Time Expression Recognition Using a Time-related Tagging Scheme

Xiaoshi Zhong and Erik Cambria
School of Computer Science and Engineering
Nanyang Technological University, Singapore
{xszhong,cambria}@ntu.edu.sg

ABSTRACT

We find from four datasets that time expressions are formed by loose structure and the words used to express time information can differentiate time expressions from common text. The findings drive us to design a learning method named TOMN to model time expressions. TOMN defines a time-related tagging scheme named TOMN scheme with four tags, namely T, O, M, and N, indicating the constituents of time expression, namely Time token, Modifier, Numeral, and the words Outside time expression. In modeling, TOMN assigns a word with a TOMN tag under conditional random fields with minimal features. Essentially, our constituent-based TOMN scheme overcomes the problem of *inconsistent tag assignment* that is caused by the conventional position-based tagging schemes (e.g., BIO scheme and BILOU scheme). Experiments show that TOMN is equally or more effective than state-of-the-art methods on various datasets, and much more robust on cross-datasets. Moreover, our analysis can explain many empirical results and observations reported in other works about time expression recognition and named entity recognition.

ACM Reference Format:
Xiaoshi Zhong and Erik Cambria. 2018. Time Expression Recognition Using a Time-related Tagging Scheme. In *Proceedings of the 27th International World Wide Web Conference, Lyon, France, April 2018 (WWW),* 10 pages.
https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Time information plays an increasingly important role in data mining, information retrieval, and natural language processing [1, 8]. Researchers from these fields have devoted tremendous effort to specify standards for time expression annotation [14, 18, 30, 32], build annotated corpora for time expression [18, 27, 31], and recognize time expressions from free text [5–7, 11, 42, 43, 45].

We analyze four datasets (i.e., TimeBank [31], Gigaword [28], WikiWars [27], and Tweets [47]) for the characteristics of time expressions and have two findings. First, time expressions are formed by loose structure, with more than 53.5% of distinct time tokens appearing in different positions within time expressions. Second, time tokens can differentiate time expressions from common text; more than 91.8% of time expressions include at least one time token while no more than 0.7% of common text contain time tokens.

WWW, April 2018, Lyon, France

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

1) September/U 2) September/B 2006/L
3) 2006/B September/L 4) 1/B September/I 2006/L

(a) BILOU tag assignment: '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 the finding that time tokens can differentiate 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) TOMN tag assignment: 'September' in different positions within labeled time expressions is consistently assigned with the same tag of T. The consistent tag assignment protects the 'September's predictive power.

Figure 1: BILOU and TOMN tag assignment during training. BILOU scheme is based on the position within the chunk, while TOMN scheme is based on the constituent of the chunk. Here inconsistent tag assignment is defined as that during training, a word is assigned with different tags simply because the word appears in different positions within labeled chunks.[2]

The findings motivate us to design a learning method named TOMN to model time expressions. Specifically, TOMN defines a time-related tagging scheme named TOMN scheme,[1] consisting of four tags, namely T, O, M, and N, indicating the constituents of time expression, namely Time token, Modifier, Numeral, and the words Outside time expression. Time tokens include the words that explicitly express information about time, such as '2006,' 'month,' and 'September.' Modifiers are the words that modify time tokens and appear around them; for example, 'last' modifies 'month' in 'last month.' Numerals are the ordinals and numbers (except the year like '2006'). TOMN models time expressions under conditional random fields (CRFs) [19] with only a kind of pre-tag features and the lemma features; it assigns a word with a TOMN tag.

TOMN scheme can keep the tag assignment consistent during training and therefore overcomes the problem of inconsistent tag assignment. (In a supervised learning procedure, tag assignment occurs in feature extraction during training and in tag prediction. We focus on the training stage to analyze the impact of tag assignment.) The loose structure by which time expressions are formed exhibits in two aspects. First, many time expressions consist 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 '1 September 2006' by a numeral before it and another time token after it. Second, some time expressions

[1] 'TOMN' denotes our method and 'TOMN scheme' denotes the tagging scheme that TOMN defines.
[2] The definition of inconsistent tag assignment can be generalized as that during training, a unit in different labeled instances is assigned with different tags for some reason while the unit should be consistently assigned with the same tag. The unit of interest can be a word, a phrase, a relation, a webpage, or a group of words as a whole, etc.

can change their word order without changing their meanings. For example, 'September 2006' and '2006 September' express the same meaning. The conventional tagging schemes like BILOU [33] are based on *the position within the chunk*, namely a Unit-word chunk, and the Beginning, Inside, and Last word of a multi-word chunk. Under BILOU scheme, a word that appears in different positions within labeled time expressions is assigned with different tags; for example, the above 'September' can be assigned with U, B, L, or I. See Figure 1(a). The inconsistent tag assignment causes difficulty for statistical models to model time expressions. First, inconsistent tag assignment reduces the predictive power of lexicon, and this contradicts the finding that time tokens can differentiate time expressions from common text. Second, inconsistent tag assignment might cause the problem of tag imbalance. Our TOMN scheme instead is based on **the constituent of the chunk** (i.e., Time token, Modifier, and Numeral of time expression) 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 'September' is consistently assigned with T. See Figure 1(b). With consistent tag assignment, TOMN scheme avoids the potential tag imbalance and protects time tokens' predictive power. Besides, TOMN scheme models a word with fewer tags than BILOU scheme does, and therefore reduces the complexity of the trained model.

We evaluate TOMN against five state-of-the-art methods (i.e., HeidelTime [39], SUTime [9], SynTime [47], ClearTK [4], and UWTime [21]) on three datasets (i.e., TE-3 [42], WikiWars [27], and Tweets [47]).[3] Experiments show that TOMN is equally or more effective than the state-of-the-art methods, and much more robust on cross-dataset performance. Experiments also show the advantage of TOMN scheme over the position-based tagging schemes.

In addition, we find that the named entities in CoNLL03 English NER dataset [34] demonstrate common characteristics similar to the time expressions (see Section 6 for details). The finding suggests that the problem of inconsistent tag assignment widely exists in the position-based tagging schemes and that our idea of defining constituent-based tagging scheme should be widely effective for general entities. In the future we will further develop that idea.

To summarize, we make the contributions as follows.

- We analyze four diverse datasets for the characteristics of time expressions and have two important findings about their organization and constituent words.
- We discover a fundamental problem underlying in the position-based tagging schemes: inconsistent tag assignment. To overcome that problem we define a constituent-based tagging scheme to model time expressions. Our method provides an idea to model target entities based on their constituents.
- We conduct experiments on various datasets, and the results show the effectiveness, efficiency, and robustness of our method compared with state-of-the-art methods. The results also show the advantage of our constituent-based tagging scheme over the position-based tagging schemes.
- The analysis in this work can help explain many empirical results and observations reported in other works about time expression recognition and named entity recognition.

[3] We follow [47] not to use the Gigaword dataset in our experiments because its labels are automatically generated by other taggers but not the ground truth.

2 RELATED WORK

Time expression identification aims to automatically identify time expressions from free text and it can be divided into two subtasks, namely recognition and normalization. In this paper we focus on the recognition. Methods for time expression recognition can be categorized into rule-based methods and learning-based methods.

Rule-based Methods. Rule-based methods like TempEx, GUTime, HeidelTime, and SUTime mainly handcraft deterministic rules to identify time expressions. TempEx and GUTime use both handcrafted rules and machine-learnt rules to resolve time expressions [26, 44]. HeidelTime manually designs rules with time resources to recognize time expressions [39]. SUTime designs deterministic rules at three levels (i.e., individual word level, chunk level, and time expression level) for time expression recognition [9]. A recent type-based time tagger, SynTime, designs general heuristic rules with a token type system to recognize time expressions [47].

TOMN uses the token regular expressions, similar to SUTime [9] and SynTime [47], and further groups them into three token types, similar to SynTime. While SynTime further defines 21 token types for the constituent words of time expression, TOMN uses the three general token types that are helpful for a learning method to connect the words with low frequencies to the words with high frequencies. Moreover, TOMN leverages the statistical information from entire corpus to improve the precisions and can alleviate the deterministic role of deterministic rules and heuristic rules.

Learning-based Methods. Learning-based methods in TempEval series mainly extract features from text (e.g., character features, word features, syntactic features, and semantic features), and on the features apply statistical models (e.g., CRFs, logistic regression, maximum entropy, Markov logic network, and support vector machines) to model time expressions [4, 15, 23, 41]. Besides the standard methods, Angeli et al., and Angeli and Uszkoreit exploit an EM-style approach with compositional grammar to learn latent time parsers [2, 3]. Lee et al. leverage combinatory categorial grammar (CCG) [36] and define a collection of lexicon with linguistic context to model time expressions [21].

Unlike [4, 15, 17, 23, 41] which use the 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; which can enhance the impact of the significant features and reduce the impact of the insignificant features. Unlike [2, 3, 21] which use fixed structure information, TOMN uses the loose structure information by grouping the constituent words of time expression under three token types, which 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. Methods for time expression normalization are mainly based on rules [4, 15, 23, 39, 41, 44]. Since the rule methods are highly similar, Llorens et al. suggest to construct a large shared knowledge base for public use [22]. Lee et al., Angeli et al., and Angeli and Uszkoreit instead combine grammar rules and machine learning techniques to normalize time expressions [2, 3, 21]. TOMN focuses on the recognition and leaves the normalization to those highly similar rule methods or to the future work.

Table 1: Dataset statistics ('timex' stands for time expression.)

Dataset	#Documents	#Words	#Timex
TimeBank	183	61,418	1,243
Gigaword	2,452	666,309	12,739
WikiWars	22	119,468	2,671
Tweets	942	18,199	1,127

Table 2: Percentage of distinct 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

3 TIME EXPRESSION ANALYSIS

Datasets. We analyze the time expressions from four datasets: TimeBank, Gigaword, WikiWars, and Tweets. TimeBank is a benchmark dataset and consists of 183 news articles [31]. Gigaword consists of 2,452 news articles with automatically annotated time expressions [28]. WikiWars is constructed by collecting articles about war from Wikipedia [27]. Tweets consists of 942 tweets collected from Twitter [47]. The four datasets cover comprehensive data (TimeBank, Gigaword, and Tweets) and specific domain data (WikiWars) as well as formal text (TimeBank, Gigaword, and WikiWars) and informal text (Tweets). Table 1 summarizes the statistics of the four datasets.

3.1 Findings

Although the four datasets differ in source, domain, corpus size, and text type, their time expressions demonstrate some common characteristics. We find such two common characteristics of time expressions about their organization and constituent words.

Finding 1. *Time expressions are formed by loose structure; more than 53.5% of time tokens appear in different positions within time expressions.*

We find that time expressions are formed by loose structure and the loose structure exhibits in two aspects. First, many time expressions consist 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 '1 September 2006' by a numeral before it and another time token after it. Second, some time expressions can change their word order without changing their meanings. For example, 'September 2006' can be written as '2006 September' with the same meaning. From the point of view of the position within time expressions, the 'September' may appear as the (i) beginning or (ii) inside word of time expression when time expressions are modeled by BIO scheme; or it may appear as (1) a unit-word time expression, or the (2) beginning, (3) inside, (4) last word of a multi-word time expression when time expressions are modeled by BILOU scheme.

Table 3: Percentage of time expression's constituents 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 2 reports the percentage of distinct time tokens and modifiers that appear in different positions within time expressions. 'Distinct' here means ignoring the word variants and frequencies 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 BIO scheme and at least two of the four different positions under BILOU scheme. For each dataset, under BIO scheme, more than 53.5% of distinct time tokens appear in different positions; and under BILOU scheme, more than 61.4% of distinct time tokens appear in different positions. The number of modifiers that appear in different positions is more than 27.5%. When BIO or BILOU scheme is applied 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 Section 4.1 for details).

Finding 2. *Time tokens can differentiate time expressions from common text while modifiers and numerals cannot.*

Table 3 reports the percentage of time expression's constituent words appearing 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 Equation (1) and P_{text} is by Equation (2), where $T \in \{time\ \ token,\ \ modifier,\ \ numeral\}$.

$$P_{timex}(T) = \frac{\#timex\ \ that\ \ contain\ \ T}{\#total\ \ timex} \tag{1}$$

$$P_{text}(T) = \frac{\#tokens\ \ that\ \ are\ \ T}{\#total\ \ tokens} \tag{2}$$

From the second column of Table 3 we can see 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. [47]. (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 differentiate 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 differentiate time expressions from common text.

Looking at the Tweets dataset, we can see that the P_{timex} of the time tokens (96.0%) is relatively high while the P_{timex} of the modifiers (21.4%) and numerals (18.8%) are much lower than the ones of other datasets. This indicates that in Twitter people tend to use time expressions with fewer modifiers and numerals.

3.2 Properties

In addition to our findings, Zhong et al. also conclude some characteristics about time expressions from the same four datasets [47]. The characteristics concluded by Zhong et al. are helpful for our analysis, so we briefly describe them here and report them as time expressions' properties.[4]

Property 1. *"Time expressions are very short. More than 80% of time expressions contain no more than three words and more than 90% contain no more than four words." [47]*

Time expressions across different datasets follow a similar length distribution and on average, time expressions contain two words. For the one-word time expressions, the percentage of which in TimeBank is 40.3%; in Gigaword the percentage is 53.9%; in WikiWars it is 36.2%; and in Tweets it is 62.9%.

Property 2. *"Only a small group of time-related keywords are used to express time information." [47]*

There are only about 70 distinct time tokens used in individual dataset, regardless of the corpus sizes and the number of time expressions, and only 123 distinct time tokens across different datasets. That means time expressions are highly overlapped at their time tokens within individual dataset and across different datasets.

Property 3. *"POS information could not distinguish time expressions from common words, but within time expressions, POS tags can help distinguish their constituents." [47]*

Among the top 40 part-of-speech (POS) tags in time expressions (10×4 datasets), 37 tags whose percentage over the corresponding tags of the whole text is lower than 20%.

4 TOMN: TIME-RELATED TAGGING SCHEME

Figure 2 shows the overview of TOMN, including three parts: TOMN scheme, TmnRegex, and time expression recognition. TOMN scheme is a time-related tagging scheme with four tags (top-left side). TmnRegex is a set of regular expressions for time-related words (bottom-left side). Time expressions are modeled under a CRFs framework with the help of TmnRegex and TOMN scheme (right-hand side).

4.1 TOMN Scheme

Finding 1 shows that time expressions are formed by loose structure and suggests us to explore an appropriate scheme to model time expressions. We define a time-related tagging scheme, namely TOMN scheme, consisting of four tags: T, O, M, and N; they indicate the constituents of time expression, namely Time token, Modifier, and Numeral, and the words Outside time expression.

Conventional tagging schemes like BIO[5] [35] and BILOU[6] [33] are based on *the position within the chunk*. BIO refers to the Beginning, Inside, and Outside of a chunk; BILOU refers to a Unit-word chunk, and the Beginning, Inside, Last word of a multi-word chunk. TOMN scheme instead is based on **the constituent of the chunk**, indicating the constituents of time expression. Following we use BILOU scheme, the newer version, as representative of the conventional position-based tagging schemes for analysis.

[4]One of Zhong et al. concluded properties (i.e., Finding 2 in [47]) is incorporated into our Finding 2.
[5]The BIO scheme in this paper denotes the standard IOB2 scheme described in [35].
[6]The BILOU scheme is also widely known as the IOBES scheme.

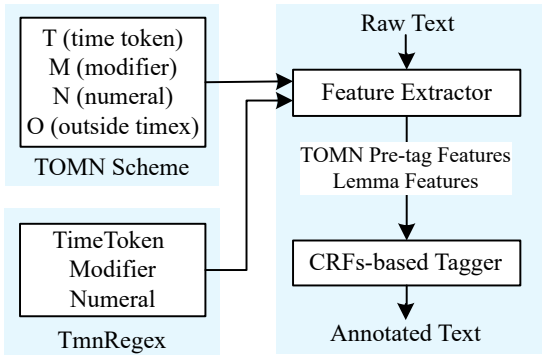


Figure 2: Overview of TOMN. Top-left side shows the TOMN scheme, consisting of four tags. Bottom-left side is the TmnRegex, a set of regular expressions for time-related words. Right-hand side shows the time expression modeling, with the help of TmnRegex and TOMN scheme.

Using BILOU scheme for time expression recognition leads to inconsistent tag assignment. (A typical supervised learning procedure involves tag assignment in two stages; one is in feature extraction during training and the other is in tag prediction. We focus on the training stage to analyze the impact of tag assignment under different kinds of tagging schemes.) Finding 1 shows that time expressions are formed by loose structure, exhibiting in loose collocations and exchangeable order. Under BILOU scheme, both of 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 training data. During feature extraction, they are assigned with the BILOU tags as 'September/U,' 'September/B 2006/L,' '2006/B September/L,' and '1/B September/I 2006/L' (see Figure 1(a)). The four 'September' have the same word (the word itself) and express the same meaning (the ninth month of the year), but because they appear in different positions within 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. First, inconsistent tag assignment reduces the predictive power of lexicon. A word assigned with different tags causes confusion to model the word. If a word is assigned with different tags in equal number, then the word itself cannot provide any information useful 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 Finding 2 which shows that time tokens can differentiate time expressions from common text. Second, inconsistent tag assignment may cause another problem: tag imbalance. If a tag of a word dominates in 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 the trained model to correctly predict the data collected from the style of 'September 1, 2006' in which 'September' should be predicted as B.

TOMN scheme overcomes the problem of inconsistent tag assignment and its derivational problems. 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 expression (see Section 4.2) and same word plays same constituent role in time expressions, therefore, the same word is assigned with the same TOMN tag, regardless of its frequency and its positions within time expressions. For example, 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 Figure 1(b)). The four 'September' have the same tag of 'T' and statistical models need only to model them as 'T,' without any confusion. With consistent tag assignment, TOMN scheme avoids the potential tag imbalance and protects time tokens' predictive power.

Besides, TOMN scheme models a word by fewer tags than BILOU scheme does. BILOU scheme typically models a time token by four tags (i.e., U, B, L, or I) and models a modifier/numeral by five tags (i.e., U, B, L, I, or O), while TOMN scheme models a time token by one tag (i.e., T) and models a modifier/numeral by two tags (i.e., M or N if the modifier/numeral appears inside time expressions and O if it appears outside time expressions). Compared with BILOU scheme, TOMN scheme reduces the complexity of the trained model.

4.2 TmnRegex

Property 2 indicates a small group of words that are used in time expressions. TOMN uses three time-related token types, namely time token, modifier, and numeral, to group those words. The three token types are consistent with three of the above four tags (i.e., T, M, and N), and are similar to the ones defined in SynTime [47].

Time tokens explicitly express information about time, such as year (e.g., '2006'), month (e.g., 'September'), date (e.g., '2006-09-01'), and time units (e.g., 'month'). Modifiers are the words that modify time tokens and appear around them; for example, the two modifiers 'the' and 'last' modify the time token 'month' in 'the last month.' Numerals include ordinals and numbers, except those that are recognized as year (e.g., '2006'). Token types are defined on tokens themselves and are not necessarily relevant to their context. For example, '2006' alone expresses time information, so it is a time token; although the '1' in '1 September 2006' implies the day, itself alone does not express time information, so it is a numeral.

The three token types with the words they group form a set of token regular expressions, and the set of token regular expressions is denoted by TmnRegex. TmnRegex is constructed by importing token regular expressions for its time token, modifier, and numeral from SUTime.[7] Like SynTime [47], TmnRegex collects from SUTime only the regular expressions at the level of token. TmnRegex contains only 115 distinct time tokens, 57 modifiers, and 58 numerals, without counting the words with changing digits.

4.3 Time Expression Recognition

Time expression recognition mainly consists of two stages: (1) feature extraction and (2) model learning and tagging. When extracting features we set a guideline that the features should be able to help differentiate time expressions from common text and help build connections among time expressions.

[7] https://github.com/stanfordnlp/CoreNLP/tree/master/src/edu/stanford/nlp/time/rules

4.3.1 Feature Extraction. The features we extract include two kinds: TOMN pre-tag features and lemma features. When extracting features we use w_i to denote the i-th word in text.

TOMN Pre-tag Features. Finding 2 shows that time tokens can differentiate time expressions from common text while modifiers and numerals cannot, therefore, how to leverage the information of these words becomes crucial. In our consideration, they are treated as pre-tag features using the TOMN scheme. Specifically, a time token is pre-tagged by the tag of T, a modifier is pre-tagged by M, and a numeral is by N; other common words are by O. The assignment of pre-tags is conducted by simply looking up the words at TmnRegex.

The last four columns of Table 3 indicate that modifiers and numerals constantly appear in time expressions and in common text. To distinguish where a modifier or numeral appears, we conduct a checking for the modifiers and numerals (the words with pre-tag of M or N (M/N)) to record whether they directly or indirectly modify any time token. 'Indirectly' here means a M/N together with other M/N 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 the time tokens. For each time token we search its left side without exceeding its previous time token and search its right side without exceeding its following time token. When searching a side of a time token, if encounter a M/N, then record the M/N and continue searching; if encounter a word that is not M/N, then stop the searching for this side of this time token. After the checking, those M/N that modify time tokens are recorded; for example, the 'two' in 'two months' is recorded while in 'two apples' is not. The checking is treated as a feature for modeling.

For the pre-tag features we extract them in a 5-word window of w_i, namely the pre-tags of w_{i-2}, w_{i-1}, w_i, w_{i+1}, and w_{i+2}. For the checking feature we consider only the current word w_i.

In the training phase we consider the TOMN pre-tag features for only the words within labeled time expressions. In the test phase the TOMN pre-tag features are extracted for all the words in text.

Lemma Features. The lemma features include the word shape of w_i in a 5-word window, namely the lemmas of 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.

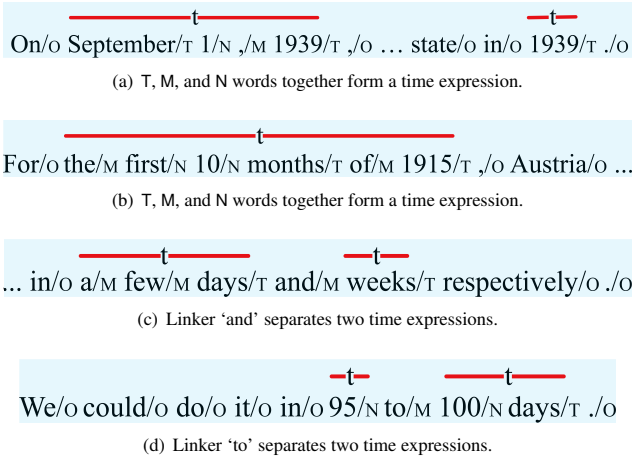
The lemma features are extracted for all the words in text in both of the training phase and the test phase.

We do not consider the features of characters nor word variants because they cannot help build connections among time expressions but hurt the modeling; for example, 'Sept.' and 'September' express the same thing but computer does not treat them as the same thing.

We also do not consider the POS features nor other syntactic features. Property 3 indicates that POS tags cannot help differentiate time expressions from common text, and experiments confirm that POS tags do not improve the performance. On the other hand, Finding 1 shows that time expressions are formed by loose structure, which together with Property 3 suggests that other syntactic features (e.g., syntactic dependency) that rely on POS tags and fixed linguistic structure cannot provide extra useful information for a CRFs-based

Table 4: Features of word w_i used for modeling

1. TOMN pre-tags of w_i in a 5-word window, namely the pre-tags 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 directly or indirectly modifies any time token
3. Lemmas of w_i in a 5-word window



(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.

(d) Linker 'to' separates two time expressions.

Figure 3: Examples of time expression extraction. The label t indicates time expression.

learning method, which already considers the dependency, to differentiate time expressions from common text. We therefore do not use those syntactic features in our model.

Feature Values. For the TOMN pre-tag features we separate the features with binary values. The theory of scales of measurement suggests that non-ordinal attributes should be transformed to separate dimensions [37]. TOMN pre-tags are non-ordinal, therefore, each of TOMN pre-tags as well as the checking feature works as a separate feature. For the lemma features we follow the traditional use to incorporate multiple values under a feature.

Table 4 summarizes the features that are used for time expression modeling. Typically up to 11 features are extracted for a word.

4.3.2 Model Learning and Tagging. TOMN models time expressions through the feature vectors under a CRFs framework [19]. In implementation, we use Stanford Tagger[8] for the lemma features and use CRFSuite[9] with default setting for model learning. For the tagging, each word is assigned with one of TOMN tags, namely T, O, M, or N. Note that the TOMN scheme is used in feature extraction as a kind of features and in sequence tagging as labeling tags.

Time Expression Extraction. After sequence tagging, those T, M, and N words (or non-O words) that appear together are extracted as a time expression. See Figure 3(a) and 3(b). A special kind of modifiers, i.e., the linker 'to,' '-,' 'or,' and 'and' separates the non-O words into parallel time expressions. See Figure 3(c) and 3(d).

[8]http://nlp.stanford.edu/software/tagger.shtml
[9]http://www.chokkan.org/software/crfsuite/

5 EXPERIMENTS

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

5.1 Experiment Setting

Datasets. The three datasets used in our experiments are TE-3, WikiWars, and Tweets. TE-3 uses the TimeBank corpus as training set and the Platinum corpus as test set. TimeBank consists of 183 news articles and Platinum consists of 20 new articles; they are comprehensive corpora in formal text and described in TempEval-3 [42]. WikiWars is a specific domain dataset in formal text, consisting of 22 English Wikipedia articles about famous wars [27]. Tweets is a comprehensive dataset in informal text, with 942 tweets that contain time expressions [47]. For WikiWars and Tweets, we follow previous works [21, 47] to set their training sets and test sets. The performance of a method (rule-based or learning-based) on a dataset is reported on the dataset's test set.

Baseline Methods. We evaluate TOMN against five state-of-the-art methods, including three rule-based methods, HeidelTime, SUTime, and SynTime, and two learning-based methods, ClearTK and UWTime. HeidelTime [39] and SUTime [9] use predefined deterministic rules and achieve the best results in relaxed match while ClearTK [4] uses a CRFs framework with BIO scheme and achieves the best result in strict match in TempEval-3 [42]. UWTime uses combinatory categorial grammar (CCG) to predefine linguistic structure for time expressions and achieves better results than HeidelTime on TE-3 and WikiWars datasets [21]. SynTime uses a set of general heuristic rules and achieves good results on TE-3, WikiWars, and Tweets datasets [47]. SynTime has two versions, a basic version and an expanded version. Because the expanded version requires extra manual annotation for each dataset, for fair comparison, we use the basic version to ensure that the token regular expressions used in SynTime and TOMN are comparable.

Evaluation Metrics. We report the results in *Precision*, *Recall*, and F_1 under *strict match* and *relaxed match* by following the TempEval-3 with their evaluation toolkit[10] [42]. Strict match means exact match between the extracted time expressions and the ground truth while relaxed match means that there exists overlap between them.

5.2 Experiment Results

Table 5 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 results, and much better than other baselines which achieve at most 4 best or second best results. For each measure, TOMN achieves either best results or comparable results to the best 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.

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

[10]http://www.cs.rochester.edu/~naushad/tempeval3/tools.zip

Table 5: Performance of TOMN and baseline methods. We make bold the best results and underline the second best results. Some results are reported directly from the sources indicated by the references where the results are publicly available.

Dataset	Method	Strict Match			Relaxed Match		
		$Pr.$	$Re.$	F_1	$Pr.$	$Re.$	F_1
TE-3	HeidelTime[40]	83.85	78.99	81.34	93.08	87.68	90.30
	SUTime[10]	78.72	80.43	79.57	89.36	91.30	90.32
	SynTime[47]	91.43	**92.75**	**92.09**	94.29	**95.65**	**94.96**
	ClearTK[4]	85.90	79.70	82.70	93.75	86.96	90.23
	UWTime[21]	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[38]	**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[47]	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[21]	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[47]	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

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 [47].

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.

Table 6: Cross-dataset performance on the test set of TE-3. 'Training' indicates the dataset whose training set is used for training. Color background indicates the single-dataset results.

Training	Method	Strict Match			Relaxed Match		
		$Pr.$	$Re.$	F_1	$Pr.$	$Re.$	F_1
TE-3	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	ClearTK	65.67	63.77	64.71	87.31	84.78	86.03
	UWTime	76.92	72.46	74.63	88.46	83.33	85.82
	TOMN	**84.06**	**84.06**	**84.06**	**93.48**	**93.48**	**93.48**
Tweets	ClearTK	72.59	71.01	71.79	**93.33**	91.30	92.31
	UWTime	80.00	72.46	76.05	92.80	84.06	88.21
	TOMN	**85.42**	**89.13**	**87.23**	91.67	**95.65**	**93.62**

Table 7: Cross-dataset performance on the test set of WikiWars.

Training	Method	Strict Match			Relaxed Match		
		$Pr.$	$Re.$	F_1	$Pr.$	$Re.$	F_1
TE-3	ClearTK	74.38	60.76	66.89	**97.54**	79.68	87.71
	UWTime	**87.01**	**79.34**	**83.00**	96.07	87.60	91.64
	TOMN	82.18	75.65	79.07	96.26	**87.93**	**91.90**
WikiWars	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	ClearTK	57.75	54.73	56.20	91.93	87.12	**89.46**
	UWTime	**80.28**	62.81	**70.48**	**94.37**	73.83	82.84
	TOMN	60.29	**66.00**	63.02	84.74	**92.76**	88.57

Table 8: Cross-dataset performance on the test set of Tweets.

Training	Method	Strict Match			Relaxed Match		
		$Pr.$	$Re.$	F_1	$Pr.$	$Re.$	F_1
TE-3	ClearTK	81.16	47.26	59.73	**97.10**	56.54	71.47
	UWTime	89.66	65.82	75.91	94.83	69.62	80.29
	TOMN	**92.92**	**88.61**	**90.71**	96.90	**92.41**	**94.60**
WikiWars	ClearTK	72.48	45.57	55.96	95.30	59.92	73.58
	UWTime	**87.43**	61.60	72.28	**95.81**	67.61	79.21
	TOMN	85.00	**86.08**	**85.53**	93.75	**94.94**	**94.34**
Tweets	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**

5.2.2 Cross-dataset Performance. We conduct cross-dataset experiments and compare TOMN with the learning-based methods that require training. Specifically, a method is trained on the training set of one dataset and then tested on the test sets of other datasets. Since the three datasets used in our experiments are quite diverse, the cross-dataset experiments on the three datasets therefore can evaluate a learning method's robustness. Table 6 reports the cross-dataset performance on TE-3; Table 7 reports the performance on WikiWars; and Table 8 on Tweets. For comparison, Table 6, 7, and 8 also report the performance on single-dataset; 'single-dataset' here means the training set and the test set belong to the same dataset. The single-dataset results are reported directly from Table 5.

On (the test set of) TE-3, TOMN achieves at least 84.0% in strict F_1 and at least 93.4% in relaxed F_1. (See the rows of WikiWars

and Tweets in Table 6.) On Tweets, TOMN achieves at least 85.5% and 94.3% respectively. (See the rows of TE-3 and WikiWars in Table 8.) It significantly outperforms ClearTK and UWTime. On WikiWars, TOMN achieves comparable results with ClearTK and UWTime in relaxed match but performs worse than UWTime in strict match; especially when trained on Tweets, TOMN achieve only 63.0% in strict F_1, 7.5% lower than the one of UWTime. (See the rows of TE-3 and Tweets in Table 7.) Tweets contains many short time expressions (62.9% one-word time expressions) and uses fewer modifiers and numerals in time expressions while WikiWars includes quite a few long time expressions (only 36.2% one-word time expressions) and some descriptive time expressions. For these reasons, TOMN trained on Tweets cannot fully recognize the long and descriptive time expressions in WikiWars. UWTime instead predefines linguistic structure, which contributes significantly to exact recognition of those long and descriptive time expressions.

Look at the single-dataset and cross-dataset performance in relaxed match. TOMN achieves similar performance, regardless of which dataset it is trained on; in relaxed F_1, TOMN achieves about 93.9% on TE-3, about 91.6% on WikiWars, and about 94.8% on Tweets. By contrast, ClearTK and UWTime perform well on single-dataset but much worse on cross-dataset; especially on Tweets, their relaxed F_1 drops from at least 87.0% when trained on Tweets to at most 80.3% when trained on other datasets. This indicates that TOMN is much more robust than ClearTK and UWTime.

The robustness of TOMN can be explained by Finding 2 and Property 2. Finding 2 indicates that time tokens are capable of predicting time expressions and Property 2 indicates that time expressions highly overlap at their time tokens within individual dataset and across different datasets. That means, the time tokens from one dataset can help recognize the time tokens from other datasets. Therefore, in terms of relaxed match, the cross-dataset performance should be comparable to the single-dataset performance.

5.2.3 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 9.

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 the lemma features that are used in TOMN.[11]

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 systems are reported as 'BIO' and 'BILOU' in Table 9. 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 (see Property 1); 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

Table 9: Impact of factors. '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.

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**

[11]The BIO and BILOU schemes can be extracted with other features, but we using the BIO and BILOU schemes here is to conduct controlled experiments to analyze the impact of TOMN scheme as labeling tags. So we extract the same features in TOMN to the BIO and BILOU schemes

Table 10: Running time that TOMN and the learning-based methods cost to go through a whole process (unit: seconds)

Method	TE-3	WikiWars	Tweets
ClearTK	152	223	86
UWTime	864	1,050	160
TOMN	36	48	42

time tokens (see Property 1); 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. The reason is that the TOMN pre-tag features provide useful information to recognize time tokens. The strict match on TE-3 and WikiWars decreases dramatically, which indicates that the 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.

5.2.4 Computational Efficiency. HeidelTime and SUTime run nearly in real time; SynTime in real time. Table 10 reports the running time that TOMN and the learning-based methods cost to go through a whole process (including training and test) on the three datasets on a Mac OS laptop (1.4GHz Processor and 8GB Memory). We can see that TOMN is much more efficient than ClearTK and UWTime. Consider only the test, TOMN runs in real time.

6 DISCUSSION

The analysis of time expressions can explain a lot of empirical observations reported in other works about time expression recognition. UzZaman et al. report that using an extra large dataset does not improve the performance [42]; Bethard reports that using TimeBank alone performs better than using TimeBank and AQUAINT datasets together [4]; and Filannino et al. report that features of gazetteers, shallow parsing, and propositional noun phrases do not contribute significant improvement [15]. These observations can be explained by the findings and properties illustrated in Section 3. Finding 1, 2 and Property 2, 3 together suggest that additional gazetteers, large corpus, and more datasets provide no further useful information but repeated time tokens and their loose combinations, and that those syntactic features cannot provide extra useful information for a CRFs-based learning method to model time expressions.

The analysis of tagging schemes can explain the empirical observations in other works about the impact of the BIO (or IOB2) and BILOU (or IOBES) schemes in named entity recognition (NER). Ratinov and Roth report that BILOU scheme outperforms BIO scheme on MUC-7 and CoNLL03 NER datasets [33]; Dai et al. report that IOBES scheme performs better than IOB2 scheme in drug name recognition [13]. When looking at their results, however, we find that the improvements are rather slight, most of them are within 1%; and in some cases, BIO scheme performs better than BILOU scheme. Lample et al. confirm that they do not observe significant improvement of IOBES scheme over IOB2 scheme on CoNLL03 NER dataset [20]. These observations can be explained by our analysis of tagging schemes in Section 4.1 and 5.2.3. Basically, BIO and BILOU schemes are based on the position within the chunk and implicitly assume that target entities should be formed by fixed structure and even fixed collocations. But entities as part of language are actually flexible. When applied to entity recognition, the BIO and BILOU schemes would more or less suffer from the problem of *inconsistent tag assignment*. We analyze the named entities in CoNLL03 (English NER) dataset [34] as example. We find that for each of the CoNLL03 training, development, and test sets, more than 53.7% of distinct words appear in different positions within named entities; more than 93.7% of named entities each has at least one word not appearing in common text; and the named entities on average contain 1.45 words, with 63.2% one-word named entities. The percentage 53.7% is similar to the one of distinct time tokens in time expressions (53.5%; see Finding 1); the 93.7% is similar to the one of time expressions that contain time tokens (91.8%; see Finding 2); and the length distribution is similar to the one of time expressions in Tweets (see Property 1). That means named entities demonstrate common characteristics similar to time expressions. This suggests that when modeling named entities, like modeling time expressions, the BIO and BILOU schemes would either suffer from the problem of inconsistent tag assignment or be roughly equivalent if they were reduced to the constituent-based BO and UO schemes. In either case, the difference between the two schemes impacts slightly.

When analyzing the CoNLL03 dataset (which contains four entity types: PER, LOC, ORG, and MISC), we find that some named entities are annotated with different entity types. In the training set, for example, 'Wimbledon' is annotated 4 times with LOC, 8 times with ORG, and 18 times with MISC. Such named entities (including several polysemy) in the training set, development set, and test set reach relatively high percentage of respective 6.9%, 4.4%, and 6.5%. The inconsistent annotation and inconsistent tag assignment may be able to explain why most state-of-the-art NER methods achieve the F_1 at around 94.5% on the development set and around 91.5% on the test set [12, 20, 24, 25, 29, 33, 46], and why more than 10 years' effort improves the F_1 by only 0.8% on the development set (from 2003's 93.9% [16] to current 94.7% [25]) and by only 2.9% on the test set (from 2003's 88.7% [16] to current 91.6% [12]). The two inconsistency problems seem to limit the upper bound of the performance on development set at near 94.5% and the one on test set at near 91.5%. This suggests that to further improve the performance on current CoNLL03 dataset with current methods is difficult and unreliable. Instead of continuing to fine-tune current methods, we should try to correct the inconsistent annotation and address the problem of inconsistent tag assignment.

REFERENCES

[1] Omar Alonso, Jannik Strotgen, Ricardo Baeza-Yates, and Michael Gertz. 2011. Temporal Information Retrieval: Challenges and Opportunities. In *Proceedings of 1st International Temporal Web Analytics Workshop*. 1–8.

[2] Gabor Angeli, Christopher D. Manning, and Daniel Jurafsky. 2012. 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*. 446–455.

[3] Gabor Angeli and Jakob Uszkoreit. 2013. Language-Independent Discriminative Parsing of Temporal Expressions. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*. 83–92.

[4] Steven Bethard. 2013. ClearTK-TimeML: A minimalist approach to TempEval 2013. In *Proceedings of the 7th International Workshop on Semantic Evaluation*. 10–14.

[5] Steven Bethard, Leon Derczynski, Guergana Savova, James Pustejovsky, and Marc Verhagen. 2015. SemEval-2015 Task 6: Clinical TempEval. In *Proceedings of the 9th International Workshop on Semantic Evaluation*. 806–814.

[6] Steven Bethard, Guergana Savova, Wei-Te Chen, Leon Derczynski, James Pustejovsky, and Marc Verhagen. 2016. SemEval-2016 Task 12: Clinical TempEval. In *Proceedings of SemEval-2016*. 1052–1062.

[7] Steven Bethard, Guergana Savova, Martha Palmer, and James Pustejovsky. 2017. SemEval-2017 Task 12: Clinical TempEval. In *Proceedings of the 11th International Workshop on Semantic Evaluation*. 565–572.

[8] Ricardo Campos, Gael Dias, Alipio M. Jorge, and Adam Jatowt. 2014. Survey of temporal information retrieval and related applications. *Comput. Surveys* 47, 2 (2014), 15:1–41.

[9] Angel X. Chang and Christopher D. Manning. 2012. SUTime: A Library for Recognizing and Normalizing Time Expressions. In *Proceedings of 8th International Conference on Language Resources and Evaluation*. 3735–3740.

[10] Angel X. Chang and Christopher D. Manning. 2013. SUTime: Evaluation in TempEval-3. In *Proceedings of second Joint Conference on Lexical and Computational Semantics (SEM)*. 78–82.

[11] Nancy A. Chinchor. 1997. MUC-7 Named Entity Task Definition. In *Proceedings of the 7th Message Understanding Conference*, Vol. 29.

[12] Jason P.C. Chiu and Eric Nichols. 2016. Named Entity Recognition with Bidirectional LSTM-CNNs. *Transactions of the Association for Computational Linguistics* 4 (2016), 357–370.

[13] Hong-Jie Dai, Po-Ting Lai, Yung-Chun Chang, and Richard Tzong-Han Tsai. 2015. Enhancing of chemical compound and drug name recognition using representative tag scheme and fine-grained tokenization. *Journal of Cheminformatics* 7.S1, S14 (2015), 1–10.

[14] Lisa Ferro, L. Gerber, I. Mani, Beth Sundheim, and G. Wilson. 2005. *TIDES 2005 Standard for the Annotation of Temporal Expressions*. Technical Report. MITRE.

[15] Michele Filannino, Gavin Brown, and Goran Nenadic. 2013. ManTIME: Temporal expression identification and normalization in the TempEval-3 challenge. In *Proceedings of the 7th International Workshop on Semantic Evaluation*. 53–57.

[16] Radu Florian, Abe Ittycheriah, Hongyan Jing, and Tong Zhang. 2003. Named Entity Recognition through Classifier Combination. In *Proceedings of the 7th Conference on Natural Language Learning*. 168–171.

[17] Kadri Hacioglu, Ying Chen, and Benjamin Douglas. 2005. Automatic Time Expression Labeling for English and Chinese Text. In *Proceedings of the 6th International Conference on Intelligent Text Processing and Computational Linguistics*. 548–559.

[18] William F. Styler IV, Steven Bethard, Sean Finan, Martha Palmer, Sameer Pradhan, Piet C de Groen, Brad Erickson, Timothy Miller, Chen Lin, Guergana Savova, and James Pustejovsky. 2014. Temporal Annotation in the Clinical Domain. *Transactions of the Association for Computational Linguistics* 2 (2014), 143–154.

[19] John Lafferty, Andrew McCallum, and Fernando Pereira. 2001. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *Proceedings of International Conference on Machine Learning*. 281–289.

[20] Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural Architecture for Named Entity Recognition. In *Proceedings of the 15th Annual Conference of the North American Chapter of the Association for Computational Linguistics*. 260–270.

[21] Kenton Lee, Yoav Artzi, Jesse Dodge, and Luke Zettlemoyer. 2014. Context-dependent Semantic Parsing for Time Expressions. In *Proceedings of the 52th Annual Meeting of the Association for Computational Linguistics*. 1437–1447.

[22] Hector Llorens, Leon Derczynski, Robert Gaizauskas, and Estela Saquete. 2012. TIMEN: An Open Temporal Expression Normalisation Resource. In *Proceedings of International Conference on Language Resources and Evaluation*. 3044–3051.

[23] Hector Llorens, Estela Saquete, and Borja Navarro. 2010. TIPSem (English and Spanish): Evaluating CRFs and Semantic Roles in TempEval-2. In *Proceedings of the 5th International Workshop on Semantic Evaluation*. 284–291.

[24] Gang Luo, Xiaojiang Huang, Chin-Yew Lin, and Zaiqing Nie. 2015. Joint Named Entity Recognition and Disambiguation. In *Proceedings of the 2005 Conference on Empirical Methods in Natural Language Processing*. 879–888.

[25] Xuezhe Ma and Eduard Hovy. 2016. 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)*. 1064–1074.

[26] Inderjeet Mani and George Wilson. 2000. Robust Temporal Processing of News. In *Proceedings of the 38th annual meeting on Association for Computational Linguistics. Association for Computational Linguistics*. 69–76.

[27] Pawel Mazur and Robert Dale. 2010. WikiWars: A New Corpus for Research on Temporal Expressions. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*. 913–922.

[28] Robert Parker, David Graff, Junbo Kong, Ke Chen, and Kazuaki Maeda. 2011. Engilish Gigaword Fifth Edition. (2011).

[29] Alexandre Passos, Vineet Kumar, and Andrew McCallum. 2014. Lexicon Infused Phrase Embeddings for Named Entity Resolution. In *Proceedings of the 8th Conference on Computational Language Learning*. 78–86.

[30] James Pustejovsky, Jose Castano, Robert Ingria, Roser Sauri, Robert Gaizauskas, Andrea Setzer, Graham Katz, and Dragomir Radev. 2003. TimeML: Robust Specification of Event and Temporal Expressions in Text. *New Directions in Question Answering* 3 (2003), 28–34.

[31] James Pustejovsky, Patrick Hanks, Roser Sauri, Andrew See, Robert Gaizauskas, Andrea Setzer, Beth Sundheim, Dragomir Radev, David Day, Lisa Ferro, and Marcia Lazo. 2003. The TIMEBANK Corpus. *Corpus Linguistics* 2003 (2003), 647–656.

[32] James Pustejovsky, Kiyong Lee, Harry Bunt, and Laurent Romary. 2010. ISO-TimeML: An International Standard for Semantic Annotation. In *Proceedings of the Seventh conference on International Language Resouces and Evaluation (LREC'10)*. 394–397.

[33] Lev Ratinov and Dan Roth. 2009. Design Challenges and Misconceptions in Named Entity Recognition. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning*. 147–155.

[34] Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2013 Shared Task: Language-Independent Named Entity Recognition. In *Proceedings of the 7th Conference on Natural Language Learning*. 142–147.

[35] Erik F. Tjong Kim Sang and Jorn Veenstra. 1999. Representing Text Chunks. In *Proceedings of the ninth Conference on European Chapter of the Association for Computational Linguistics*. 173–179.

[36] Mark Steedman. 1996. *Surface Structure and Interpretation*. The MIT Press.

[37] S. S. Stevens. 1946. On the Theory of Scales of Measurement. *Science* 103, 2684 (1946), 677–680.

[38] Jannik Strotgen, Thomas Bogel, Julian Zell, Ayser Armiti, Tran Van Canh, and Michael Gertz. 2014. Extending HeidelTime for Temporal Expressions Referring to Historic Dates. In *Proceedings of 9th International Conference on Language Resources and Evaluation*. 2390–2397.

[39] Jannik Strötgen and Michael Gertz. 2010. HeidelTime: High Quality Rule-based Extraction and Normalization of Temporal Expressions. In *Proceedings of the 5th International Workshop on Semantic Evaluation (SemEval'10)*. Association for Computational Linguistics, Stroudsburg, PA, USA, 321–324.

[40] Jannik Strotgen, Julian Zell, and Michael Gertz. 2013. HeidelTime: Tuning English and Developing Spanish Resources. In *Proceedings of second Joint Conference on Lexical and Computational Semantics (SEM)*. 15–19.

[41] Naushad UzZaman and James F. Allen. 2010. TRIPS and TRIOS System for TempEval-2: Extracting Temporal Information from Text. In *Proceedings of the 5th International Workshop on Semantic Evaluation*. 276–283.

[42] Naushad UzZaman, Hector Llorens, Leon Derczynski, Marc Verhagen, James Allen, and James Pustejovsky. 2013. SemEval-2013 Task 1: TempEval-3: Evaluating Time Expressions, Events, and Temporal Relations. In *Proceedings of the 7th International Workshop on Semantic Evaluation*. 1–9.

[43] Marc Verhagen, Robert Gaizauskas, Frank Schilder, Mark Hepple, Graham Katz, and James Pustejovsky. 2007. SemEval-2007 Task 15: TempEval Temporal Relation Identification. In *Proceedings of the 4th International Workshop on Semantic Evaluation*. 75–80.

[44] Marc Verhagen, Inderjeet Mani, Roser Sauri, Robert Knippen, Seok Bae Jang, Jessica Littman, Anna Rumshisky, John Phillips, Inderjeet Mani, Roser Sauri, Robert Knippen, Seok Bae Jang, Jessica Littman, Anna Rumshisky, John Phillips, and James Pustejovsky. 2005. Automating Temporal Annotation with TARQI. In *Proceedings of the ACL Interactive Poster and Demonstration Sessions*. 81–84.

[45] Marc Verhagen, Roser Sauri, Tommaso Caselli, and James Pustejovsky. 2010. SemEval-2010 Task 13: TempEval-2. In *Proceedings of the 5th International Workshop on Semantic Evaluation*. 57–62.

[46] Mingbin Xu, Hui Jiang, and Sedtawut Watcharawittayakul. 2017. A Local Detection Approach for Named Entity Recognition and Mention Detection. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*. 1237–1247.

[47] Xiaoshi Zhong, Aixin Sun, and Erik Cambria. 2017. 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*. 420–429.