# Time-Aware Representation Learning for Time-Sensitive Question Answering

**Anonymous ACL submission**

## Abstract

Time is one of the crucial factors in real-world question answering (QA) problems. However, language models have a problem in understanding the relationships between time specifiers, such as 'after' and 'before', and numbers, since existing QA datasets do not include a sufficient number of time expressions. To address this issue, we propose a **T**ime-**C**ontext dependent **S**pan **E**xtraction (TCSE) task and a time-context dependent data generation framework for model training. Moreover, we present a metric to evaluate the time awareness of the QA model using TCSE. The TCSE task consists of a question and four sentence candidates generated by a pre-defined template. Candidates are classified as correct or incorrect based on time and context. The model is trained to extract the answer span from the sentence that is both correct in time and context. The model trained with TCSE outperforms baseline models up to 6.97 of the F1-score in the TimeQA dataset.

## 1 Introduction

Question Answering (QA) models (Devlin et al., 2019; Clark et al., 2020) have achieved significant success in recent years. However, most existing QA models fail to understand time (Chen et al., 2021) since most QA datasets (Rajpurkar et al., 2018; Kwiatkowski et al., 2019) lack temporal information. Ignoring temporal constraints when answering questions can lead to inaccurate or unreliable results. For instance, as shown in Figure 1, neglecting the time while extracting the answer may lead to the selection of an incorrect entity, 'Katie'.

To overcome this limitation, language models must be able to incorporate temporal information into their comprehension of the context in which a question is asked. This requires the model to recognize temporal expressions within the text and understand the relationship between the time specifiers and numerical values. For example, asking about anything that happened 'after 2020' and 'before 2020' are entirely different, even though they include the same number. Therefore, models must be capable of comprehending the connection between time specifiers and numbers beyond simple numerical comparisons.

Q: Who **worked in the Salvation Army** *before 1995*?
A: Harry

| Context / Time | Correct | Incorrect |
|---|---|---|
| Correct | Harry **joined the Salvation Army** *in 1991* (BC) | *In 1991*, Brian **began his football career at Rovers** (TC) |
| Incorrect | Katie **joined the Salvation Army** *in 2002* (CC) | *In 2002*, Paul **began his football career at Rovers** (BI) |

Figure 1: Example case of Time-Context dependent Span Extraction (TCSE) task. For each time-sensitive question, the passage consists of four types of sentences that depend on whether the sentences match the time and context of the question. Each tag indicates Both Correct (BC), Time Correct (TC), Context Correct (CC), and Both Incorrect (BI), respectively. The target span is 'Harry' in this example.

This study aims to investigate methods for enhancing the performance of QA models in time-sensitive tasks. Specifically, we aim to develop a model that can process temporal information and utilize it to answer time-sensitive questions precisely. Injecting time awareness and numeracy into QA models is challenging since there are many possible temporal expressions, and the model must consider time information as an independent part of the context. Therefore, it is necessary to train the model by incorporating a new task, the Time-Context dependent Span Extraction (TCSE), and generating synthetic data for TCSE.

In this paper, our contributions are:

- We propose a TCSE task and generate syn-

thetic data to enhance temporal reasoning ability to understand time expressions. We will release the synthetic dataset and code to facilitate further research in time-sensitive QA.

- We demonstrate that training the model with TCSE can improve the time awareness of QA models.
- We introduce a new metric to evaluate QA models in terms of time and context awareness.

## 2 Related Work

Several previous works have addressed the issue of temporal reasoning in question answering using knowledge graphs. Zhang (2022) proposed a novel framework for handling complex temporal questions that involve time ordering, such as "Who held the position of President of the USA before WWII?". Shang et al. (2022) jointly train the model using text with timestamps. However, these approaches may not be sufficient for time-sensitive QA tasks requiring understanding temporal information in the text, as temporal knowledge graphs typically handle only well-structured time information such as (Barack Obama, position held, President of USA, [2009, 2017]).

Despite these efforts, there remains a gap in research regarding handling various time expressions and numerical reasoning in time-sensitive QA tasks. Chen et al. (2021) attempted to address this gap by constructing a dataset containing time-sensitive question-context pairs that involve time-evolving events. Their analysis revealed that existing language models often fail to adequately consider temporal constraints in such tasks, resulting in significantly lower performance than humans.

## 3 Method

We present an approach to improve the performance of models in time-sensitive QA tasks by proposing a Time-Context dependent Span Extraction (TCSE) task. TCSE is for evaluating language models' time awareness, and training models to learn the relationship between time specifiers and numerical values.

### 3.1 Synthetic Time-sensitive Data Generation

Data generation for TSCE involves constructing question-context templates and generating time-context dependent data. A question-context template is a pair of a question in which the time constraint is masked, and a context in which time information and target entity are masked, as shown in Figure 2.

We extract time-related sentences such as "He joined the Salvation Army in 1853" from Wikipedia articles. Then, a question is generated for each extracted sentence by using the question generation model (Raffel et al., 2020). We create a template of the question and sentence pair by replacing the person entity and time expression with special tokens, `'[NAME]' and '[TIME]'`, respectively.

To obtain time-sensitive question-context pairs, we utilize a time pair generation process and we employ the 'names' Python module that randomly generates a person's name.

We generate random time pairs through rule-based matching of time specifiers and years. To simplify template generation, we assume that all events continue indefinitely when generating year numbers. We adopt seven time specifiers {in, after, since, before, until, between, from}. We generate positive time expressions that match the time range of the question and negative time expressions that does not. We exclusively use the time specifier 'in' when generating time expressions for the context to facilitate model training. For example with rule-based matching, if the question time is 'before 1995', then positive time is the year smaller than 1995, and negative time is the year greater than 1995. We randomly select one of the context templates to obtain a negative context.

As depicted in Figure 2, we get positive and negative context and time for each question. This allows us to produce sentences that are correct in both context and time (BC), only in context (CC), only in time (TC), and are incorrect in both (BI) for the corresponding question.

### 3.2 Time-context Dependent Span Extraction

We train the model in a multi-task setting using both reading comprehension and TCSE tasks. The loss for the reading comprehension task, denoted as $L_{RC}$, is calculated by the sum of cross-entropy loss between ground truth and predicted distribution of start and end indices. Similarly, the TCSE task adopts the same loss function, but with the answer span set as the target entity in 'BC' context. As such, the final loss is defined as a weighted sum of

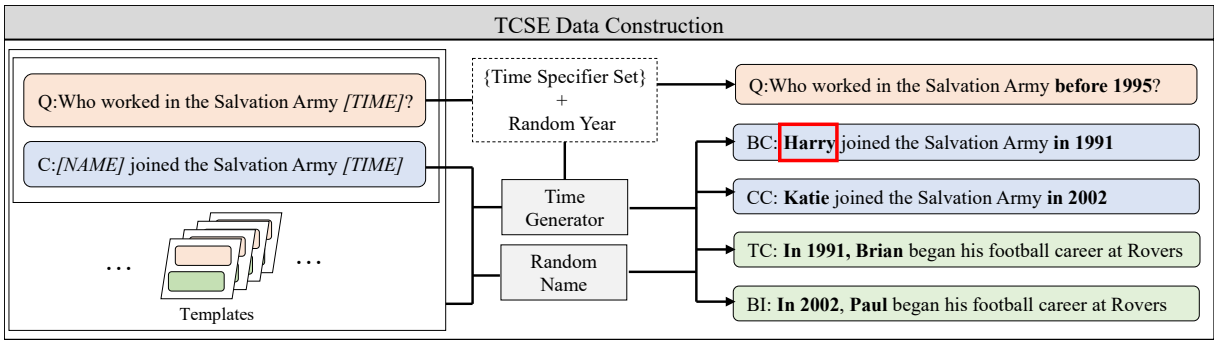



Figure 2: Four types of candidates, namely BC, CC, TC, and BI, are derived from question-context templates via time expression generation and random name.

answer-span prediction loss and TCSE loss:

$$L_{total} = L_{RC} + \lambda * L_{TCSE} \quad (1)$$

### 3.3 Evaluation Metric of Time Awareness

We propose a new evaluation metric for measuring the time awareness of the model leveraging TCSE. Since the TCSE dataset labels which sentence is correct in terms of the time or the context, it is possible to determine whether the model extracted the answer from the correct time or context. Specifically, if the model correctly extracts the answer from BC or TC sentence, it indicates that the model finds the answer in the correct time range. Similarly, if the model extracts the answer from BC or CC sentences, it indicates that the model identified the correct context. Therefore, the Time Awareness (TA) and the Context Awareness (CA) scores are calculated by the ratio of cases in which the model extracts the answer in the correct time range or context, respectively. Awareness Scores are calculated with the following equations:

$$\text{TA} = \frac{|BC| + |TC|}{(\#\ \text{of questions})}$$
$$\text{CA} = \frac{|BC| + |CC|}{(\#\ \text{of questions})} \quad (2)$$

where |BC|, |TC|, |CC| indicate the number of questions that the model extracts the answer in BC, TC, CC, respectively. Then, Time-Context Awareness Score (TCAS) is calculated as the harmonic mean of TA and CA:

$$\text{TCAS} = 2 \times \frac{TA \times CA}{TA + CA} \quad (3)$$

TCAS allows for a comprehensive evaluation of a model's performance in terms of both time and context awareness. We validate TCSE in Appendix D

## 4 Experimental Setup

### 4.1 Dataset

**TimeQA** (Chen et al., 2021) is a reading comprehension dataset that involves complex temporal reasoning. TimeQA consists of two subsets, easy and hard-mode, which differ in the level of difficulty of temporal reasoning required. We use a hard-mode dataset as it involves reasoning with more complex time expressions, such as matching two time ranges. The resolution of questions in the hard-mode dataset is not attainable through text-matching only.

To evaluate the time and context awareness of the model using TCSE task, we generate a TCSE test set following the same process used for generating the training data for TCSE. We extract time-related sentences from Wikipedia pages not included in the training data to avoid context overlap.

As a result, we generated 10,302 templates, and we generated 118,104 TCSE data for training, and 9323 TCSE data for tests from templates.

### 4.2 Baselines

**BERT** (Devlin et al., 2019) is a large pre-trained language model largely used in QA tasks. In our experiments, we use the BERT base model fine-tuned with SQuAD2.0 (Rajpurkar et al., 2018)
**BigBird** (Zaheer et al., 2020) is a language model that was developed to handle long sequence input by using sparse attention. In our experiments, we use the RoBERTa (Liu et al., 2019) base BigBird model fine-tuned with Natural Questions (NQ) (Kwiatkowski et al., 2019).

## 5 Result and Discussion

We present the experimental results described in the previous section. We show that the model trained

| Model | $BERT_{base}$ | | $BigBird_{RoBERTa}$ | |
|---|---|---|---|---|
| Metric | EM | F1 | EM | F1 |
| Baseline | 15.5 | 16.01 | 25.93 | 35.92 |
| +TimeQA | 19.95 | 26.25 | 44.61 | 53.56 |
| +(TimeQA +TCSE) | **23.13** | **33.22** | **46.14** | **54.48** |

Table 1: Performance of baseline models, model trained with timeQA data, and model trained with the proposed method. We evaluate the model on the TimeQA test dataset; three runs average all results. Our method outperforms the baseline model.

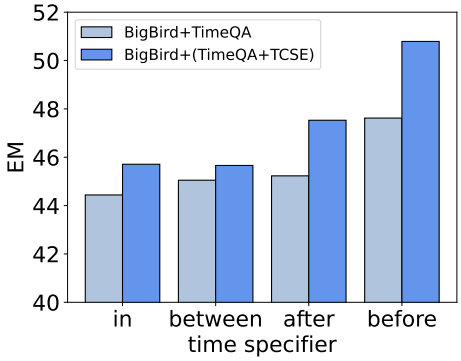



Figure 3: EM score on TimeQA according to the time specifier included in the question.

| | F1 | TA | CA | TCAS |
|---|---|---|---|---|
| Baseline | 35.92 | 51.48 | **88.78** | 65.16 |
| +TimeQA | **53.56** | **67.96** | 79.32 | **73.21** |

Table 2: Comparison among the F1-score in TimeQA, and score in TCSE task: Time Awareness (TA), Context Awareness (CA), and Time-Context Awareness Score (TCAS) of $BigBird_{RoBERTa}$ model.

with TCSE outperforms baseline models in a time-sensitive QA task. We also demonstrate that TCSE can be employed to assess the time and context awareness of QA models.

### 5.1 Time-sensitive Question Answering

We evaluate time-sensitive QA performance on TimeQA (Chen et al., 2021) dataset. We show the result in Table 1, demonstrating that training the model with TCSE outperforms the baseline models. BERT model further trained on TCSE shows a significant performance improvement of 6.97 F1-score compared to the model trained only on TimeQA, which suggests that the model learns time-aware representations with TCSE. The performance gap between BERT and BigBird can be attributed to their maximum input length difference.

### 5.2 Analysis on Time Specifier

We analyze model performance on TimeQA according to the time specifier included in the question. Figure 3 shows the EM score difference for four kinds of time specifiers: {in, between, after, before}. There are comparatively substantial improvements in model performance on time specifiers 'after' and 'before'. This improvement demonstrates that TCSE effectively trains the model to understand the time range.

### 5.3 Time and Context Awareness

We evaluate the model's time awareness and context awareness using the TCAS metric. Table 2 indicates that the F1-score and TA exhibit similar trends, implying that TA is a reliable indicator of time awareness. We observed that training with TimeQA resulted in a decrease in contextual understanding, as evidenced by an 9.46-point drop in CA. The results suggest the importance of learning time expressions while maintaining contextual understanding. We utilized the TCAS metric to provide an overall assessment of the model's performance. We found that the model's contextual awareness decreased, but its time awareness improved significantly, resulting in improved TCAS score. We do not perform TCAS on models trained with TCSE, because the model has already learned the TCSE task.

## 6 Conclusion

In this paper, we demonstrated that existing QA models are inadequate in understanding time expressions. To address this problem, we proposed TCSE, which enables models to learn time expressions while maintaining their understanding of context. We constructed question-context templates to generate time-context dependent data for TCSE and trained the model in a multi-task learning setting. Our experimental results showed that TCSE improves the performance of QA models on TimeQA. Also, we proposed a new evaluation metric, TCAS, and showed a gap in performance between models in terms of time and contextual understanding. Future research should focus on advancing temporal reasoning capabilities beyond the comprehension of simple temporal expressions.

## Ethical Consideration

This paper presents a synthetic data generation framework that modifies time information and name while retaining the original text. Notably, this approach does not produce any unintended harmful effects, as it does not alter the semantic content of the original text beyond the specified modifications.

## Limitations

Limitation of our approach is that TCSE does not cover all kinds of time because we construct TCSE data with only seven time specifiers. Although it is possible to enhance the model's time awareness by adding additional time expressions, our experiments showed that the inclusion of only these seven led to a performance improvement.

## References


Wenhu Chen, Xinyi Wang, and William Yang Wang. 2021. A dataset for answering time-sensitive questions. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. Electra: Pre-training text encoders as discriminators rather than generators. In *International Conference on Learning Representations*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

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

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

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551.

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

Chao Shang, Guangtao Wang, Peng Qi, and Jing Huang. 2022. Improving time sensitivity for question answering over temporal knowledge graphs. *arXiv preprint arXiv:2203.00255*.

Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, et al. 2020. Big bird: Transformers for longer sequences. *Advances in neural information processing systems*, 33:17283–17297.

Bryan Zhang. 2022. Improve MT for search with selected translation memory using search signals. In *Proceedings of the 15th Biennial Conference of the Association for Machine Translation in the Americas (Volume 2: Users and Providers Track and Government Track)*, pages 123–131, Orlando, USA. Association for Machine Translation in the Americas.

# Appendix

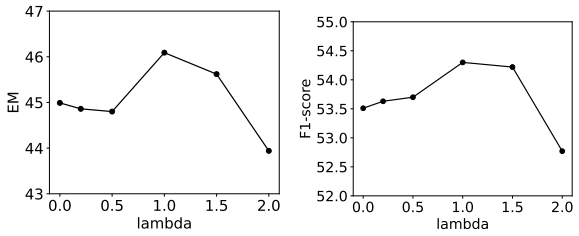
## A Qualitative Analysis

To clearly understand our model's improvement in time awareness, we present a case study on the TimeQA dataset in Table 5. Our model successfully finds the correct answer in the context with the correct time range. The model correctly answered a challenging question that required verifying whether the time range 'between 1831 and 1833' matches with 'from 1829 to 1835'. Furthermore, our model recognizes that a sentence containing the correct context but with an incorrect time range does not yield an answer.

## B Hyper Parameter Setting

### B.1 Analysis on $\lambda$

We observe the changes in model performance according to the value of $\lambda$. Figure 4 shows that the model performance increases with an increase in $\lambda$ until it reached a value of 1.0. However, the model performance decreases when $\lambda$ was set to a value greater than one due to overfitting the TCSE task.




(a) EM (b) F1

Figure 4: Analysis on $\lambda$ for time-sensitive question answering for TimeQA dataset with $Bigbird_{RoBERTa}$ model. We increase lambda from 0 to 2.0: {0, 0.2, 0.5, 1.0, 1.5, 2.0}. Increasing lambda improves time-sensitive question answering performance until $\lambda = 1.0$ and then decreases.

### B.2 Effect of TCSE Dataset Size

| | EM | F1 |
|---|---|---|
| BigBird+TimeQA | 44.61 | 53.56 |
| +$TCSE_1$ | 45.16 | 53.87 |
| +$TCSE_2$ | **46.5** | 54.4 |
| +$TCSE_4$ | 46.14 | **54.48** |

Table 3: Effect of TCSE according to the ratio of dataset size. $TCSE_k$ denotes that it employs TCSE data corresponding to k times the number of TimeQA dataset

To investigate the effect of the dataset size of TCSE on the model performance, we observe changes in performance according to the number of TCSE data. As shown in Table 3, utilizing a larger TCSE data than that of TimeQA yields a more substantial improvement on TimeQA.

## C Handling Long Sequence Input

Since TimeQA (Chen et al., 2021) also contains long passages of more than 10,000 tokens, we split them into length intervals that correspond to the maximum input length of the models. During training, we use the context span that contains the indices of the answer span for answerable questions and only the first context span for unanswerable questions. We select the final answer as the maximum logit value among each split context during inference.

## D Validation of TCAS

We verify that TCAS serves as a reliable evaluation metric for assessing a model in terms of time and context. We train the $BERT_{base}$ model using SQuAD2.0 and TimeQA datasets, seperately. We present the results of the zero-shot TCSE task on synthetic test dataset in Table 4. The model trained with SQuAD2.0 achieved a higher CA score, while the model trained with TimeQA achieved a higher TA score. Consequently, TA and CA score effectively reflect the model performance in contextual and temporal comprehension, respectively. TCAS provides a comprehensive assessment of the model's overall performance in both temporal and contextual aspects.

| Training dataset | TA | CA | TCAS |
|---|---|---|---|
| SQuAD2.0 | 52.64 | **95.99** | **67.99** |
| TimeQA | **62.12** | 74.91 | 67.92 |

Table 4: We calculate TA, CA, and TCAS of the $BERT_{base}$ model trained with SQuAD2.0 and TimeQA, respectively.

## E Implementation Details

We followed the implementation detail of TimeQA[1] to train models using the TimeQA dataset. Baseline models are trained using Quadro RTX A6000

[1] https://github.com/wenhuchen/Time-Sensitive-QA.git

| Question | Passage | BigBird +TimeQA | BigBird +(TimeQA+ TCSE) |
|---|---|---|---|
| A: What position did John Pope take **between Sep 1831 and Nov 1833**? | ... He served as a member of the Kentucky Senate *from 1825 to 1829* , and ... ... **From 1829 to 1835** , he served as the Governor of Arkansas Territory . ... | member of the Kentucky Senate | Governor of Arkansas Territory |
| B: Sarah Bond was an employee for whom **in Feb 2011**? | ... Bond was appointed Assistant Professor of Classics at the University of Iowa in 2014 , after holding an assistant professorship in Ancient and Early Medieval History at Marquette University *from 2012* . ... | Marquette University | - |

Table 5: A case study on TimeQA dataset: proposed model successfully (A) extracts the answer with the correct time and context and (B) detects an unanswerable question.

48GB, with a training batch size of 4, and a learning rate of 2e-5. Model fine-tuning per epoch took approximately 5 hours for BERT[2] and 12 hours for BigBird[3].

[2] https://huggingface.co/bert-base-uncased

[3] https://huggingface.co/vasudevgupta/bigbird-roberta-natural-questions