TimeBERT: Enhancing Pre-Trained Language Representations with Temporal Information

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

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

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

ABSTRACT

Time is an important aspect of text documents, which has been widely exploited in natural language processing and has strong influence, for example, in temporal information retrieval, where the temporal information of queries or documents needs to be identified for relevance estimation. Event-related tasks like event ordering, which aims to order events by their occurrence time, needs to be widely exploited in natural language processing and has strong gains on different downstream NLP tasks or applications for which time is of importance.\(^1\)

CCS CONCEPTS

- Information systems → Content analysis and feature selection.

KEYWORDS

pre-trained language model, temporal information, temporal news collection, time-aware language representation

1 INTRODUCTION

Temporal signals constitute one of the most significant features for many types of text documents, for example, news articles. They can be leveraged to organize and search 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 [10, 25] 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 [14, 46], timeline summarization [3, 12, 35, 45, 50], event occurrence time prediction [55], temporal clustering and information retrieval [2, 10, 11], question answering [37, 53] and named entity recognition [1, 39] are other example tasks where utilizing temporal information proved beneficial.

Pre-trained transformer-based [52] language models such as BERT [15], XLNet [59], GPT [9, 38] 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 paper, we propose 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 our proposed model could simultaneously utilize 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.

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

1. We investigate the effectiveness of incorporating temporal information into pre-trained language representation 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 next 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 representation 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 state-of-the-art performance, and thus has capability
to be successfully applied in many applications for which
time is crucial.

The remainder of this paper is structured as follows. In the next
section, we overview the related work. In Section 3, we introduce
the details of our method. Section 4 describes experimental set-
tings, while Section 5 provides experimental results. In Section 6
we demonstrate how the proposed language model can improve
other downstream tasks or applications on the example of temporal
question answering. Finally, we conclude the paper and outline our
future work in Section 7.

2 RELATED WORK

According to Campos et al. [10], 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 (e.g., a document published today2
about WWW II would have its timestamp being May 19, 2022 and
the focus of time 1939-1945). Similarly, the query timestamp refers
to the time when the query was issued, while the focus time [22]
is the content time of the query or of the document (e.g., “winter
olympics 1978” query issued 3 month ago would have its timestamp
in February, 2022 and would refer to the time period when the
winter sports took place in Edmonton, Alberta in 1987). In addition,
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 [22]. In order to estimate the
correct focus time of a document, approaches need to evaluate the
time to which individual sentences refer. 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 updateliness. On the other hand, the focus time can help
to strengthen our understanding of document content; for
example, events developments and their causal relations can be
understood by analyzing the relations between different content
temporal information. In the recent years, exploiting these two
kinds of temporal information in documents and queries has been
 gaining increased importance in information retrieval and NLP.

Their interplay can be utilized to develop time-specific search
and exploration applications [4, 10, 25], such as temporal web search
[44], temporal question answering [53, 54], search results diversifi-
cation [8, 47] and clustering [3, 49], summarization [6], historical
event ordering [19] and so on.

BERT [15] has emerged as one of key breakthroughs that
contributed to the recent success and development of pre-trained lan-
}

2We use "2022/05/19" as the exact date of "today" being the submission deadline.
et al. [18] propose the first unsupervised approach to use contextu-
alized embeddings from BERT to model lexical semantic changes. 
Dhingra et al. [16] propose a simple modification to pre-training that 
parametrizes MLM with timestamp information using 
temporally-scoped knowledge, and test the proposed language 
model on question answering task. [40, 41] mainly address the tasks 
of temporal change detection, that needs to identify which words 
undergo semantic changes and to what extent. More specifically, 
Rosin and Radinsky [41] extend the self-attention mechanism of 
the transformer architecture [52] by incorporating timestamp in-
formation, which is used to compute attention scores. Rosin et al. 
[40] further train BERT by using the concatenation of timestamp 
and text sequences as input, which reaches state-of-the-art perfor-
mance on semantic change detection. As we can see, these models 
mainly focus on the problem of lexical semantic change and the 
timestamp at only the coarse granularity (i.e., year or even decade) 
is utilized, despite the fact that the content time actually constitutes 
an important feature.

Similar to the above pre-training models, our proposal is also a 
transformer-based [52] language representation model. However, 
unlike all the aforementioned approaches, it exploits both times-
tamp and content time during pre-training on a temporal news 
collection - following a design proposed to achieve good perfor-
mance on two downstream time-related tasks: event occurrence 
time estimation and document timestamp estimation. Building such 
a language model that can further help to make better use of the 
temporal information in various applications is of great importance, 
especially in temporal information retrieval field, timeline construc-
tion, question answering over temporal collections, and other NLP 
tasks that rely on temporal signals.

3 METHOD

In this section, we present TimeBERT, the proposed pre-trained 
language representation model based on transformer encoder [52]. 
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 expres-
sions embedded in the content) to construct time-aware language 
representation. Our approach is inspired by BERT model [15], 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 [36], which is used by RoBERTa [34]), they still 
utilize the same training technique as on the synchronic document 
collections, which essentially ignores the temporal aspects of doc-
uments. 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 

\[ \text{Suppose there is a token sequence } X = (x_1, x_2, \ldots, x_n), \] 

where \( x_i \) (\( 1 \leq i \leq n \)) indicates a token in the vocabulary. Firstly, the 
temporal expressions in the entire document content are recog-
nized using spaCy\(^5\) (as indicated by the gray font in the bottom 
in Figure 1). The recognized temporal expressions\(^*\) set is denoted 
by \( T = (t_1, t_2, \ldots, t_m) \), 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 \( \alpha \), 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 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 ex-
ample). Finally, same as in BERT, 80\% of the sampled tokens are 
replaced with [MASK], 10\% with random tokens, and 10\% with the 
original tokens.

Through this masking scheme, we encourage the model to be 
more focused on the content temporal information and the rela-
tions 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

---

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

2 The selected example is the news article published in The New York Times in 
1990/01/05, with title ‘Conductors’ Conference To Include Stern Tribute’.

3 https://spacy.io/
As described in Section 2, timestamp temporal information can be applied in retrieval, for example, it has been widely utilized in temporal information retrieval for estimating document relevance scores [27, 33, 54]. 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 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 1 uses month granularity.

3.2 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. As described in Section 2, timestamp temporal information can be applied in retrieval, for example, it has been widely utilized in temporal information retrieval for estimating document relevance scores [27, 33, 54].

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 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 1 uses month granularity.

Jatowt and Au Yeung [21] 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. [55] 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 model trained on coarse granularity (e.g., year granularity) might not perform well on difficult time-related tasks. In Section 5.4, we analyze the effect of different choices of temporal granularity \( g \).

3.3 Temporal Information Replacement (TIR)

We also experiment with other ways in which temporal information of documents could be utilized while pre-training. The last pre-training task we investigate has been inspired by WKLM [56]. 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 [13], a popular tool for recognizing and normalizing temporal expressions, is utilized to detect and then group temporal expressions into boundaries. 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 2, “1990” is replaced by “2000” (note that both are of the same granularity), while “today through Sunday” is not replaced.

For example, the timestamp of the document published today is “2022/05/19” 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.

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.
temporal expression) are concatenated and used to make binary prediction (“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.

4 EXPERIMENTAL SETTINGS

4.1 Pre-training Dataset and Implementation

For the experiments, we use the New York Times Annotated Corpus (NYT corpus) [42] 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 [10, 26]. Figure 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 [15] 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.\(^9\) The maximum sequence length was 512, while the batch size was 8. We took AdamW [29] as the optimizer and set 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.\(^10\)

4.2 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 [55], WOTD dataset [19]) 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) [10, 23, 25], 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 how event occurrence time can be used is in question answering. In Question Answering over temporal document collections, 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 [53, 54]. Table 1 presents examples of the four datasets. The details of these datasets are:

1. **EventTime** [55]: This dataset consists of descriptions and occurrence times of 22,398 events (between January 1987 and June 2007) that were originally collected from two resources: Wikipedia year pages\(^11\) and On This Day website\(^12\). We will compare our approach with the SOTA method for this dataset. As the SOTA method [55] 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 same input setting. The top-1 relevant document of each event in the NYT corpus is firstly extracted using the same retrieval method (BM25) as in [55], 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) [19]: This dataset was scraped from Wikipedia’s On this day webpages\(^13\), 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).\(^14\) 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

\(^9\)The experiments took about 80 hours on 1 NVIDIA A100 GPU.

\(^10\)These two hyperparameters’ values of the pre-training tasks are also used in the released TimeBERT version. In Section 5.3 and 5.4 we will study the effect of temporal masking ratio and temporal granularity of TimeBERT.

\(^11\)https://en.wikipedia.org/wiki/List_of_years

\(^12\)https://www.onthisday.com/dates-by-year.php

\(^13\)https://en.wikipedia.org/wiki/Wikipedia:On_this_day/Today, accessed 05/2022

\(^14\)For example, the contextual information of the WOTD example in Table 1 is “The Loyalists never controlled territory unless the British Army occupied it.”

Table 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 Commissioners appointed to inquire into the alleged corrupt practices at Norwich hav made, their report. It commences with a tribute... | 1876-03-20      |
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 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 [42] mentioned in Section 4.1.

(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 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 needs to be estimated at finer granularities (e.g., month or day), as the number of labels will also greatly increase. For example, for TDA-Timestamp under day granularity, the label count equals to 29,551 which corresponds to the number of days in the dataset. In addition, as [55] and [19] use 80:10:10 split ratio to divide EventTime and WOTD, we also divide the constructed NYT-Timestamp, and TDA-Timestamp using the same ratio.

### 4.3 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.

### 4.4 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 [15], which has been trained on BooksCorpus [62] 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 [55] and [19], respectively. Note the two 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.

6. **TimeBERT**: Our final language model TimeBERT trained on the NYT corpus for 10 epochs using TAML and DTP tasks. Note that we also study degenerated versions of our proposed model in the ablation studies which will be reported in Sec. 5.2.

### 4.5 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 EXPERIMENTAL RESULTS

#### 5.1 Main Results

5.1.1 **Event Occurrence Time Estimation**. Table 2 and Table 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 models18 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.19

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 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

---

18The SOTA methods [55] and [19] are not based on language models.
19Especially on EventTime dataset for the case when the time is need to be predicted under month/day granularity and on WOTD dataset.
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, and 5.83%, 20.45% on EventTime-WithTop1Doc, respectively. When compared with SOTA [55], 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 SOTA [55] utilizes both temporal signals and obtains comparative results, their TEP-Trans model can only estimate the time within the time frame of the underlying knowledge source that is being used. In addition, they use 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 [19] 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 [19] and BERT when using contextual information. The two latter methods do not utilize news article 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.

5.1.2 Document Timestamp Estimation. Table 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 has 7,438 day labels, TDA-Timestamp has 2,627 month labels and 29,551 day labels. In all the cases, the results are the same. In the next section, we will discuss how to improve the performance of document timestamp estimation.

When considering WOTD dataset, 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 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.

5.2 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 and Table 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 5.1, for example, the test of TDA-Timestamp at month or day granularities is removed). -DTP, -TAML, -MLM indicate the corresponding models trained using only DTP, TAML 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 and Table 6, TimeBERT, which uses TAML 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 our model. When considering the models that use only one of the pre-training objectives of TimeBERT, -DTP and -TAML, 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.

5.3 Effect of Different Temporal Masking Ratios in TAML

Temporal masking ratio \( \alpha (0.0 \leq \alpha \leq 1.0) \) is an important hyperparameter of TAML task, which determines how many temporal expressions in the document are sampled during masking. For example, when \( \alpha \) 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 1, the year information of the timestamp, “1990”, is repeated in the first sentence of the document.). On the other hand, when \( \alpha \) 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 \( \alpha \), we pre-train TimeBERT using different \( \alpha \) values using TAML and DTP tasks for 3 epochs. Figure 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 \( \alpha \) 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 \( \alpha \) equals to 0.3, and it produces the best results on WOTD and TDA-Timestamp when \( \alpha \) equals to 0.2, 0.1 respectively.22

---

20Since the authors use NYT corpus as the knowledge source, the model can only estimate the time of events happened between 1987 and 2007.

21Contextual information contains the relevant sentences extracted from Wikipedia as the external knowledge, as explained in Section 4.2.

22The released TimeBERT version uses \( \alpha \) value equal to 0.3.
Table 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 |
| RG    | ACC 4.77  | MAE 6.92 | 0.41 | 81.60 | 0.01 | 2484.48 |
|       | ACC 4.77  | MAE 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 | 1544.56 |
| SOTA [55] | -     | -    | -   | 0.01 | 0.01 | 48.51 |
|       | 40.93     | 3.01 | 30.89 | 36.19 | 16.42 | 1235.67 |
| BERT-THR | 25.40 | 3.23 | 6.83 | 40.45 | 0.99 | 1751.92 |
|       | 36.47     | 3.94 | 17.01 | 31.72 | 0.09 | 1564.95 |
| 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 3: Performance of different models on WOTD dataset with/without contextual information.

| Model | NO CI | CI |
|-------|-------|----|
|       | ACC ME | ACC ME |
| RG    | 6.16   | 217.72 |
|       | 8.15   | 217.35 |
| BERT  | 7.20   | 52.58 |
|       | 9.60   | 41.16 |
| BERT-NYT | 8.08 | 53.75 |
|       | 19.97  | 36.47 |
| SOTA [19] | 11.40 | - |
|       | 13.10  | - |
| BERT-THR | 10.13 | 54.92 |
|       | 18.36  | 35.99 |
| TimeBERT | 11.60 | 48.34 |
|       | 23.85  | 33.70 |

Figure 4: TimeBERT performance (accuracy in the top plot and MAE in the bottom plot) with different temporal masking ratios on four datasets.

5.4 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 7 and Table 8, we can observe that TimeBERT pre-trained using month granularities achieves most of the best results, 23 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 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. 24

6 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 potential application can be: generating a timeline summary for a specific news story [57, 61] or for a given entity [51], temporal image retrieval [17] that helps users to find relevant images which satisfy the temporal intent behind their queries (e.g., query ‘iPhone13’ should returned images showing the right device model released in recent years), or event detection and ordering [14, 46], temporal clustering and information retrieval [2, 10, 11], question answering [37, 53], etc.

We next demonstrate how the proposed TimeBERT model could be utilized in one such application. In particular, we test a temporal question answering system called QANA [54], 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 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-Top1Doc dataset of month granularity. Similar to the way of making EventTime-Top1Doc 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

23The released TimeBERT version uses g set to month granularity.

24Note 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 4: Performance of different models for document timestamp estimation on two datasets.

| Model   | NYT-Timestamp |        | TDA-Timestamp |        |
|---------|---------------|--------|---------------|--------|
|         | Year | Month | Day | Year | Month | Day | Year | Month | Day |
|         | ACC  | MAE  | ACC  | MAE  | ACC  | MAE  | ACC  | MAE  | ACC  |
| RO      | 4.77 | 7.06 | 5.41 | 8.17 | 0.01 | 2485.5 | 4.45 | 75.39 | 0.04 | 873.88 | 0.00 | 11253.72 |
| BERT    | 35.00 | 1.64 | 2.56 | 22.74 | 0.10 | 1815.89 | 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 |

Table 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  | Day   |
|           | ACC       | MAE    | ACC   | MAE   | ACC   | MAE   |
| -DTP      | 26.95     | 3.32   | 9.38  | 38.56 | 10.01 | 65.73 | 18.50 | 46.86 |
| -TAMLMM   | 23.05     | 3.37   | 6.67  | 41.16 | 9.43  | 53.48 | 19.82 | 38.74 |
| -MLM+TDF  | 21.52     | 3.45   | 3.71  | 44.47 | 8.66  | 55.66 | 18.80 | 40.85 |
| -MLM+DTP  | 26.13     | 3.28   | 8.44  | 39.68 | 11.16 | 58.40 | 19.32 | 43.24 |
| TimeBERT  | 29.51     | 3.17   | 10.80 | 36.14 | 11.16 | 51.09 | 22.47 | 36.80 |

Table 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.

Table 8: 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   |
|           | 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 9: Performance of different models in QA task.

| Model  | Top 1 | Top 5 | Top 10 | Top 15 |
|--------|-------|-------|--------|--------|
|        | EM    | F1    | EM    | F1    |
| QANA   | 21.00 | 28.90 | 28.20 | 36.85 |
| TimeBERT | 22.40 | 29.31 | 29.20 | 37.14 |

are then ordered and used as “start” and “end” information of the estimated question time scope. The estimated time scope is then utilized for reranking 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 taken from [54]. 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 9, QANA+TimeBERT outperforms QANA for all the different N values. Considering the top 1 document, the new extended model achieves 6.67% improvement on EM and 1.42% on F1.

7 CONCLUSIONS

Time is of particular importance in many types of documents, and multiple IR or NLP methods have utilized temporal signals from text documents. In this paper, 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 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).
Jiexin Wang, Adam Jatowt, and Masatoshi Yoshikawa. 2021. 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. 398–407.

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

Rui Yan, Xiaoqun Wan, Jahna Otterbacher, Liang Kong, Xiaoming Li, and Yan Zhang. 2011. 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. 745–754.

An Yang, Quan Wang, Jing Liu, Kai Liu, Yazan Lyu, Hua Wu, Qiaogao She, and Sujuan Li. 2019. 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. 2346–2357.

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

Yi Yu, Adam Jatowt, Antoine Doucet, Kazunari Sugiyama, and Masatoshi Yoshikawa. 2021. 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). 377–387.

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

Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. 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. 19–27.