

Timeline-based Sentence Decomposition with In-Context Learning for Temporal Fact Extraction

Jianhao Chen¹, Haoyuan Ouyang¹, Junyang Ren¹, Wentao Ding², Wei Hu¹, Yuzhong Qu¹

¹State Key Laboratory for Novel Software Technology, Nanjing University, China

²State Key Laboratory of General Artificial Intelligence

Beijing Institute for General Artificial Intelligence (BIGAI), China

{jh_chen, hyouyang, jyren}@smail.nju.edu.cn

dingwentao@bigai.ai, {whu,yzqu}@nju.edu.cn

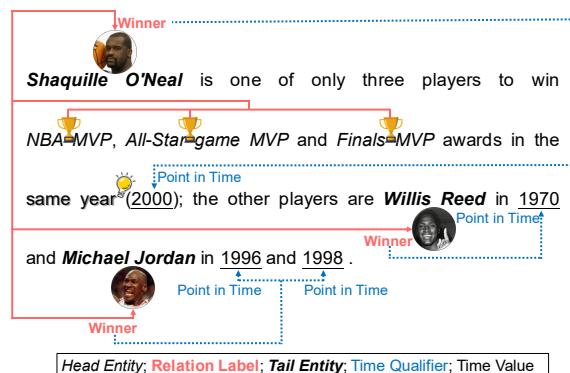
Abstract

Facts extraction is pivotal for constructing knowledge graphs. Recently, the increasing demand for temporal facts in downstream tasks has led to the emergence of the task of temporal fact extraction. In this paper, we specifically address the extraction of temporal facts from natural language text. Previous studies fail to handle the challenge of establishing time-to-fact correspondences in complex sentences. To overcome this hurdle, we propose a timeline-based sentence decomposition strategy using large language models (LLMs) with in-context learning, ensuring a fine-grained understanding of the timeline associated with various facts. In addition, we evaluate the performance of LLMs for direct temporal fact extraction and get unsatisfactory results. To this end, we introduce TSDRE¹, a method that incorporates the decomposition capabilities of LLMs into the traditional fine-tuning of smaller pre-trained language models (PLMs). To support the evaluation, we construct ComplexTRED, a complex temporal fact extraction dataset. Our experiments show that TSDRE achieves state-of-the-art results on both HyperRED-Temporal and ComplexTRED datasets.

1 Introduction

Acquiring knowledge has long been a fundamental challenge in the field of artificial intelligence. Typically, the acquired knowledge is stored in knowledge graphs (KGs) as triples, comprising a head entity, a relation, and a tail entity. Recently, there has been a significant surge in the need (Chen et al., 2023; Liu et al., 2023; Ding et al., 2022; Xu et al., 2023; Li et al., 2022; Mezni, 2022; Xiong et al., 2024) for acquiring temporal facts. For instance, users may seek information from the KGs, such as “When did Michael Jordan win the NBA Finals MVP?”. However, traditional triples like

¹The released source code and documentation are available at <https://github.com/JianhaoChen-nju/TSDRE>



Extracted Facts:

Head Entity	Relation Label	Tail Entity	Time Qualifier	Time Value
NBA MVP	Winner	Shaquille O'Neal	Point in Time	2000
All-Star game MVP	Winner	Shaquille O'Neal	Point in Time	2000
Finals MVP	Winner	Shaquille O'Neal	Point in Time	2000
NBA MVP	Winner	Willis Reed	Point in Time	1970
All-Star game MVP	Winner	Willis Reed	Point in Time	1970
Finals MVP	Winner	Willis Reed	Point in Time	1970
NBA MVP	Winner	Michael Jordan	Point in Time	1996
All-Star game MVP	Winner	Michael Jordan	Point in Time	1996
Finals MVP	Winner	Michael Jordan	Point in Time	1996
NBA MVP	Winner	Michael Jordan	Point in Time	1998
All-Star game MVP	Winner	Michael Jordan	Point in Time	1998
Finals MVP	Winner	Michael Jordan	Point in Time	1998

Figure 1: A difficult temporal fact extraction example, which contains 12 temporal facts in one sentence.

(NBA Finals MVP, winner, Michael Jordan) cannot meet such needs due to the lack of time dimension. This can be solved by adding time information like (Point In Time, 1996) to the triple. A temporal fact can be formalized as a quintuple $(e_{head}, r, e_{tail}, q, t)$, where the time qualifier q and time value t indicate the time dimension information of the triple.

Although many works (Wang et al., 2011; Liu et al., 2021; Chia et al., 2022) have explored the task of temporal fact extraction, the challenge of establishing correspondence between time and triples remains unresolved. Figure 1 shows a difficult example of temporal fact extraction. It can be seen that the difficulty of this example mainly comes from the use of “the other” to imply that the other two players also won three awards in one year. Besides, natural language expressions are typically succinct, with “Michael Jordan in 1996 and 1998”

alone expressing six temporal facts. Existing methods cannot handle this situation without explicitly stating what event occurred at a certain point in time, leading to suboptimal performance in addressing complex narratives with interwoven timelines.

To address this gap, we propose a timeline-based sentence decomposition strategy. Our strategy involves breaking down sentences according to their timelines to capture the temporal dimensions of facts, ensuring a fine-grained understanding of the timeline associated with various facts. In the past, sentence decomposition often required a large amount of training corpus. But now, the in-context learning capability of large language models (LLMs) empowers us to perform timeline-based sentence decomposition without training corpus.

To the best of our knowledge, we are the first to investigate the application of LLMs for temporal fact extraction tasks. We conduct an evaluation to assess the performance of employing LLMs directly for temporal fact extraction. However, it is unsatisfactory that the performance of LLMs does not surpass the traditional approach of fine-tuning smaller PLMs. To enhance the performance of temporal fact extraction methods, we come up with the idea of combining the surprising sentence decomposition capability of LLMs with the traditional way of fine-tuning small pre-trained language models. Experiments show that our method based on the combining idea achieves SOTA results.

In addition to research on extraction methods, there are currently few benchmarks (Chia et al., 2022; Wang et al., 2012) for temporal fact extraction, and they do not pay much attention to sentences involving complex time related narratives (we call them complex sentences for short). To this end, we construct a temporal fact extraction dataset composed of complex sentences for evaluation.

In summary, our main contributions in this paper are outlined as follows:

1. We summarize the unique challenge of the temporal fact extraction task and propose a timeline-based sentence decomposition method on natural language using in-context learning enhanced by human feedback.
2. We propose timeline-based sentence decomposition (TSD) for the temporal fact extraction task. Evaluation results demonstrate that TSD indeed helps models understand the correspondence between events and time.
3. We conduct evaluations of the methods that utilize LLMs in the task of temporal extraction. Moreover, we build a novel dataset ComplexTRED, which consists of 19,148 complex sentences with multiple time expressions or temporal facts.

2 Related Work

2.1 Temporal Fact Extraction

Temporal fact extraction methods are mainly divided into two categories: pattern-based methods and deep learning-based methods.

Pattern-based methods Existing work attempts to treat relationships and qualifiers as a whole to mine corresponding patterns in text, or use sequence annotation and classification methods. T-Yago (Wang et al., 2010) and a similar study (Kuzey and Weikum, 2012) extract temporal instances from semi-structured data like Wikipedia’s Infoboxes, Categories, and Lists, limiting coverage to freely available text. Pravda (Wang et al., 2011) uses textual patterns to represent candidate facts and labels them through a graph-based label propagation algorithm. Liu et al. (2021) applies the idea of distant supervision, leveraging existing temporal facts to learn corresponding patterns from web text and subsequently applying them to the extraction process.

Deep learning-based methods CubeRE (Chia et al., 2022) first employs a sequence labeling approach to identify entities and time, followed by classification of the relations and qualifiers between them.

Previous Studies have not fully addressed the challenges posed by complex sentences in temporal fact extraction. Moreover, we are the first to explore the application of LLMs for the task of temporal fact extraction.

2.2 Sentence Decomposition

Sentence decomposition is a common method used for tasks in the NLP field. From a technical perspective, sentence decomposition can be roughly divided into two categories: supervised learning methods (Huang et al., 2023) and rule-based methods (Hu et al., 2021). Supervised learning requires sufficient training corpus, while rule-based methods are labor-intensive and often suffer from poor coverage issues. In the era of large models, we

explore using in-context learning for sentence decomposition, which does not require task-specific training data and has sufficient coverage.

2.3 In-Context Learning

Large language models, exemplified by the GPT series (Brown et al., 2020; OpenAI, 2023) and the LLaMA (Touvron et al., 2023a,b) family, are known to have impressive in-context learning abilities. These LLMs have been proven to be able to solve a completely new problem with a small number of examples without the need for task-specific training leading to a surge in exploration within the domain of in-context learning.

Generally, research on in-context learning can be categorized into two main areas. The first focuses on the strategy for selecting examples (Li et al., 2023) to enhance performance, while the second delves into the interpretability aspects (Han et al., 2023) of in-context learning. Our study focuses on example selection. We iteratively constructed demonstrative examples for task scenarios without training corpus. Moreover, we incorporate human feedback to guide the model and prevent common errors.

3 Preliminary

3.1 Problem Formulation

Given an input sentence of n words $s = \{x_1, x_2, \dots, x_n\}$, the objective of the temporal fact extraction task is to extract all existing temporal facts in s . Formally, a temporal fact is represented as $(e_{head}, r, e_{tail}, q, t)$. An entity e is a consecutive span of words where $e = \{x_i, x_{i+1}, \dots, x_j\}$, $i, j \in \{1, \dots, n\}$. r represents the relation between head entity e_{head} and tail entity e_{tail} . $r \in R$, where R is the predefined set of relation labels. The qualifier q and the time value t indicates the time dimension of the relation triplet (e_{head}, r, e_{tail}) . $q \in Q$, where Q is the predefined set of qualifier labels.

3.2 Complex Sentence in Temporal Fact Extraction

In this paper, we delineate the distinctions between simple and complex sentences in the context of temporal fact extraction. Prior research has shown a lack of emphasis on complex sentences. However, this paper shifts its focus to the temporal fact extraction of complex sentences.

A simple sentence comprises only one time element and one temporal fact, presenting relatively

lower difficulty in relation extraction and time selection. Example 1 provides a concrete instance of a simple sentence in the context of temporal fact extraction.

Example 1. *Peter Whittle was elected a fellow of the Royal Society in 1981 .*

Extracted Temporal Facts:

(Peter Whittle, member of, Royal Society, start time, 1981)

On the other hand, a complex sentence involves two or more time elements or two or more facts, often introducing challenges in time selection or relation extraction. Understanding the correspondence between time and facts becomes more challenging in the presence of multiple time references compared to a situation with a single time reference. Figure 1 has shown a complex sentence with more than two time elements.

Example 2 provides another concrete instance of a complex sentence in the context of temporal fact extraction.

Example 2. *20 November 1883, Jules Ferry succeeds ChallemeL-Lacour as Minister of Foreign Affairs.*

Extracted Temporal Facts:

(Jules Ferry, position held, Minister of Foreign Affairs, start time, 20 November 1883)

(ChallemeL-Lacour, position held, Minister of Foreign Affairs, end time, 20 November 1883)

(Jules Ferry, replaces, ChallemeL-Lacour, point in time, 20 November 1883)

Although there is only one point in time in Example 2, this sentence contains three temporal facts. This is mainly because the word "succeeds" expresses the relationship of succession at this moment. Implicit expressions such as these imply the connection of events in the time dimension, which brings challenges to the complete extraction of temporal facts.

4 Method

In this section, we begin by presenting two direct approaches to leverage LLMs for temporal fact extraction. Following that, we introduce our extraction methods, encompassing a timeline-based sentence decomposition strategy and the fine-tuning of generative models using decomposition results for training.

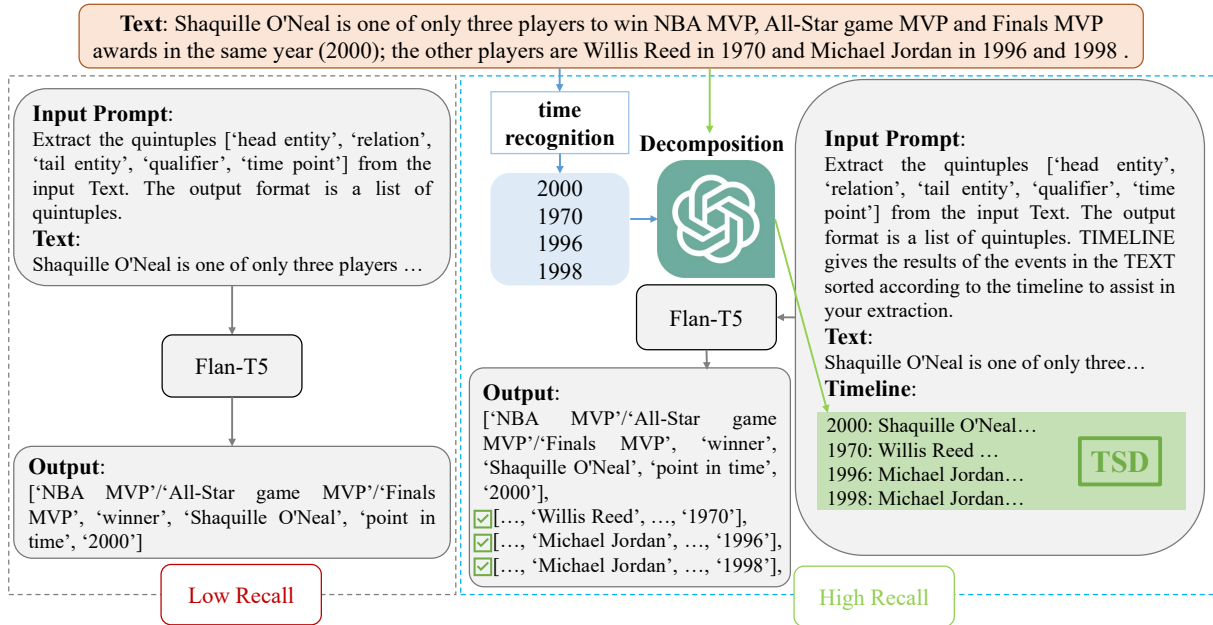


Figure 2: Framework comparison between Flan-T5 and TSDRE. Leveraging Timeline-based sentence decomposition for training can significantly improve the recall of the Flan-T5.

4.1 In-Context Learning with ChatGPT3.5 and Fine-tuning Open-Source LLM

We try to employ LLMs directly for temporal fact extraction through in-context learning and fine-tuning.

Specifically, we apply in-context learning to ChatGPT3.5. To construct the prompt, we first give a task description: Extract all the quintuples [subject, relation, object, qualifier, time point] from the input text, followed by the specified relationship list and qualifier list. We then select 48 examples at random from the train set to ensure that all relations and qualifiers have at least one-shot examples. The complete prompt is more than 2,000 tokens and will be shown in Appendix E.

We also try to LoRA fine-tune Llama2 (7B). We still give Llama2 the task description as the instruction to make Llama2 better understand the task.

4.2 Timeline-based Sentence Decomposition

Timeline-based sentence decomposition can be used to understand and process texts containing temporal information. It helps organize and present the timeline in sentences, making it easier for models to understand and follow the development of events.

An example of decomposition is as follows:

Text: Shaquille O'Neal is one of only three players to win NBA MVP, All-Star



Figure 3: Player awards are presented in a timeline.

game MVP, and Finals MVP awards in the same year (2000); the other players are Willis Reed in 1970 and Michael Jordan in 1996 and 1998.

Time: ['2000', '1970', '1996', '1998']

Decomposition: 2000: Shaquille O'Neal is one of only three players to win NBA MVP, All-Star game MVP, and Finals MVP awards in the same year. 1970: Willis Reed won NBA MVP, All-Star game MVP, and Finals MVP awards in the same year. 1996: Michael Jordan won NBA MVP, All-Star game MVP, and Finals MVP awards in the same year. 1998: Michael Jordan won NBA MVP, All-Star game MVP, and Finals MVP awards in the same year.

Due to the absence of annotated data, we consider leveraging the in-context learning capability of LLMs to accomplish the decomposition task. Before starting in-context learning, we use SU-Time (Chang and Manning, 2012) to identify time expressions in sentences, as specialized tools excel in time recognition compared to large LLMs. With-

out even a decomposition example, we iteratively construct the prompt for in-context learning. First, we only give ChatGPT3.5 the task description and one test sentence:

Instruction: First, I will give you a Text, and secondly, I will give you the time contained in the Text. You need to sort out the events that occurred at each time I provided based on the Text. The Decomposition format is required to contain a time point and corresponding events in each sentence. Each sentence contains the complete elements of the event, such as subject, predicate, and object.

We undertake this exploration to understand the output format preferences of ChatGPT3.5. This knowledge will guide us in uncovering the potential of the model to achieve optimal performance by aligning with these preferences. Subsequently, we make slight modifications to the output to generate a demonstrative example that is both accurate and in line with the model’s preferred output format. After several iterations like this, we get the demonstrative examples needed for in-context learning.

Human Feedback Enhanced In-Context Learning We find that even though we have shown ChatGPT3.5 high-quality demonstrative examples, it still makes mistakes when outputting. We add negative examples in the prompt to help the model avoid common mistakes. We carefully select the examples that we believe are representative of the mistakes made by ChatGPT3.5. Subsequently, we add human feedback to point out the mistakes and provide corrected answers. The complete prompt will be shown in Appendix E.

4.3 Fine-tuning Models with Timeline-based Sentence Decomposition

We combine the natural language understanding and reasoning capabilities emerging from very large LMs with smaller LMs fine-tuned for specific tasks. Specifically, we fine-tune generative language models (Llama2 and Flan-T5) with timeline-based sentence decomposition. We splice the text in the dataset with the decomposition results generated by ChatGPT3.5 as a new input for training. The following is an example of the input we provide for models:

Input: Text: Lamberto Visconti di Eldizio (died 1225) was the Judge of Gallura from

1206 , when he married the heiress Elena , to his own death . Decomposition: 1225: Lamberto Visconti di Eldizio passed away and ended serving as the Judge of Gallura. 1206: Lamberto Visconti di Eldizio became the Judge of Gallura and married the heiress Elena.

Figure 2 illustrates the distinction between training directly and training with the utilization of temporal-based sentence decomposition (TSD) information in the framework. We name the method of fine-tuning Flan-T5 using TSD as TSDRE.

5 ComplexTRED: A Complex Temporal Fact Extraction Dataset

There are very few temporal fact extraction datasets available to date. Pravda (Wang et al., 2012) lacks annotation labels and therefore cannot be used for supervised learning. The Wiki-People Dataset (Liu et al., 2021) is not open source. HyperRED (Chia et al., 2022) is a hyper-relational fact extraction dataset, 48% of which are temporal facts. However, many samples in HyperRED are overly simplistic and insufficient to depict the potential difficulties in complex practical scenarios. Specifically, the majority of sentences in HyperRED consist solely of a single temporal expression, and their respective extraction results comprise a singular temporal fact.

We need a complex temporal fact extraction dataset to evaluate our method and train existing models on their ability to extract temporal facts from complex temporal sentences. Below we will introduce data collection and dataset statistics.

5.1 Data Collection

It is very challenging to construct a large-scale, diverse complex temporal fact extraction dataset. The two major difficulties are the number of complex temporal sentences and high-quality annotated labels. In order to collect enough data, we use two methods to obtain data: using distant supervision to obtain the alignment of web text and temporal fact in KG, and manually correcting the samples in HyperRED. We will introduce our means of controlling the quality of the dataset in the introduction of each method.

Distant Supervision We collect the introduction sections of 343,603 DBpedia articles, the full text of 401,796 Wikipedia articles, and 3,002,373 Wikidata temporal facts for alignment. Specifically, we use DBpedia Spotlight (Mendes et al., 2011) for

entity linking and SUTime (Chang and Manning, 2012) to extract time entities from DBpedia and Wikidata articles. In distant supervision, if a sentence contains the head entity, tail entity and temporal entity of a temporal fact, then we align the fact to the sentence.

Distant supervision encounters two significant challenges: noise and incomplete facts (The sentence may express facts that are not in the knowledge graph). However, by aligning the two entities plus time, the noise problem is greatly mitigated. To tackle the incomplete facts problem, we leverage ChatGPT to complete the facts contained in the sentence. Subsequently, we organize 50 computer science undergraduate students to manually label the correctness of the added facts by ChatGPT3.5. Finally, we obtain about 17,000 complex sentences through distant supervision.

Manual Correction from HyperRED Samples in HyperRED are alignment with Wikipedia text and Wikidata data, which are from the same source as our samples. Therefore, we correct the labels of some complex sentences from HyperRED and absorb them into our dataset. We first select about 7,000 sentences in HyperRED that contain more than or equal to two time expressions as candidates. Then we organize 70 computer science undergraduates to correct the labels of each sentence, and the corrected samples will be added to our dataset. Finally, we get 2,589 corrected samples.

5.2 Dataset Statistics

To ensure that the train set, dev set, and test set data adhere to an independent and identical distribution, we employ stratified sampling on the original data based on the relation types. Our goal is to maintain an 8:1:1 ratio for each relation type in the train, dev, and test set, respectively. The resulting splitting of the dataset is presented in Table 1. We manually check all samples in the dev and test set to ensure the dataset’s quality.

The analysis of the sample is shown in the Table 2. ComplexTRED significantly outperforms HyperRED-Temporal in terms of sentence length, temporal expressions, and temporal facts contained per sentence, which reflects the difficulty of the dataset to a certain extent.

Datasets		#Sent.	#Facts	#Rel.
HyperR.	Train	17,004	23,507	45
	Dev	432	582	37
	Test	1,712	2,391	41
ComplexT.	Train	16,573	33,632	40
	Dev	1,679	4,025	40
	Test	1,584	3,964	39

Table 1: Statistics about the number of sentences, temporal facts, and relations of the two datasets.

Metrics	HyperR.	ComplexT.
Sentence length	31.47	41.27
Time expressions	1.95	3.46
Temporal facts	1.38	2.10

Table 2: Sample-level comparison between HyperRED-Temporal and ComplexTRED. Each metric is calculated on an average per sample basis.

6 Experiments

6.1 Experimental Setup

Datasets We assess temporal fact extraction using HyperRED (Chia et al., 2022), a publicly available benchmark, and ComplexTRED, a dataset specifically crafted by us for complex temporal fact extraction. We select samples in HyperRED whose labels are temporal facts, delete the relations that have little correlation with time, and finally form HyperRED-Temporal as the evaluation dataset. The statistics of the two datasets are shown in Table 1.

Compared Methods We compare our approach with the SOTA extraction method reported on CubeRE (Chia et al., 2022) and several baseline methods. We evaluated the performance of directly employing LLMs for temporal fact extraction, including LoRA fine-tuning the 7B version of Llama2 (Touvron et al., 2023b) and naive in-context learning with ChatGPT3.5. We also tried to leverage our Timeline-based Sentence Decomposition (TSD) to enhance LoRA fine-tuned Llama2.

In addition to directly employing LLMs, we also evaluated the performance of Flan-T5-Large (Chung et al., 2022), a relatively smaller pre-trained language model (PLM). We transplant the method of Wadhwa et al. (2023) on relation extraction (triple extraction) to temporal fact extraction, which trains Flan-T5 with LLM-generated explanations. We use ChatGPT3.5 to generate the

Methods	HyperRED-Temporal			ComplexTRED		
	Pr	Re	F_1	Pr	Re	F_1
ChatGPT3.5 [†]	11.26	20.08	14.43	19.01	23.57	21.05
Llama2 w LoRA [†] (Touvron et al., 2023b)	56.95	31.87	40.87	35.72	17.74	23.71
Llama2 w LoRA [†] + TSD [†] (OURS)	56.60	48.06	51.98	40.64	27.18	32.58
CubeRE (Chia et al., 2022)	55.31	49.64	52.33	41.69	27.92	33.44
Flan-T5 (Chung et al., 2022)	<u>66.64</u>	<u>61.06</u>	<u>63.73</u>	47.64	<u>34.86</u>	<u>40.26</u>
Flan-T5 + Explanation [†] (Wadhwa et al., 2023)	64.99	58.85	61.76	<u>47.69</u>	34.07	39.75
TSDRE [†] w Flan-T5 (OURS)	68.61	64.91	66.71	48.99	37.61	42.55

Table 3: Main results on HyperRED-Temporal and ComplexTRED. [†] denotes that the corresponding foundation model is a LLM.

explanations required for training instead of GPT-3 in the Wadhwa et al. (2023), because ChatGPT3.5 has not yet come out when Wadhwa et al. (2023) was published and we believe ChatGPT3.5 performs better than GPT-3.

Evaluation metrics We report the precision (denoted as P), recall (denoted as R), and F_1 score of the evaluation results. When calculating metrics, we employ an exact match (at string level) approach without following Wadhwa et al. (2023)’s method of utilizing manual assessment for the outputs of large-scale models.

6.2 Main results

We first report the results of methods directly based on LLMs, and subsequently, we report the results of approaches that involve combining LLMs and smaller PLMs.

LLMs results. Based on Table 3, it is evident that both Fine-tuned open-source LLMs and in-context learning LLMs do not perform satisfactorily on both datasets. Among them, in-context ChatGPT3.5 consistently gets the lowest F_1 score. It is reasonable to achieve such performance without training. HyperRED-Temporal includes 45 relation types and 4 qualifier types, while ComplexTRED includes 40 relation types and 4 qualifier types. Despite providing detailed task settings for ChatGPT3.5, it is expected to be exceedingly challenging to provide accurate answers across all 180 (45*4)/160 (40*4) categories based solely on a few-shot examples. Fine-tuned Llama2 (7B) performs better than in-context ChatGPT3.5, but its F_1 score still falls considerably short of CubeRE. This could be attributed to the insufficient training samples to effectively support such a large language model, leading to suboptimal fitting. Additionally,

we conducted experiments using our decomposition method to enhance Llama2 training. This resulted in Llama2 achieving an 11-point F_1 score improvement on HyperRED-Temporal, and a 9-point F_1 score improvement on ComplexTRED. This result indicates that integrating our decomposition method into training enables the model to better learn the relationships between sentence features and temporal facts. Another possible reason we speculate is that high-quality decomposition results are suitable as training data and make the training of Llama2 more sufficient.

Results of combining LLMs and smaller PLMs.

Flan-T5 (Large) achieved surprising results on both datasets. We believe that compared to LLMs, smaller PLMs are easier to fine-tune and fit when the training data is insufficient. However, enhancing Flan-T5 with ChatGPT3.5-generated explanation resulted in a tiny drop in F_1 on both datasets. This method may be disadvantageous under exact match, which we have mentioned in the evaluation metrics in Section 6.1. Another reason we believe is that the temporal fact extraction task is inherently more challenging to interpret compared to the fact (triple) extraction task. Finally, our method TSDRE, which enhances Flan-T5 with timeline-based sentence decomposition (TSD), achieves state-of-the-art results on both datasets. We achieve remarkable results by combining the strengths of both large and small LMs: Large LMs excel at effectively organizing timelines in natural language, while small LMs prove more adept at precise fine-tuning for specific tasks.

6.3 Decomposition Quality

We randomly select 100 sentences and invite three experts to evaluate our decomposition results of

these sentences. We still use precision and recall as evaluation metrics. The goal of our decomposition is to divide different events in the text into different points in time. In this scenario, True Positive refers to the count of events correctly classified at their respective time points in the prediction results. False Positive indicates the count of events erroneously classified at the wrong time point. False Negative represents the count of events that were not predicted. It must be pointed out that the relatively ambiguous aspect is that opinions differ among individuals regarding what should be considered an event in natural language text.

	Precision	Recall
Prompt	93.80	93.53
Prompt + feedback	94.65	95.57

Table 4: Human evaluation of Timeline-based sentence decomposition results.

Table 4 shows the scores of manual evaluation. Overall, our decomposition results surpass 90 in both precision and recall metrics, indicating that we have acquired high-quality auxiliary information for training. This establishes a solid foundation for enhancing model performance. Moreover, prompt enhanced by human feedback has further improved precision and recall. This shows that human feedback on demonstrative examples can enable LLMs to understand the task requirement better and avoid some common errors.

6.4 Error Analysis

We select 50 sentences from each of the two datasets whose F_1 scores are less than 1 to analyze TSDRE’s performance. The results are illustrated in Table 5. We classify error types according to the

Main Errors	HyperR.	ComplexT.
NER	14%	24%
- totally wrong	8%	12%
- overlapped	6%	12%
Relation Extraction	20%	22%
Qualifier Classification	0%	0%
Time Selection	2%	6%
False Negative	30%	22%
Missing Facts	34%	26%

Table 5: The statistics of main errors of sampled sentences.

elements of the quintuple as Named Entity Recognition (NER) error, Relation Extraction error, Time Selection error, Qualifier Classification error, False

Negative and Missing Facts. It is worth noting that the error rates of Time selection and Qualifier Classification are very low, showing that TSDRE perform well when faced with time-to-fact correspondences. In addition, under relaxed standards, the prediction results of entities overlapping with the answer entities and the prediction results of false negatives can both be counted as correct. This means that the actual performance of the model is much better than the scores under the exact match measurement. Finally, completely wrong NER, wrong Relation Extraction, and Missing Facts are still the legacy problems of fact (triple) extraction.

6.5 Case Study

As is shown in Figure 2, Flan-T5 fails to capture information regarding the awards of the other two players, aside from O’Neal. However, after we incorporate decomposition into training, Flan-T5 successfully outputs all temporal facts. Smaller language models do have limited learning capabilities for implicit expressions, so introducing LLMs with powerful understanding capabilities for natural language can effectively make up for this shortcoming.

7 Conclusion

In this paper, we explore the application of large language models (LLMs) in the extraction of temporal facts. Our attempts indicate that directly employing LLMs for temporal fact extraction falls short of achieving satisfactory results. To tackle this issue, we introduce a timeline-based sentence decomposition (TSD) method. Building upon this, we propose TSDRE, which employs a relatively smaller PLM as its foundation, combined with LLM-driven TSD to achieve the extraction. Experiments demonstrate that TSDRE achieves SOTA results on two datasets and incorporating TSD into the training process can enhance the performance of LLMs on temporal fact extraction tasks. In the future, an interesting topic would be to explore the extraction of temporal facts from text that necessitate inferring the occurrence time based on existing temporal references, such as “three days later”, which has not yet received widespread attention.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (No. 62272219) and the Collaborative Innovation Center of Novel Software Technology & Industrialization.

Limitations

Our contribution does have important limitations. First, our decomposition results rely on ChatGPT for completion, and decomposition without training the open-source LLMs cannot achieve the desired results.

Second, we only tested the effect of ChatGPT on the GPT3.5-turbo model, but not on the latest GPT4 or GPT4-turbo, due to the significantly higher cost involved.

Third, for document-level temporal fact extraction, when combined with time-based sentence decomposition results, the input may exceed the maximum length allowed by the generative model.

Finally, our dataset construction inevitably introduces noise problems due to the use of distant supervision. Additionally, due to limited resources, we only checked the validation set and test set of ComplexTRED, which may result in some noise issues in the training set.

Ethics Statement

First and foremost, our proposed TSDRE method strives to enhance the overall performance of RE models in extracting temporal facts. However, given the inherent black-box nature of the generative model, it is inevitable that the extracted facts may possess certain quality issues. Hence, when employing our method to extract temporal facts and utilize them for downstream tasks, users must exercise caution in discerning the authenticity of these facts in order to mitigate potential real-world consequences arising from erroneous information.

Secondly, for dataset construction, we have gathered text from Wikipedia and DBpedia, as well as facts from Wikidata. These are publicly available datasets commonly utilized for dataset construction. Wikidata facts are under the Creative Commons CC0 License², while the texts obtained from both Wikipedia and DBpedia are licensed under the Creative Commons Attribution-ShareAlike 3.0 Unported License^{3,4}. Thus, we are able to freely utilize this data to construct our dataset, and our dataset will be released under the same license. Furthermore, we have organized some human annotators throughout the dataset construction process, and

each annotator has been duly compensated based on their respective working hours.

References

- Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. [Language models are few-shot learners](#). In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.
- Angel X. Chang and Christopher D. Manning. 2012. [Sutime: A library for recognizing and normalizing time expressions](#). In *Proceedings of the Eighth International Conference on Language Resources and Evaluation, LREC 2012, Istanbul, Turkey, May 23-25, 2012*, pages 3735–3740. European Language Resources Association (ELRA).
- Ziyang Chen, Jinzhi Liao, and Xiang Zhao. 2023. [Multi-granularity temporal question answering over knowledge graphs](#). In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 11378–11392. Association for Computational Linguistics.
- Yew Ken Chia, Lidong Bing, Sharifah Mahani Aljunied, Luo Si, and Soujanya Poria. 2022. [A dataset for hyper-relational extraction and a cube-filling approach](#). In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 10114–10133. Association for Computational Linguistics.
- Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Y. Zhao, Yanping Huang, Andrew M. Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. [Scaling instruction-finetuned language models](#). *CoRR*, abs/2210.11416.
- Wentao Ding, Hao Chen, Huayu Li, and Yuzhong Qu. 2022. [Semantic framework based query generation for temporal question answering over knowledge graphs](#). In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*,

²<https://www.wikidata.org/wiki/Wikidata:Licensing#Uses>

³<https://www.dbpedia-spotlight.org/licenses>

⁴<https://en.wikipedia.org/wiki/Wikipedia:Copyrights>

- EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022, pages 1867–1877. Association for Computational Linguistics.
- Volker Gast, Lennart Bierkandt, Stephan Druskat, and Christoph Rzymiski. 2016. Enriching timebank: Towards a more precise annotation of temporal relations in a text. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 3844–3850.
- Xiaochuang Han, Daniel Simig, Todor Mihaylov, Yulia Tsvetkov, Asli Celikyilmaz, and Tianlu Wang. 2023. [Understanding in-context learning via supportive pre-training data](#). In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 12660–12673. Association for Computational Linguistics.
- Xixin Hu, Yiheng Shu, Xiang Huang, and Yuzhong Qu. 2021. [Edg-based question decomposition for complex question answering over knowledge bases](#). In *The Semantic Web - ISWC 2021 - 20th International Semantic Web Conference, ISWC 2021, Virtual Event, October 24-28, 2021, Proceedings*, volume 12922 of *Lecture Notes in Computer Science*, pages 128–145. Springer.
- Xiang Huang, Sitao Cheng, Yiheng Shu, Yuheng Bao, and Yuzhong Qu. 2023. [Question decomposition tree for answering complex questions over knowledge bases](#). In *Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI 2023, Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence, IAAI 2023, Thirteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2023, Washington, DC, USA, February 7-14, 2023*, pages 12924–12932. AAAI Press.
- Erdal Kuzey and Gerhard Weikum. 2012. [Extraction of temporal facts and events from wikipedia](#). In *2nd Temporal Web Analytics Workshop, TempWeb '12, Lyon, France, April 16-17, 2012*, pages 25–32. ACM.
- Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.
- Xiaonan Li, Kai Lv, Hang Yan, Tianyang Lin, Wei Zhu, Yuan Ni, Guotong Xie, Xiaoling Wang, and Xipeng Qiu. 2023. [Unified demonstration retriever for in-context learning](#). In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 4644–4668. Association for Computational Linguistics.
- Yujia Li, Shiliang Sun, and Jing Zhao. 2022. [Tirgn: Time-guided recurrent graph network with local-global historical patterns for temporal knowledge graph reasoning](#). In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI 2022, Vienna, Austria, 23-29 July 2022*, pages 2152–2158. ijcai.org.
- Yonghao Liu, Di Liang, Mengyu Li, Fausto Giunchiglia, Ximing Li, Sirui Wang, Wei Wu, Lan Huang, Xiaoyue Feng, and Renchu Guan. 2023. [Local and global: Temporal question answering via information fusion](#). In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI 2023, 19th-25th August 2023, Macao, SAR, China*, pages 5141–5149. ijcai.org.
- Yu Liu, Wen Hua, and Xiaofang Zhou. 2021. [Temporal knowledge extraction from large-scale text corpus](#). *World Wide Web*, 24(1):135–156.
- Pablo N. Mendes, Max Jakob, Andrés García-Silva, and Christian Bizer. 2011. [Dbpedia spotlight: shedding light on the web of documents](#). In *Proceedings the 7th International Conference on Semantic Systems, I-SEMANTICS 2011, Graz, Austria, September 7-9, 2011*, ACM International Conference Proceeding Series, pages 1–8. ACM.
- Haithem Mezni. 2022. [Temporal knowledge graph embedding for effective service recommendation](#). *IEEE Trans. Serv. Comput.*, 15(5):3077–3088.
- OpenAI. 2023. [GPT-4 technical report](#). *CoRR*, abs/2303.08774.
- Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurélien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023a. [Llama: Open and efficient foundation language models](#). *CoRR*, abs/2302.13971.
- Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shrutu Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurélien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023b. [Llama 2: Open foundation and fine-tuned chat models](#). *CoRR*, abs/2307.09288.
- Naushad UzZaman, Hector Llorens, James Allen, Leon Derczynski, Marc Verhagen, and James Pustejovsky.

2012. Tempeval-3: Evaluating events, time expressions, and temporal relations. *arXiv preprint arXiv:1206.5333*.

Somin Wadhwa, Silvio Amir, and Byron C. Wallace. 2023. [Revisiting relation extraction in the era of large language models](#). In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 15566–15589. Association for Computational Linguistics.

Yafang Wang, Maximilian Dylla, Marc Spaniol, and Gerhard Weikum. 2012. [Coupling label propagation and constraints for temporal fact extraction](#). In *The 50th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference, July 8-14, 2012, Jeju Island, Korea - Volume 2: Short Papers*, pages 233–237. The Association for Computer Linguistics.

Yafang Wang, Bin Yang, Lizhen Qu, Marc Spaniol, and Gerhard Weikum. 2011. [Harvesting facts from textual web sources by constrained label propagation](#). In *Proceedings of the 20th ACM Conference on Information and Knowledge Management, CIKM 2011, Glasgow, United Kingdom, October 24-28, 2011*, pages 837–846. ACM.

Yafang Wang, Mingjie Zhu, Lizhen Qu, Marc Spaniol, and Gerhard Weikum. 2010. [Timely YAGO: harvesting, querying, and visualizing temporal knowledge from wikipedia](#). In *EDBT 2010, 13th International Conference on Extending Database Technology, Lausanne, Switzerland, March 22-26, 2010, Proceedings*, volume 426 of *ACM International Conference Proceeding Series*, pages 697–700. ACM.

Siheng Xiong, Ali Payani, Ramana Kompella, and Farmarz Fekri. 2024. [Large language models can learn temporal reasoning](#).

Yi Xu, Junjie Ou, Hui Xu, and Luoyi Fu. 2023. [Temporal knowledge graph reasoning with historical contrastive learning](#). In *Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI 2023, Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence, IAAI 2023, Thirteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2023, Washington, DC, USA, February 7-14, 2023*, pages 4765–4773. AAAI Press.

A Environments and Parameters

TSDRE’s results were achieved using a Python implementation running on a workstation with an Intel(R) Xeon(R) Gold 5222 CPU @ 3.80GHz, 376GB RAM and 3 NVIDIA RTX3090 graphics cards. Llama2’s results were achieved on a workstation with an Intel(R) Xeon(R) Gold 6248 CPU @ 2.50GHz, 472GB RAM, and 8 NVIDIA Tesla V100 graphics cards. The hyperparameter settings of CubeRE, Flan-T5, Llama2 and BART are shown

in Table 6. When applying LoRA to fine-tune Llama2, we used a rank of 8 and an alpha value of 32. Besides, We set the temperature value of ChatGPT3.5 to 0 to facilitate reproduction.

B TSDRE with BART

In this paper, we propose a pipeline architecture TSDRE which is backbone-free. We have reported in Section 6.2 that our model TSDRE w Flan-T5 achieves the SOTA results. We test the performance of replacing Flan-T5 Large (770M) with a slightly weaker generative model BART Large (340M). The results are shown in table 7. From the experimental results, it can be seen that the performance of BART as base model is indeed lower than that of Flan-T5 (mentioned in Table 3). However, TSD still enhances the performance of BART.

C Error Analysis

Here we present specific examples of three major types of errors.

C.1 NER Error

Input: Fifteen locomotives of British Rail Class 83 were built between 1960 and 1962 by English Electric at Vulcan Foundry , as part of British Rail ’s policy to develop a standard electric locomotive .

Prediction: [['British Rail Class 83', 'manufacturer', 'Vulcan Foundry', 'end time', '1962'], ['British Rail Class 83', 'manufacturer', 'Vulcan Foundry', 'start time', '1960']]

Error Analysis: The tail entity should be 'English Electric'.

C.2 Relation Extraction Error

Input: He was appointed manager of the Highland Railway ’ s Lochgorm Works in 1903 , and promoted to Assistant to the Chief Mechanical Engineer , Peter Drummond in 1903 .

Prediction: [['Peter Drummond', 'employer', 'Highland Railway', 'start time', '1903'], ['Peter Drummond', 'employer', 'Highland Railway', 'end time', '1903']]

Error Analysis: Relation should be 'director/ manager'.

Models	Data	Batch Sizes	Warm-up	Learning Rates	Max Epochs
CubeRE	HyperRED	32	0.2	5e-5	30
	ComplexTRED	32	0.2	5e-5	30
Flan-T5	HyperRED	2	0.12	2e-5	4
	ComplexTRED	2	0.12	2e-5	4
Llama2	HyperRED	4	default	1e-4	3
	ComplexTRED	4	default	1e-4	3
BART	HyperRED	2	0.12	2e-5	4
	ComplexTRED	2	0.12	2e-5	4

Table 6: Hyperparameters for CubeRE, Flan-T5, Llama2 and BART.

Methods	HyperRED-Temporal			ComplexTRED		
	Pr	Re	F ₁	Pr	Re	F ₁
BART Large (340M) (Lewis et al., 2019)	68.85	48.89	57.18	43.68	17.52	25.00
TSDRE [†] w BART Large (340M)	71.51	50.27	59.04	45.15	18.45	26.20

Table 7: Performance of BART as base model on HyperRED-Temporal and ComplexTRED. † denotes that the corresponding foundation model is a LLM.

C.3 Missing Facts

Input: Sir Hubert Edward Henry Jerningham , (18 October 1842 - 3 April 1914) was a British Liberal Party politician and Governor of Mauritius 1892 - 1897 , then Governor of Trinidad and Tobago between 1897 and 1900 .

Prediction: [['Hubert Edward Henry Jerningham', 'position held', 'Governor of Trinidad and Tobago', 'start time', '1897'], ['Hubert Edward Henry Jerningham', 'position held', 'Governor of Mauritius', 'end time', '1897'], ['Hubert Edward Henry Jerningham', 'position held', 'Governor of Mauritius', 'start time', '1892']]

Error Analysis: ['Hubert Edward Henry Jerningham', 'position held', 'Governor of Trinidad and Tobago', 'end time', '1900'] missed.

D Difference from the temporal relation extraction task

Indeed, it is easy to confuse temporal relation extraction with temporal fact extraction based on their names, but they are actually very different tasks. Specifically, temporal relation extraction aims to identifying the temporal relation (e.g., BEFORE, AFTER, OVERLAPS) between events and times, while temporal fact extractions (please refer to Section 3.1) aims to to extract facts with temporal

attributes (e.g., start_time, end_time). Therefore, datasets used for temporal relation extraction tasks such as TimeBank (Gast et al., 2016) and TempEval (UzZaman et al., 2012) are not suitable for evaluating our task.

E Prompt

E.1 In-Context ChatGPT3.5 Prompt

Task: Extract all the quintuples [subject, relation, object, qualifier, time point] from the input text.

Task requirements: Extract all the quintuples [subject, relation, object, qualifier, time point] from the input text.

Here are some concrete examples, the output format is a list of quintuples:

Input: He received the 1921 Nobel Prize in Physics for his services to theoretical physics ; in particular his discovery of the law of the photoelectric effect , a pivotal step in the evolution of quantum theory .

Output: [['Nobel Prize', 'winner', 'He', 'point in time', '1921'], ['He', 'award received', 'Nobel Prize', 'point in time', '1921']]

Input: It won the 1991 Nebula Award for Best Novelette and was nominated for the 1991 Hugo Award for Best Novelette .

Output: [['It', 'award received', 'Nebula

Award', 'point in time', '1991'], ['It', 'nominated for', 'Best Novelette', 'point in time', '1991']]

Input: They are currently the only club in Ulster to have won an All - Ireland Senior Club Hurling Championship , which they first won in 1983 .

Output: [['All - Ireland Senior Club Hurling Championship', 'winner', 'They', 'point in time', '1983']]

Input: Alexander Mackenzie , PC (January 28 , 1822 2013 April 17 , 1892) , was a building contractor and newspaper editor , and was the second Prime Minister of Canada , from November 7 , 1873 to October 8 , 1878 .

Output: [['Canada', 'head of government', 'Alexander Mackenzie', 'end time', 'October 8 , 1878'], ['Canada', 'head of government', 'Alexander Mackenzie', 'start time', 'November 7 , 1873'], ['Alexander Mackenzie', 'position held', 'Prime Minister', 'end time', 'October 8 , 1878'], ['Alexander Mackenzie', 'position held', 'Prime Minister', 'start time', 'November 7 , 1873']]

Input: There has been a resident Treasury or Downing Street cat employed as a mouser and pet since the reign of Henry VIII , when Cardinal Wolsey placed his cat by his side while acting in his judicial capacity as Lord Chancellor , an office he assumed in 1515 .

Output: [['Cardinal Wolsey', 'position held', 'Lord Chancellor', 'start time', '1515']]

Input: He had 24 caps for Japan , from 1974 to 1984 , scoring 3 tries , 5 conversions , 14 penalties and 3 drop goals , in an aggregate of 73 points .

Output: [['He', 'member of sports team', 'Japan', 'start time', '1974'], ['He', 'member of sports team', 'Japan', 'end time', '1984']]

Input: He later played professional football in the American Football League , appearing in 42 games as a tackle and defensive end for the New York Titans (later renamed the Jets) from 1960 to 1962 .

Output: [['He', 'member of sports team', 'New York Titans', 'start time', '1960'],

['He', 'member of sports team', 'New York Titans', 'end time', '1962']]

Input: " Suedehead " is the debut solo single from Morrissey , released in February 1988 .

Output: [['Suedehead', 'performer', 'Morrissey', 'publication date', 'February 1988']]

Input: It closed on 1 December 2003 when operation of the line was suspended between Kabe Station and Sandankyō14d Station .

Output: [['It', 'adjacent station', 'Kabe', 'end time', '1 December 2003'], ['It', 'adjacent station', 'Sandankyō14d', 'end time', '1 December 2003']]

Input: The 2008 presidential campaign of Barack Obama , then junior United States Senator from Illinois , was announced at an event on February 10 , 2007 in Springfield , Illinois .

Output: [['Barack Obama', 'candidacy in election', '2008 presidential campaign', 'start time', 'February 10 , 2007']]

Input: Among his victories were in reconquering Ji ' an in Jiangxi Province in 1856 , as well as leading the assault on the Taiping capital at Nanjing in 1864 .

Output: [['Nanjing', 'capital of', 'Taiping', 'end time', '1864']]

Input: He was also one of the original correspondents on Comedy Central ' s The Daily Show from 1996 to 1998 .

Output: [['The Daily Show', 'cast member', 'He', 'end time', '1998'], ['The Daily Show', 'cast member', 'He', 'start time', '1996']]

Input: He was International President of WWF from 1996 to 1999 succeeding Prince Philip , the Duke of Edinburgh .

Output: [['WWF', 'chairperson', 'He', 'end time', '1999'], ['WWF', 'chairperson', 'He', 'start time', '1996']]

Input: Timothy Fok Tsun - ting (born 14 February 1946 in Hong Kong) , GBS , JP , the eldest son of Henry Fok , is a Member of the Legislative Council of Hong Kong , representing the Sports , Performing

Arts , Culture and Publication functional constituency .

Output: [['Henry Fok', 'child', 'Timothy Fok Tsun - ting', 'start time', '14 February 1946']]

Input: The 2013 Philadelphia Eagles season was the franchise ' s 81st season in the National Football League , and the first under head coach Chip Kelly .

Output: [['Chip Kelly', 'coach of sports team', 'Philadelphia Eagles', 'start time', '2013']]

Input: Shek Kip Mei Station served as a terminus in the very early phase of the Kwun Tong Line (Shek Kip Mei to Kwun Tong , 1 October 1979 to 31 December 1979) .

Output: [['Shek Kip Mei', 'connecting line', 'Kwun Tong Line', 'start time', '1 October 1979']]

Input: Kōf6nigsberg was transferred to Soviet control in 1945 after World War II .

Output: [['Kōf6nigsberg', 'country', 'Soviet', 'start time', '1945']]

Input: She was the Director of the Walter and Eliza Hall Institute of Medical Research (WEHI) , from 1996 until 30 June 2009 and remains a faculty member , having rejoined the institute ' s Molecular Genetics of Cancer Division .

Output: [['WEHI', 'director / manager', 'She', 'end time', '30 June 2009'], ['WEHI', 'director / manager', 'She', 'start time', '1996']]

Input: He previously served as Commander , United States Transportation Command from September 2005 to August 2008 .

Output: [['United States Transportation Command', 'director / manager', 'He', 'end time', 'August 2008'], ['United States Transportation Command', 'director / manager', 'He', 'start time', 'September 2005']]

Input: He graduated from Pennsylvania State University in State College , PA in 1969 , and earned a J . D .

Output: [['He', 'educated at', 'Pennsylvania State University', 'end time', '1969']]

Input: She married in 1936 , and took up her first post in Liverpool University

, where she studied for the rest of her working life .

Output: [['She', 'employer', 'Liverpool University', 'start time', '1936']]

Input: Air Union was merged with four other French airlines to become Air France on 7 October 1933 .

Output: [['Air Union', 'followed by', 'Air France', 'point in time', '7 October 1933']]

Input: The Duchy of Magdeburg (German : Herzogtum Magdeburg) was a province of Brandenburg - Prussia from 1680 to 1701 and a province of the German Kingdom of Prussia from 1701 to 1807 .

Output: [['Brandenburg - Prussia', 'followed by', 'Prussia', 'point in time', '1701'], ['Magdeburg', 'located in the administrative territorial entity', 'Prussia', 'start time', '1701'], ['Magdeburg', 'located in the administrative territorial entity', 'Brandenburg - Prussia', 'start time', '1680']]

Input: In 1998 , the studio moved from Studio City , California to Burbank in celebration of a new facility , and was renamed Nickelodeon Animation Studio .

Output: [['Nickelodeon Animation Studio', 'headquarters location', 'Studio City , California', 'end time', '1998'], ['Nickelodeon Animation Studio', 'headquarters location', 'Burbank', 'start time', '1998']]

Input: Kimera Walusimbi was Kabaka of the Kingdom of Buganda between 1374 and 1404 .

Output: [['Kimera', 'noble title', 'Kabaka', 'start time', '1374'], ['Kimera', 'noble title', 'Kabaka', 'end time', '1404']]

Input: For the start of the 1982 season , the Minnesota Vikings moved from Metropolitan Stadium to the Hubert H . Humphrey Metrodome .

Output: [['Metrodome', 'occupant', 'Minnesota Vikings', 'start time', '1982'], ['Minnesota Vikings', 'home venue', 'Metrodome', 'start time', '1982']]

Input: It was established as the official legislature of Kampuchea on January 5 ,

1976 , consisting of 250 members .

Output: [['Kampuchea', 'legislative body', 'It', 'start time', 'January 5 , 1976']]

Input: They were designed by R . J . Billinton and built at Brighton works from 1895 to 1897 .

Output: [['They', 'manufacturer', 'Brighton works', 'start time', '1895']]

Input: Dee Palmer (formerly David Palmer ; born 2 July 1937) is an English composer , arranger , and keyboardist best known for having been a member of the progressive rock group Jethro Tull from 1977 to 1980 .

Output: [['David Palmer', 'member of', 'Jethro Tull', 'end time', '1980'], ['David Palmer', 'member of', 'Jethro Tull', 'start time', '1977']]

Input: In 1931 , he joined Joseph Lyons and several other members in leaving the Labor Party and joining with the Nationalists to create the United Australia Party .

Output: [['Joseph Lyons', 'member of political party', 'United Australia Party', 'start time', '1931'], ['Joseph Lyons', 'member of political party', 'Labor Party', 'end time', '1931']]

Input: He was in the United States Army during World War II , from 1943 to 1946 .

Output: [['He', 'military branch', 'United States Army', 'end time', '1946'], ['He', 'military branch', 'United States Army', 'start time', '1943']]

Input: She was named after the title character of the 1866 opera Mignon , written by her godfather , French composer Ambroise Thomas .

Output: [['Ambroise Thomas', 'notable work', 'Mignon', 'publication date', '1866']]

Input: It was best known as the home of the Detroit Red Wings hockey team of the National Hockey League from its opening until 1979 .

Output: [['Detroit Red Wings', 'home venue', 'It', 'end time', '1979'], ['It', 'occupant', 'Detroit Red Wings', 'end time', '1979']]

Input: He became a solicitor in 1900 and a barrister in 1913 , being a member of

both King ' s Inns , Dublin , and Gray ' s Inn , London .

Output: [['He', 'occupation', 'solicitor', 'start time', '1900'], ['He', 'occupation', 'barrister', 'start time', '1913']]

Input: The Portuguese Air Force (PoAF) operated 50 LTV A - 7 Corsair II aircraft in the anti - ship , air interdiction and air defense roles between 1981 and 1999 .

Output: [['LTV A - 7 Corsair II', 'operator', 'Portuguese Air Force', 'start time', '1981'], ['LTV A - 7 Corsair II', 'operator', 'Portuguese Air Force', 'end time', '1999']]

Input: Brickleberry is an American animated comedy that premiered on September 25 , 2012 on Comedy Central .

Output: [['Brickleberry', 'original broadcaster', 'Comedy Central', 'start time', 'September 25 , 2012']]

Input: Shellen joined Google in 2003 when the company acquired Pyra Labs , which developed the Blogger blogging platform .

Output: [['Pyra Labs', 'owned by', 'Google', 'start time', '2003']]

Input: Volkswagen purchased the Bugatti trademark in June 1998 and incorporated Bugatti Automobiles S . A . S .

Output: [['Bugatti', 'owned by', 'Volkswagen', 'start time', 'June 1998']]

Input: In 1986 , the company was acquired by Penguin Group and split into two imprints : Dutton and Dutton Children ' s Books .

Output: [['Dutton', 'parent organization', 'Penguin Group', 'start time', '1986'], ['Dutton Children ' s Books', 'parent organization', 'Penguin Group', 'start time', '1986']]

Input: In 1990 , following the Iraqi invasion of Kuwait , Saudi Arabia participated in the Gulf War to expel Iraqi forces from the country .

Output: [['Gulf War', 'participant', 'Saudi Arabia', 'point in time', '1990']]

Input: On November 18 , 1928 the first Mickey Mouse cartoon released to the

public , Steamboat Willie , debuted at the Colony .

Output: [['Mickey Mouse', 'present in work', 'Steamboat Willie', 'point in time', 'November 18 , 1928']]

Input: The Third Republic of South Korea was replaced in 1972 by the Fourth Republic of South Korea under the Third Republic of South Korea ' s president Park Chung - hee .

Output: [['Fourth Republic of South Korea', 'replaces', 'Third Republic of South Korea', 'point in time', '1972']]

Input: He returned to favor in 1942 and was recalled to Moscow .

Output: [['He', 'residence', 'Moscow', 'start time', '1942']]

Input: On 20 July 2012 , the Constable welcomed the Olympic Torch to London at the Tower one week in advance of the London 2012 Summer Olympic Games , as part of the Olympic torch relay .

Output: [['London', 'significant event', 'London 2012 Summer Olympic Games', 'point in time', '20 July 2012']]

Input: and Max Verstappen , who in 2015 became the youngest driver in Formula One history at just 17 years old .

Output: [['Max Verstappen', 'sport', 'Formula One', 'start time', '2015']]

Input: She was married to internationally famous writer Jorge Amado from 1945 until his death in 2001 .

Output: [['She', 'spouse', 'Jorge Amado', 'start time', '1945'], ['She', 'spouse', 'Jorge Amado', 'end time', '2001']]

Input: It was first listed on the London Stock Exchange in 2005 is now a constituent of the FTSE 100 Index .

Output: [['It', 'stock exchange', 'London Stock Exchange', 'start time', '2005']]

Input: The Valencia Street Circuit (Valencian : Circuit Urbà de València , Spanish : Circuito Urbano de Valencia) was a street circuit in Valencia , Spain which hosted the Formula One European Grand Prix for five years (2008 2013 2012) .

Output: [['Valencia Street Circuit', 'used by', 'Formula One', 'start time', '2008'], ['Valencia Street Circuit',

'used by', 'Formula One', 'end time', '2012']]

Now please extract the quintuples from the following input:

Input:

E.2 Decomposition Prompt with Feedback

First, I will give you a TEXT, and secondly, I will give you the TIME contained in the TEXT. You need to sort out the events that occurred at each time I provided based on the TEXT. The DECOMPOSITION format is required to contain a time point and corresponding events in each sentence. Each sentence contains the complete elements of the event, such as subject, predicate, and object. DECOMPOSITION ends with </s>. There are two evaluation criteria for this task: Precision and Recall. The definitions of Precision and Recall under this task are as follows:

Precision: Whether the time at which each event occurred is given correctly.

Recall: Whether all events that occurred at each time are given completely.

I will show you some examples. At the same time, i will give you human feedback for decomposition results in these examples. I need you to learn from these examples and feedback how to decompose TEXT correctly and completely:
EXAMPLE1:

TEXT: The Duchy of Magdeburg (German : Herzogtum Magdeburg) was a province of Brandenburg - Prussia from 1680 to 1701 and a province of the German Kingdom of Prussia from 1701 to 1807 .

TIME: ['from 1680 to 1701', 'from 1701 to 1807']

DECOMPOSITION: From 1680 to 1701: The Duchy of Magdeburg was a province of Brandenburg-Prussia. From 1701 to 1807: The Duchy of Magdeburg was a province of the German Kingdom of Prussia. </s>

feedback: Precision: very good. Recall: very good.

EXAMPLE2:

TEXT: From its foundation in 1921 until moving to Ashton Gate in 2014 the Memorial Ground was the home to Bristol Rugby .

TIME: ['1921', '2014']

DECOMPOSITION: 1921: The Memorial Ground was founded and became the home to Bristol Rugby. 2014: The Memorial Ground moved to Ashton Gate. </s>

feedback: Precision: Not good. It was not the Memorial Ground that was founded in 1921 but Bristol Rugby. Besides, it was not the Memorial Ground but Bristol Rugby that moved to Ashton Gate. Recall: Not good. Two events are not recalled: In 1921 Bristol Rugby was founded and in 2014 Bristol Rugby moved to Ashton Gate. A correct and complete decomposition is as follows: 1921: Bristol Rugby was founded and the Memorial Ground the home to Bristol Rugby. 2014: Bristol Rugby moved to Ashton Gate.

EXAMPLE3:

TEXT: Alexander Mackenzie , PC (January 28 , 1822 2013 April 17 , 1892) , was a building contractor and newspaper editor , and was the second Prime Minister of Canada , from November 7 , 1873 to October 8 , 1878 .

TIME: ['January 28, 1822', 'November 7, 1873', 'October 8, 1878', 'April 17, 1892']

DECOMPOSITION: January 28, 1822: Alexander Mackenzie, PC, was born. November 7, 1873: Alexander Mackenzie started serving as the second Prime Minister of Canada. October 8, 1878: Alexander Mackenzie ended his term as the second Prime Minister of Canada. April 17, 1892: Alexander Mackenzie passed away. </s>

Feedback: Precision: very good. Recall: very good.

EXAMPLE4:

TEXT: Pope Benedict XV (Latin : Benedictus XV ; Italian : Benedetto XV) born Giacomo Paolo Giovanni Battista della Chiesa , (21 November 1854 2013 22 January 1922) , was Pope from 3 September 1914 until his death in 1922 .

TIME: ['21 November 1854', '22 January 1922', '3 September 1914', '1922']

DECOMPOSITION: 21 November 1854: Pope Benedict XV, born Giacomo Paolo Giovanni Battista della Chiesa, was born. 22 January 1922: Pope Benedict XV passed

away. 3 September 1914: Pope Benedict XV started serving as Pope. 1922: Pope Benedict XV ended serving as Pope. </s>

feedback: Precision: very good. Recall: not good. In 22 January 1922, Pope Benedict XV passed away and was no longer Pope. In 1922: Pope Benedict XV passed away and ended serving as Pope. A correct and complete decomposition is as follows: 21 November 1854: Pope Benedict XV, born Giacomo Paolo Giovanni Battista della Chiesa, was born. 22 January 1922: Pope Benedict XV passed away and ended serving as Pope. 3 September 1914: Pope Benedict XV started serving as Pope. 1922: Pope Benedict XV passed away and ended serving as Pope.

EXAMPLE5:

TEXT: He received : Serbian NIN Prize 1978 and Andrić Award 1987 , Swedish PEN Tucholsky Award (1993) , German Herder Prize (1995) , Montenegrin Njegoš Award (2009) and 13th July award (2004) , Slovenian Vilenica Award (2003) , Croatian Vladimir Nazor Award (2008) etc .

TIME: ['1978', '1987', '1993', '1995', '2003', '2004', '2008', '2009']

DECOMPOSITION: 1978: He received the Serbian NIN Prize. 1987: He received the Andrić Award. 1993: He received the Swedish PEN Tucholsky Award. 1995: He received the German Herder Prize. 2003: He received the Slovenian Vilenica Award. 2004: He received the 13th July award. 2008: He received the Croatian Vladimir Nazor Award. 2009: He received the Montenegrin Njegoš Award. </s>

feedback: Precision: very good. Recall: very good.

EXAMPLE6:

TEXT: It is one of the most famous old districts in Poland today and was the center of Poland ' s political life from 1038 until King Sigismund III Vasa relocated his court to Warsaw in 1596 .

TIME: ['1038', '1596']

DECOMPOSITION: 1038: It was the center of Poland's political life. 1596: King Sigismund III Vasa relocated his court to Warsaw. </s>

feedback: Precision: very good. Recall:

not good. In 1596 it was no longer the center of Poland ' s political life. A correct and complete decomposition is as follows: 1038: It was the center of Poland's political life. 1596: It was no longer the center of Poland ' s political life since King Sigismund III Vasa relocated his court to Warsaw.

EXAMPLE7:

TEXT: He played for Gloucestershire between 1920 and 1921 .

TIME: ['1920', '1921']

DECOMPOSITION: 1920: He played for Gloucestershire. 1921: He played for Gloucestershire. </s>

feedback: Precision: not good. This decomposition does not convey the meaning that he started playing for Gloucestershire in 1920 and ended playing for Gloucestershire in 1921.

Recall: very good. A correct and complete decomposition is as follows: 1920: He started playing for Gloucestershire. 1921: He ended playing for Gloucestershire.

Now process the following TEXT and follow my examples above for the DECOMPOSITION format:

F Relations Deleted from HyperRED

There are 57 types of relationships with time qualifier in the HyperRED data set. We have deleted the following relations:

Distant supervision of these relations results in excessive noise:

Shares_Border_With, Country_Of_Citizenship, Instance_Of, League, Subclass_Of, Part_Of, Partner_In_Business_Or_Sport.

Easily confused relations:

Head_Of_State
(Head_Of_Government), Location (Located_In_The_Administrative_Territorial_Entity), Part_Of (Member_of), Participating_team (Participant), Voice_Actor (Performer).

In ComplexTRED, we delete 5 relations from HyperRED-Temporal:

Distant supervision of these relations results in excessive noise:

Followed_By (Replaces), Award_Received (Winner), Head_Of_Government (Position held),

Easily confused relations: