CXTime: A General Rule-based Method for Chinese Time Expression Recognition and Normalization

Xiaoshi Zhong[1], Yufei Zhao[2], Liya Bi[1], and Erik Cambria[3]
[1]*School of Computer Science and Technology, Beijing Institute of Technology, China*
[2]*Jincheng Power Supply Company, State Grid Shanxi Electric Power Company, China*
[3]*College of Computing and Data Science, Nanyang Technological University, Singapore*
xszhong@bit.edu.cn; yfzhao167@163.com; 1120230592@bit.edu.cn; cambria@ntu.edu.sg

***Abstract*—Time expression (timex) recognition and normalization (TERN) is a crucial task for many downstream applications. Although TERN has been extensively studied in English, it has received limited attention in Chinese. Moreover, high-quality resources for Chinese TERN are scarce. To address these challenges, we construct a new dataset collected from Chinese social media and manually annotate Chinese timexes, and then analyze Chinese timexes using both our annotated dataset and an existing benchmark dataset. From the analysis we summarize seven statistical characteristics, such as: (1) Chinese timexes typically contain at least one time token; (2) strong mappings exist between time tokens and Chinese timex types; (3) Chinese timex types follow a priority relationship; and (4) timex normalized values conform to a small set of structured formats. Based on these characteristics, we propose CXTime, a general heuristic rule-based method for Chinese TERN. Experiments on diverse datasets demonstrate that CXTime significantly outperforms representative state-of-the-art models.**

***Index Terms*—Chinese timex, time tokens, token triples, mapping relations, priority relationship**

I. Introduction

TERN plays a fundamental role in natural language understanding, as it provides essential temporal grounding for tasks such as event timeline summarization [6], and temporal reasoning [7]. Despite extensive research on timex processing in English, Chinese TERN receives limited attention and it remains a challenging problem. Chinese TERN aims to recognize Chinese timexes from unstructured text and standardize the recognized timexes into structured formats. For instance, given the sentence "该活动于 2013 年 4 月 15 日 8 点开始，持续时间大约一个月" (*The event started at 8:00 on April 15, 2013, and lasted for approximately one month*)[1] as shown in Figure 1, the goal of Chinese TERN is to recognize "2013 年 4 月 15 日 8 点" (*8:00 on April 15, 2013*) and "大约一个月" (*approximately one month*) as timexes from the sentence, and then normalize the first timex to the type of TIME and the standard value of "2013-04-15T08:00:00" and the second timex to DURATION and "P1M".

[1]After each Chinese example, we provide its English translation in parentheses.

Compared with English, Chinese temporal meanings are often encoded through flexible combinations of characters whose semantics are sensitive to token order, making similar expressions convey entirely different time information. For example, the two Chinese timexes "3 天后" (*three days later*) and "后 3 天" (*next three days*) contain the same temporal tokens but refer to different time spans: the former denotes a time point relative to the reference time, while the latter often functions as a duration modifier. These characteristics increase the difficulty of accurately recognizing and normalizing Chinese timexes in text.

Existing methods for Chinese TERN have been dominated by deterministic rule-based methods [26]–[28]. These approaches are effective in well-defined domains, but their coverage and robustness remain limited when facing informal expressions. Learning-based methods have been introduced to formulate timex recognition as a sequence labeling problem, yet largely depend on rule-based components for normalization [12], [29]. Moreover, the availability of high-quality, publicly accessible resources for Chinese TERN research is still severely limited.

To address these challenges, we construct a large-scale dataset, termed **WBPost**[2], consisting of informal texts collected from Weibo (a Chinese social media platform). We then conduct a systematic analysis of Chinese timexes from WBPost and three versions of the existing benchmark dataset **TempEval-2 Chinese** (**TE2C**) [25]: **TE2C-Original**, **TE2C-Clean** [2], and **TE2C-Improved** [15]. From our analysis we summarize seven statistical characteristics about Chinese timex constituents (see Section III-B). These characteristics are highly consistent with those of English timexes reported in [34], [35], suggesting that timexes exhibit consistent structural properties across languages.

Motivated by these observations and inspired by [34], we propose CXTime, a general heuristic rule-based method for Chinese TERN. CXTime explicitly models three kinds of meta time information (see Section IV-A), and applies them to a structured pipeline (see Section IV-B) to recognize Chinese timexes from text and normalize them into

[2]See Section V-A1 and Table I for WBPost construction and statistics.

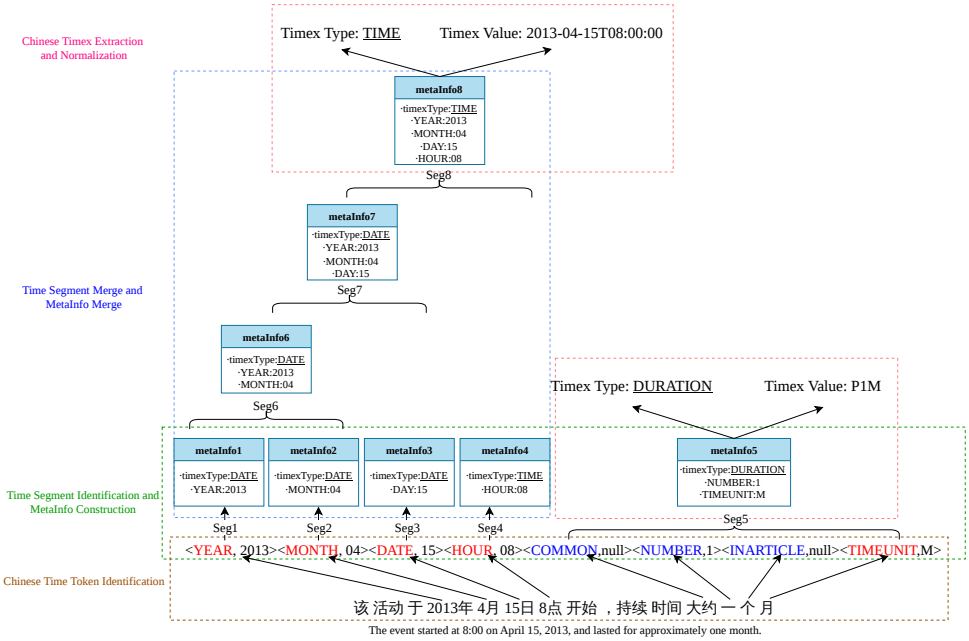


Fig. 1. An example of CXTime recognizing Chinese timexes from unstructured text and normalizing them into standard type and value formats. CXTime first identifies time tokens and assigns each time token a <*token type*, *token value*> pair, then searches the surroundings of time tokens to form time segments and initializes a MetaInfo instance for each time segment. After that CXTime merges some time segments and corresponding MetaInfo instances, and finally extracts timexes from final time segments and determines the timexes' types and values.

standard type and value formats. As CXTime operates on abstract token types and is independent of surface lexicons, it is independent of specific domains and genres. Moreover, it is lightweight for Chinese timex processing.

We evaluate CXTime under both end-to-end TERN and pure timex normalization settings, and compare it to several types of representative state-of-the-art methods: rule-based [15], learning-based [8], [17], masked language model-based [13], and LLM-based [10]. Experimental results demonstrate that CXTime significantly and consistently outperforms the state-of-the-art baselines in both Chinese TERN and pure Chinese timex normalization. Moreover, its use of general heuristic rules enables CXTime to handle diverse Chinese timexes while remaining lightweight, and capable of processing in real-time.

Both WBPost and CXTime are publicly available at https://github.com/xszhong/CXTime.

II. Related Work

TERN has been widely studied for English [32], [33], evolving from rule-based systems to neural and LLM-based approaches. By contrast, Chinese TERN remains relatively under-explored due to the flexible structure of Chinese timexes and the scarcity of annotated corpora.

Rule-based approaches represent the earliest paradigm for TERN [5], [23], [34]. For Chinese TERN, early research proposes linguistic frameworks tailored to Chinese temporal concepts [11], [16]. Subsequent studies incorporate linguistic features such as part-of-speech tags and temporal keywords to improve recognition accuracy [27], [28]. [15] further extend HeidelTime to Chinese using language-specific rules and resources.

To reduce reliance on handcrafted rules, learning-based approaches have been increasingly applied to TERN [1], [14], [31]. For Chinese TERN, neural sequence models such as BiLSTM-CRF are increasingly adopted [29], [30]. Because normalization still relies heavily on heuristic rules, later hybrid approaches combine neural recognition with rule-based normalization [12], [17], [20].

Recent studies apply transformer-based models and LLMs to timex processing. Transformer encoders improve contextualized timex recognition [4], [19], while multilingual neural models and prompt-based methods are explored for timex normalization [10], [13], [21]. While these methods provide flexibility, their performance remains sensitive to prompt design and annotated data quality.

Inspired by [34], [35], we propose to design general heuristic rules for Chinese TERN. These general heuristic rules are flexible and interpretable, and they are independent of specific domains and text genres.

III. Data Analysis and Observations

A. Datasets

We employs two Chinese corpora annotated for Chinese timex analysis: **TE2C** [25] and **WBPost**. TE2C has been developed into three versions: **Original**, **Clean** [2], and **Improved** [15]; WBPost is our annotated corpus. The two corpora represent two complementary evaluation scenarios: **formal, well-written, small-scale news text** versus **noisy, user-generated, large-scale social media text**. Table I summarizes the statistics of the TE2C and WBPost corpora.

TABLE I

STATISTICS OF THE TE2C AND WBPOST CORPORA. "DURA" INDICATES THE TIMEX TYPE "DURATION". **For data analysis in Section III, statistical observations are summarized on the training sets, while in Section V, experimental results are reported on the test sets.**

Dataset	Type	#Docs	#Timex	#DATE	#TIME	#SET	#DURA
TE2C-Original	train	44	746	623	10	0	113
	test	15	190	160	0	3	27
TE2C-Clean	train	44	661	555	10	0	96
	test	15	143	128	0	0	15
TE2C-Improved	train	44	765	628	10	2	125
	test	15	193	166	0	4	23
WBPost	train	2,709	3,565	1,880	493	209	983
	test	678	912	469	280	42	121

TABLE II

MAPPING RELATIONS BETWEEN TIME TOKENS AND TIMEX TYPES

Time Token	Timex Type
YEAR, MONTH, WEEK, DATE, TIMELINE, ERA, SEASON, DECADE, HOLIDAY	DATE
TIME, DAYTIME, HOUR, MINUTE, SECOND	TIME
PERIODICAL	SET
DURATION	DURATION

B. Observations

We conduct a comprehensive analysis on **the training sets** of the TE2C-Original, TE2C-Clean, TE2C-Improved, and WBPost datasets, with a focus on the structural properties of Chinese timexes. Our analysis suggests that Chinese timexes in diverse datasets exhibit consistent patterns in component combinations, syntactic realization, and distributional characteristics. In what follows, we briefly summarize the statistical characteristics about the constituents of Chinese timexes.[3]

Observation 1: **Chinese timexes are typically short and compact.**

Observation 2: **Most Chinese timexes explicitly contain time tokens.**

Observation 3: **Chinese timexes are composed from a small set of time tokens.**

Observation 4: **Chinese timexes exhibit strong part-of-speech regularities.**

Observation 5: **Strong mapping relations exist between time tokens and timex types: a specific type of time token primarily appears in a specific type of timex.**

We find strong mapping relations existing between time tokens and timex types, and Table II summarizes these mapping relations.

Observation 6: **There exists a general priority relationship among the four timex types: DATE < TIME < DURATION < SET.**

Observation 7: **Standard timex values have only a couple of formats and these formats are mainly composed of time tokens and numerals.**

These characteristics of Chinese timexes are highly consistent with those of English timexes reported in [34], [35].

[3]Due to space limitation, we do not present the detailed statistics of the observations in this paper. They are provided in the extended version.

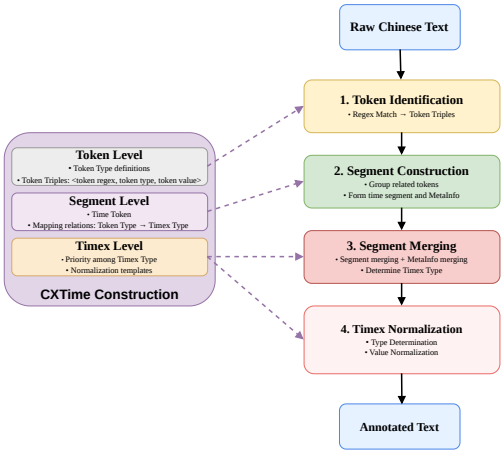


Fig. 2. Framework of CXTime for Chinese TERN. The left-hand side shows the CXTime construction with three kinds of meta time information: (1) token triples, (2) mapping relations between time tokens and timex types, and (3) priority among timex types. The right-hand side shows the main steps of CXTime recognizing and normalizing Chinese timexes into standard type and value formats.

IV. CXTIME: CHINESE TERN WITH GENERAL HEURISTIC RULES

The framework of CXTime for Chinese TERN in practice mainly contains two components, as illustrated in Figure 2: (1) CXTime construction and (2) Chinese timex extraction and normalization.

A. CXTime Construction

The construction of CXTime is initialized with three kinds of meta time information: (1) token triples, (2) mapping relations between token types and timex types, as described in Observation 5, and (3) priority relationship among timex types, as analyzed in Observation 6.

1) Token Triples: Similar to [34], CXTime defines *token triples* for Chinese time-related tokens. Each token triple represents a time token as a combination of a token regular expression (a.k.a., "*regex*"), a token type, and an associated value when applicable:

$$\text{Token triple} := <\textit{token regex}, \textit{token type}, \textit{token value}>$$

CXTime distinguishes three categories of token types:

Time Token. CXTime defines 17 kinds of time tokens: YEAR, MONTH, WEEK, DATE, TIME, HOUR, MINUTE, SECOND, DAYTIME, TIMELINE, TIMEUNIT, SEASON, DECADE, HOLIDAY, PERIODICAL, DURATION, and ERA.

Modifier. CXTime organizes modifiers into 7 categories: COMMON, QUANTITATIVE, FREQUENT, prefix and suffix temporal operators (OPERATE_{Pre} and OPERATE_{Suf}), linkage symbols (LINKAGE), and punctuation (COMMA).

Numeral. Numerals in Chinese, including *cardinal numbers* (e.g., "1" and "二" (*two*)) and *ordinal numbers* (e.g., "第一" (*first*) and "第二" (*second*)), are treated as a separate token category. They are essential components of many timexes, such as "2023 年" (*the year 2023*) and "第十周" (*the tenth week*).

Table III presents several examples of token triples.

TABLE III
SOME EXAMPLES OF TOKEN TRIPLES USED IN CXTIME. "NULL" INDICATES THAT THE TOKEN DOES NOT HAVE A VALUE.

Token Triple
<"一月 |1 月" (*January*) , MONTH , 01>
<"下午" (*afternoon*) , DAYTIME , AF>
<"每年" (*every year*) , PERIODICAL , XXXX>
<"大约 | 左右" (*approximately*) , COMMON , null>

2) Mapping from Time Tokens to Timex Types: CXTime defines explicit mapping relations between token types and timex types, as shown in Table II.

3) Priority Relationship among Timex Types: Observation 6 summarizes a general preference among the four timex types: DATE <TIME <DURATION <SET. This priority is utilized to determine the timex type of a timex when the timex contains two or more time tokens.

B. Chinese Timex Recognition and Normalization

CXTime performs Chinese TERN by leveraging the above three kinds of meta time information. Figure 1 demonstrates an example of this procedure, which consists of the following four main steps:

1) Chinese Time Token Identification: CXTime utilizes its token triples to identify time tokens from unstructured text and assign each time token with a <*token type*, *token value*> pair. In the meantime, CXTime also identifies the modifiers and numerals from unstructured text.

As illustrated in Figure 1, the raw sentence is decomposed into five time tokens: YEAR, MONTH, DATE, HOUR, and TIMEUNIT. For time tokens and numerals, the matching process further derives their standardized values (e.g., "4 月" (*April*) → <MONTH, 04>). Modifiers such as "大约" (*approximately*) are assigned with null values.

2) Time Segment Identification and MetaInfo Construction: CXTime identifies time segments from the identified time tokens and instantiates a MetaInfo structure to store meta time information for each time segment. This process consists of the following two sub-steps:

Time Segment Identification. Initially, each identified time token forms a time segment. If the time token corresponds to a self-contained timex, such as DURATION or PERIODICAL, then no further expansion is required. Otherwise, CXTime searches both sides of the time token to absorb compatible modifiers and numerals. During left-side expansion, COMMON, OPERATE_{Pre}, NUMBER, and ORDINAL are allowed to absorb. During right-side expansion, OPERATE_{Suf}, NUMBER, and ORDINAL are absorbed. Expansion on either side terminates when encountering a COMMA, LINKAGE, non-modifier/numeral token, or another time token. The expansion is constrained to not cross adjacent time tokens.

As illustrated in Figure 1, the example sentence is decomposed into five time segments: "2013 年" (*2013*), "4 月" (*April*), "15 日" (*the 15th day*), "8 点" (*8 o'clock*), and "大约一个月" (*approximately one month*).

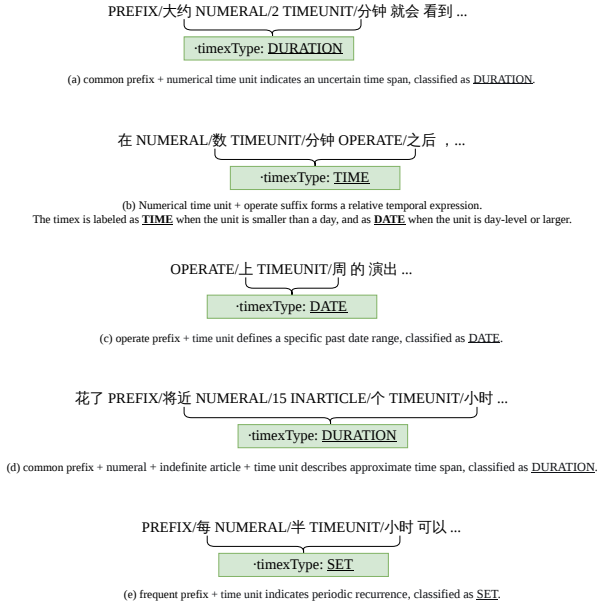


Fig. 3. Examples of timex type settings for time segments that contain TIMEUNIT modified by different prefixes and suffixes.

MetaInfo Construction. For each identified time segment, CXTime creates a MetaInfo instance to store its meta time information. The attributes defined in MetaInfo are similar to the ones defined in [34] for English timexes.

During initialization, CXTime stores the normalized values derived from <*token type*, *token value*> pairs into their corresponding MetaInfo attributes. For example, as shown in Figure 1, for the time segment "4 月" (*April*), which has a <MONTH, 04> pair, the value "04" is set to the attribute ·MONTH in metaInfo2. The attribute ·timexType of a MetaInfo instance is set based on the time token of its time segment using the mapping relations summarized in Table II.

Particularly, when a TIMEUNIT co-occurs with NUMERAL and QUANTITATIVE or approximation modifiers (e.g., "大约 2 分钟" (*about 2 minutes*) and "将近 15 个小时" (*nearly 15 hours*)) as shown in Figures 3(a) and 3(d)), the time segment denotes a bounded time span and is set to DURATION. If a numerical TIMEUNIT is followed by an OPERATE_{Suf} (e.g., "数分钟之后" (*several minutes later*) in Figure 3(b)), the time segment is treated as a relative timex: time segments involving sub-day TIMEUNIT are set to TIME, whereas those involving day-level TIMEUNIT are set to DATE. By contrast, time segments where an OPERATE_{Pre} directly modifies a TIMEUNIT (e.g., "上周" (*last week*) in Figure 3(c)) are interpreted as referring to a specific calendar interval and are therefore set to DATE. When a TIMEUNIT is modified by FREQUENT (e.g., "每半小时" (*every half hour*) in Figure 3(e)), the time segment denotes periodic recurrence and is set to SET.

3) Time Segment Merge and MetaInfo Merge: A timex may consist of multiple time tokens. CXTime performs a merging step to combine such time segments into a larger time segment and merge their corresponding MetaInfo



Fig. 4. Example time segments and timexes. The above labels are from time segment identification while the below labels are for timex extraction.

instances into a new MetaInfo instance. Therefore, this step includes the following two modules:

Time Segment Merge. A stand-alone time segment is treated as a timex (e.g., see Figure 4(a)). If two time segments are adjacent with no intervening tokens, CXTime directly merges them into a larger time segment (e.g., Figure 4(c)). If two time segments overlap at a shared boundary, the shared token is examined to determine whether to merge them. When the shared token is not a LINKAGE, the two time segments are merged (e.g., Figure 4(b)), as the overlap typically arises from shared modifiers or numerals introduced during time segment expansion. If the shared token is a LINKAGE, such as a range connector, CXTime treats the preceding segment as a timex candidate (e.g., Figure 4(d)) and continues scanning from the following time segment.

MetaInfo Merge. When time segments are merged, their corresponding MetaInfo instances are also merged into a new MetaInfo instance. In this merging process, CXTime first determines the ·timexType of the new MetaInfo instance by selecting the higher-priority timex type from the two MetaInfo instances to be merged.

For attributes used for timex values, Observation 7 indicates that different MetaInfo instances within a single timex typically contribute complementary attributes rather than conflict ones. As a result, all these attributes from individual MetaInfo instances are preserved in the new MetaInfo instance. For example, as shown in Figure 1, when metaInfo7 and metaInfo4 are merged into metaInfo8, all their attributes are preserved in metaInfo8.

4) Timex Extraction and Normalization: CXTime extracts a Chinese timex from each final time segment and assigns it a timex type and a normalized value based on the information stored in the final MetaInfo instance.

Timex Type Determination. The type of a timex is directly determined by the "·timexType" attribute of the final MetaInfo instance. For example, as shown in Figure 1, the timex type of "大约一个月" (*approximately one month*) is set by the one of metaInfo5, i.e., DURATION.

Timex Value Normalization. CXTime generates the normalized value of a timex from its final MetaInfo instance via *value format selection* and *value instantiation*.

In the first module, CXTime determines the appropriate value format for a timex. Based on the timex type, CXTime selects the corresponding candidate formats and then determines the final format by examining which attributes are present in the final MetaInfo instance. Only formats whose required attribute compositions are fully satisfied are considered valid. Continuing the example in Figure 1, CXTime produces two final MetaInfo instances corresponding to two timexes. For the timex "大约一个月" (*approximately one month*), the final MetaInfo instance is assigned with DURATION. CXTime selects duration-level candidate formats and determines the final format based on the presence of the time unit attribute together with its numerical value. As a result, the format PnM" is selected.

In the second module, CXTime instantiates the selected value format by filling in its slots using the "*attribute*:*value*" pairs stored in the MetaInfo instance. In the running example in Figure 1, for the DURATION timex, the numerical value "1" is instantiated into the selected duration format, producing the normalized value "P1M".

V. Experiments

A. Experimental Setup

1) Datasets: The **WBPost** dataset is our annotated dataset that is derived from a large-scale corpus called CED [22]. Specifically, we collect all the raw Weibo posts from the CED dataset and employ the PKUSEG toolkit [18] to segment these posts. For the segmented Weibo posts, we train two students with the knowledge about TimeML specification and TE2C annotation, and the two students work as independent annotators to manually label timex spans, timex types, and normalization values. After the two annotators' labeling, a third annotator who is familiar with TERN serves as an adjudicator to resolve disagreements and finalize the timex annotations.

After annotation, we randomly split WBPost into training, validation, and test sets at a ratio of 7 : 1 : 2. All experimental results are reported on the test set.

The statistics of the four datasets is reported in Table I.

2) Baseline Models: We compare CXTime to six representative state-of-the-art baseline methods spanning multiple paradigms for Chinese timex processing, including rule-based methods, hybrid approaches, neural sequence labeling models, and LLM-based methods: HeidelTime [15], PTime [9], ARTime [8], LXD2021 [17], BiLSTM-CRF[4], LANGE2023 [13], and GLS2024[5] [10].

[4]Its code is available at https://github.com/zjy-ucas/ChineseNER.

[5]GLS2024 has three variants: (1) **GLS2024**$_{E2E}$, (2) **GLS2024**$_{Doc}$, and (3) **GLS2024**$_{Sen}$. We use GLS2024$_{E2E}$ to report Chinese TERN performance while use GLS2024$_{Doc}$ and GLS2024$_{Sen}$ to report Chinese timex normalization performances. All configurations are tested on the DeepSeek model with a deterministic setting (temperature = 0). A system/user dual-prompt structure is adopted, and the maximum output length is set to 4096 tokens.

TABLE IV
OVERALL PERFORMANCE OF CXTIME AND BASELINES ON TE3C-ORIGINAL, TE3C-CLEAN, TE3C-IMPROVED, AND WBPOST. THE BEST RESULTS ARE HIGHLIGHTED IN BOLD AND THE SECOND BEST ARE UNDERLINED. RESULTS ARE OBTAINED EITHER FROM OUR RE-IMPLEMENTATION OR BY RUNNING PUBLICLY AVAILABLE IMPLEMENTATIONS UNDER A UNIFIED EVALUATION PROTOCOL.

Dataset	Method	Timex Recognition						Normalization	
		Strict Match			Relaxed Match			Type	Value
		Pre.	*Rec.*	F_1	*Pre.*	*Rec.*	F_1	F_1	F_1
TE2C-Original	HeidelTime	85.79	85.34	85.56	92.63	92.15	92.39	86.09	63.52
	PTime	71.07	59.16	64.57	89.94	74.87	81.71	-	-
	LXD2021	83.70	80.63	82.13	90.22	86.91	88.53	84.80	-
	BiLSTM-CRF	84.97	76.96	80.77	92.49	83.77	87.91	75.82	-
	LANGE2023	75.17	57.07	64.88	**96.55**	73.30	83.33	76.79	35.12
	GLS2024$_{E2E}$	66.34	70.16	68.19	82.67	87.43	84.99	81.42	53.94
	CXTime	**90.40**	**93.72**	**92.03**	92.93	**96.34**	**94.60**	**92.54**	**68.38**
TE2C-Clean	HeidelTime	**70.53**	93.71	80.48	72.11	95.8	82.28	79.28	72.67
	PTime	67.86	66.43	67.14	**77.86**	76.22	77.03	-	-
	LXD2021	65.22	83.92	73.39	67.39	86.71	75.84	73.39	-
	BiLSTM-CRF	58.64	66.43	62.30	72.22	81.82	76.72	60.98	-
	LANGE2023	53.79	54.55	54.17	68.28	69.23	68.75	63.89	40.97
	GLS2024$_{E2E}$	59.26	78.32	67.47	63.49	83.92	72.29	71.08	60.84
	CXTime	70.20	**97.20**	**81.52**	71.72	**99.30**	**83.28**	**82.70**	**77.42**
TE2C-Improved	HeidelTime	88.95	87.56	88.25	95.79	94.30	95.04	91.38	82.51
	PTime	76.71	58.03	66.08	**97.26**	73.58	83.78	-	-
	LXD2021	83.70	79.79	81.70	90.76	86.53	88.59	84.35	-
	BiLSTM-CRF	80.90	74.61	77.63	94.94	87.56	91.11	73.32	-
	LANGE2023	75.86	56.99	65.09	97.24	73.06	83.43	78.70	50.30
	GLS2024$_{E2E}$	71.74	68.39	70.03	89.13	84.97	87.00	84.88	69.50
	CXTime	**90.91**	**93.26**	**92.07**	94.44	**96.89**	**95.65**	**94.12**	**89.51**
WBPost	HeidelTime	81.75	49.12	61.37	94.53	56.80	70.96	62.60	53.01
	PTime	81.46	66.01	72.93	95.94	77.74	85.89	-	-
	LXD2021	76.69	50.88	61.17	90.08	59.76	71.85	65.92	-
	BiLSTM-CRF	90.98	85.20	88.00	96.72	90.57	93.54	85.73	-
	LANGE2023	80.61	63.38	70.96	96.23	75.66	84.71	77.96	55.99
	GLS2024$_{E2E}$	67.94	79.71	73.36	77.01	90.35	83.15	76.69	69.63
	CXTime	**91.24**	**89.04**	**90.12**	**97.75**	**95.39**	**96.56**	**93.78**	**87.13**

3) Evaluation Metrics: We adopt the TempEval-3 evaluation script [24] to report system performance.

Pure Normalization Evaluation In pure normalization setting, texts with gold-standard timex spans are provided as input and only type classification and value normalization are evaluated. Following [3], the accuracy of type classification is defined as the proportion of timexes whose types are correctly classified; similarly, the accuracy of type normalization is the proportion of timexes whose values are correctly normalized. Under this setting, *Acc.* is mathematically equivalent to both *Pre.* and *Rec.*, and equals to the value of F_1 [34]. Therefore, we report F_1 for pure normalization experiments.

B. Experimental Results

Table IV reports the overall performance of CXTime and baselines in Chinese TERN on the four datasets. It shows that CXTime achieves 28 best and 2 second-best results out of 32 measures, significantly outperforming all compared baselines in both timex recognition and timex normalization across datasets. In timex recognition, CXTime obtains all highest *Rec.* and F_1 under both strict and relaxed matches and outperforms all baselines by margins of 2.59∼8.38 points in *Rec.* and 0.61∼6.47 points in F_1 across the four datasets.[6] This indicates that (1) the set of token triples used in CXTime cover most of tokens in Chinese timexes across diverse datasets and verifies Observation 3 that Chinese timexes are composed of a small set of time-related keywords; and (2) the general heuristic rules designed in CXTime successfully recognizes most of Chinese timexes from free text.

[6]For each evaluation measure, the margin between CXTime and the baselines is calculated as the performance difference between CXTime and the strongest baseline.

In type classification, CXTime achieves all highest F_1 and outperforms all baselines by margins of 2.74∼8.05 points. This verifies Observation 5 that strong mapping relations exist between time tokens and timex types and verifies Observation 6 that a general priority relationship exists among the four timex types. For value normalization, CXTime consistently attains the highest F_1 on all datasets, exceeding all baselines by 4.75∼17.50 points.

a) CXTime vs. Rule-based Methods: As shown in Figure IV, CXTime outperforms HeidelTime in both timex recognition and normalization. While CXTime performs slightly worse than HeidelTime in *Pre.* on some datasets (e.g., relaxed *Pre.* on TE2C-Clean and TE2C-Improved), it achieves substantially higher *Rec.* and F_1. Moreover, CXTime attains higher F_1 in both type classification and value normalization across all four datasets.

Particularly, for timex recognition, CXTime consistently surpasses HeidelTime by 3.49∼39.92 points in strict *Rec.* and 1.04∼28.75 points in strict F_1. For timex normalization, similarly, CXTime outperforms HeidelTime in terms of F_1 by 2.74∼31.18 points in type classification and 4.75∼34.12 points in value normalization across datasets.

A plausible reason for these results is that HeidelTime suffers from its deterministic rules that could not cover flexible structures of Chinese timexes; by contrast, CXTime addresses this limitation through token-level abstraction using token triples and general heuristic rules.

b) CXTime vs. Learning-based Methods: CXTime significantly and consistently outperforms the three learning-based methods (i.e., LXD2021, BiLSTM-CRF, and PTime) in both timex recognition and normalization across all four datasets. Particularly, for timex recognition, CXTime substantially surpasses learning-based baselines by 3.84∼13.47 points in *Rec.* and 2.12∼10.37 points in F_1. For timex normalization, CXTime significantly outperforms learning-based baselines on type classification by 7.74∼9.77 points in F_1 across datasets.

These results are attributed to the fact that learning-based methods often struggle with boundary detection and token-level coverage, particularly for irregular and colloquial Chinese timexes. Their performance heavily depends on training data, which limits generalization and their predictions lack interpretability. By contrast, CXTime achieves much higher *Rec.*, resulting in consistently superior F_1 in both timex recognition and timex normalization. These results also highlight the advantage of modeling timex value patterns over purely learning-based approaches for timex normalization.

c) CXTime vs. Transformer-based and LLM-based Methods: Figures IV shows that CXTime generally outperforms the two transformer- and LLM-based methods (i.e., LANGE2023 and GLS2024$_{E2E}$) in both TERN subtasks across datasets.

Particularly, for timex recognition, CXTime achieves *Rec.* improvements of 5.04∼24.87 points and F_1 improvements of 8.65∼23.84 points compared to LANGE2023 and

TABLE V
PERFORMANCE OF PURE TIMEX NORMALIZATION, IN WHICH ALL CHINESE TIMEXES ARE ASSUMED TO BE PERFECTLY RECOGNIZED WITH 100% *PRECISION*, *RECALL*, AND F_1.

Method	TE2C-Original		TE2C-Clean		TE2C-Improved		WBPost	
	Type	Value	Type	Value	Type	Value	Type	Value
ARTime	85.97	**79.40**	80.83	71.67	71.29	62.71	82.91	73.01
LANGE2023	-	40.84	-	57.34	-	51.30	-	51.10
$GLS2024_{Doc}$	-	68.59	-	81.82	-	75.65	-	83.66
$GLS2024_{Sen}$	-	68.25	-	84.21	-	78.01	-	85.12
CXTime	**93.72**	68.59	**98.60**	**90.91**	**95.83**	**90.62**	**93.91**	**85.94**

$GLS2024_{E2E}$ across datasets. For timex normalization, similarly, CXTime achieves F_1 gains of 9.24∼15.82 points in type classification and 14.44∼20.01 points in value normalization.

A possible reason for these results is that transformer- and LLM-based methods effectively leverage contextual information but have inherent limitations in Chinese TERN. Masked language-model-based methods predict tokens based on context, making them prone to boundary detection errors in noisy text; they also lack explicit constraints for normalized values and often produce non-standard outputs for rare or complex Chinese timexes. Prompt-based methods offer more flexible outputs but are sensitive to prompt design and the quality of few-shot examples. By contrast, CXTime explicitly models token-level meta time information (Observations 3, 5, 6, and 7), which ensures stable recognition and accurate normalization. By exploiting these meta time information, CXTime achieves higher *Rec.* span detection and consistent type classification and value normalization across datasets, while reducing reliance on large-scale training data or carefully engineered prompts.

1) Performance in Pure Timex Normalization: We further examine the performance of CXTime in **pure timex normalization**. The results are reported in Table V.

Except for value normalization on TE2C-Original, CXTime significantly outperforms the four baselines in both sub-tasks across all four datasets. For type classification, CXTime achieves F_1 improvements of 7.75∼24.54 points across datasets. This further verifies Observations 5 and 6 that strong mapping relations between time tokens and timex types and priority relationship among timex types are particularly effective for type classification. For value normalization, CXTime achieves best performance on TE2C-Clean, TE2C-Improved, and WBPost, with F_1 gains of 0.82∼12.61 points. CXTime's inferior performance compared to ARTime on TE2C-Original is likely attributable to annotation noise, as timex value normalization is highly sensitive to annotation quality and TE2C-Original contains noisy annotations.

2) Ablation Study: Token triples and mapping relations between time tokens to timex types are necessary in CXTime, therefore, we analyze the impact of the priority relationship among timex types as well as the POS tags.

a) Impact of Priority among Timex Types: To evaluate the impact of the priority among timex types, we remove the priority component by assigning the ·timexType of the first MetaInfo instance as the final timex type. These results are denoted by "w/o Priority" in Table VI. It shows that removing the priority component leads to substantial performance degradation in both type classification and value normalization across datasets. These results justify the importance of the priority component in CXTime for both sub-tasks of timex normalization.

TABLE VI
IMPACT OF THE PRIORITY RELATIONSHIP AMONG TIMEX TYPES AND POS TAGS IN CXTIME.

Method	TE2C-Original		TE2C-Clean		TE2C-Improved		WBPost	
	Type	Value	Type	Value	Type	Value	Type	Value
CXTime	92.54	68.38	82.70	77.42	94.12	89.51	93.78	**87.13**
w/o Priority	82.26	63.24	75.66	72.14	85.42	81.33	77.80	72.70
w/o POS	82.03	59.91	72.02	66.84	83.94	77.98	89.34	82.86

b) Impact of POS Tags: To evaluate the impact of POS information, we remove POS tags from CXTime and perform timex normalization without leveraging POS cues. These results are denoted by "w/o POS" in Table VI. It shows that excluding POS tags consistently degrades CXTime's performance in both type classification and value normalization across datasets. These results highlight the critical role of POS tags in CXTime for both type classification and value normalization.

3) Error Analysis: There are mainly three types of errors in CXTime's evaluation:

Annotation Errors: A portion of the errors originates from annotation noise in the datasets. In TE2C-Original, TE2C-Clean, and TE2C-Improved, some documents contain incomplete document creation times (DCTs), such as "1998-00-00", where only the year information is reliable while the month and day fields are unspecified.

Recognition Errors: CXTime may produce errors in the timex recognition stage, particularly in determining precise timex boundaries and coverage of timex patterns.

Reference Date Inference Errors: Evaluation errors also arise from reference date inference, particularly for relative timexes that depend on implicit contextual anchors. Although CXTime leverages reference time information, it may still fail to accurately infer the correct reference time from context in certain cases.

VI. CONCLUSION

We construct a large-scale annotated dataset for TERN research, and summarize several characteristics about structures and constituents of Chinese timexes. Motivated by these characteristics, we propose a general heuristic rule-based method to recognize and normalize Chinese timexes from unstructured text. Experiments on diverse datasets demonstrate that CXTime significantly outperforms representative state-of-the-art models in Chinese TERN. More importantly, CXTime is independent of specific tokens, domains, and text genres; it is lightweight and extensible, which makes it suitable for practical use.

References

[1] G. Angeli, C. Manning, and D. Jurafsky, "Parsing time: Learning to interpret time expressions," in *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Jun. 2012, pp. 446–455.

[2] G. Angeli and J. Uszkoreit, "Language-independent discriminative parsing of temporal expressions," in *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Aug. 2013, pp. 83–92.

[3] S. Bethard, "A synchronous context free grammar for time normalization," in *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, Oct. 2013, pp. 821–826.

[4] Y. Cao, W. Groves, T. K. Saha, J. Tetreault, A. Jaimes, H. Peng, and P. Yu, "XLTime: A cross-lingual knowledge transfer framework for temporal expression extraction," in *Findings of the Association for Computational Linguistics: NAACL 2022*, Jul. 2022, pp. 1931–1942.

[5] A. X. Chang and C. D. Manning, "Sutime: A library for recognizing and normalizing time expressions," in *Proceedings of 8th International Conference on Language Resources and Evaluation*, 2012, pp. 3735–3740.

[6] X. Chen, M. Li, S. Gao, Z. Chan, D. Zhao, X. Gao, X. Zhang, and R. Yan, "Follow the timeline! generating an abstractive and extractive timeline summary in chronological order," *ACM Trans. Inf. Syst.*, vol. 41, no. 1, Jan. 2023.

[7] Z. Chu, J. Chen, Q. Chen, W. Yu, H. Wang, M. Liu, and B. Qin, "TimeBench: A comprehensive evaluation of temporal reasoning abilities in large language models," in *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Aug. 2024, pp. 1204–1228.

[8] W. Ding, J. Chen, J. Li, and Y. Qu, "Automatic rule generation for time expression normalization," in *Findings of the Association for Computational Linguistics: EMNLP 2021*, Nov. 2021, pp. 3135–3144.

[9] W. Ding, G. Gao, L. Shi, and Y. Qu, "A pattern-based approach to recognizing time expressions," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, 2019, pp. 6335–6342.

[10] A. Gautam, L. Lange, and J. Strötgen, "Discourse-aware in-context learning for temporal expression normalization," in *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, Jun. 2024, pp. 306–315.

[11] K. Hacioglu, Y. Chen, and B. Douglas, "Automatic time expression labeling for english and chinese text," in *Proceedings of the 6th International Conference on Intelligent Text Processing and Computational Linguistics*, 2005, pp. 548–559.

[12] T. Hao, X. Pan, Z. Gu, Y. Qu, and H. Weng, "A pattern learning-based method for temporal expression extraction and normalization from multi-lingual heterogeneous clinical texts," *BMC medical informatics and decision making*, vol. 18, no. Suppl 1, p. 22, 2018.

[13] L. Lange, J. Strötgen, H. Adel, and D. Klakow, "Multilingual normalization of temporal expressions with masked language models," in *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, May 2023, pp. 1174–1186.

[14] K. Lee, Y. Artzi, J. Dodge, and L. Zettlemoyer, "Context-dependent semantic parsing for time expressions," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Jun. 2014, pp. 1437–1447.

[15] H. Li, J. Strötgen, J. Zell, and M. Gertz, "Chinese temporal tagging with HeidelTime," in *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, volume 2: Short Papers*, Apr. 2014, pp. 133–137.

[16] W. Li, K.-F. Wong, and C. Yuan, "Toward automatic chinese temporal information extraction," *Journal of the American Society for Information Science and Technology*, vol. 52, no. 9, pp. 748–762, 2001.

[17] X. Liu, Z. Che, and W. Cui, "A multi-level hybrid method for dynamic extraction and normalization of chinese temporal expressions," in *2021 IEEE International Conference on Power, Intelligent Computing and Systems (ICPICS)*, 2021, pp. 498–503.

[18] R. Luo, J. Xu, Y. Zhang, Z. Zhang, X. Ren, and X. Sun, "Pkuseg: A toolkit for multi-domain chinese word segmentation." *CoRR*, vol. abs/1906.11455, 2019.

[19] K. Ma, Y. Tan, M. Tian, X. Xie, Q. Qiu, S. Li, and X. Wang, "Extraction of temporal information from social media messages using the bert model," *Earth Science Informatics*, vol. 15, no. 1, pp. 573–584, 2022.

[20] X. Pan, B. Chen, H. Weng, Y. Gong, and Y. Qu, "Temporal expression classification and normalization from chinese narrative clinical texts: Pattern learning approach," *JMIR Medical Informatics*, vol. 8, no. 7, 2020.

[21] A. Sánchez de Castro, L. Araujo, and J. Martinez-Romo, "A novel methodology for enhancing cross-language and domain adaptability in temporal expression normalization," *Computational Linguistics*, pp. 1–33, 06 2025.

[22] C. Song, C. Tu, C. Yang, Z. Liu, and M. Sun, "Ced: Credible early detection of social media rumors," *arXiv preprint arXiv:1811.04175*, 2018.

[23] J. Strötgen and M. Gertz, "Heideltime: High quality rule-based extraction and normalization of temporal expressions," in *Proceedings of the 5th International Workshop on Semantic Evaluation (SemEval'10)*. Stroudsburg, PA, USA: Association for Computational Linguistics, 2010, pp. 321–324.

[24] N. UzZaman, H. Llorens, L. Derczynski, M. Verhagen, J. Allen, and J. Pustejovsky, "Semeval-2013 task 1: Tempeval-3: Evaluating time expressions, events, and temporal relations," in *Proceedings of the 7th International Workshop on Semantic Evaluation*, 2013, pp. 1–9.

[25] M. Verhagen, R. Saurí, T. Caselli, and J. Pustejovsky, "SemEval-2010 task 13: TempEval-2," in *Proceedings of the 5th International Workshop on Semantic Evaluation*, Jul. 2010, pp. 57–62.

[26] M. Wu, W. Li, Q. Lu, and B. Li, "CTEMP: A Chinese temporal parser for extracting and normalizing temporal information," in *Second International Joint Conference on Natural Language Processing: Full Papers*, 2005.

[27] H. Xu and C.-R. Huang, "A rule system for Chinese time entity recognition by comprehensive linguistic study," in *Proceedings of the Sixth International Joint Conference on Natural Language Processing*, Oct. 2013, pp. 795–801.

[28] Z. Ya-yao, L. Yao-fa, and L. Jie-jun, "Recognition and normalization of chinese time expressions based on rules." *Journal of Guangdong University of Technology*, vol. 31, no. 3, 2014.

[29] B. Yin and B. Jin, "A multi-label classification method on chinese temporal expressions based on character embedding," in *Proceedings of the 4th International Confernce on Information Science and Control Engineering*, 2017, pp. 51–54.

[30] H. Zhang, L. Liu, S. Cheng, and W. Shi, "Distant supervision for chinese temporal tagging," in *Knowledge Graph and Semantic Computing. Knowledge Computing and Language Understanding*, 2019, pp. 14–27.

[31] X. Zhong and E. Cambria, "Time expression recognition using a constituent-based tagging scheme," in *Proceedings of the 2018 World Wide Web Conference*, Lyon, France, 2018, pp. 983–992.

[32] ——, *Time expression and named entity recognition*. Springer, 2021, vol. 10.

[33] ——, "Time expression recognition and normalization: A survey," *Artificial Intelligence Review*, vol. 56, no. 9, pp. 9115–9140, 2023.

[34] X. Zhong, C. Jin, M. An, and E. Cambria, "Xtime: A general rule-based method for time expression recognition and normalization," *Knowledge-Based Systems*, vol. 297, p. 111921, 2024.

[35] X. Zhong, A. Sun, and E. Cambria, "Time expression analysis and recognition using syntactic token types and general heuristic rules," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Jul. 2017, pp. 420–429.