Time-Aware Representation Learning for Time-Sensitive Question Answering

Jungbin Son
KAIST
sonjbin@kaist.ac.kr

Alice Oh
KAIST
alice.oh@kaist.edu

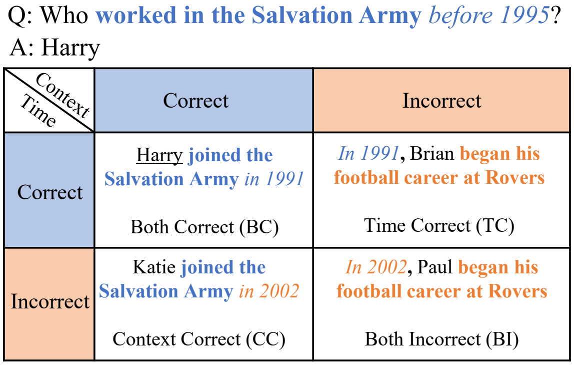
Abstract

Time is one of the crucial factors in real-world question answering (QA) problems. However, language models have difficulty understanding the relationships between time specifiers, such as 'after' and 'before', and numbers, since existing QA datasets do not include sufficient time expressions. To address this issue, we propose a **T**ime-**C**ontext aware **Q**uestion **A**nswering (TCQA) framework. We suggest a **T**ime-**C**ontext dependent **S**pan **E**xtraction (TCSE) task, and build 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 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 TCQA outperforms baseline models up to 8.5 of the F1-score in the TimeQA dataset.[1]

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 (Chen et al., 2022). 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* Both Correct (BC)	*In 1991*, Brian **began his football career at Rovers** Time Correct (TC)
Incorrect	Katie **joined the Salvation Army** *in 2002* Context Correct (CC)	*In 2002*, Paul **began his football career at Rovers** Both Incorrect (BI)

Figure 1: Example case of Time-Context dependent Span Extraction (TCSE) task. The passage consists of four types of sentences that depend on whether the sentences match the time and context of the question. 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, we propose a Time-Context aware Question Answering (TCQA) framework to achieve this issue. We train the model through Time-Context Dependent Span Extraction (TCSE) task and contrastive time representation learning.

In this paper, our contributions are:

[1] Our dataset and code are available at https://github.com/sonjbin/TCQA

Findings of the Association for Computational Linguistics: EMNLP 2023, pages 70–77
December 6-10, 2023 ©2023 Association for Computational Linguistics

- We propose a TCQA framework that involves TCSE and contrastive time representation learning, and generate synthetic data to enhance temporal reasoning ability to understand time expressions.
- We demonstrate that training the model with TCQA 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. Shang et al. (2022) proposed a novel framework for handling complex temporal questions that involve time ordering. Saxena et al. (2021) jointly train the model using text with timestamps. However, these approaches may not be sufficient for time-sensitive QA tasks, as temporal knowledge graphs typically handle only 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-passage pairs. 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 aware QA (TCQA) framework.

3.1 Synthetic Time-Sensitive Data Generation

Data generation for TCSE involves constructing question-context templates. A question-context template is a pair of questions 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 from Wikipedia articles. Then, a question is generated for each extracted sentence by using the 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 Aware Question Answering

3.2.1 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.

3.2.2 Contrastive Representation Learning

In order to enhance the time-awareness, we employ contrastive learning. We construct contrastive samples with TCSE data by pairing questions and contexts. For each sample within TCSE data, only BC context over four contexts corresponds to the question. Therefore, one positive pair (BC) and

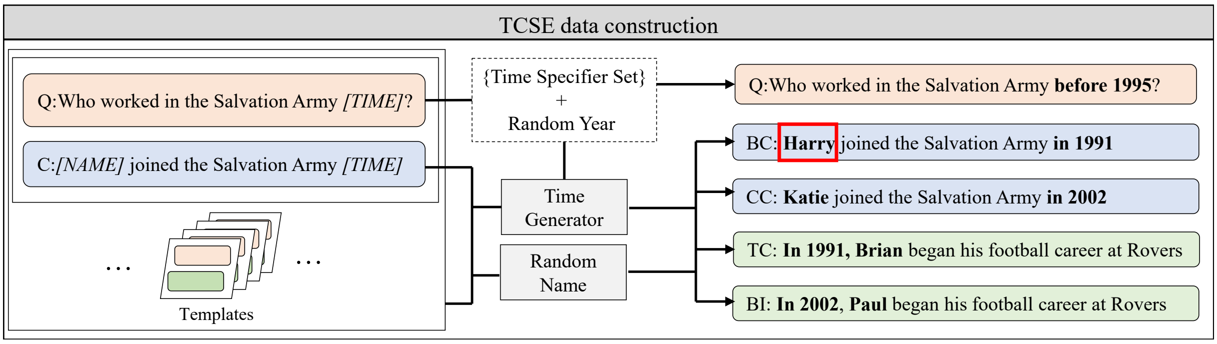

TCSE data construction
Q:Who worked in the Salvation Army [TIME]?
C:[NAME] joined the Salvation Army [TIME]
...
Templates
...
{Time Specifier Set} + Random Year
Time Generator
Random Name
Q:Who worked in the Salvation Army before 1995?
BC: Harry joined the Salvation Army in 1991
CC: Katie joined the Salvation Army in 2002
TC: In 1991, Brian began his football career at Rovers
BI: In 2002, Paul began his football career at Rovers

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.

three negative pairs (TC, CC, BI) are constructed for each TCSE data sample.

Contrastive loss is calculated based on the cosine similarity between the embedding of question (v_q) and context (v_c) for each pair. It is desirable for the distance between positive pairs to be minimized, while the distance between negative pairs should be maximized. Consequently, the contrastive label, denoted as Y, is assigned: Y=1 signifies that the context corresponds to a positive sample of the question, whereas Y=0 indicates that the context represents a negative sample. Subsequently, the contrastive loss is computed as follows:

$$s_i = CosineDistance(v_{q_i}, v_{c_i}) \tag{1}$$

$$\begin{aligned} L_{Contrast} = \sum_i [&w_p Y * exp(s_i) \\ &+ w_n (1 - Y) * exp(1 - s_i))] \end{aligned} \tag{2}$$

w_p and w_n is the weight of the positive and negative sample, respectively.

3.2.3 Joint Training

The final loss is defined as a weighted sum of answer-span prediction loss (L_{RC}), TCSE loss (L_{TCSE}) and contrastive loss ($L_{Contrast}$):

$$L_{total} = L_{RC} + \lambda_T * L_{TCSE} + \lambda_C * L_{Contrast} \tag{3}$$

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:

$$\begin{aligned} \text{TA} &= \frac{|BC| + |TC|}{(\#\ \text{of}\ \text{questions})} \\ \text{CA} &= \frac{|BC| + |CC|}{(\#\ \text{of}\ \text{questions})} \end{aligned} \tag{4}$$

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 (TC-score) is calculated as the harmonic mean of TA and CA:

$$\text{TC-score} = 2 \times \frac{TA \times CA}{TA + CA} \tag{5}$$

TC-score allows for a comprehensive evaluation of a model's performance in terms of both time and context awareness.

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.

To evaluate the TC-score of the model, we generate a test set of TCSE task using time-related

Model	$BERT_{base}$		$RoBERTa_{base}$		$ALBERT_{base}$		$BigBird_{RoBERTa}$	
Metric	EM	F1	EM	F1	EM	F1	EM	F1
Baseline	19.95	26.25	29.89	38.5	24.66	34.5	44.43	53.21
+TCQA	**25.63**	**34.75**	**30.86**	**39.03**	**27.36**	**35.48**	**46.31**	**54.26**

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.

sentences from Wikipedia pages not included in the training data.

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), **RoBERTa** (Liu et al., 2019) and **ALBERT** (Lan et al., 2019) is a large pre-trained language model largely used in QA tasks. In our experiments, we use the 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. 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

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 TCQA outperforms the baseline models. BERT model further trained on TCQA shows a significant performance improvement of 8.5 F1-score compared to the model trained only on TimeQA. This improvement is the result of the model learning to distinguish correct time expressions. The performance gap between BigBird and others can be attributed to their maximum input length difference.

5.2 Time and Context Awareness

We evaluate the model's time awareness and context awareness using the TC-score. 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 TC-score 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 TC-score. We do not perform TC-score on models trained with TCQA, because the model has already learned the TCSE task. Alternatively, TCQA is assessed using an alternative approach in Appendix 5.5.

FT dataset	F1	TA	CA	TC-score
NQ	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 (TC-score) of $BigBird_{RoBERTa}$ model according to the data used for fine-tuning.

5.3 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 TCQA effectively trains the model to understand the time range. However, the performance improvements on time specifier 'between' is comparatively low since it is more difficult as it requires simultaneous consideration of two distinct time ranges.

5.4 Ablation Study

An ablation study was conducted to assess the impact of contrastive representation learning. We

Model	$BERT_{base}$		$RoBERTa_{base}$		$ALBERT_{base}$		$BigBird_{RoBERTa}$	
Metric	EM	F1	EM	F1	EM	F1	EM	F1
w/o CRL	23.1	31.99	30.7	38.16	25.83	34.65	**46.7**	**54.44**
w/ CRL	**25.63**	**34.75**	**30.86**	**39.03**	**27.36**	**35.48**	46.31	54.26

Table 3: Results of the ablation study of the contrastive learning

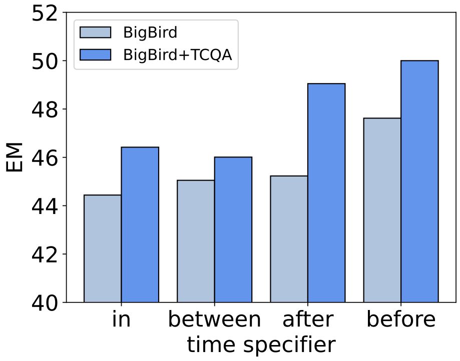


Figure 3: EM score on TimeQA of Bigbird according to the time specifier included in the question.

compare the performance of the model trained with and without adding contrastive loss in Table 3. Adding contrastive loss resulted in an improvement in the model performance of BERT, RoBERa and ALBERT model. However, it led to a slight decrease in performance for the Bigbird model. The reason for this discrepancy of result is that the vector embedding size of the Big Bird model is eight times greater than that of the other models. Consequently, we can infer that the data used for contrastive learning was insufficient.

5.5 Reading Comprehension Performance

We conduct a comparative analysis of models on the SQuAD v2 dataset to investigate the effect of TCQA on context awareness. Table 4 demonstrate that TCQA mitigates the degradation in context awareness resulting from fine-tuning on TimeQA. This improvement caused by training the model using time-context dependent data.

	EM	F1
FT on TimeQA	34.74	46.49
+ TCQA	**36.12**	**48.1**

Table 4: Performance of BigBird model on SQuAD v2 development set.

6 Conclusion

In this paper, we demonstrated that existing QA models are inadequate in understanding time expressions. To address this problem, we proposed TCQA, 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 contrastive learning, and jointly trained the model. Our experimental results showed that TCQA improves the performance of QA models on TimeQA. Additionally, we proposed a new evaluation metric, TC-score, 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 expressions because we construct the 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 Informa-*

tion Processing Systems Datasets and Benchmarks Track (Round 2).

Ziyang Chen, Xiang Zhao, Jinzhi Liao, Xinyi Li, and Evangelos Kanoulas. 2022. Temporal knowledge graph question answering via subgraph reasoning. *Knowledge-Based Systems*, 251:109134.

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.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. Albert: A lite bert for self-supervised learning of language representations. In *ICLR*.

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. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789, Melbourne, Australia. Association for Computational Linguistics.

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

Chao Shang, Guangtao Wang, Peng Qi, and Jing Huang. 2022. Improving time sensitivity for question answering over temporal knowledge graphs. In *Annual Meeting of the Association for Computational Linguistics*.

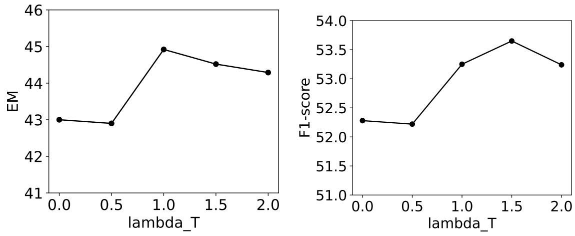
Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, et al. 2020. Big bird: Transformers for longer sequences. *Advances in neural information processing systems*, 33:17283–17297.

Appendix

A Hyper Parameter Setting

A.1 Analysis on λ_T and λ_C

We observe the changes in model performance according to the value of λ_T. Figure 4 shows that the EM and F1-score increases with an increase in λ_T until it reaches a value of 1.0 and 1.5, respectively. However, the model performance decreases when λ_T was set to a value greater than due to overfitting the TCSE task.

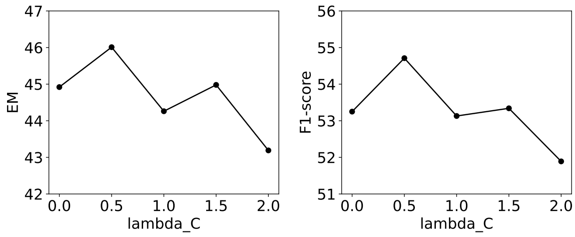


(a) EM

(b) F1

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

Additionally, we conduct an analysis of the changes in model performance as determined by the value of λ_C while maintaining a λ_T value fixed to 1.0. Model performance tends to decrease for λ_C values greater than 0.5.



(a) EM

(b) F1

Figure 5: Analysis on λ_C for time-sensitive question answering for TimeQA dataset with $Bigbird_{RoBERTa}$ model. We increase lambda from 0 to 2.0: {0, 0.5, 1.0, 1.5, 2.0}.

A.2 Effect of TCSE Dataset Size

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 5, utilizing a larger TCSE data than that of TimeQA yields a more substantial improvement on TimeQA until using two times the number of TimeQA dataset.

	EM	F1
$Bigbird_{RoBERTa}$	44.43	53.21
$+TCSE_1$	45.31	53.92
$+TCSE_2$	**46.7**	**54.44**
$+TCSE_4$	45.96	54.26

Table 5: 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.

B Qualitative Analysis

B.1 Case Study

To clearly understand our model's improvement in time awareness, we present a case study on the TimeQA dataset in Table 6. 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.2 Example of TCSE Data

We present examples of TCSE data to substantiate the reliability of synthetic data. Table 7 demonstrates the effectively generated questions and their corresponding sentences. Every sentence, 'BC', 'TC', 'CC', 'BI' , for each question are effectively generated and appropriately labeled in accordance with temporal and contextual considerations.

C Handling Long Sequence Input

Since TimeQA (Chen et al., 2021) 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.

Question	Passage	Baseline	Baseline +TCQA
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	Unanswerable

Table 6: 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.

Question	Example
Who was General Counsel in 1481?	**BC**: ... in 1473 ... Mitchell became General Counsel and a managing director. **TC**: ... Abendroth, who was banned from working as a legal trainee in 1473... **CC**: ... in 1483 ... Kimberly became General Counsel and a managing director. **BI**: ... Jefferson Rash also attended the Bilderberg conference in 1483 in St.
Who played Asian Cup finals between 1566 and 1569?	**BC:** In 1567, Julio participated with the team in the finals of the Asian Cup... **TC:** Jon Kyl, who had represented the district in 1567, said... **CC:** In 1578, Betty participated with the team in the finals of the Asian Cup... **BI:** ... golf womens Izod products were put on hiatus in 1578 , but...
Who was loaned to Wolves before 2013?	**BC**: Matthew had a second loan spell at Wolves, as well as ... in 2007... **TC**: Rita went through...the last time an NFL team had done that was in 2007 ... **CC**: Martin had a second loan spell at Wolves, as well as ... in 2020... **BI**: Bell also played in a 3–1 defeat and a draw with West Germany in 2020...
Who was governor of ohio from 2003 to 2004?	**BC**: David did not return ... until he served as Governor of Ohio in 2004 ... **TC**: He also played in 2004 AFC Asian Cup , as well as ... **CC**: Sandra did not return... until he served as Governor of Ohio in 2011 ... **BI**: Vecsei ... project designer on the construction of ... in 2011 ...

Table 7: Example of TCSE data for each time specifiers: in, between, before, from

D Implementation Details

We followed the implementation detail of TimeQA[2] to train models using the TimeQA dataset. Baseline models are trained using Quadro RTX A6000 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[3] and 12 hours for BigBird[4].

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

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

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