dropped (463,726 questions removed).

6. Remove questions with unclear pronouns, for example, "*What was the name of the agency that she worked for in the Agriculture Department?*" (63,300 questions removed).

We describe here the details of removal of questions with unclear pronouns. We first utilize part-of-speech tagger in spaCy to obtain the fine-grained POS information of each token in the generated questions. The questions whose tokens are classified as "PRP" or "PRP$" are collected as the initial set of unclear-pronoun questions. Then we utilize the novel coreference resolution tool (NeuralCoref [23]) to obtain the coreference results of each sentence in the question set. For example, for the question "*When did Sampras win his first Grand Slam?*", the information that 'his' points to 'Sampras' is derived. Then we apply several heuristic rules to collect only clear-pronoun questions. A sentence is considered correct if its pronoun points to named entities appearing inside the question's content (e.g., 'Sampras' in the previous example), or if the question asks about the actual resolution of the pronoun (e.g., "*Who dived into rough waters near her home in Maui to save a Japanese woman?*"), etc.

Then, this module transforms the relative temporal information of the QA pairs by the following two steps:

1. Transform relative temporal information in questions to absolute temporal information. For example, *"How many votes did President Clinton have in New Jersey last year?"* is transformed to *"How many votes did President Clinton have in New Jersey in 1996?"* (140,658 questions transformed).

2. Transform relative temporal information of the answers of generated questions to absolute temporal information. We apply the same approach as in the previous step. For example, the answers to questions *"When did Rabbi Riskin write about protests by West Bank settlers in Israel?"* and *"When were the three teenagers convicted of murdering Patrick Daly?"*, which are "Aug. 7" and "yesterday", respectively, are transformed to "August 07, 1995" and "June 15, 1993", by incorporating the articles' publication dates: '1995-08-12' and '1993/06/16' (279,671 answers transformed in this way).

We describe here the details of relative temporal information transformation. We first apply SUTime [21] to recognize temporal expressions, and we use the publication date information of the articles, which include the paragraphs used to generate the question, as the reference date to transform the relative temporal information. Note that we do not transform all the temporal expressions in the entire corpus, since this would be too time-consuming. Additionally, this would change the original contents of the articles in the corpus, the situation which we

try to avoid. Any systems that will use our dataset should see only the original, unchanged content of NYT's news articles for answering our dataset's questions. We expect that models which need to use temporal expressions should utilize article timestamps to resolve temporal expressions.

6.2.4 General & Temporal Ambiguity Filtering Module

Filtering by Content Specificity. Sentence specificity is often pragmatically defined as the level of detail of the information contained in the sentence [89, 99]. In contrast to specific sentences that contain informative messages, general sentences do not reveal much specific information (e.g., overview statements). In the examples shown below, the first sentence is general as it is clearly less informative than the second sentence (specific one), and is not suitable to be used for question generation.

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

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

Thus, in this step, we aim to remove questions that have been generated from general sentences. We use the training dataset from Ko et al. [74], which is composed of three publicly available, labeled datasets [89, 90, 100]. The combined dataset contains 4,342 sentences taken from news articles together with their sentence-level binary labels (general vs. specific). We partition this dataset randomly into the training set (90%), and the test set (10%). We next fine-tune three Transformer-based classifiers: BERT-based model [28], RoBERTa-base model [98] and ALBERT-base model [81], such that each classifier consists of the corresponding pre-trained language model followed by a dropout layer and a fully connected layer. We finally choose RoBERTa-base model [98] as our specificity-determining model because it achieves the best results on the test set - 84.49% accuracy. Finally, we discard all questions whose underlying sentences from which they were generated have been classified by the above-described approach as general. This filtering step removed 952,398 questions. Few examples of the removed general questions are given in Table 6.2.

Table 6.1. Temporal ambiguity of example questions.

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

Filtering by Temporally Ambiguity. When manually analyzing the resulting dataset we have observed that some questions are problematic due to their temporal ambiguity, e.g., *"How many people were killed by a car bomb in Baghdad?"*. Such questions can be matched to several distinct events. The first and the second generated example questions in Table 6.1 exhibit such characteristics; the correct answers of such questions should be actually a list of answers rather than a single answer. However, the datasets having multiple correct answers for each question are quite rare in the current ODQA field [181] (we are only aware of AMBIGQA dataset [107] which contains multiple possible answers to ambiguous questions). This might be because it would not be clear how to rank systems as some of the ground-truth answers might be more preferred than others. In our case, for example, some events related to the ambiguous questions could be more important or more popular than other related events. Also, and perhaps more importantly, finding all the correct answers to such questions is quite difficult, if not impossible, within a large news collection (especially an archival one that spans two decades such as ours). Hence, we decided to remove temporally ambiguous questions, however we will make them available for the community to download as a separate data, should anyone be interested in studying questions of this type.

We define temporally ambiguous questions as ones that have multiple correct and different answers over time. Note that temporally ambiguous questions are specific to temporal datasets like ours, and consequently they have not been studied before. Since there is no readily available dataset for detecting temporally ambiguous questions, we have manually labeled 5,500 questions obtained from the

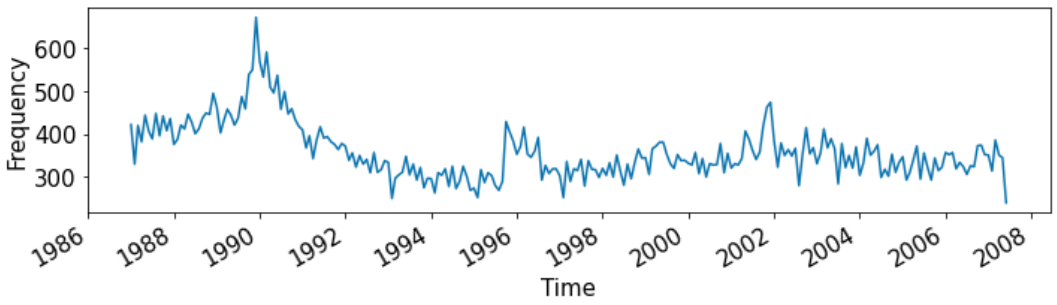

Figure 6.2. Distribution of articles used in ArchivalQA

previous filtering steps.* Then, we again fine-tuned three Transformer-based classifiers, same as when training the specificity-evaluating model. The BERT-based model [28] has been finally chosen as it performs best on the test set achieving 81.82% accuracy. We then used it to remove 1,823,880 questions classified as temporally ambiguous.† Similarly, in Table 6.2, we also give few examples of the removed ambiguous questions.

Table 6.2. Examples of Questions Removed by the General & Temporal Ambiguity Filtering Module.

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

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

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

6.2.5 Triple-based Filtering Module

In the final module, we aim to remove remaining poor quality data instances by analyzing the entire $\langle question, answer, paragraph \rangle$ triples. Some instances are still problematic due to several reasons (e.g., questions with incorrect answers, questions containing information not found in paragraphs, or other wrong questions that have not been filtered out by the previous filtering stages). To construct the last filter we first created a dedicated dataset by asking 10 annotators to label 10k samples selected from the results obtained after applying the previously-introduced filtering stages. The labels were either "Good" or "Bad" based on $\langle question, answer, paragraph \rangle$ triples.[‡] The annotators had to not only consider the particular problems we discussed before, but also check whether the questions are grounded in their paragraphs and whether they can be answered by their answers, and whether the questions are grammatically correct or not. The dataset, which contains 5,699 "Good" questions and 4,301 "Bad" questions, was then randomly split into the training set (90%), and the test set (10%). Then, we trained a RoBERTa-base model [98] that takes the triples as the input after adding a special token ([SEP]) to the question-answer pair and paragraph of each sample. We set a high threshold that permits only the predicted good triples with probabilities higher than 0.99 be chosen as the final good triples. This last filtering step resulted in the precision of finding good triples to be 86.74% on our test set. Finally, we removed 534,612 questions whose corresponding triples were classified as bad.

6.3 Dataset Analysis

6.3.1 Data Statistics

After all the above filtering steps, we have finally obtained the dataset which includes 532,444 question-answer pairs that were derived from 313,100 paragraphs of 88,431 news articles. About half of the questions (263,292) come from the randomly selected articles, and the other questions (269,152) are based on articles that were selected based on Wikipedia events. This provenance information is recorded for each question. Paragraph IDs are also appended to each

[‡]This dataset is also available.

Table 6.3. Basic statistics of ArchivalQA

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

question-answer pair to let ODQA systems explicitly train their IR components. We partition the entire dataset randomly into the training set (80%, 425,956 examples), the development set (10%, 53,244 examples), and the test set (10%, 53,244 examples). More detailed dataset statistics are presented in Table 6.3. Table 6.4 shows few examples. Figure 6.2 shows also the temporal distribution of documents used for producing ArchivalQA questions.

We have also analyzed the named entity types[§] of the answers in the dataset. As shown in the left pie chart in Figure 6.3, the answers that belong to PERSON, ORG, DATE, GPE and NORP[¶] account for a large part of ArchivalQA. Further, the right hand side's pie chart in Figure 6.3 shows the distribution of 9 event categories of the questions that are classified by another dedicated classifier prepared by us, which has been trained based on the event dataset created by Sumikawa and Jatowt [150] achieving 85.86% accuracy. We can see that ArchivalQA contains questions related to diverse event categories, while the "arts & culture", "politics & elections", "armed conflicts & attacks", "law and crime" and "business & economy" events account for a large portion of questions. Figure 6.4 presents also the distribution of frequent trigram prefixes. While nearly half of SQuAD questions are "what" questions [128], the distribution of ArchivalQA is more evenly spread across multiple question types.

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

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

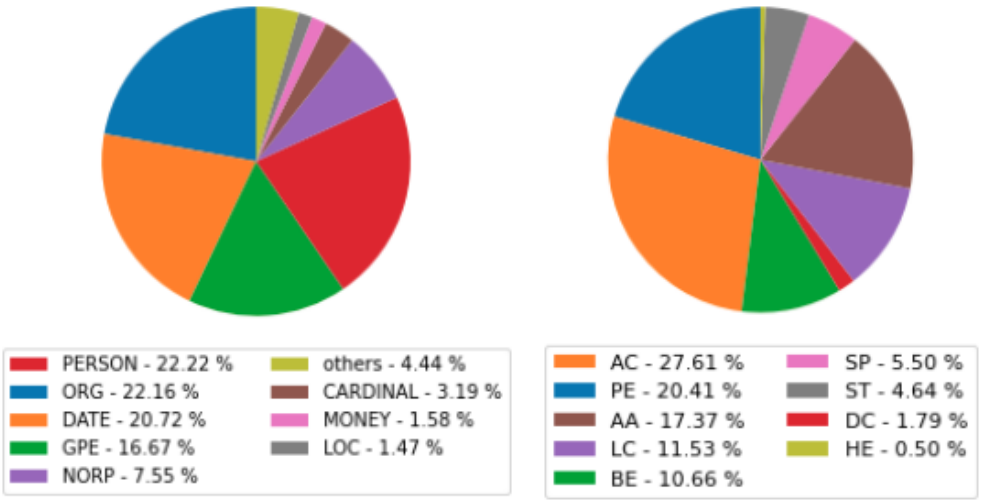

Figure 6.3. Left: Answers' named entity distribution ("others": named entities that account for a very small part (< 1%)). Right: Questions' category distribution ("AC": "arts & culture", "PE": "politics & elections", "AA": "armed conflicts & attacks", "LC": "law and crime", "BE": "business & economy", "SP": "sport", "ST": "science & technology", "DC": "disasters & accidents", "HE": "health & environment").

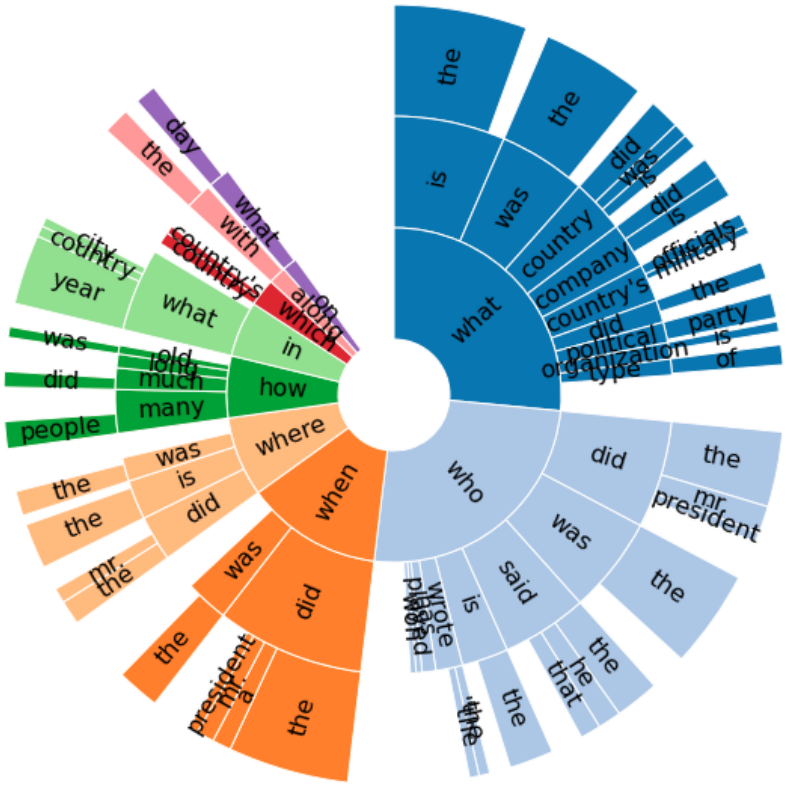

Figure 6.4. Trigram prefixes of ArchivalQA questions

Table 6.4. ArchivalQA Dataset Examples. org_answer, answer_start, trans_que, trans_ans, and source represent the original answer text, its start index in the document, flag indicating whether the question has been transformed, flag showing whether the answer has been transformed and the selection method of the document used for producing the question, respectively. para_id contains concatenated information of the document ID (the metadata of each article in the NYT corpus) and the ith paragraph used to generate the question.

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

6.3.2 Model Performance

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

1. DrQA-Wiki [22]: DrQA combines a search component based on bigram hashing and TF-IDF matching with a multi-layer recurrent neural network model trained to extract answers from articles. We first test the DrQA model which uses Wikipedia as the knowledge source (DrQA's default knowledge source). With this setting we would like to test if Wikipedia alone could be sufficient for answering questions about the historical events.

2. DrQA-NYT [22]: DrQA model which uses NYT.

3. DrQA-NYT-TempRes [22]: DrQA model which uses NYT and transforms the answers with relative temporal information by an approach similar to the one we used for transforming relative temporal information in Syntactic & Temporal Filtering/Transforming Module (see relative temporal information transformation steps of Section 6.2.3).

Table 6.5. Models' performance on ArchivalQA

Model	**EM**	**F1**
DrQA-Wiki [22]	7.53	11.64
DrQA-NYT [22]	38.13	46.12
DrQA-NYT-TempRes [22]	44.84	53.06
BERTserini-Wiki [171]	10.19	16.25
BERTserini-NYT [171]	54.30	66.05
BERTserini-NYT-TempRes [171]	**56.34**	**68.93**
DPR-NYT [67]	44.30	56.64
DPR-NYT-TempRes [67]	49.27	60.72

4. BERTserini-Wiki [171]: BERTserini tackles end-to-end question answering by combining BERT [28] with the Anserini [170] IR toolkit, with BM25 as the ranking function. We also first test BERTserini model using Wikipedia (BERTserini's default knowledge source).

5. BERTserini-NYT [171]: BERTserini model which uses NYT.

6. BERTserini-NYT-TempRes [171]: BERTserini model which uses NYT archive and transforms the relative temporal answers.

7. DPR-NYT [67][‖]: Unlike previous ODQA approaches, this end-to-end QA model incorporates BERT [28] reader module[**] with dense retriever module that has been trained for 15 epochs using ArchivalQA dataset and NYT corpus. In the retriever module, the paragraphs and questions are represented by dense vector representations, computed using two BERT networks. The ranking function is given by the dot product between the query and passage representations.

8. DPR-NYT-TempRes [67]: DPR model which uses NYT archive and transforms the relative temporal answers.

We measure the performance of the above-listed models using exact match (EM) and F1 score - the two standard measures commonly used in QA research.

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

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

Table 6.6. Human evaluation results of ArchivalQA

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

The results of all the models are given in Table 6.5. Firstly, we can observe that the models that utilize Wikipedia as the knowledge source perform much worse than the models that use NYT corpus, which is due to many questions being about minor things or events that Wikipedia does not seem to record (or it describes them only shallowly). Secondly, the models that resolve implicit temporal answers perform better than the ones without this step. Temporal information resolution is then clearly important. Thirdly, we notice that BERTserini models outperform DrQA models by large margins. There are two possible reasons, one is that DrQA models retrieve the entire long articles containing many non-relevant sentences rather than short paragraphs; the other is that DrQA uses RNN-base reader component rather than a better choice which would be the BERT-base reader component. Finally, DPR models which use dense vector representations for retrieval also achieve relatively good results on both metrics. Future work on explicitly incorporating ODQA models with temporal information (e.g., timestamp information) or on combining dense retrieval with sparse retrieval could be studied to further improve the performance.

6.3.3 Human Evaluation

We finally conduct human evaluation on ArchivalQA to study the quality of the generated questions. We randomly sampled 5K question-answer pairs along with their original paragraphs and publication dates and asked 10 graduate students for their evaluation. The evaluators were requested to rate the generated questions from 1 (very bad) to 5 (very good) on four criteria: *Fluency* measures if a question is grammatically correct and is fluent to read. *Answerability* indicates if a question can be answered by the given answer. *Relevance* measures whether a question is grounded in the given passage, while *Non-ambiguity* defines if a question is non-ambiguous. The average scores for each evaluation metric are shown in Table 6.6. Our model achieves high performance over all the metrics, especially on *Fluency* and *Relevance*. In addition, the *Non-ambiguity* result is high, indicating that large majority of the questions are non-ambiguous.

Table 6.7. Statistics of the dataset used in Triple-based Filtering

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

We then examine the effectiveness of the General & Temporal Ambiguity Filtering Module by analyzing reasons as for why 10 annotators labelled 10k data samples as "Bad" for the Triple-based Filtering Module. As shown in Table 6.7, among 10k questions, there are 390 (3.90%) questions labelled as "Bad" due to specificity problems, and 806 (8.06%) questions have temporal ambiguity problems.[††] These relatively small numbers suggest that the General & Temporal Ambiguity Filtering Module should have removed most of the questions with specificity or ambiguity issues. The final filtering step using the Triple-based Filtering Module is supposed to remove the remaining "Bad" questions by analyzing $\langle question, answer, paragraph \rangle$ at the same time.

6.4 Sub-Dataset Creation

We also distinguish subparts of the dataset based on the question difficulty levels and the containment of temporal expressions, which we believe could be used for training/testing ODQA systems with diverse strengths and abilities. Table 6.8 presents few randomly sampled examples for each of the four subdivisions of our dataset which we describe below.

6.4.1 Difficult/Easy Questions Dataset

We created two sub-datasets (called ArchivalQAEasy and ArchivalQAHard) based on the difficulty levels of their questions, such that 100,000 are easy and another 100,000 are difficult questions. We use the open-source Anserini IR toolkit with BM25 as the ranking function to create these subsets. The samples are labeled

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

Table 6.8. ArchivalQA Sub-Dataset Examples

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

Table 6.9. Performance of different models over different Sub-Datasets

Model	ArchivalQAEasy		ArchivalQAHard		ArchivalQATime		ArchivalQANoTime	
	EM	F1	EM	F1	EM	F1	EM	F1
DrQA-NYT [22]	42.10	51.97	22.81	31.24	31.32	42.17	39.59	47.18
DrQA-NYT-TempRes [22]	48.41	57.26	27.37	34.02	33.19	44.01	46.39	54.91
BERTserini-NYT [171]	59.15	69.16	25.00	33.73	50.65	63.24	55.36	68.37
BERTserini-NYT-TempRes [171]	**61.80**	**71.56**	29.88	38.44	**51.12**	**65.67**	**58.27**	**70.19**
DPR-NYT [67]	46.24	59.63	39.99	48.03	42.29	53.73	45.28	57.92
DPR-NYT-TempRes [67]	52.10	64.51	**41.65**	**48.96**	42.91	54.27	51.13	62.75

as easy if the paragraphs used to generate the questions appeared within the top 10 retrieved documents; otherwise they are considered difficult. We then partitioned both these two sub-datasets randomly into the training set (80%, 80,000 examples), the development set (10%, 10,000 examples), and the test set (10%, 10,000 examples).

6.4.2 Division based on Time Expressions

We created the next two sub-datasets based on the temporal characteristics of their questions. In particular, we constructed two sub-datasets containing 75,000 questions with temporal expressions and 75,000 without temporal expressions (called ArchivalQATime and ArchivalQANoTime, respectively). We used SUTime [21] combined with our handcrafted rules to collect the former questions, while the latter were randomly chosen questions without temporal expressions.

Note that questions with temporal expressions should let ODQA systems limit the search time scope from the entire time frame of the news archive to the narrower time periods specified by the temporal expressions contained in these questions. For example, for the question *"Which team won the 1990 World Series?"*, the accurate answers could be just searched within news articles published during (or perhaps also some time after) 1990. Same as with ArchivalQAEasy and ArchivalQAHard, both ArchivalQATime and ArchivalQANoTime were randomly split into the training (80%, 60,000 examples), development (10%, 7,500 examples), and test sets (10%, 7,500 examples).

6.4.3 Model Performance on Sub-Datasets

Table 6.9 presents the performance of different ODQA models over the four sub-datasets discussed above. We can see that all the models achieve better results on ArchivalQAEasy than on ArchivalQAHard, indicating that the questions of ArchivalQAHard tend to be indeed harder to answer. For example, the improvement of BERTserini-NYT-TempRes is in the range of 106.83% and 86.16% on EM and F1 metrics, respectively. However, DPR models using dense vector representations for retrieving relevant paragraphs are subject to a small performance drop on two sub-datasets (ArchivalQAEasy and ArchivalQAHard) and they manage to surpass the other ODQA approaches that use sparse retrievers by large margins on ArchivalQAHard. For example, when considering DPR-NYT-TempRes model on ArchivalQAHard and ArchivalQAEasy, the improvements are only 25.09% and 31.76% on EM and F1, respectively. When comparing DPR-NYT-TempRes with BERTserini-NYT-TempRes on ArchivalQAHard, the improvements are 39.39% and 27.37% on EM and F1 metrics, respectively. This is likely because questions in ArchivalQAHard contain less lexical overlap with the NYT articles while DPR excels at semantic representation and handles lexical variations well. When considering ArchivalQATime and ArchivalQANoTime, the models perform slightly better on ArchivalQANoTime. A possible reason for that can be that such temporal signals are currently just used as usual textual information (rather than being utilized as time selectors) which can even cause harm, despite the fact that time expressions actually constitute an important feature. Future models should pay special attention to such important temporal signals to find more relevant documents, which has been widely leveraged in temporal information retrieval

[4, 17, 61]. QANA model Wang et al. [161, 162] we introduced previously, has already used such temporal signals to answer temporally-scoped questions about the past.

6.5 Dataset Use

Our dataset can be used in several ways. First, ODQA models can use the questions, answers and paragraphs[‡‡] for training their IR and MRC modules [32, 67] on a novel kind of data that poses challenges in terms of highly changing contexts of different years, high temporal periodicity of events and rich temporal signals in terms of document timestamps and temporal expressions embedded in document content, for example, training ODQA models with time-aware dense retriever components that use the important temporal signals (e.g., timestamp information). As shown in chapter 3, the proposed QANA model [161, 162], that utilize such complex temporal signals (using Temporal IR approaches or others) achieve better results than other ODQA approaches. In addition, it is now possible to further improve QANA by replacing its sparse retriever module with dense retriever module.

When it comes to the underlying news dataset, most systems would use our QA pairs against the NYT corpus. They might however potentially use other temporal news collections that temporally align with the NYT collection (i.e., ones that also span 1987-2007), although naturally this would make the task more challenging. It might be even feasible to consider answering our questions using synchronic knowledge bases such as Wikipedia, although as we have observed earlier, Wikipedia seems to lack a lot of detailed information on the past. The questions in our dataset are often specific and minor, and relate to relatively old events, hence they may be different than questions in other popular ODQA datasets. Such questions can be particularly valuable considering that the true utility of QA systems lies in answering hard questions that humans cannot (at least easily) answer by themselves. Finally, system testing and comparison can be made to be more fine-grained based on the question difficulty and the occurrence of temporal components contained in questions. Also, another practical application could be to use our generated questions for education, e.g., for evaluating

[‡‡]Note that another way to use the dataset is to train models without using the paragraph information [85].

students knowledge and stimulating self-learning in history courses.

6.6 Summary

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

In the future, we will further improve QANA by replacing its sparse retriever module with dense retriever module by using ArchivalQA dataset. In addition, we plan to extend our dataset by incorporating also multi-hop questions in order to foster multi-hop question answering research [103] on archival news collections.

CHAPTER 7

Conclusion and Future Work

7.1 Conclusion

In this thesis, we focus on the temporal news collections and we aim to benefit from better utilization of such valuable resources. In addition, time, that could be leveraged to organize and search relevant information, is one of the most significant dimensions especially in news domain. To this end, we first propose three distinct methods of addressing different research problems by exploiting temporal information over temporal historical collections. In the final topic, we additionally construct a large-scale ODQA dataset over temporal news collections, with the objective to promote the development of QA research over news archives. The contributions in the four research topics described in this dissertation are listed as follows.

1. **Exploiting Temporal Information in Question Answering.**

    - We describe a novel subtask of QA, which uses temporal news collections as the knowledge source.
    - We introduce an effective ODQA system called QANA for answering event-related questions over temporal news collections, by exploiting diverse temporal characteristics of both questions and documents. In

addition, it is also the first study to adapt and improve concepts from temporal information retrieval to the QA research domain.

- We create and provide the test sets for automatically answering questions about the history.
- We conduct extensive experimental evaluation of our proposed model using dedicated test sets and a document collection spanning 20 years.

2. **Exploiting Temporal Information in Event Occurrence Time Estimation.**

   - We propose a novel TEP-Trans model based on Transformer architecture and multivariate time series analysis which is able to estimate the event occurrence time at different temporal granularities based on a long-term news archive as the underlying knowledge source.
   - We construct a large dataset of past events and perform extensive experiments to prove the effectiveness of our model.
   - We show that our model can be successfully applied on the downstream IR/NLP tasks such as ODQA task to further improve their performance.

3. **Exploiting Temporal Information in Constructing Time-aware Language Representation.**

   - We investigate the effectiveness of incorporating temporal information into pre-trained language models using different pre-training tasks, and we demonstrate that injecting such information via specially designed time-oriented pre-training tasks can benefit various downstream time-related tasks.
   - We propose a novel pre-trained language model called TimeBERT, which is trained through two new pre-training tasks that involve two kinds of temporal aspects. To our best knowledge, this is the first work to investigate both types of temporal information (timestamp and content time signals in news articles) when constructing language models.
   - We conduct extensive experiments on diverse time-related tasks that involve the two temporal dimensions of documents or queries. The results demonstrate that TimeBERT achieves a new SOTA performance, and

thus has capability to be successfully applied in many applications for which time is crucial.

4. **Creating a Large-scale ODQA Dataset over Temporal News Collections.**

   - We propose one of the largest ODQA datasets for news collections, which is not only spanning the longest time period compared to other QA datasets, but it also provides detailed questions on the events that occurred from 14 to 34 years ago.
   - We propose an approach to generate large datasets in an inexpensive way, whose resulting questions tend to be non-ambiguous and of good quality, thus having only a single potential answer. Compared with other QG methods, most of our generated questions are clear and non-ambiguous, and thus they can be especially useful in education, e.g., to support generating questions for exams.
   - We undertake comprehensive analysis of the generated dataset, which does not only show the quality and utility of the resulting data, but also proves the effectiveness of our QG framework.

7.2 Future Directions

The four research topics in this thesis are intended to inspire more interests and attention in methods using temporal news collections, especially those exploiting two distinct temporal information. Several promising research avenues can be further explored in future work.

1. **Enriching collected temporal information by leveraging temporal relations.** The temporal relations associated with the temporal expressions are important, which can denote explicit temporal relations held between two abstract entities (time and event, time and time, or event and event). For example, "before 1999", "until Sunday", etc. Considering the temporal relations is a very interesting work, that enriches temporal information which might be colelcted and might further improve the performance of models in different tasks.

2. **Utilizing external knowledge sources to obtain more relevant information.** In the four topics, only a single knowledge source is used, that might contain limited information in some events. Utilizing more knowledge sources and exploring effective approaches of combining knowledge sources is a potential and interesting direction, that can improve the model performance. For example, federated QA systems over multiple news archives, could likely surpass QANA in answering questions of temporal nature, or Wikidate temporal information could be utilized in addition to information collected from raw text.

3. **Investigating different roles of temporal information in different articles.** There are various types of articles in temporal news collections, such as sports, politics, economy, etc. The temporal information (timestamp and content time) plays different roles in different types of articles and considering the differences of temporal information in these domains or genres can help us know the better utilization of such information.

4. **Building ODQA datasets of multi-hop question answering over temporal news collections.** Multi-hop question answering is one of the most researched tasks over the recent years. The ability to answer multi-hop questions and perform multi-step reasoning could significantly improve the utility of NLP systems. In the future we plan to extend our ArchivalQA dataset by incorporating multi-hop questions in order to foster multi-hop QA research on temporal news collections.

Acknowledgements

Although only my name appears on the cover, many people have contributed to the completion of this dissertation, that I would like to express my sincere gratitude to them who have made this thesis possible.

First and foremost I would like to express my deepest gratitude to my supervisor, Professor Masatoshi Yoshikawa, for his careful and kind guidance and constant support in overcoming numerous obstacles I have been facing through my PhD study. I am also very thankful to him for giving me this lifetime opportunity to study in Kyoto University, that the five years in Japan has been one of the most enriching experience of my life and I will forever cherish the memories and friends I have made here. I hope that one day I would become as good an advisor to others as he has been to me.

My special thanks goes to my co-supervisor, Professor Adam Jatowt, who is now working at the Department of Computer Science in University of Innsbruck. I am deeply grateful to him and touched by his inspiring ideas, consistent guidance and many nights reviewing and modifying our work delicately. We usually discussed the research problems twice a week and it was very enjoyable talking with him, that I learned enormously from the discussions. Most importantly, he helped me to find the joy of doing research and inspired me to grow as a research scientist.

I am deeply grateful to my advisors, Professor Keishi Tajima and Associate Professor Donghui Lin for their guidance and encouragement throughout the doctoral course. They gave me lost of valuable comments and academic advice, that helped me to look at the problems more comprehensively. I would like to show my great appreciation to Professor Sadao Kurohashi for being a member of the thesis committee.

I would also like to thank the stuff members and all students in the Yoshikawa & Ma Laboratory for their help in both academic areas and my daily life. Thanks to Associate Professor Qiang Ma, Assistant Professor Kazunari Sugiyama and Assistant Professor Yang Cao, for having enlightening discussions with me on my research. Thanks to all the lab mates for making the valuable comments and for all the fun we have had in the lab. Thanks to the lab secretaries: Ms. Chie Nomura, Ms. Yoko Nukii, Ms. Yoko Nakahara and Ms. Michiyo Kai, for their hard working and kind support. I really appreciated Ms. Nomura and Ms. Nukii who helped me much in applying research funding and dealing with various procedures.

I want to express my appreciation to Nishimura International Scholarship Foundation and "Support for Pioneering Graduate Students presented by the Kyoto University Graduate Division", for their financial supports which greatly reduce my financial burden.

Finally, I must express my very profound gratitude to my parents Danhua Wu and Linfeng Wang, for their unconditional love and strong support in my life, without which all my achievements would not be possible. Lastly, thanks to my better half, Simin Ouyang, for her continuous encouragement and unfailing love.

Jiexin Wang, June 2022

Bibliography

[1] Prabal Agarwal, Jannik Strötgen, Luciano Del Corro, Johannes Hoffart, and Gerhard Weikum. Dianed: time-aware named entity disambiguation for diachronic corpora. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 686–693, 2018.

[2] 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.

[3] Omar Alonso, Michael Gertz, and Ricardo Baeza-Yates. Clustering and exploring search results using timeline constructions. In *Proceedings of the 18th ACM conference on Information and knowledge management*, pages 97–106, 2009.

[4] Omar Alonso, Jannik Strötgen, Ricardo Baeza-Yates, and Michael Gertz. Temporal information retrieval: Challenges and opportunities. *Twaw*, 11: 1–8, 2011.

[5] Sophia Althammer, Mark Buckley, Sebastian Hofstätter, and Allan Hanbury. Linguistically informed masking for representation learning in the patent domain. *arXiv preprint arXiv:2106.05768*, 2021.

[6] Giuseppe Amodeo, Giambattista Amati, and Giorgio Gambosi. On relevance, time and query expansion. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, pages 1973–1976, 2011.

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

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

[9] Cristina Barros, Elena Lloret, Estela Saquete, and Borja Navarro-Colorado. Natsum: Narrative abstractive summarization through cross-document timeline generation. *Information Processing & Management*, 56(5):1775–1793, 2019.

[10] Iz Beltagy, Kyle Lo, and Arman Cohan. Scibert: A pretrained language model for scientific text. *arXiv preprint arXiv:1903.10676*, 2019.

[11] Klaus Berberich and Srikanta Bedathur. Temporal diversification of search results. In *Proceedings of the SIGIR 2013 workshop on time-aware information access*, 2013.

[12] 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.

[13] 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.

[14] Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020.

[15] Ricardo Campos, Alípio Mário Jorge, Gaël Dias, and Célia Nunes. Disambiguating implicit temporal queries by clustering top relevant dates in web snippets. In *2012 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology*, volume 1, pages 1–8. IEEE, 2012.

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

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

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

[19] Ricardo Campos, Arian Pasquali, Adam Jatowt, Vítor Mangaravite, and Alípio Mário Jorge. Automatic generation of timelines for past-web events. In *The Past Web*, pages 225–242. Springer, 2021.

[20] Nathanael Chambers. Labeling documents with timestamps: Learning from their time expressions. Technical report, NAVAL ACADEMY ANNAPOLIS MD DEPT OF COMPUTER SCIENCE, 2012.

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

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

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

[24] Na Dai and Brian D Davison. Freshness matters: in flowers, food, and web authority. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, pages 114–121, 2010.

[25] Wisam Dakka, Luis Gravano, and Panagiotis Ipeirotis. Answering general time-sensitive queries. *IEEE Transactions on Knowledge and Data Engineering*, 24(2):220–235, 2010.

[26] Supratim Das, Arunav Mishra, Klaus Berberich, and Vinay Setty. Estimating event focus time using neural word embeddings. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 2039–2042, 2017.

[27] Leon Derczynski. *Automatically Ordering Events and Times in Text*, volume 677. 01 2017. ISBN 978-3-319-47240-9. doi: 10.1007/978-3-319-47241-6.

[28] 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.

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

[30] Bhuwan Dhingra, Jeremy R Cole, Julian Martin Eisenschlos, Daniel Gillick, Jacob Eisenstein, and William W Cohen. Time-aware language models as temporal knowledge bases. *arXiv preprint arXiv:2106.15110*, 2021.

[31] Gaël Dias, José G Moreno, Adam Jatowt, and Ricardo Campos. Temporal web image retrieval. In *International Symposium on String Processing and Information Retrieval*, pages 199–204. Springer, 2012.

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

[33] Anlei Dong, Yi Chang, Zhaohui Zheng, Gilad Mishne, Jing Bai, Ruiqiang Zhang, Karolina Buchner, Ciya Liao, and Fernando Diaz. Towards recency ranking in web search. In *Proceedings of the third ACM international conference on Web search and data mining*, pages 11–20, 2010.

[34] Xinya Du and Claire Cardie. Event extraction by answering (almost) natural questions, 2020.

[35] Nan Duan, Duyu Tang, Peng Chen, and Ming Zhou. Question generation for question answering. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 866–874, 2017.

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

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

[38] Jonathan L Elsas and Susan T Dumais. Leveraging temporal dynamics of document content in relevance ranking. In *Proceedings of the third ACM international conference on Web search and data mining*, pages 1–10, 2010.

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

[40] Eric Foner. *Free Soil, Free Labor, Free Men: The Ideology of the Republican Party Before the Civil War: With a New Introductory Essay*. OUP USA, 1995.

[41] 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.

[42] Yifan Gao, Lidong Bing, Wang Chen, Michael R Lyu, and Irwin King. Difficulty controllable generation of reading comprehension questions. *arXiv preprint arXiv:1807.03586*, 2018.

[43] Demian Gholipour Ghalandari and Georgiana Ifrim. Examining the state-of-the-art in news timeline summarization. *arXiv preprint arXiv:2005.10107*, 2020.

[44] Mario Giulianelli, Marco Del Tredici, and Raquel Fernández. Analysing lexical semantic change with contextualised word representations. *arXiv preprint arXiv:2004.14118*, 2020.

[45] Dhruv Gupta and Klaus Berberich. Identifying time intervals of interest to queries. In *Proceedings of the 23rd ACM international conference on conference on information and knowledge management*, pages 1835–1838, 2014.

[46] Dhruv Gupta and Klaus Berberich. Diversifying search results using time. In *European Conference on Information Retrieval*, pages 789–795. Springer, 2016.

[47] Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. Realm: Retrieval-augmented language model pre-training. *arXiv preprint arXiv:2002.08909*, 2020.

[48] 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.

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

[50] Or Honovich, Lucas Torroba Hennigen, Omri Abend, and Shay B Cohen. Machine reading of historical events. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7486–7497, 2020.

[51] Wenpeng Hu, Bing Liu, Jinwen Ma, Dongyan Zhao, and Rui Yan. Aspect-based question generation. 2018.

[52] Kexin Huang, Jaan Altosaar, and Rajesh Ranganath. Clinicalbert: Modeling clinical notes and predicting hospital readmission. *arXiv preprint arXiv:1904.05342*, 2019.

[53] Adam Jatowt and Ching-man Au Yeung. Extracting collective expectations about the future from large text collections. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, pages 1259–1264, 2011.

[54] Adam Jatowt, Ching-Man Au Yeung, and Katsumi Tanaka. Estimating document focus time. In *Proceedings of the 22nd ACM International Conference on Information and Knowledge Management*, page 2273–2278, 2013.

[55] Adam Jatowt, Ching-Man Au Yeung, and Katsumi Tanaka. Estimating document focus time. In *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*, pages 2273–2278, 2013.

[56] 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.

[57] Rosie Jones and Fernando Diaz. Temporal profiles of queries. *ACM Transactions on Information Systems (TOIS)*, 25(3):14–es, 2007.

[58] FM Jong, Henning Rode, and Djoerd Hiemstra. Temporal language models for the disclosure of historical text. 2005.

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

[60] Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S Weld, Luke Zettlemoyer, and Omer Levy. Spanbert: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77, 2020.

[61] Nattiya Kanhabua and Avishek Anand. Temporal information retrieval. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, pages 1235–1238, 2016.

[62] Nattiya Kanhabua and Kjetil Nørvåg. Improving temporal language models for determining time of non-timestamped documents. In *International conference on theory and practice of digital libraries*, pages 358–370. Springer, 2008.

[63] Nattiya Kanhabua and Kjetil Nørvåg. Using temporal language models for document dating. In *Joint European conference on machine learning and knowledge discovery in databases*, pages 738–741. Springer, 2009.

[64] 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.

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

[66] Nattiya Kanhabua, Roi Blanco, and Kjetil Nørvåg. Temporal information retrieval. *Foundations and Trends® in Information Retrieval*, 9(2):91–208, 2015. ISSN 1554-0669. doi: 10.1561/1500000043. URL `http://dx.doi.org/10.1561/1500000043`.

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

[68] Pei Ke, Haozhe Ji, Siyang Liu, Xiaoyan Zhu, and Minlie Huang. Sentilare: Sentiment-aware language representation learning with linguistic knowledge. *arXiv preprint arXiv:1911.02493*, 2019.

[69] Tae-Young Kim and Sung-Bae Cho. Predicting the household power consumption using cnn-lstm hybrid networks. In *International Conference on Intelligent Data Engineering and Automated Learning*, pages 481–490. Springer, 2018.

[70] Tae-Young Kim and Sung-Bae Cho. Predicting residential energy consumption using cnn-lstm neural networks. *Energy*, 182:72–81, 2019.

[71] Yanghoon Kim, Hwanhee Lee, Joongbo Shin, and Kyomin Jung. Improving neural question generation using answer separation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6602–6609, 2019.

[72] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

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

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

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

[76] 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. doi: 10.1145/3295750.3298931. URL `http://doi.acm.org/10.1145/3295750.3298931`.

[77] Dimitrios Kotsakos, Theodoros Lappas, Dimitrios Kotzias, Dimitrios Gunopulos, Nattiya Kanhabua, and Kjetil Nørvåg. A burstiness-aware approach for document dating. In *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*, pages 1003–1006, 2014.

[78] Ghader Kurdi, Jared Leo, Bijan Parsia, Uli Sattler, and Salam Al-Emari. A systematic review of automatic question generation for educational purposes. *International Journal of Artificial Intelligence in Education*, 30(1): 121–204, 2020.

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

[80] Guillaume Lample and Alexis Conneau. Cross-lingual language model pretraining. *arXiv preprint arXiv:1901.07291*, 2019.

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

[82] Theodoros Lappas, Benjamin Arai, Manolis Platakis, Dimitrios Kotsakos, and Dimitrios Gunopulos. On burstiness-aware search for document sequences. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 477–486, 2009.

[83] Jinhyuk Lee, Seongjun Yun, Hyunjae Kim, Miyoung Ko, and Jaewoo Kang. Ranking paragraphs for improving answer recall in open-domain question answering. *arXiv preprint arXiv:1810.00494*, 2018.

[84] Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4): 1234–1240, 2020.

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

[86] Artuur Leeuwenberg and Marie-Francine Moens. Temporal information extraction by predicting relative time-lines. *arXiv preprint arXiv:1808.09401*, 2018.

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

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

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

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

[91] 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.

[92] Xin Li and Dan Roth. Learning question classifiers. In *Proceedings of the 19th international conference on Computational linguistics-Volume 1*, pages 1–7. Association for Computational Linguistics, 2002.

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

[94] Min Lin, Qiang Chen, and Shuicheng Yan. Network in network. *arXiv preprint arXiv:1312.4400*, 2013.

[95] Tao Lin, Tian Guo, and Karl Aberer. Hybrid neural networks for learning the trend in time series. In *Proceedings of the twenty-sixth international joint conference on artificial intelligence*, number CONF, pages 2273–2279, 2017.

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

[97] Shanshan Liu, Xin Zhang, Sheng Zhang, Hui Wang, and Weiming Zhang. Neural machine reading comprehension: Methods and trends. *Applied Sciences*, 9(18):3698, 2019.

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

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

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

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

[102] S. Martschat and M. Katja. A temporally sensitive submodularity framework for timeline summarization. In *CoNLL*, pages 230–240, 2018.

[103] Vaibhac Mavi, Anubhav Jangra, and Adam Jatowt. A survey on multi-hop question answering and generation. *arXiv preprint arXiv:2204.09140*, 2022.

[104] Bryan McCann, Nitish Shirish Keskar, Caiming Xiong, and Richard Socher. The natural language decathlon: Multitask learning as question answering. *arXiv preprint arXiv:1806.08730*, 2018.

[105] 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.

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

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

[108] Arunav Mishra and Klaus Berberich. Event digest: A holistic view on past events. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, pages 493–502, 2016.

[109] Dan Moldovan, Christine Clark, and Sanda Harabagiu. Temporal context representation and reasoning. In *INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE*, volume 19, page 1099. Citeseer, 2005.

[110] Christian Morbidoni, Alessandro Cucchiarelli, and Domenico Ursino. Leveraging linked entities to estimate focus time of short texts. In *Proceedings of the 22nd International Database Engineering & Applications Symposium*, pages 282–286, 2018.

[111] Sebastian Nagel. Cc-news. *URL: http://web. archive. org/save/http://commoncrawl. org/2016/10/newsdatasetavailable*, 2016.

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

[113] Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. Ms marco: A human generated machine reading comprehension dataset. *arXiv preprint arXiv:1611.09268*, 2016.

[114] 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.

[115] Kai Niklas. Unsupervised post-correction of ocr errors. *Master's thesis. Leibniz Universität Hannover*, 2010.

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

[117] Marius Pasca. Towards temporal web search. In *SAC*, page 1117–1121, 2008. ISBN 9781595937537.

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

[119] Maria-Hendrike Peetz, Edgar Meij, and Maarten de Rijke. Using temporal bursts for query modeling. *Information retrieval*, 17(1):74–108, 2014.

[120] 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.

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

[122] James Pustejovsky, Robert Knippen, Jessica Littman, and Roser Saurí. Temporal and event information in natural language text. *Language resources and evaluation*, 39(2-3):123–164, 2005.

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

[124] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[125] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*, 2019.

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

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

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

[129] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*, 2019.

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

[131] Shruti Rijhwani and Daniel Preotiuc-Pietro. Temporally-informed analysis of named entity recognition. In *Proceedings of ACL*, pages 1–13, 2020.

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

[133] Guy D Rosin and Kira Radinsky. Temporal attention for language models. *arXiv preprint arXiv:2202.02093*, 2022.

[134] Guy D Rosin, Ido Guy, and Kira Radinsky. Time masking for temporal language models. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*, pages 833–841, 2022.

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

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

[137] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019.

[138] Estela Saquete, J Luis Vicedo, Patricio Martínez-Barco, Rafael Munoz, and Hector Llorens. Enhancing qa systems with complex temporal question processing capabilities. *Journal of Artificial Intelligence Research*, 35:775–811, 2009.

[139] Estela Saquete Boró, Patricio Martinez-Barco, Rafael Munoz, Jose-Luis Vicedo, et al. Splitting complex temporal questions for question answering systems. Association for Computational Linguistics (ACL), 2004.

[140] Apoorv Saxena, Soumen Chakrabarti, and Partha Talukdar. Question answering over temporal knowledge graphs. *arXiv preprint arXiv:2106.01515*, 2021.

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

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

[143] Shashank Shrivastava, Mitesh Khapra, and Sutanu Chakraborti. A concept driven graph based approach for estimating the focus time of a document. In *International Conference on Mining Intelligence and Knowledge Exploration*, pages 250–260. Springer, 2017.

[144] Jaspreet Singh, Wolfgang Nejdl, and Avishek Anand. History by diversity: Helping historians search news archives. *Proceedings of the 2016 ACM Conference on Human Information Interaction and Retrieval, CHIIR 2016, Carrboro, North Carolina, USA, March 13-17, 2016*, pages 183–192, 2016.

[145] 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.

[146] Michael Stack. Full text search of web archive collections. *Proc. of IWAW*, 2006.

[147] Julius Steen and Katja Markert. Abstractive timeline summarization. In *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, pages 21–31, 2019.

[148] Jannik Strötgen and Michael Gertz. Event-centric search and exploration in document collections. In *Proceedings of JCDL*, pages 223–232, 2012.

[149] Jannik Strötgen and Michael Gertz. Temporal tagging on different domains: Challenges, strategies, and gold standards. In *LREC*, volume 12, pages 3746–3753, 2012.

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

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

[152] Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Xuyi Chen, Han Zhang, Xin Tian, Danxiang Zhu, Hao Tian, and Hua Wu. Ernie: Enhanced representation through knowledge integration. *arXiv preprint arXiv:1904.09223*, 2019.

[153] Goutham Swapna, Soman Kp, and Ravi Vinayakumar. Automated detection of diabetes using cnn and cnn-lstm network and heart rate signals. *Procedia computer science*, 132:1253–1262, 2018.

[154] Giang Tran, Mohammad Alrifai, and Eelco Herder. Timeline summarization from relevant headlines. In *ECIR*, pages 245–256. Springer, 2015.

[155] Tuan A Tran, Claudia Niederée, Nattiya Kanhabua, Ujwal Gadiraju, and Avishek Anand. Balancing novelty and salience: Adaptive learning to rank entities for timeline summarization of high-impact events. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, pages 1201–1210, 2015.

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

[157] Shikhar Vashishth, Shib Sankar Dasgupta, Swayambhu Nath Ray, and Partha Talukdar. Dating documents using graph convolution networks. *arXiv preprint arXiv:1902.00175*, 2019.

[158] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[159] 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.

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

[161] Jiexin Wang, Adam Jatowt, Michael Färber, and Masatoshi Yoshikawa. Answering event-related questions over long-term news article archives. In *European Conference on Information Retrieval*, pages 774–789. Springer, 2020.

[162] Jiexin Wang, Adam Jatowt, Michael Färber, and Masatoshi Yoshikawa. Improving question answering for event-focused questions in temporal collections of news articles. *Information Retrieval Journal*, pages 1–26, 2021.

[163] Jiexin Wang, Adam Jatowt, and Masatoshi Yoshikawa. Event occurrence date estimation based on multivariate time series analysis over temporal document collections. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 398–407, 2021.

[164] 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.

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

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

[167] Wenhan Xiong, Jingfei Du, William Yang Wang, and Veselin Stoyanov. Pretrained encyclopedia: Weakly supervised knowledge-pretrained language model. *arXiv preprint arXiv:1912.09637*, 2019.

[168] Rui Yan, Xiaojun Wan, Jahna Otterbacher, Liang Kong, Xiaoming Li, and Yan Zhang. Evolutionary timeline summarization: a balanced optimization framework via iterative substitution. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, pages 745–754, 2011.

[169] An Yang, Quan Wang, Jing Liu, Kai Liu, Yajuan Lyu, Hua Wu, Qiaoqiao She, and Sujian Li. Enhancing pre-trained language representations with rich knowledge for machine reading comprehension. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2346–2357, 2019.

[170] Peilin Yang, Hui Fang, and Jimmy Lin. 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*, pages 1253–1256, 2017.

[171] 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.

[172] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, 32, 2019.

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

[174] Yi Yu, Adam Jatowt, Antoine Doucet, Kazunari Sugiyama, and Masatoshi Yoshikawa. Multi-timeline summarization (mtls): Improving timeline summarization by generating multiple summaries. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 377–387, 2021.

[175] MohammadSadegh Zahedi, Abolfazl Aleahmad, Maseud Rahgozar, Farhad Oroumchian, and Arastoo Bozorgi. Time sensitive blog retrieval using temporal properties of queries. *Journal of Information Science*, 43(1):103–121, 2017.

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

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

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

[179] Xin Wayne Zhao, Yanwei Guo, Rui Yan, Yulan He, and Xiaoming Li. Timeline generation with social attention. In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*, pages 1061–1064, 2013.

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

[181] Fengbin Zhu, Wenqiang Lei, Chao Wang, Jianming Zheng, Soujanya Poria, and Tat-Seng Chua. Retrieving and reading: A comprehensive survey on open-domain question answering. *arXiv preprint arXiv:2101.00774*, 2021.

[182] Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the IEEE international conference on computer vision*, pages 19–27, 2015.

Selected List of Publications

- **Journals**

  [1] Jiexin Wang, Adam Jatowt, Michael Farber and Masatoshi Yoshikawa. Improving Question Answering for Event-focused Questions in Temporal Collections of News Articles. *Information Retrieval Journal (IRJ)*, 24, 1 (2021), 29–54.

- **International Conferences**

  [2] Jiexin Wang, Adam Jatowt, Michael Farber and Masatoshi Yoshikawa. Answering event-related questions over long-term news article archives. *Proceedings of the 42nd European Conference on Information Retrieval (ECIR 2020)*, Springer LNCS, pp. 774-789 (2020). (Industry Impact Honorable Mention)

  [3] Jiexin Wang, Adam Jatowt, Masatoshi Yoshikawa. Event Occurrence Date Estimation based on Multivariate Time Series Analysis over Temporal Document Collections. *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2021)*, ACM Press, pp. 398-407 (2021).

  [4] Jiexin Wang, Adam Jatowt, Masatoshi Yoshikawa. ArchivalQA: A Large-scale Benchmark Dataset for Open-Domain Question Answering over Historical News Collections. *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2022).*

  [5] Jiexin Wang, Adam Jatowt, Masatoshi Yoshikawa. TimeBERT: Enhancing Pre-Trained Language Representations with Temporal Inform-

ation. *(Submitted).*

- **Domestic Conferences and Workshops**

  [6] Jiexin Wang, Adam Jatowt, Michael Farber and Masatoshi Yoshikawa. Answering Questions on Long-term News Article Archives. *The 11th Forum on Data Engineering and Information Management (DEIM), 2019. (Student Presentation Award)*

  [7] Jiexin Wang, Adam Jatowt, Masatoshi Yoshikawa. Answering "When" Questions over News Article Archives. *The 12th Forum on Data Engineering and Information Management (DEIM), 2020.*