Cognitive Computation (2020) 12:844–862
https://doi.org/10.1007/s12559-020-09714-8

Extracting Time Expressions and Named Entities with Constituent-Based Tagging Schemes

Xiaoshi Zhong[1] · Erik Cambria[1] · Amir Hussain[2]

Received: 21 August 2019 / Accepted: 21 January 2020 / Published online: 10 May 2020

Abstract

Time expressions and named entities play important roles in data mining, information retrieval, and natural language processing. However, the conventional position-based tagging schemes (e.g., the BIO and BILOU schemes) that previous research used to model time expressions and named entities suffer from the problem of *inconsistent tag assignment*. To overcome the problem of inconsistent tag assignment, we designed a new type of tagging schemes to model time expressions and named entities based on their constituents. Specifically, to model time expressions, we defined a constituent-based tagging scheme termed TOMN scheme with four tags, namely T, O, M, and N, indicating the defined constituents of time expressions, namely *time token*, *modifier*, *numeral*, and the words *outside* time expressions. To model named entities, we defined a constituent-based tagging scheme termed UGTO scheme with four tags, namely U, G, T, and O, indicating the defined constituents of named entities, namely *uncommon word*, *general modifier*, *trigger word*, and the words *outside* named entities. In modeling, our TOMN and UGTO schemes model time expressions and named entities under conditional random fields with minimal features according to an in-depth analysis for the characteristics of time expressions and named entities. Experiments on diverse datasets demonstrate that our proposed methods perform equally with or more effectively than representative state-of-the-art methods on both time expression extraction and named entity extraction.

Keywords Inconsistent tag assignment · Position-based tagging scheme · Constituent-based tagging scheme · Named entities · Time expressions · Intrinsic characteristics

Introduction

Time expressions and named entities play increasingly important roles in the fields of data mining, information retrieval, and natural language processing [51, 66, 80]. There are many linguistic tasks that are involved in time expressions and named entities, such as temporal relation extraction [8, 45], timeline construction [16, 33], temporal information retrieval [2, 7], named entity recognition [11, 66], named entity typing [22, 37, 44], entity linking [27, 36], and domain-specific entity recognition [29, 73].

Researchers from various fields have devoted tremendous effort to specify standards for the annotation of time expressions [18, 59, 61] and named entities [12, 66], build annotated corpora for time expressions [48, 60] and named entities [57, 66], and recognize time expressions and named entities from unstructured text [66, 75, 76, 78].

To better understand the intrinsic characteristics of time expressions and named entities, we analyzed four diverse datasets about time expressions and two benchmark datasets about named entities. According to these characteristics, we proposed two methods to extract time expressions and named entities from unstructured text.

Analysis and Extraction of Time Expressions

The four datasets we used to analyze time expressions include TimeBank [60], Gigaword [54], WikiWars [48], and Tweets [82]. From our analysis, we found two characteristics about their organization and constituent words. Firstly, time expressions are formed by loose structure; more than 53.5% of unique time tokens appearing in different positions

This paper is an extension of the following conference paper: Xiaoshi Zhong and Erik Cambria. 2018. Time Expression Recognition Using a Constituent-based Tagging Scheme. In *Proceedings of the 2018 World Wide Web Conference* , Association for Computing Machinery, Lyon, France, pages 983–992.

✉ Xiaoshi Zhong
xszhong@ntu.edu.sg

Extended author information available on the last page of the article.

Fig. 1 Tag assignment of the BILOU and TOMN schemes. The BILOU scheme is based on the positions within a labeled chunk, while the TOMN scheme is based on the constituents of a labeled chunk. Here, inconsistent tag assignment is defined as that during the training stage, a word is assigned with different tags simply because this word appears in different positions within labeled chunks

1) September/U 2) September/B 2006/L

3) 2006/B September/L 4) 1/B September/I 2006/L

(a) Tag assignment of BILOU scheme: "September" in different positions within labeled time expressions is assigned with different tags of U, B, L, or I. The inconsistent tag assignment reduces the predictive power of "September," and this contradicts that characteristic that time tokens can distinguish time expressions from common text.

1) September/T 2) September/T 2006/T

3) 2006/T September/T 4) 1/N September/T 2006/T

(b) Tag assignment of TOMN scheme: "September" in different positions within labeled time expressions is consistently assigned with the same tag of T. The consistent tag assignment protects the predictive power of "September."

within time expressions. Secondly, time tokens can distinguish time expressions from common text; more than 91.8% of time expressions contain at least one time token while no more than 0.7% of common text contain time tokens.

These two characteristics motivated us to design a learning-based method termed TOMN to model time expressions. Specifically, TOMN defines a constituent-based tagging scheme termed TOMN scheme that consists of four tags, namely T, O, M, and N, indicating the constituents of time expressions, namely *time token*, *modifier*, *numeral*, and the words *outside* of time expressions. Time tokens include those words that explicitly express information about time, such as "month" and "September." Modifiers include those words that modify time tokens and appear around them; for example, "last" modifies "month" in "last month." Numerals include ordinals and numbers. TOMN models time expressions under conditional random fields (CRFs) [30] with only a kind of pre-tag features and the lemma features. In modeling, a word is assigned with one of the four TOMN tags.

TOMN scheme can keep the tag assignment consistent during training and therefore overcomes the problem of inconsistent tag assignment.[1] The loose structure by which time expressions are formed exhibits in two aspects. Firstly, many time expressions consist of loose collocations. Secondly, some time expressions can change their word order without changing their meanings. The conventional tagging schemes like BILOU [63] are based on *the positions within a labeled chunk*. Under the BILOU scheme, a word that appears in different positions within labeled time expressions is assigned with different tags. For example, the time token "September" in the four time expressions shown in Fig. 1a can be assigned with U, B, L, or I. The inconsistent tag assignment causes difficulty for statistical models to model time expressions. Firstly, inconsistent tag assignment reduces the predictive power of lexicons, and this contradicts the characteristic that time tokens can distinguish time expressions from common text. Secondly, inconsistent tag assignment might cause the problem of tag imbalance. Our TOMN scheme instead is based on *the constituents of a labeled chunk* and assigns the same constituent word with the same tag, regardless of its frequency and its positions within time expressions. Under TOMN scheme, for example, the above time token "September" in the four time expressions is consistently assigned with T (see Fig. 1b). With consistent tag assignment, TOMN scheme protects the predictive power of time tokens and avoids the potential tag imbalance.

We evaluate TOMN against five state-of-the-art methods on three datasets. Experimental results demonstrate that TOMN performs equally or more effectively than these state-of-the-art methods, and much more robust on cross-dataset performance compared with those learning-based baselines. Experimental results also demonstrate the advantage of our constituent-based TOMN scheme over the conventional position-based tagging schemes (see "Time Expression Extraction" for details).

Analysis and Extraction of Named Entities

The two benchmark datasets we used to analyze named entities are CoNLL2003 [66] and OntoNotes*, which is a derived version of OntoNotes5 corpus [57, 58]. From our analysis we found two common characteristics about named entities. Firstly, 92.2% of named entities contain *uncommon words*, which include those words that mainly appear in named entities while hardly appear in common text. Secondly, named entities are mainly made up of proper nouns; in the whole text, more than 84.8% of proper nouns appear in named entities, and within named entities, more than 80.1% of words are proper nouns.

These two characteristics motivated us to design a CRFs-based learning method termed UGTO to extract named entities from unstructured text. UGTO defines a constituent-based tagging scheme termed UGTO scheme that consists of

[1] In a supervised-learning procedure, tag assignment occurs in two stages: (1) feature extraction in the training stage and (2) tag prediction in the testing stage. We focus on the training stage to analyze the impact of tag assignment.

four tags: U, G, T, and O. U encodes the *uncommon words*, such as "Boston" and "Africans." G encodes the *generic modifiers* while T encodes the *trigger words*. Generic modifiers (e.g., "of" and "and") can appear in several types of named entities while trigger words appear in a specific type of named entities; for example, the trigger word "University" appears in "Boston University." O encodes those words Outside named entities. In modeling, UGTO assigns one word with one of the UGTO tags under a CRFs framework with only the UGTO pre-tag features and some basic features.

We evaluate UGTO on two benchmark datasets against two representative state-of-the-art baselines. Experimental results demonstrate the effectiveness and efficiency of UGTO in comparison with the two baselines. Experimental results also demonstrate that traditional methods with simple handcrafted features can achieve state-of-the-art performance on named entity extraction, compared with a state-of-the-art neural-network-based method, and that joint modeling named entity extraction and classification does not improve the performance of named entity extraction, in both our model and the baselines (see "Named Entity Extraction" for details).

Contributions

To summarize, we made in this paper the following contributions.

- We summarized from four diverse datasets two common characteristics about time expressions, and summarized from two benchmark datasets two common characteristics about named entities.
- We identified a fundamental problem underlying in the conventional position-based tagging schemes: inconsistent tag assignment. To overcome that problem, we defined a new types of constituent-based tagging schemes to model time expressions and named entities.
- We conducted extensive experiments on diverse datasets, and the experimental results demonstrate the effectiveness and efficiency of our proposed methods against state-of-the-art baselines on the extractions of time expressions and named entities. Experimental results also demonstrate that joint modeling named entity extraction and classification does not improve the performance of named entity extraction, in both our model and baselines.

Related Works

The works that are related to our work include the tasks of time expression extraction and normalization as well as named entity extraction and classification.

Time Expression Extraction and Normalization

Time expression identification is an information-extraction task whose goal is to automatically identify time expressions from unstructured text and it can be divided into two subtasks: time expression extraction and time expression normalization. Methods that were developed for time expression extraction can be classified into two kinds: rule-based methods and learning-based methods.

Time Expression Extraction

Rule-Based Methods Rule-based methods like TempEx, GUTime, HeidelTime, and SUTime mainly handcrafted deterministic rules to identify time expressions. TempEx and GUTime used both handcrafted rules and machine-learnt rules to resolve time expressions [46, 77]. HeidelTime manually designed rules with time resources to recognize time expressions [71]. SUTime designed deterministic rules at three levels (i.e., individual word, chunk, and time expression levels) for time expression extraction [9]. SynTime designed general heuristic rules with a token type system to recognize time expressions [82].

TOMN uses token regular expressions, similar to SUTime [9] and SynTime [82], and further groups them into three token types, similar to SynTime. While SynTime further defines 21 token types for the constituent words of time expressions, TOMN uses those three general token types that are helpful for a learning method to connect those words with low frequencies to those words with high frequencies. Moreover, TOMN leverages statistical information from an entire corpus to improve the precision of the extraction and alleviate the deterministic role of deterministic rules and heuristic rules.

Learning-Based Methods Learning-based methods in the TempEval series mainly derived features from text (e.g., character features, word features, syntactic features, and semantic features), and applied statistical models (e.g., CRFs, logistic regression, maximum entropy, Markov logic network, and support vector machines) to model time expressions [5, 19, 41, 74]. Besides the standard methods, Angeli et al. [3] and Angeli and Uszkoreit [4] exploited an EM-style approach with compositional grammar to learn a latent time parser. Lee et al. [32] leveraged combinatory categorial grammar (CCG) [70] and defined a collection of lexicon with linguistic context to model time expressions.

Unlike [5, 19, 24, 41, 74] which used standard features, TOMN derives features according to the characteristics of time expressions and uses only a kind of pre-tag features and the lemma features. Such strategy can enhance the impact of those significant features and reduce the impact of those insignificant features. Unlike [3, 4, 32] which used fixed structure information, TOMN uses the loose structure information by grouping the constituent words of time expressions under three token types; this strategy can fully account for the loose structure of time expressions. More importantly, TOMN models time expressions under a CRFs framework with a constituent-based tagging scheme, which can keep the tag assignment consistent.

Time Expression Normalization

The methods that were developed for time expression normalization are mainly based on rules [5, 19, 41, 71, 74, 77]. Since those rule methods are highly similar, Llorens et al. [40] suggested to construct a large knowledge base for public use. Angeli et al. [3], Angeli et al. [4], and Lee et al. [32] combined grammar rules and machine learning techniques to normalize time expressions. TOMN focuses on the extraction and leaves the normalization to those highly similar rule methods.

Named Entity Extraction and Classification

Named Entity Recognition

The research of named entity recognition has a long history. Nadeau and Sekine review the development of the early years [51] in terms of languages (e.g., English and Chinese) [23, 66, 79], text genres and domains (e.g., scientific and journalistic) [47, 56], statistical learning techniques (e.g., CRFs and maximum entropy models) [6, 49], engineering features (e.g., lexical features and dictionary features) [13, 69], and shared task evaluations (e.g., MUC, CoNLL, and ACE) [11, 17, 23, 66].

Before the deep learning era, there were also works that concern several aspects of NER, like leveraging unlabeled data for NER [34], leveraging external knowledge for NER [28, 63], nested NER [1, 21], and NER in informal text [39, 65].

In the deep learning era, many researchers use neural networks and word embeddings to develop variants of models on CoNLL03 dataset [14, 15, 26, 31, 35, 38, 42, 43, 55, 62, 68, 72].

UGTO benefits from the traditional methods by utilizing some of their basic features (e.g., lexical and POS features), and refines the significant features (i.e., UGTO pre-tag features) according to an in-depth analysis for the characteristics of named entities. Unlike neural network based methods that mainly compute the semantic similarities among words, UGTO focuses on the difference between named entities and the common text. Unlike most NER methods that jointly model entity extraction and classification, our analysis and experiments show that the joint NER task does not improve the performance of entity extraction but simply costs additional runtime, in both our model and representative models.

Named Entity Classification

The studies of named entity classification (also known as entity typing) assume that named entities are already extracted from text [22, 37, 44, 52, 64]. These studies leverage features like bag of words, POS tags, head words, and n-gram strings, many of which are similar to those derived for NER. A key difference between these studies and NER is that they do not formulate entity classification as a problem of sequence tagging but treat a whole named entity as a unit. We focused on entity extraction and leave entity classification to future work.

Data Analysis

In this section, we firstly reported two common characteristics of time expressions from analyzing four diverse datasets, and then reported two characteristics about named entities from two benchmark datasets.

Time Expression Analysis

Datasets

The four datasets we used to analyze time expressions include TimeBank [60], Gigaword [54], WikiWars [48], and Tweets [82]. TimeBank is a benchmark dataset with 183 news articles. Gigaword consists of 2,452 news articles with automatically annotated time expressions. WikiWars is constructed by collecting articles from Wikipedia about famous wars. Tweets consists of 942 tweets collected from Twitter. The four datasets cover comprehensive data (TimeBank, Gigaword, and Tweets) and domain-specific data (WikiWars) as well as formal text (TimeBank, Gigaword, and WikiWars) and informal text (Tweets). Table 1 summarizes the statistics of these four datasets.

Table 1 Statistics of the four datasets ("timex" stands for time expression)

Dataset	No. of documents	No. of words	No. of timex
TimeBank	183	61,418	1243
Gigaword	2452	666,309	12,739
WikiWars	22	119,468	2671
Tweets	942	18,199	1127

Characteristics

Although the four datasets are different from each other in source, domain, corpus size, and text type, their time expressions demonstrate some common characteristics. We found such two common characteristics of time expressions about their organization and constituent words.

Characteristic 1 *Time tokens can distinguish time expressions from common text while modifiers and numerals cannot.*

Table 2 reports the percentage of the constituent words of time expressions that appear in time expressions (P_{timex}) and in common text (P_{text}). "Common text" here means the whole text with time expressions excluded. P_{timex} is defined by Eq. 1 and P_{text} is defined by Eq. 2, where $T \in \{timetoken, modifier, numeral\}$.

$$P_{timex}(T) = \frac{\textit{no. of timex that contain T}}{\textit{no. of total timex}} \tag{1}$$

$$P_{text}(T) = \frac{\textit{no. of tokens that are T}}{\textit{no. of total tokens}} \tag{2}$$

The second column of Table 2 shows that more than 91.8% of time expressions contain at least one time token; the percentage 91.8% is consistent with the one analyzed by Zhong et al. [82]. (Some time expressions without time token depend on other time expressions; for example, "95" depends on "100 days" in "95 to 100 days.") By contrast, the third column shows that no more than 0.7% of common text contain time tokens. This indicates that time tokens can distinguish time expressions from common text. On the other hand, the last four columns show that on average, 32.1% of time expressions and 21.1% of common text contain modifiers and 24.9% of time expressions and 4.1% of common text contain numerals. This indicates that modifiers and numerals cannot distinguish time expressions from common text.

Characteristic 2 *Time expressions are formed by loose structure; more than 53.5% of time tokens appear in different positions within time expressions.*

We found that time expressions are formed by loose structure, and the loose structure exhibits in the following two aspects. Firstly, many time expressions are composed of loose collocations. For example, the time token "September" can form a time expression by itself, or forms "September 2006" by another time token appearing after it, or forms "1 September 2006" by a numeral appearing before it and another time token appearing after it. Secondly, some time expressions can change their word order without changing their meanings. For example, "September 2006" can be written as "2006 September" without changing its meaning. From the point of view of the positions within time expressions, the "September" may appear as the (i) beginning or (ii) inside word of time expressions when time expressions are modeled by the BIO scheme; or it may appear as (1) a unit-word time expression, or the (2) beginning, (3) inside, (4) last word of multi-word time expressions when time expressions are modeled by the BILOU scheme.

Table 3 reports the percentage of unique time tokens and modifiers that appear in different positions within labeled time expressions. "Unique" here means ignoring the variants and frequencies of a word during counting; for example, "month," "months," and "mths" are treated the same and are counted only once. "Different positions" means the two different positions under the BIO scheme and at least two of the four different positions under the BILOU scheme. For each dataset, under the BIO scheme, more than 53.5% of unique time tokens appear in different positions; under the BILOU scheme, more than 61.4% of unique time tokens appear in different positions. The number of modifiers that appear in different positions is more than 27.5%. When the BIO or BILOU scheme is used to model time expressions, the appearance in different positions leads to inconsistent tag assignment, and the inconsistent tag assignment causes difficulty for statistical models to model time expressions. We need to explore an appropriate tagging scheme (see "TOMN Scheme" for details).

Table 2 Percentage of the constituents of time expressions that appear in time expressions (P_{timex}) and in common text (P_{text})

Dataset	Time token		Modifer		Numeral	
	P_{timex}	P_{text}	P_{timex}	P_{text}	P_{timex}	P_{text}
TimeBank	94.61	0.34	47.39	22.56	22.61	3.16
Gigaword	96.44	0.65	28.05	22.82	20.24	2.03
WikiWars	91.81	0.14	31.64	26.14	38.01	9.82
Tweets	96.01	0.50	21.38	13.03	18.81	1.28

Table 3 Percentage of unique time tokens and modifiers that appear in different positions within time expressions

Dataset	BIO scheme		BILOU scheme	
	Time token	Modifier	Time token	Modifier
TimeBank	58.18	33.33	63.64	33.33
Gigaword	61.29	45.83	77.05	46.00
WikiWars	53.57	26.19	61.40	29.55
Tweets	67.21	27.59	72.58	27.59

Named Entity Analysis

Datasets

The two benchmark datasets we used to analyze named entities are CoNLL03 [66] and OntoNotes*, which is a derived version of OntoNotes5 corpus [57]. The original CoNLL03 and OntoNotes5 datasets include data in English and other languages, but here we focus on the English data.

CoNLL03 is a benchmark dataset derived from Reuters RCV1 corpus, with 1,393 news articles between August 1996 and August 1997 [66]. It contains 4 entity types: PER, LOC, ORG, and MISC.

OntoNotes* is a dataset derived from OntoNotes5 corpus [57], which is developed for named entity analysis and consists of 3370 articles collected from different sources (e.g., newswire and web data) and contains 18 entity types.[2]

Although OntoNotes5 is a benchmark dataset, we found its annotation far from perfect. For example, "OntoNotes Named Entity Guidelines (Version 14.0)" states that ORDINAL includes all the ordinal numbers and CARDINAL includes the whole numbers, fractions, and decimals, but we found in the common text 3,588 numeral words, which is 7.1% of the total numeral words. Besides, some sequences are annotated inconsistently. For "the Cold War," for example, in some cases the whole sequence is annotated as an entity (i.e., "<ENAMEX>the Cold War</ENAMEX>"; where "ENAMEX" is the annotation mark) while in some cases only the "Cold War" is an entity (i.e., "the <ENAMEX>Cold War</ENAMEX>").

To get a high-quality dataset for analysis, we derived a dataset, which is termed OntoNotes*, from OntoNotes5 by removing those entity types whose entities are mainly composed of numbers and ordinals,[3] and moving all the "the" at the beginning of entities and all the "'s" at the end of entities outside their entities (e.g., all the "<ENAMEX>the Cold War 's</ENAMEX>" are changed to "the <ENAMEX>Cold War</ENAMEX> 's").

In splitting datasets into training, development, and test sets, we followed the setting by [66] for CoNLL03 and the setting[4] by OntoNotes5's author for OntoNotes*. Table 4 summarizes the statistics of these two datasets.

Table 4 Dataset statistics

Dataset		No. of Docs	No. of words	No. of entities	No. of types
CoNLL03	Train	946	203,621	23,499	4
	Dev.	216	51,362	5,942	
	Test	231	46,435	5,648	
	Whole	*1393*	*301,418*	*35,089*	
OntoNotes*	Train	2,729	1,578,195	81,222	11
	Dev.	406	246,009	12,721	
	Test	235	155,330	7,537	
	Whole	*3370*	*1,979,534*	*101,480*	

"Whole" denotes the whole dataset

Characteristics

Although these two datasets vary in source, corpus size, and text genre, we will see that their named entities demonstrate some common characteristics.

Characteristic 3 *Most named entities contain uncommon word(s); more than 92.2% of named entities have at least one word that hardly appears in common text.*

Table 5 reports the percentage of named entities that have words hardly appearing in common text (case sensitive). "Common text" here means the whole text with named entities excluded. The percentage is computed within a set that contains named entities and common text; the set can be a whole dataset (e.g., the CoNLL03 dataset) or only a splitting set (e.g., CoNLL03's training set). Within a set, for a word w, the rate of its occurrence in named entities over the one in the whole text is defined by Eq. 3.

$$r(w) = \frac{f_{entity}(w)}{f_{entity}(w) + f_{common}(w)} \tag{3}$$

where $f_{entity}(w)$ denotes w's occurrence in named entities while $f_{common}(w)$ denotes its occurrence in common text. If $r(w)$ reaches a threshold R, then the word w is treated as hardly appearing in common text. For CoNLL03 and its splitting sets, R is set to 1, which means the word w does not appear in common text. For OntoNotes* and its splitting sets, R is set to 0.95, because its annotation is imperfect: its

Table 5 Percentage of named entities that have at least one word that hardly appears in common text

	Whole	Train	Dev.	Test
CoNLL03	97.77	98.77	99.19	98.62
OntoNotes*	92.91	92.20	95.22	95.61

[2] OntoNotes5's 18 entity types include CARDINAL, DATE, EVENT, FAC, GPE, LANGUAGE, LAW, LOC, MONEY, NORP, ORDINAL, ORG, PERCENT, PERSON, PRODUCT, QUANTITY, TIME, WORK_OF_ART.

[3] Those removed entity types are CARDINAL, DATE, MONEY, ORDINAL, PERCENT, QUANTITY, TIME.

[4] https://github.com/ontonotes/conll-formatted-ontonotes-5.0

common text contains some words that should be treated as named entities, such as “American.” We call such kind of words *uncommon words*.

From Table 5 we can see that for a set, more than 92.2% of its named entities contain at least one uncommon word. This phenomenon of uncommon words widely exists in the CoNLL03 and OntoNotes* datasets as well as their training, development, and test sets. An implication of this phenomenon is that for a dataset, the uncommon words of its development and test sets also hardly appear in the common text of its training set. This suggests that those words of the test set that hardly appear in the common text of the training set tend to predict named entities.

Characteristic 4 *Named entities are mainly made up of proper nouns. In the whole text, more than 84.8% of proper nouns appear in named entities; within named entities, more than 80.1% of the words are proper nouns.*

Table 6 lists the top 4 POS tags appearing in named entities, and their percentages over the whole tags in named entities (p_{entity}) and over the corresponding tags in the whole text (p_{whole}):

$$p_{entity}(t) = \frac{f_{entity}(t)}{\sum_{t_i} f_{entity}(t_i)} \tag{4}$$

$$p_{whole}(t) = \frac{f_{entity}(t)}{f_{entity}(t) + f_{common}(t)} \tag{5}$$

where $f_{entity}(t)$ denotes the occurrence of tag t in named entities while $f_{common}(t)$ denotes its occurrence in common text.

We can see that the top 4 POS tags in both CoNLL03 and OntoNotes* are the same and they are NNP, JJ, NN, and NNPS. The p_{entity} of proper nouns (including NNP and NNPS) reaches more than 80.1%, and this indicates that named entities are mainly made up of proper nouns. The p_{whole} of proper nouns reaches more than 84.8%, and this indicates that in the whole text, the proper nouns mainly appear in named entities.[5] Within named entities, those JJ words are mainly the nationality words and those NN words are some common nouns.

Table 6 Top 4 POS tags in named entities and their percentage within named entities (p_{entity}) and over the corresponding tags in the whole text (p_{whole})

CoNLL03			OntoNotes*		
POS	p_{entity}	p_{whole}	POS	p_{entity}	p_{whole}
NNP	83.81	84.82	NNP	77.67	85.88
JJ	5.82	17.57	JJ	4.60	6.77
NN	4.89	6.46	NN	4.57	2.91
NNPS	1.55	94.12	NNPS	2.50	93.04

Methodology

This section describes the method TOMN we proposed to extract time expressions from unstructured text and the method UGTO we proposed to extract named entities.

TOMN: Time Expression Extraction with Constituent-Based TOMN Scheme

Figure 2 shows the overview of TOMN. It mainly includes three parts: TOMN scheme, TmnRegex, and time expression modeling. The TOMN scheme is a constituent-based tagging scheme with four tags. TmnRegex is a set of regular expressions for time-related words. Time expressions are modeled under a CRFs framework with the help of TmnRegex and the TOMN scheme.

TOMN Scheme

Characteristic 2 suggests us to explore an appropriate tagging scheme to model time expressions. We defined a constituent-based tagging scheme termed TOMN scheme with four tags: T, O, M, and N; they indicate the constituents of time expressions, namely *time token*, *modifier*, *numeral*, and the words *outside* time expressions.

Conventional tagging schemes like the BIO scheme[6] [67] and the BILOU scheme[7] [63] are based on *the positions within a labeled chunk*. BIO refers to the beginning, inside, and outside of a chunk; BILOU refers to a unit-word chunk, or the beginning, inside, last word of a multi-word chunk. The TOMN scheme instead is based on *the constituents of a labeled chunk*, indicating the constituent words of time expressions. Next, we use the BILOU scheme as the representative of the conventional position-based tagging schemes for analysis.

Using the BILOU scheme for time expression extraction leads to inconsistent tag assignment. Characteristic 2 indicates that time expressions are formed by loose structure, which exhibits in two aspects: loose collocations

[5] The p_{whole} of proper nouns does not reach 100% mainly because each individual dataset is concerned with certain types of named entities and partly because some NNP* words are POS tagging errors, e.g., “SURPRISE DEFEAT” is tagged as “NNP NNP,” but it should be tagged as “JJ NN.”

[6] The BIO scheme in this paper denotes the standard IOB2 scheme described in [67].

[7] The BILOU scheme is also widely known as the BIOES or IOBES scheme.

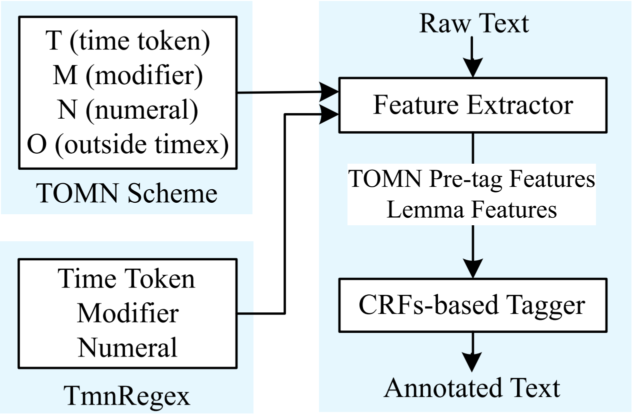


Fig. 2 Overview of TOMN. Top-left side shows the TOMN scheme, which consists of four tags. Bottom-left side is the TmnRegex, a set of regular expressions for time-related words. Right-hand side shows the time expression modeling, with the help of TmnRegex and the TOMN scheme

and exchangeable order. Under the BILOU scheme, both loose collocations and exchangeable order lead to the problem of inconsistent tag assignment. Suppose "September," "September 2006," "2006 September," and "1 September 2006" are four manually labeled time expressions in the training data. During feature extraction, they are tagged as "September/U," "September/B 2006/L," "2006/B September/L," and "1/B September/I 2006/L" (see Fig. 1a). The four "September" have the same word and express the same meaning, but because they appear in different positions within labeled time expressions, they are assigned with different tags (i.e., U, B, L, and I).

The inconsistent tag assignment causes difficulty for statistical models to model time expressions. Firstly, inconsistent tag assignment reduces the predictive power of lexicon. A word assigned with different tags causes confusion to model that word. If a word is assigned with different tags in equal number, then that word itself cannot provide any useful information to determine which tag should be assigned to it. Reducing the predictive power of lexicon indicates reducing the predictive power of time tokens, and this contradicts Characteristic 1 which describes that time tokens can distinguish time expressions from common text. Secondly, inconsistent tag assignment may cause another problem: tag imbalance. If a tag of a word dominates in the training data, then all the instances of that word in test data will be predicted as that tag. For example, "1 September 2006" can be written as "September 1, 2006" in some cultures. If the training data are collected from the style of "1 September 2006" in which most "September" are assigned with I, then it is difficult for a trained model to correctly predict the data that are collected from the style of "September 1, 2006" in which "September" should be predicted as B.

Our TOMN scheme overcomes the problem of inconsistent tag assignment. The TOMN scheme assigns a tag to a word according to the constituent role that the word plays in time expressions. Since our TmnRegex well defines the constituent words of time expressions (see "TmnRegex") and same word plays same constituent role in time expressions, therefore, the same word is assigned with the same tag, regardless of its frequency and its positions within time expressions. For example, the TOMN scheme assigns the above four time expressions as "September/T," "2006/T September/T," "September/T 2006/T," and "1/N September/T 2006/T" (see Fig. 1b). The four "September" have the same tag of T and statistical models can model them without any confusion. With consistent tag assignment, the TOMN scheme protects the predictive power of time tokens and avoids the potential tag imbalance.

TmnRegex

TOMN uses three token types, namely time token, modifier, and numeral, to group those time-related words. These three token types corresponds to three of the above four tags (i.e., T, M, and N), and are same to the ones defined by Zhong et al. [82]. Time tokens explicitly express information about time, such as month (e.g., "September"), and time units (e.g., "month"). Modifiers include thos words that modify time tokens in time expressions; for example, the two modifiers "the" and "last" modify the time token "month" in "the last month." Numerals include ordinals and numbers.

The three token types with the words they group form a set of token regular expressions, which is denoted by TmnRegex. TmnRegex is constructed by importing token regular expressions from SUTime [10]. Note that TmnRegex collects from SUTime only the regular expressions at the level of token, the same as SynTime [82] did, and it contains 115 unique time tokens, 57 modifiers, and 58 numerals, without counting the words with changing digits.

Time Expression Extraction

Time expression extraction mainly includes two parts: (1) feature extraction and (2) model learning and tagging.

Feature Extraction The features we extracted for time expression extraction include two kinds: TOMN pre-tag features and lemma features. During feature extraction we used w_i to denote the i-th word in the text.

TOMN Pre-tag Features: Characteristic 1 suggests that time tokens can distinguish time expressions from common text while modifiers and numerals cannot, therefore, how to leverage the information of these words becomes crucial. In practice, we treated them as a kind of pre-tag features under the TOMN scheme. Specifically, a time token is pre-tagged

Table 7 Features extracted for the word w_i in time expression modeling

1	TOMN pre-tags in a 5-word window of w_i, namely pre-tag of w_{i-2}, w_{i-1}, w_i, w_{i+1}, and w_{i+2}
2	If w_i is a M or N, then check whether it modifies any time token
3	Lemmas in a 5-word window of w_i, namely lemmas of w_{i-2}, w_{i-1}, w_i, w_{i+1}, and w_{i+2}

by T, a modifier is pre-tagged by M, and a numeral is pre-tagged by N; other common words are pre-tagged by O. The assignment of pre-tags is conducted by simply looking up the words at TmnRegex.

The last four columns of Table 2 suggests that modifiers and numerals constantly appear in time expressions and in common text. To distinguish where a modifier or numeral appears, we conducted a checking for the modifiers and numerals (those words assigned with the pre-tag of M or N (denoted as M/N)) to record whether they directly or indirectly modify any time token. "Indirectly" here means a M/N word together with other M/N words modifies a time token; for example, in "last two months," "last" (M) together with "two" (N) modifies "months" (T). The checking is a loop searching relying on time tokens. For each time token we search its left side without exceeding the previous time token and search its right side without exceeding the next time token. When searching a side of a time token, if encounter a M/N word, then record this M/N word and continue searching; if encounter a word that is not a M/N word, then stop the searching for this side of this time token. After the checking, those M/N words that modify time tokens are recorded; for example, the modifier "two" in "two months" is recorded while in "two apples" it is not recorded. The checking is treated as a feature for modeling.

Lemma Features: The lemma features include the word shape of w_i in a 5-word window, namely w_{i-2}, w_{i-1}, w_i, w_{i+1}, and w_{i+2}. If w_i contains changing digit(s), then we set its lemma by its token type. For example, the lemma of "20:16" is set by TIME. We use five special lemma for the words with changing digits: YEAR, DATE, TIME, DECADE, and NUMERAL. The lemma features can help build connections among time expressions; for example, the two different words "20:16" and "19:25:33" are connected at TIME.

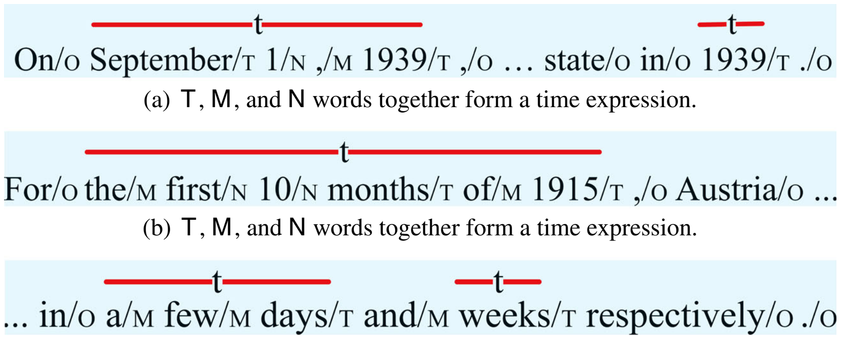
Table 7 summarizes the features extracted for w_i to modeling time expressions. For the TOMN pre-tag features, we extracted them in a 5-word window of w_i. For the checking feature, we only considered the current word w_i. For the lemma features, we extracted them for all the words in text in both training and test phases.

Model Learning and Tagging During modeling and tagging, each word is assigned with one of the TOMN tags. Note that the TOMN scheme is used in feature extraction as a kind of pre-tag features as well as in sequence tagging as labeling tags.

After sequence tagging, those T, M, and N words (or non-O words) that appear together are extracted as a time expression (see Fig. 3a and b). A special kind of modifiers, i.e., the linkers "to," "-," "or," and "and" separate those non-O words into parallel time expressions (see Fig. 3c).

UGTO: Named Entity Extraction with Constituent-Based UGTO Scheme

Characteristic 3 and 4 suggest that for a dataset, those words of its test sets that hardly appear in the common text of its training set tend to predict named entities, and they are mainly proper nouns. This is our main idea for named entity extraction. Figure 4 visualizes this idea with a simple example: in the unannotated test set, words like "Boston" and "Reuters" hardly appear in the training set's common text and tend to predict named entities. Such words are also called *uncommon words* and they include two kinds: the first kind appears in the training set's named entities (e.g., "Boston") while the second kind does not (e.g., "Reuters"). The remaining of this section illustrates how we developed our idea in UGTO.


t
On/O September/T 1/N ,/M 1939/T ,/O … state/O in/O 1939/T ./O
t
For/O the/M first/N 10/N months/T of/M 1915/T ,/O Austria/O …
t
… in/O a/M few/M days/T and/M weeks/T respectively/O ./O

(a) T, M, and N words together form a time expression.

(b) T, M, and N words together form a time expression.

(c) Linker "and" separates two time expressions.

Fig. 3 Examples of time expression extraction. The symbol *t* indicates time expressions

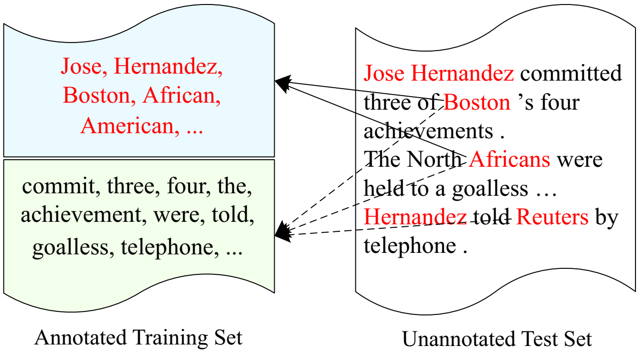

Jose, Hernandez, Boston, African, American, ...
commit, three, four, the, achievement, were, told, goalless, telephone, ...
Jose Hernandez committed three of Boston 's four achievements . The North Africans were held to a goalless ... Hernandez told Reuters by telephone .
Annotated Training Set
Unannotated Test Set

Fig. 4 Main idea: those words (red font) of the test set that hardly appear in the common text of the training set (bottom-left) tend to predict named entities. Such words include two kinds: the first kind (e.g., "Boston") appears in training set's named entities (top-left) while the second kind (e.g., "Reuters") does not. The training set is annotated, indicated by the colored background, while the test set is not. Solid arrow denotes appearing in the training's named entities while dashed arrow denotes hardly appearing in the training set's common text

UGTO models named entities under a CRFs framework and follows a typical CRFs procedure. It mainly includes four components: (1) uncommon word induction, (2) word lexicon, (3) UGTO scheme, and (4) named entity modeling.

Uncommon Word Induction

We induced two kinds of uncommon words from the annotated training set and the unannotated test set.

For each dataset, the first kind of uncommon words is induced from the annotated training set. At the beginning, there is an empty list L. For each word w in the named entities of its training set, its rate $r(w)$ of hardly appearing in common text is calculated by Eq. 3. If $r(w)$ reaches a threshold R, then w is added to L. Like the setting in "Characteristics," R is set to 1 for CoNLL03 and 0.95 for OntoNotes*.

The second kind of uncommon words is induced from the unannotated test set. They include those words (excluding those in L) that appear in the unannotated test set and do not appear in the common text of the training set. Inducing them is to extract out-of-vocabulary named entities. This kind of uncommon words can be viewed as the information from the unannotated data, and note that they can be only used in the test phase, because the unannotated test set is not available in the training phase.

Word Lexicon

Word lexicon includes two kinds of entity-related words: entity token and modifier. Entity tokens are collected from external sources; some entity tokens are from the entity list provided by the CoNLL03 shared task [66] and some are from Wikipedia.[8] Modifiers are collected from the training set according to the annotation guideline of each dataset; they include two kinds: generic modifier and trigger word. Generic modifiers can modify several types of entity tokens, such as "of" and "and," while trigger words modify a specific type of entity tokens, such as "Mr." modifying PER entity tokens.

Table 8 Number of word lexicon

Word lexicon	Number
Entity token	9658
Generic modifier	17
PER trigger word	31
Other trigger word	116

We put all the entity tokens together, without using their entity types (e.g., PER, LOC, and ORG), so as to remove the impact of the information carried in entity types. For the trigger words, we separated PER trigger words from other trigger words because PER trigger words appear outside named entities while other trigger words appear inside named entities.

Unlike previous works that used lexicon in word sequences [28, 63], we used lexicon in words. For example, we did not use "Boston University" but used "Boston" and "University." Table 8 summarizes the number of the word lexicon.

UGTO Scheme

The constituent-based UGTO scheme consists of four tags: U, G, T, and O; they indicate the constituents of named entities: *uncommon word*, *generic modifier*, *trigger word*, and the words *outside* named entities. U encodes uncommon words and entity tokens. G encodes generic modifiers while T encodes trigger words.

Named Entity Extraction

Like time expression extraction, named entity extraction also includes two parts: (1) feature extraction and (2) model learning and tagging.

Feature Extraction The features we extracted for named entity extraction include three kinds: UGT pre-tag features, word cluster features, and basic lexical & POS features. The ith word in the text is denoted by w_i.

UGTO Pre-tag Features: UGTO pre-tag features are designed to encode the information of those uncommon words and word lexicon under our UGTO scheme.

[8] https://en.wikipedia.org/wiki/Lists_of_cities_by_country and https://en.wikipedia.org/wiki/Lists_of_people_by_nationality.

Table 9 Features extracted for the word w_i in named entity modeling

1	UGTO pre-tags in a 5-word window of w_i, namely pre-tag of w_{i-2}, w_{i-1}, w_i, w_{i+1}, and w_{i+2}
2	Whether w_i is matched by any entity token; whether w_i is hyphenized by any entity token
3	Prefix paths of 4, 8, and 12 bits from a hierarchical word clusters for w_i
4	w_i itself, its lowercase, its lemma, whether the first letter is capitalized, where it is the beginning of a sentence, POS tag

Specifically, a word is encoded by U if it satisfies two conditions: (1) it appears in the list L induced in "Uncommon Word Induction" (i.e., the first kind of uncommon words) or does not appear in the common text of training set (i.e., the second kind of uncommon words[9]); (2) it has a POS tag of NNP* or is matched by the entity tokens or is hyphenized by at least one entity token (e.g., "U.S.-based" and "English-oriented"). A word is encoded by G if it is matched by any of generic modifiers. A word is encoded by TP if it is matched by any of PER trigger words; a word is encoded by T if it is matched by other trigger words.

Besides UGTO pre-tag features, we used two features to indicate (1) whether a word is matched by any of the entity tokens and (2) whether a word is hyphenized by any of the entity tokens.

Word Cluster Features: Previous works have demonstrated that word clusters are useful for many information extraction tasks [34, 50]. We followed those words to derive the prefix paths of 4, 8, and 12 bits from a hierarchical word clusters as features for a word. In practice, we used the publicly available word clusters: the bllip-clusters for the CoNLL03 dataset and the one trained by OntoNotes 5.0 corpus [57] for the OntoNotes* dataset.

Lexical & POS Features: The lexical & POS features are widely used for named entity modeling and we extracted three kinds of such features for w_i: (1) the word w_i itself, its lowercase, and its lemma; (2) whether its first letter is capitalized and whether it is the beginning of a sentence; and (3) its POS tag.

Table 9 summarizes the features extracted for w_i to modeling named entities. For the UGTO pre-tag features and lexical & POS features, we extracted them in a 5-word window of w_i, namely the features of w_{i-2}, w_{i-1}, w_i, w_{i+1}, and w_{i+2}. For the word cluster features we consider them for only the w_i.

Model Learning and Tagging UGTO uses Stanford Tagger to get word lemma and POS tags and uses Java version of CRFSuite with its default parameters for modeling. Note that the UGTO scheme is used in feature extraction as a kind of pre-tag features as well as in sequence tagging as labeling tags.

After sequence tagging, we extracted named entities from tagged sequences. For the models excluding entity types from labeling tags (in Experiment 1), those U, G, and T words that appear together form a named entity (see Example (1)∼(3) in Table 14). For the models incorporating entity types into labeling tags (in Experiment 2), those consecutive words that are tagged with the same entity type form a named entity (see Example (4)∼(6) in Table 14).

Experiments

Time Expression Extraction

We conducted experiments to evaluate TOMN against five state-of-the-art methods, namely HeidelTime (with the Colloquial setting for Tweets), SUTime, SynTime, ClearTK-TimeML (short as "ClearTK"), and UWTime, on three datasets, namely TE-3, WikiWars, and Tweets.[10]

Experimental Setup

Datasets The three datasets used for the experiments of time expression extraction are TE-3, WikiWars, and Tweets. TE-3 uses the TimeBank corpus as the training set and the Platinum corpus as the test set [75]. WikiWars is a domain-specific dataset in formal text, consisting of 22 English Wikipedia articles about famous wars [48]. Tweets is a comprehensive dataset in informal text, with 942 tweets that contain time expressions [82]. The performance of a method on a dataset is reported on the test set of that dataset.

Baseline Methods We evaluated TOMN against five state-of-the-art methods, including three rule-based methods (i.e., HeidelTime, SUTime, and SynTime) and two learning-based methods (i.e., ClearTK and UWTime). HeidelTime [71] and SUTime [9] use predefined deterministic rules and achieve the best results in the relaxed match while ClearTK [5] uses a CRFs framework with the BIO scheme and achieves the best result in the strict match in TempEval-3 [75]. UWTime uses combinatory categorial grammar (CCG) to predefine linguistic structure for time expressions

[9] Note that this kind of uncommon words are not available in the training phase because they are extracted from the unannotated test set.

[10] We followed [82] not to use the Gigaword dataset in experiments because its labels are not ground-truth labels, but are automatically generated by other taggers.

and achieves better results than HeidelTime on the TE-3 and WikiWars datasets [32]. SynTime uses a set of general heuristic rules and achieves good results on the TE-3, WikiWars, and Tweets datasets [82]. SynTime has two versions, a basic version and an expanded version. Because the expanded version requires extra manual annotations, for fair comparison, we used the basic version to ensure that the token regular expressions used in SynTime and TOMN are comparable.

Evaluation Metrics We reported results in the three standard metrics $Precision$, $Recall$, and F_1 under *strict match* and *relaxed match* by using the evaluation toolkit of TempEval-3 [75].

Experiment Results

Table 10 reports the performance of TOMN and baseline methods. Among the 18 measures, TOMN achieves 13 best or second best results. It is better than SynTime which achieves 10 best or second best ones, and much better than other baselines which achieve at most 4 best or second best. For each measure, TOMN achieves either the best or comparable results. Especially for the F_1, TOMN performs the best in strict F_1 on Tweets and in relaxed F_1 on WikiWars; for other F_1, TOMN performs comparably (most are within 0.5% difference) to the corresponding best results.

TOMN vs. Baseline Methods We further compared TOMN with the rule-based methods and the learning-based methods.

TOMN vs. Rule-based Baselines. On TE-3 and Tweets, TOMN achieves comparable results with SynTime. On WikiWars, TOMN achieves the F_1 with 2.0 to 2.3% absolute increase over SynTime. This indicates that compared with SynTime, TOMN is equally effective on comprehensive data and more effective on specific domain data. The reason is that the heuristic rules of SynTime are greedy for recalls at the cost of precisions, and the cost is expensive when it comes to specific domain data. TOMN instead leverages statistical information from entire corpus, which may miss the rare time expressions but helps recognize time expressions more precisely; especially in specific domain data, the statistical information significantly improves the precisions at little cost of recalls. For HeidelTime and SUTime, except the strict F_1 on WikiWars, TOMN outperforms them on all the datasets, with up to 15.3% absolute increase in recalls and up to 12.0% absolute increase in F_1. The reason is that the deterministic rules of HeidelTime and SUTime are designed in fixed manner, which lacks flexibility [82].

Table 10 Performance of TOMN and baseline methods

Dataset	Method	Strict match			Relaxed match		
		$Pr.$	$Re.$	F_1	$Pr.$	$Re.$	F_1
TE-3	HeidelTime	83.85	78.99	81.34	93.08	87.68	90.30
	SUTime	78.72	80.43	79.57	89.36	91.30	90.32
	SynTime	91.43	**92.75**	**92.09**	94.29	**95.65**	**94.96**
	ClearTK	85.90	79.70	82.70	93.75	86.96	90.23
	UWTime	86.10	80.40	83.10	94.60	88.40	91.40
	TOMN	**92.59**	90.58	91.58	**95.56**	93.48	94.51
WikiWars	HeidelTime	**88.20**	78.50	83.10	95.80	85.40	90.30
	SUTime	78.61	76.69	76.64	95.74	89.57	92.55
	SynTime	80.00	80.22	80.11	92.16	**92.41**	92.29
	ClearTK	87.69	80.28	**83.82**	96.80	90.54	93.56
	UWTime	87.70	78.80	83.00	**97.60**	87.60	92.30
	TOMN	84.57	**80.48**	82.47	96.23	92.35	**94.25**
Tweets	HeidelTime	**91.67**	74.26	82.05	96.88	78.48	86.71
	SUTime	77.69	79.32	78.50	88.84	90.72	89.77
	SynTime	89.52	94.07	91.74	93.55	**98.31**	**95.87**
	ClearTK	86.83	75.11	80.54	96.59	83.54	89.59
	UWTime	88.36	70.76	78.59	**97.88**	78.39	87.06
	TOMN	90.69	**94.51**	**92.56**	93.52	97.47	95.45

For each measure, we make bold the best results and underline the second best. Some results are reported directly from the sources where the results are publicly available

TOMN vs. Learning-based Baselines. Except the strict F_1 on WikiWars, TOMN outperforms ClearTK and UWTime on all three datasets in all the recalls and F_1. Especially on TE-3 and Tweets datasets, TOMN improves the recalls by at least 9.8% in strict match and at least 5.1% in relaxed match, and improves the F_1 by at least 8.5% in strict match and at least 3.1% in relaxed match. The reasons are that the fixed linguistic structure predefined in UWTime cannot fully capture the loose structure of time expressions, the BIO scheme used in ClearTK reduces the predictive power of time tokens, and some of their features (e.g., syntactic dependency) actually hurt the modeling. For the strict F_1 on WikiWars, TOMN performs slightly worse than the two learning-based methods because like SynTime, TOMN follows TimeBank and SynTime to exclude the prepositions (except "of") from time expressions while some time expressions in WikiWars include these prepositions.

Factor Analysis We conduct experiments to analyze the impact of the TOMN scheme as labeling tags and the features used in TOMN. The results are reported in Table 11.

Impact of TOMN Labeling Tags. To analyze the impact of the TOMN scheme as labeling tags, we keep all the features unchanged except change the labeling tags from TOMN scheme to BIO scheme to get a BIO system and to BILOU scheme to get a BILOU system. The BIO and BILOU systems use the same TOMN pre-tag features and lemma features that are used in TOMN.

The tag assignment of BIO and BILOU schemes during feature extraction in the training stage follows their traditional use; for example, a unit-word time expression is assigned with B under BIO scheme while it is assigned with U under BILOU scheme. When extracting time expressions from tagged sequence in the test stage, we adopt two strategies. One strategy follows their traditional use in which time expressions are extracted according to the tags of words; for example, a U word under BILOU scheme is extracted as a time expression. The other strategy follows the one used for TOMN in which the non-O words that appear together are extracted as a time expression. The traditional strategy is denoted by "$trad$" while the non-O strategy is by "*nono*." The results of the BIO and BILOU

Table 11 Impact of factors

Dataset	Method	Strict match			Relaxed match		
		$Pr.$	$Re.$	F_1	$Pr.$	$Re.$	F_1
TE-3	TOMN	**92.59**	**90.58**	**91.58**	95.56	93.48	**94.51**
	BIO$_{trad}$	83.06	74.64	78.63	94.35	84.78	89.31
	BIO$_{nono}$	84.68	76.09	80.15	94.35	84.78	89.31
	BILOU$_{trad}$	84.75	72.46	78.12	94.92	81.16	87.50
	BILOU$_{nono}$	86.44	73.91	79.69	94.92	81.16	87.50
	−PreTag	89.36	60.87	72.41	**95.74**	65.22	77.59
	−Lemma	81.56	83.33	82.44	92.20	**94.20**	93.19
WikiWars	TOMN	84.57	**80.48**	**82.47**	96.23	92.35	**94.25**
	BIO$_{trad}$	77.75	71.03	74.24	93.39	85.31	89.17
	BIO$_{nono}$	77.75	71.03	74.24	93.39	85.31	89.17
	BILOU$_{trad}$	79.56	72.03	75.61	93.56	84.71	88.91
	BILOU$_{nono}$	79.78	72.23	75.82	93.56	84.71	88.91
	−PreTag	**87.22**	70.02	77.68	**99.25**	79.68	88.39
	−Lemma	74.80	75.25	75.03	92.20	**92.56**	92.28
Tweets	TOMN	90.69	94.51	**92.56**	93.52	97.47	**95.45**
	BIO$_{trad}$	89.16	93.67	91.36	92.37	97.05	94.65
	BIO$_{nono}$	90.24	93.67	91.93	93.50	97.05	95.24
	BILOU$_{trad}$	89.37	**95.78**	92.46	92.13	**98.73**	95.32
	BILOU$_{nono}$	90.65	94.09	92.34	93.50	97.06	95.24
	−PreTag	**92.41**	61.60	73.92	**98.10**	65.40	78.48
	−Lemma	90.69	94.51	**92.56**	93.52	97.47	**95.45**

"BIO" denotes the systems that replace TOMN labeling tags by BIO tags while "BILOU" denotes the systems that replace by BILOU tags. "$trad$" indicates the traditional strategy for extraction while "*nono*" indicates the non-O strategy. "−" indicates the kind of features removed from TOMN; "PreTag" denotes the TOMN pre-tag features; and "Lemma" denotes the lemma features. For each measure, the best result is made bold systems are reported as “BIO” and “BILOU” in Table 11. We can see that the non-O strategy performs almost the same as the traditional strategy, and the BIO systems achieve comparable or slightly better results compared with the BILOU systems. The reason is that time expressions on average contain about two words; in that case, BILOU scheme is reduced approximately to BLOU scheme and BIO scheme is changed approximately to BLO scheme. Between BLOU scheme and BLO scheme there is only slight difference; and under the impact of inconsistent tag assignment and TOMN pre-tag features, this slight difference affects slightly to the performance. Following we do not distinguish BILOU scheme from BIO scheme and do not distinguish non-O strategy from traditional strategy; the four methods of BIO_{trad}, BIO_{nono}, BILOU_{trad}, and BILOU_{nono} are simply represented by “BILOU.”

On TE-3 and WikiWars, TOMN significantly outperforms BILOU. TOMN achieves the recalls that are 7.0 to 14.5% absolute higher than those of BILOU and achieves the F_1 that are 5.0 to 11.4% absolute higher than those of BILOU. The reason is that the loose collocations and exchangeable order in time expressions lead BILOU scheme to suffer from the problem of inconsistent tag assignment; TOMN scheme instead overcomes that problem.

On Tweets, TOMN and BILOU achieve similar performance; the difference between their performance ranges within 1% in most measures. The reason is that 62.9% of time expressions in Tweets are one-word time expressions and 96.0% of time expressions contain time tokens, which means the one-word time expressions contain only the time tokens. In that case, TOMN scheme is reduced approximately to TO scheme and BILOU scheme is reduced approximately to UO scheme. Then UO scheme becomes a constituent-based tagging scheme in which U indicates the time token. It is equivalent to TO scheme. (BIO scheme is reduced approximately to BO scheme in which B indicates the time token. Then BO scheme is equivalent to TO scheme as well as UO scheme.)

Impact of TOMN Pre-tag Features. To analyze the impact of TOMN pre-tag features, we remove them from TOMN. After they are removed, although most of the precisions increase and even reach highest scores, all the recalls and F_1 drop dramatically, with absolute decreases of 10.4 to 32.9% in recall and 4.8 to 19.1% in F_1. That means TOMN pre-tag features significantly improve the performance and confirms the predictive power of time tokens. The results also validate that pre-tag is a good way to use those lexicon.

Impact of Lemma Features. When lemma features are removed, the performance in relaxed match on all the datasets is affected slightly. That is because TOMN pre-tag features provide useful information to recognize time tokens. The strict match on TE-3 and WikiWars decreases dramatically, indicating that lemma features heavily affect the recognition of modifiers and numerals. The strict match on Tweets is affected little because tweets tend not to use modifiers and numerals in time expressions.

Named Entity Extraction

Experimental Setup

Datasets The two benchmark datasets used for the experiments of named entity extraction are CoNLL03 [66] and OntoNotes* [57]. They are detailed in “Datasets.”

Baseline Methods The compared methods include two representative state-of-the-art methods: StanfordNER [20] and LSTM-CRF [31]. StanfordNER derives hand-crafted features under CRFs with the BIO scheme. LSTM-CRF derives automatic features learned by long short-term memory networks (LSTMs) [25] under CRFs with the IOBES scheme. We used StanfordNER as the representative of those traditional hand-crafted-feature methods and LSTM-CRF as the representative of those auto-learned-feature methods.

Evaluation Metrics We used the evaluation toolkit of the CoNLL2003 shared task [66] to report results under the three standard metrics: $Precision$, $Recall$, and F_1.

Experimental Design

We designed two kinds of experiments to evaluate UGTO against the two baselines.

– **Experiment 1** *Exclude entity types from labeling tags during the whole process.*

– **Experiment 2** *Incorporate entity types into labeling tags during modeling and tagging (i.e., training and testing, but not evaluation).*

Experiment 1 is a pure entity extraction task. In this experiment, the labeling tags of UGTO are {U, G, T, O}; the ones of StanfordNER are {B, I, O}; the ones of LSTM-CRF are {I, O, B, E, S}.

Experiment 2 is a joint named entity extraction and classification task (i.e., NER). Designing this experiment is to test *whether does named entity classification enhance named entity extraction during modeling?* In this experiment, the labeling tags for UGTO are the combination of {U, G, T, O} and entity types, such as U-PER, G-LOC, and O; similarly, the labeling tags for StanfordNER and LSTM-CRF are the combination of their basic tags and entity types, such as B-PER, I-LOC, and O.

Table 12 Named entity extraction performance of UGTO and baselines. "w/o" indicates Experiment 1 and "$w/type$" indicates Experiment 2

Dataset	Method	Dev. set			Test set		
		$Pr.$	$Re.$	F_1	$Pr.$	$Re.$	F_1
CoNLL03	StanfordNER$_{w/o}$	95.80	95.93	95.86	93.28	93.59	93.43
	StanfordNER$_{w/type}$	**96.43**	95.36	95.89	93.77	92.49	93.13
	LSTM-CRF$_{w/o}$	94.96	95.46	95.21	92.02	93.48	92.74
	LSTM-CRF$_{w/type}$	95.68	94.36	95.02	92.99	91.55	92.27
	UGTO$_{w/o}$	95.84	**96.21**	**96.02**	94.15†	**94.56**†	**94.35**†
	UGTO$_{w/type}$	96.24	95.76	96.00	**94.29**†	94.18†	94.23†
OntoNotes*	StanfordNER$_{w/o}$	92.38	91.62	92.00	93.11	91.99	92.54
	StanfordNER$_{w/type}$	**93.17**	91.17	92.16	**93.69**	90.96	92.31
	LSTM-CRF$_{w/o}$	91.41	91.86	91.64	92.35	91.91	92.13
	LSTM-CRF$_{w/type}$	92.52	90.32	91.41	93.37	90.28	91.80
	UGTO$_{w/o}$	93.28	**92.08**†	**92.67**†	93.43	**92.26**	92.84†
	UGTO$_{w/type}$	93.32	92.01†	92.66†	93.62	92.17†	**92.89**†

†Improvement of our result over the best one of baselines is statistically significant ($p < 0.05$ under t test). For each measure, the best result is made bold

We were concerned with named entity extraction and reported only the performance of named entity extraction. For Experiment 2, after named entities were extracted, we converted them to the CoNLL format and removed their entity types so as to report the performance of named entity extraction. We did the same conversion for both UGTO and the two baselines.

Experimental Results

Table 12 reports the overall performance of UGTO and the two baselines in named entity extraction.

UGTO$_{w/o}$ vs. Baselines in Experiment 1 UGTO$_{w/o}$ outperforms StanfordNER$_{w/o}$ and LSTM-CRF$_{w/o}$ on both datasets in recall and F_1. Specially , UGTO$_{w/o}$ reduces 3.86%∼14.00% of error in F_1. Compared with StanfordNER$_{w/o}$ which mainly treats the named entities of training set as a kind of dictionary, UGTO$_{w/o}$ explicitly takes into account both the named entities and common text of training set. The second kind of uncommon words can help extract more out-of-vocabulary named entities.

Let us look at LSTM-CRF. According to literature, LSTM-CRF significantly outperforms StanfordNER on the NER task [31], however, it performs comparably with or worse than UGTO$_{w/o}$ and StanfordNER$_{w/o}$ on named entity extraction. This indicates that simple hand-crafted-feature methods can achieve state-of-the-art performance on named entity extraction.

Experiment 2 vs. Experiment 1 For each of UGTO and baselines, we compared its performance in Experiment 2 with its performance in Experiment 1. On both CoNLL03 and OntoNotes* datasets, UGTO$_{w/type}$ and UGTO$_{w/o}$ perform similar; StanfordNER$_{w/type}$ and StanfordNER$_{w/o}$ perform similar; LSTM-CRF$_{w/type}$ and LSTM-CRF$_{w/o}$

Table 13 Impact of factors. "BIO" indicates the systems that replace UGTO labeling tags by BIO tags. "−" indicates removing this factor from UGTO$_{w/o}$

Dataset	Method	Dev. set			Test set		
		$Pr.$	$Re.$	F_1	$Pr.$	$Re.$	F_1
CoNLL03	UGTO$_{w/o}$	**95.84**	**96.21**	**96.02**	**94.15**	**94.56**	**94.35**
	BIO	94.78	95.14	94.96	93.66	94.02	93.83
	−UGTO PreTag	94.68	93.23	93.95	93.47	91.04	92.34
	−Word Clusters	95.09	94.96	95.02	94.01	93.23	93.62
OntoNotes*	UGTO$_{w/o}$	**93.28**	**92.08**	**92.67**	**93.43**	**92.26**	**92.84**
	BIO	92.63	91.05	91.83	92.87	91.35	92.10
	−UGTO PreTag	92.65	90.08	91.35	92.71	89.64	91.15
	−Word Clusters	92.67	90.74	91.69	93.22	92.16	92.68

For each measure, the best result is made bold

Table 14 Examples of named entity extraction

Models excluding entity types from labeling tags	
(1)	Japan/U began/O its/O Asian/U Cup/T title/O with/O a/O lucky/O 2-1/O win/O against/O ...
(2)	UK/U Department/T of/G Transport/T on/O Friday/O said/O that/O ...
(3)	Australian/U Tom/U Moody/U took/O six/O for/O ...
Models incorporating entity types into labeling tags	
(4)	Japan/U-LOC began/O its/O Asian/U-MISC Cup/T-MISC title/O with/O a/O lucky/O 2-1/O win/O against/O ...
(5)	UK/U-ORG Department/T-ORG of/G-ORG Transport/T-ORG on/O Friday/O said/O that/O ...
(6)	Australian/U-MISC Tom/U-PER Moody/U-PER took/O six/O for/O ...

Colored background indicates named entities

also perform similar. That means that the joint task of named entity extraction and classification does not improve the performance of named entity extraction, in both our model and the two baselines.

Factor Analysis in Experiment 1 We conducted controlled experiments to analyze the impact of UGTO labeling tags and the features that are used in UGTO. Their results are reported in Table 13.

Impact of UGTO Labeling Tags. To analyze the impact of UGTO labeling tags, we replaced them by BIO tags (as well as IOBES tags) and kept other factors unchanged. The BIO and IOBES schemes achieve similar results and we reported the results of the BIO scheme. UGTO$_{w/o}$ performs better than BIO, because the UGTO scheme overcomes the problem of inconsistent tag assignment [81].

Impact of UGTO Pre-tag Features. We remove the UGTO pre-tag features from UGTO$_{w/o}$ to analyze their impact. We can see that UGTO pre-tag features significantly improve the performance, with about absolute 2.0% improvements.

Impact of Word Clusters. Word clusters are helpful in UGTO (about 0.45% improvement) but not significant as their impact in some other works [34, 50, 53, 63]. The reason is that the UGTO pre-tag features play a similar role as word clusters in improving the coverage and connecting words at the abstraction level.

Error Analysis

There is a limitation in UGTO: when extracting named entities from tagged sequence, UGTO might wrongly treat several consecutive entities as a named entity. Comparing Example (3) and (6) in Table 14, for example, UGTO extracts two named entities “Australian” and “Tom Moody” as a named entity “Australian Tom Moody.”

Conclusion

In this paper, we analyzed intrinsic characteristics of time expressions from four diverse datasets and the ones of named entities from two benchmark datasets. According to these characteristics, we designed two learning-based methods under conditional random fields with a new type of constituent-based tagging schemes to extract time expressions and named entities from unstructured text. Our constituent-based tagging schemes overcome the problem of inconsistent tag assignment that is caused by the conventional position-based tagging schemes. Experiments demonstrate that our proposed methods perform either equally with or better than representative state-of-the-art models on time expression extraction and named entity extraction. Experimental results also demonstrate that the joint modeling of named entity extraction and classification does not improve the performance of named entity extraction.

Funding This research was funded by the AME Programmatic Funding (Project No. A18A2b0046) from the Agency for Science, Technology and Research (A*STAR), Singapore.

Compliance with Ethical Standards

Conflict of Interest The authors declare that they have no conflict of interest.

Ethical Approval This article does not contain any studies with human participants or animals performed by any of the authors.

References

1. Alex B, Haddow B, Grover C. Recognising nested named entities in biomedical text. In: Proceedings of the Workshop on BioNLP 2007: Biological, Translational, and Clinical Language Processing; 2007. p. 65–72.

2. Alonso O, Strotgen J, Baeza-Yates R, Gertz M. Temporal information retrieval: challenges and opportunities. In: Proceedings of 1st International Temporal Web Analytics Workshop; 2011. p. 1–8.
3. Angeli G, Manning CD, Jurafsky D. Parsing time: learning to interpret time expressions. In: Proceedings of 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies; 2012. p. 446–55.
4. Angeli G, Uszkoreit J. Language-independent discriminative parsing of temporal expressions. In: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics; 2013. p. 83–92.
5. Bethard S. ClearTK-TimeML: a minimalist approach to TempEval 2013. In: Proceedings of the 7th International Workshop on Semantic Evaluation. Minneapolis: Association for Computational Linguistics; 2013. p. 10–4.
6. Borthwick A, Sterling J, Agichtein E, Grishman R. NYU: description of the MENE named entity system as used in MUC-7. In: Proceedings of the 7th Message Understanding Conference; 1998.
7. Campos R, Dias G, Jorge AM, Jatowt A. Survey of temporal information retrieval and related applications. ACM Comput Surv. 2014;47(2):15:1–41.
8. Chambers N, Wang S, Jurafsky D. Classifying temporal relations between events. In: Proceedings of the ACL on Interactive Poster and Demonstration Sessions. Ann Arbor: Association for computational linguistics; 2007. p. 173–6.
9. Chang AX, Manning CD. SUTime: a library for recognizing and normalizing time expressions. In: Proceedings of 8th International Conference on Language Resources and Evaluation; 2012. p. 3735–40.
10. Chang AX, Manning CD. SUTime: evaluation in TempEval-3. In: Proceedings of the Second Joint Conference on Lexical and Computational Semantics (SEM); 2013. p. 78–82.
11. Chinchor NA. MUC-7 named entity task definition. In: Proceedings of the 7th Message Understanding Conference; 1998.
12. Chinchor NA. Overview of MUC-7/MET-2. In: Proceedings of the 7th Message Understanding Conference; 1998.
13. Collins M, Singer Y. Unsupervised models for named entity classification. In: Proceedings of the 1999 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora. College Park: Association for Computational Linguistics; 1999.
14. Collobert R, Weston J, Bottou L, Karlen M, Kavukcuoglu K, Kuksa PP. Natural language processing (almost) from scratch. J Mach Learn Res. 2011;12:2493–537.
15. Devlin J, Chang M-W, Lee K, Toutanova K. BERT: pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Minneapolis: Association for Computational Linguistics; 2019. p. 4171–86.
16. Do QX, Lu W, Roth D. Joint inference for event timeline construction. In: Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning; 2012. p. 677–87.
17. Doddington G, Mitchell A, Przybocki M, Ramshaw L, Strassel S, Weischedel R. The automatic content extraction (ACE) program tasks, data, and evaluation. In: Proceedings of the 2004 Conference on Language Resources and Evaluation; 2004. p. 1–4.
18. Ferro L, Gerber L, Mani I, Sundheim B, Wilson G. TIDES 2005 standard for the annotation of temporal expressions. MITRE. 2005.
19. Filannino M, Brown G, Nenadic G. ManTIME: temporal expression identification and normalization in the TempEval-3 challenge. In: Proceedings of the 7th International Workshop on Semantic Evaluation; 2013. p. 53–7.
20. Finkel JR, Grenager T, Manning C. Incorporating non-local information into information extraction systems by gibbs sampling. In: Proceedings of the 43nd Annual Meeting of the Association for Computational Linguistics; 2005. p. 363–70.
21. Finkel JR, Manning C. Nested named entity recognition. In: Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing; 2009. p. 141–50.
22. Giuliano C. Fine-grained classification of named entities exploiting latent semantic kernels. In: Proceedings of the Thirteenth Conference on Computational Natural Language Learning. Boulder: Association for Computational Linguistics; 2009. p. 201–9.
23. Grishman R, Sundheim B. Message understanding conference - 6: a brief history. In: Proceedings of the 16th International Conference on Computational Linguistics; 1996.
24. Hacioglu K, Chen Y, Douglas B. Automatic time expression labeling for English and Chinese text. In: Proceedings of the 6th International Conference on Intelligent Text Processing and Computational Linguistics. Mexico City: Springer; 2005. p. 548–59.
25. Hochreiter S, Schmidhuber J. Long short-term memory. Neur Comput. 1997;9:1735–80.
26. Huang Z, Xu W, Yu K. Bidirectional LSTM-CRF models for sequence tagging. 2015.
27. Ji H, Grishman R. Knowledge base population: successful approaches and challenges. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics; 2011. p. 1148–58.
28. Kazama J, Torisawa K. Exploiting wikipedia as external knowledge for named entity recognition. In: Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning. Prague: Association for Computational Linguistics; 2007. p. 698–707.
29. Krallinger M, Leitner F, Rabal O, Vazquez M, Oyarzabal J, Valencia A. Overview of the chemical compound and drug name recognition (CHEMDNER) task. In: BioCreative Challenge Eval Workshop; 2015. p. 2–33.
30. Lafferty J, McCallum A, Pereira F. Conditional random fields: probabilistic models for segmenting and labeling sequence data. In: Proceedings of the 18th International Conference on Machine Learning. Williams College: Morgan Kaufmann Publishers; 2001. p. 281–9.
31. Lample G, Ballesteros M, Subramanian S, Kawakami K, Dyer C. Neural architecture for named entity recognition. In: Proceedings of the 15th Annual Conference of the North American Chapter of the Association for Computational Linguistics; 2016. p. 260–70.
32. Lee K, Artzi Y, Dodge J, Zettlemoyer L. Context-dependent semantic parsing for time expressions. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics. Baltimore: Association for Computational Linguistics; 2014. p. 1437–47.
33. Li J, Cardie C. Timeline generation: tracking individuals on twitter. In: Proceedings of the 23rd International Conference on World Wide Web; 2014. p. 643–52.
34. Liang P. Semi-supervised learning for natural language. Master's Thesis. 2005.
35. Ling W, Dyer C, Black AW, Trancoso I, Fermandez R, Amir S, Marujo L, Luis T. Finding function in form: compositional character models for open vocabulary word representation. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. Lisbon: Association for Computational Linguistics; 2015. p. 1520–30.

36. Ling X, Singh S, Weld DS. Design challenges for entity linking. Trans Assoc Comput Linguist. 2015;3:315–28.
37. Ling X, Weld DS. Fine-grained entity recognition. In: Proceedings of the Twenty-Sixth Conference on Artificial Intelligence. Toronto: AAAI Press; 2012. p. 94–100.
38. Liu L, Shang J, Ren X, Xu FF, Gui H, Peng J, Han J. Empower sequence labeling with task-aware neural language model. In: Proceedings of the 32nd AAAI Conference on Artificial Intelligence. New Orleans: AAAI Press; 2018. p. 5253–60.
39. Liu X, Zhang S, Wei F, Zhou M. Recognizing named entities in tweets. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics; 2011. p. 359–67.
40. Llorens H, Derczynski L, Gaizauskas R, Saquete E. TIMEN: an open temporal expression normalisation resource. In: Proceedings of the 8th International Conference on Language Resources and Evaluation; 2012. p. 3044–51.
41. Llorens H, Saquete E, Navarro B. TIPSem (english and spanish): evaluating CRFs and semantic roles in TempEval-2. In: Proceedings of the 5th International Workshop on Semantic Evaluation; 2010. p. 284–91.
42. Luo G, Huang X, Lin C-Y, Nie Z. Joint named entity recognition and disambiguation. In: Proceedings of the 2005 Conference on Empirical Methods in Natural Language Processing; 2015. p. 879–88.
43. Ma X, Hovy E. End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (volume 1: long papers). Berlin: Association for Computational Linguistics; 2016. p. 1064–74.
44. Ma Y, Cambria E, Gao S. Label embedding for zero-shot fine--grained named entity typing. In: Proceedings of the 26th International Conference on Computational Linguistics; 2016. p. 171–80.
45. Mani I, Verhagen M, Wellner B, Lee CM, Pustejovsky J. Machine learning of temporal relations. In: Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics; 2006. p. 753–60.
46. Mani I, Wilson G. Robust temporal processing of news. In: Proceedings of the 38th annual meeting on association for computational linguistics; 2000. p. 69–76.
47. Maynard D, Tablan V, Ursu C, Cunningham H, Wilks Y. Named entity recognition from diverse text types. In: Proceedings of 2001 Recent Advances in Natural Language Processing Conference; 2001. p. 257–74.
48. Mazur P, Dale R. WikiWars: a new corpus for research on temporal expressions. In: Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing. MIT Stata Center: Association for Computational Linguistics; 2010. p. 913–22.
49. McCallum A, Li W. Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In: Proceedings of the 7th Conference on Computational Natural Language Learning. Edmonton: Association for Computational Linguistics; 2003. p. 188–91.
50. Miller S, Guinness J, Zamanian A. Name tagging with word clusters and discriminative training. In: Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics; 2004.
51. Nadeau D, Sekine S. A survey of named entity recognition and classification. Lingvisticae Investigationes. 2007;30(1):3–26.
52. Nakashole N, Tylenda T, Weikum G. Fine-grained semantic typing of emerging entities. In: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics. Sofia: Association for Computational Linguistics; 2013. p. 1488–97.
53. Owoputi O, O'Connor B, Dyer C, Gimpel K, Schneider N, Smith NA. Improved part-of-speech tagging for online conversational text with word clusters. In: Proceedings of NAACL-HLT 2013; 2013. p. 380–90.
54. Parker R, Graff D, Kong J, Chen K, Maeda K. Engilish gigaword, 5th edn. 2011.
55. Peters ME, Ammar W, Bhagavatula C, Power R. Semi-supervised suquence tagging with bidirectional language models. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics; 2017. p. 1756–65.
56. Poibeau T, Kosseim L. Proper name extraction from non-journalistic texts. Lang Comput. 2001;37:144–57.
57. Pradhan S, Moschitti A, Xue N, Ng HT, Bjorkelund A, Uryupina O, Zhang Y, Zhong Z. Towards robust linguistic analysis using OntoNotes. In: Proceedings of the 7th Conference on Computational Natural Language Learning. Sofia: Association for Computational Linguistics; 2013. p. 143–52.
58. Pradhan SS, Hovy E, Marcus M, Palmer M, Ramshaw L, Weischedel R. Ontonotes: a unified relational semantic representation. In: Proceedings of the 2007 IEEE International Conference on Semantic Computing; 2007. p. 517–26.
59. Pustejovsky J, Castano J, Ingria R, Sauri R, Gaizauskas R, Setzer A, Katz G, Radev D. TimeML: robust specification of event and temporal expressions in text. Direct Question Answer. 2003;3:28–34.
60. Pustejovsky J, Hanks P, Sauri R, See A, Gaizauskas R, Setzer A, Sundheim B, Radev D, Day D, Ferro L, Lazo M. The TIMEBANK corpus. Corpus Linguist. 2003;2003:647–56.
61. Pustejovsky J, Lee K, Bunt H, Romary L. ISO-TimeML: an international standard for semantic annotation. In: Proceedings of the Seventh Conference on International Language Resources and Evaluation (LREC'10); 2010. p. 394–7.
62. Radford A, Narasimhan K, Salimans T, Sutskever I. Improving language understanding by generative pre-training. 2018.
63. Ratinov L, Roth D. Design challenges and misconceptions in named entity recognition. In: Proceedings of the Thirteenth Conference on Computational Natural Language Learning. Boulder: Association for Computational Linguistics; 2009. p. 147–55.
64. Ren X, He W, Qu M, Huang L, Ji H, Han J. AFET: automatic fine-grained entity typing by hierarchical partial-label embedding. In: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing. Austin: Association for Computational Linguistics; 2016. p. 1369–78.
65. Ritter A, Clark S, Mausam, Etzioni O. Named entity recognition in tweets: an experimental study. In: Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing; 2011. p. 1524–34.
66. Sang EFTK, Meulder FD. Introduction to the CoNLL-2003 shared task: language-independent named entity recognition. In: Proceedings of the 7th Conference on Natural Language Learning; 2003. p. 142–7.
67. Sang EFTK, Veenstra J. Representing text chunks. In: Proceedings of the Ninth Conference on European Chapter of the Association for Computational Linguistics; 1999. p. 173–9.
68. Santos CND, Guimaraes V. Boosting named entity recognition with neural character embeddings. In: Proceedings of the 5th Named Entities Workshop. Beijing: Association for Computational Linguistics; 2015. p. 25–33.
69. Silva JFD, Kozareva Z, Lopes JGP. Cluster analysis and classification of named entities. In: Proceedings of the 4th International Conference on Language Resources and Evaluation. Lisbon: European Language Resources Association; 2004. p. 321–4.
70. Steedman M. Surface structure and interpretation. The MIT Press. 1996.

71. Strötgen J, Gertz M. HeidelTime: high quality rule-based extraction and normalization of temporal expressions. In: Proceedings of the 5th International Workshop on Semantic Evaluation (SemEval'10). Stroudsburg: Association for Computational Linguistics; 2010. p. 321–4.
72. Strubell E, Verga P, Belanger D, McCallum A. Fast and accurate entity recognition with iterated dilated convolutions. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing. Copenhagen: Association for Computational Linguistics; 2017. p. 2670–80.
73. Takeuchi K, Collier N. Bio-medical entity extraction using support vector machines. Artif Intell Med. 2005;33(2):125–37.
74. UzZaman N, Allen JF. TRIPS and TRIOS system for TempEval-2: Extracting temporal information from text. In: Proceedings of the 5th International Workshop on Semantic Evaluation; 2010. p. 276–83.
75. UzZaman N, Llorens H, Derczynski L, Verhagen M, Allen J, Pustejovsky J. SemEval-2013 task 1: TempEval-3: Evaluating time expressions, events, and temporal relations. In: Proceedings of the 7th International Workshop on Semantic Evaluation; 2013. p. 1–9.
76. Verhagen M, Gaizauskas R, Schilder F, Hepple M, Katz G, Pustejovsky J. SemEval-2007 task 15: TempEval temporal relation identification. In: Proceedings of the 4th International Workshop on Semantic Evaluation; 2007. p. 75–80.
77. Verhagen M, Mani I, Sauri R, Knippen R, Jang SB, Littman J, Rumshisky A, Phillips J, Pustejovsky J. Automating temporal annotation with TARQI. In: Proceedings of the ACL Interactive Poster and Demonstration Sessions. Ann Arbor: Association for Computational Linguistics; 2005. p. 81–4.
78. Verhagen M, Sauri R, Caselli T, Pustejovsky J. SemEval-2010 task 13: TempEval-2. In: Proceedings of the 5th International Workshop on Semantic Evaluation; 2010. p. 57–62.
79. Wang L-J, Li W-C, Chang C-H. Recognizing unregistered names for mandarin word identification. In: Proceedings of the 14th Conference on Computational Linguistics; 1992. p. 1239–43.
80. Wong K-F, Xia Y, Li W, Yuan C. An overview of temporal information extraction. Int J Comput Process Oriental Lang. 2005;18(2):137–52.
81. Zhong X, Cambria E. Time expression recognition using a constituent-based tagging scheme. In: Proceedings of the 2018 World Wide Web Conference. Lyon: Association for Computing Machinery; 2018. p. 983–92.
82. Zhong X, Sun A, Cambria E. 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. Vancouver: Association for Computational Linguistics; 2017. p. 420–9.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Affiliations

Xiaoshi Zhong[1] · **Erik Cambria**[1] · **Amir Hussain**[2]

Erik Cambria
cambria@ntu.edu.sg

Amir Hussain
a.hussain@napier.ac.uk

[1] School of Computer Science and Engineering,
Nanyang Technological University,
50 Nanyang Avenue, Singapore

[2] School of Computing, Edinburgh Napier University,
Scotland, UK