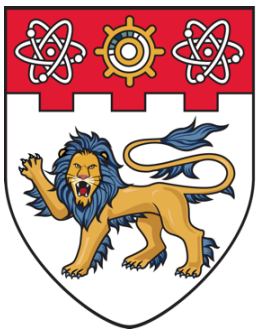


NANYANG TECHNOLOGICAL UNIVERSITY
SINGAPORE

Time Expression and Named Entity Analysis and Recognition

Xiaoshi Zhong

School of Computer Science and Engineering

A thesis submitted to the Nanyang Technological University
in partial fulfilment of the requirement for the degree of Doctor of Philosophy

January 2020

Statement of Originality

I hereby certify that the work embodied in this thesis is the result of original research, is free of plagiarised materials, and has not been submitted for a higher degree to any other University or Institution.

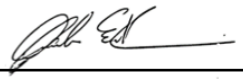
15 January 2020	Xiaoshi Zhong
Date	Xiaoshi Zhong

Supervisor Declaration Statement

I have reviewed the content and presentation style of this thesis and declare it is free of plagiarism and of sufficient grammatical clarity to be examined. To the best of my knowledge, the research and writing are those of the candidate except as acknowledged in the Author Attribution Statement. I confirm that the investigations were conducted in accord with the ethics policies and integrity standards of Nanyang Technological University and that the research data are presented honestly and without prejudice.

16 January 2020

Date



Prof. Erik Cambria

Authorship Attribution Statement

This thesis contains material from the following three peer-reviewed papers and one submission in which I am listed as the first author.

Chapter 4 and part of Section 3.1 is published with material from a conference paper:

Xiaoshi Zhong, Aixin Sun, and Erik Cambria. Time Expression Analysis and Recognition Using Syntactic Token Types and General Heuristic Rules. In *Proceedings of the 55th Annual Meetings of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 420-429, Vancouver, Canada, 2017.

The contributions of the co-authors are as follows:

- I came up with the idea, designed and performed all the experiments, conducted the data analysis, wrote the first manuscript, and revised the drafts.
- Prof. Erik Cambria and Aixin Sun revised the manuscript.

Chapter 5 and part of Section 3.1 is published with material from a conference paper:

Xiaoshi Zhong and Erik Cambria. Time Expression Recognition Using a Constituent-based Tagging Scheme. In *Proceedings of the 2018 World Wide Web Conference*, pages 983-992, Lyon, France, 2018.

The contributions of the co-authors are as follows:

- I came up with the idea, designed and performed all the experiments, conducted the data analysis, wrote the first manuscript, and revised the manuscript drafts.
- Prof. Erik Cambria proofread the manuscript.

Chapter 6 and Section 3.2 is published with material from a journal paper:

Xiaoshi Zhong, Erik Cambria, and Amir Hussain. Extracting Time Expressions and Named Entities with Constituent-based Tagging Schemes. In *Cognitive Computation*, pp. 1-19, 2020.

The contributions of the co-authors are as follows:

- I came up with the idea, designed and performed all the experiments, conducted the data analysis, wrote the first manuscript, and revised the drafts.
- Prof. Erik Cambria and Prof. Amir Hussain proofread the manuscript.

Chapter 7 contains the material from a journal submission:

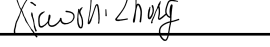
Xiaoshi Zhong, Erik Cambria, and Jagath C. Rajapakse. Power-law Distributions in Length-Frequency of Entities. Submitted to *Nature Communications*, 2020.

The contributions of the co-authors are as follows:

- I discovered this linguistic phenomenon, collected datasets, performed all the experiments, wrote the first manuscript, and revised the drafts.
- Prof. Erik Cambria and Jagath C. Rajapakse proofread the manuscript.

15 January 2020

Date



Xiaoshi Zhong

Acknowledgements

First and foremost, I would like to thank my supervisor Prof. Erik Cambria for his support, understanding, and encouragement during my Ph.D. study, which is one of the most important periods in my life. Besides being a supervisor, Erik is also a good friend and a great leader. I would also like to thank my co-supervisor Prof. Jagath C. Rajapakse for his support, especially, for his tolerance of my poor spoken English. Without the support from Erik and Jagath, it would not be possible for me to complete this dissertation. I am thankful to Prof. Jie Zhang, Prof. Shafiq Joty, and Prof. Kezhi Mao for their taking time to serve as my thesis advisory committee, and Prof. Cuntai Guan, Prof. Xavier Bresson, and Prof. Kezhi Mao for their precious time being as my thesis panel members.

I want to thank the group-mates and lab-mates in the Computational Intelligence Lab (CIL) and Biomedical Informatics Lab (BIL) for their companionship in the journey of pursuing knowledge and expanding the boundaries of our understanding. I would also want to thank the following language coaches in the Communication Cube (CommCube) of the Language and Communication Centre (LCC): Haoxin, Nicole, Sean, Kathryn, Daniel, Maitreya, Hilary, Terri, Paul, Christine, Gaia, Atiqah, Zhilian, Shalini, Hunter, Autumn, Izza, and Timothy. With their help, I have corrected many mistakes in my pronunciation and significantly improved my spoken English and communication skills. The staffs in the graduate student office, especially Ms. Chiam Poh Ling and Ms. Juliana Binte Jaapar, helped me a lot in various of affairs, and I am very grateful to them.

Finally, special thanks are given to my parents, Zhenhuan Zhong and Limei Xiao, for their unconditional love and support in the past thirty years, and to my honey, Han Li, for her unending love, trust, and support.

Contents

List of Figures

List of Tables

Acronyms and Notations

Acronyms

TER	Time expression recognition
TEN	Time expression normalization
TERN	Time expression recognition and normalization
NER	Named entity recognition
NEC	Named entity classification
NERC	Named entity recognition and classification
CRFs	Conditional random fields
POS	Part-of-Speech

Notations

SynTime	Our proposed type-based method for time expression recognition
TOMN	Our proposed learning-based method for time expression recogntiion
TOMN scheme	Our proposed constituent-based tagging scheme that TOMN defines to model time expressions
UGTO	Our proposed learning-based method for named entity recogntion
UGTO scheme	Our proposed constituent-based tagging scheme that UGTO defines to model named entities

Abstract

This dissertation presents our analysis of intrinsic characteristics of time expressions and named entities, and our use of these characteristics to design algorithms to recognize time expressions and named entities from unstructured text.

Regarding time expressions, we analyze four diverse datasets and find five common characteristics about them. Firstly, most time expressions are very short. Secondly, most time expressions contain at least one time-related word that can distinguish time expressions from common text. Thirdly, only a small group of words are used to express time information. Fourthly, words in time expressions demonstrate similar syntactic behaviour. Finally, time expressions are formed by loose structure. According to these five characteristics, we propose two methods to model time expressions. The first method is a type-based method termed SynTime. SynTime defines three main syntactic token types, namely *time token*, *modifier*, and *numeral*, to group time-related token regular expressions, and designs a small set of general heuristic rules to recognize time expressions. These heuristic rules are only relevant to token types and are independent of specific tokens, therefore, SynTime is independent of specific domains and specific text types that consist of specific tokens. Our second method is a learning-based method termed TOMN. TOMN defines a constituent-based tagging scheme with four tags, namely T, M, N, and O, indicating four types of constituent words of time expressions. In modeling, TOMN assigns a word with a TOMN tag under conditional random fields (CRFs) with minimal features. Essentially, our TOMN scheme overcomes the problem of *inconsistent tag assignment* that is caused by the conventional position-based tagging schemes (e.g., the BIO and BILOU schemes). Experimental results on three datasets demonstrate the efficiency, effectiveness, and robustness of SynTime and TOMN against four state-of-the-art methods.

Regarding named entities, we analyze two benchmark datasets and find three common characteristics about them. Firstly, most named entities contain uncommon words, which

mainly appear in named entities and hardly appear in common text. Secondly, named entities are mainly made up of proper nouns. Thirdly, named entities are formed by loose structure. These three characteristics motivate us to design a CRFs-based learning method termed UGTO to model named entities. Like TOMN, UGTO defines another constituent-based tagging scheme with four tags, namely U, G, T, and O, indicating four types of constituent words of named entities, namely *uncommon words*, *generic modifiers*, *trigger words*, and those words *outside* named entities. In modeling, our UGTO scheme models named entities under a CRF framework with minimal features. Experiments on two benchmark diverse datasets show that UGTO performs more effectively than two representative state-of-the-art methods.

When analyzing time expressions and named entities, we discover that their length, in terms of the number of words, follows a family of power-law distributions. Furthermore, we find that these power-law distributions widely appear in the length-frequency of entities in seventeen languages (e.g., Chinese, English, and German) and different types of entities (e.g., named entities, time expressions, and aspect terms). We explain this linguistic phenomenon by the principle of least effort in communication and the preference for short entities, and justify our explanation by a stochastic process, in which the probabilities are derived from real-word datasets, that reproduces power-law distributions in the length-frequency of generated entities.

Chapter 1

Introduction

Time expressions and named entities play important roles in the fields of data mining, information retrieval, and natural language processing [33, 73, 126, 143, 178, 185, 227]. They are involved in many linguistic tasks, such as temporal relation extraction [26, 127, 217], timeline construction [50, 106, 137], temporal information retrieval [2, 23], named entity recognition and classification [34, 73, 178], named entity typing [71, 114, 145], entity linking [88, 113], domain-specific entity recognition [97, 155, 209], and relation extraction and reasoning [139, 237].

Researchers from various areas have devoted tremendous effort for more than two decades to specify standards for the annotations of time expressions [56, 85, 163, 165] and named entities [34, 35, 51, 73, 177, 178], construct annotated corpora for the analyses of time expressions [85, 132, 164, 240] and named entities [34, 35, 48, 51, 73, 161, 162, 175, 177, 178, 196, 223], and recognize time expressions and named entities from unstructured text [13, 15, 16, 33, 51, 73, 177, 178, 214, 217, 219].

To better understand their intrinsic characteristics, we analyze four diverse datasets about time expressions and two benchmark datasets about named entities. According to these characteristics, we propose two methods to recognize time expressions and one method to recognize named entities from unstructured text.[1]

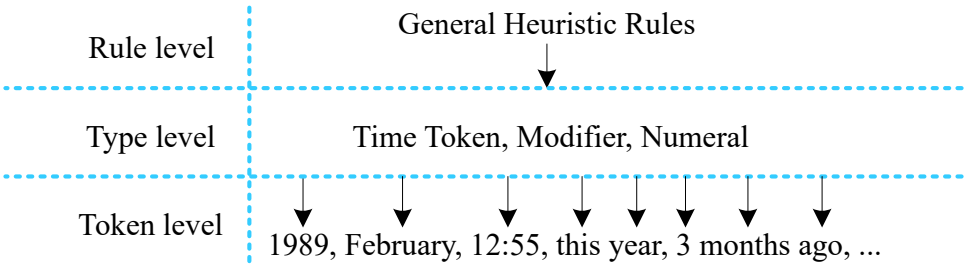
[1]Part of the content in this chapter has been published as Xiaoshi Zhong, Aixin Sun, and Erik Cambria. Time Expression Analysis and Recognition Using Syntactic Token Types and General Heuristic Rules. In *Proceedings of the 55th Annual Meetings of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 420-429, Vancouver, Canada, 2017 [240]. Xiaoshi Zhong and Erik Cambria. Time Expression Recognition Using a Constituent-based Tagging Scheme. In *Proceedings of the 2018 World Wide Web Conference*, pages 983-992, Lyon, France, 2018 [238]. Xiaoshi Zhong, Erik Cambria, and Amir Hussain. Extracting Time Expressions and Named Entities with Constituent-based Tagging Schemes. In *Cognitive Computation*, pp. 1-19, 2020 [239]. Part of the content in this chapter is also under review as Xiaoshi Zhong, Erik Cambria, and Jagath C. Rajapakse. Power-law Distributions in Length-Frequency of Entities. Submitted to *Nature Communications*, 2020.

1.1 Time Expression Analysis and Recognition

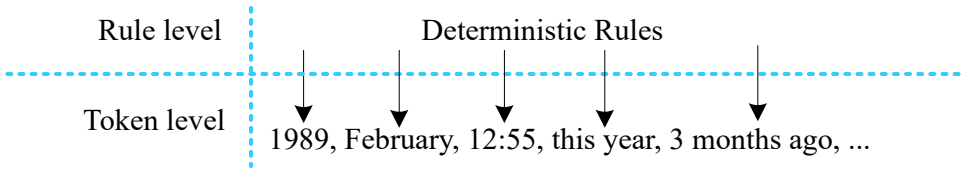
The four datasets we use to analyze time expressions are TimeBank [164], TE3-Silver [214], WikiWars [132], and Tweets [240]. From our analysis we find five common characteristics of time expressions. Firstly, most time expressions are very short, with more than 80% of time expressions containing no more than three words. Secondly, most time expressions contain time tokens that can distinguish time expressions from common text; more than 91.8% of time expressions containing at least one time tokens while no more than 0.7% of common text containing time tokens. Thirdly, those words that are used to express time information are in a small size, with only about 70 distinct time tokens in an individual dataset and about 120 distinct time tokens across these four datasets. Fourthly, words in time expressions demonstrate similar syntactic behaviour. Finally, time expressions are formed by loose structure, with more than 53.5% of distinct time tokens appearing in different positions within time expressions. The first four characteristics are related to the principle of least effort [242], which states that at both individual and collective level, people tend to act under the least effort in order to minimize the cost of energy in almost all the aspects of human actions, including language use. Time expressions are part of language and act as an interface of communication. Short expressions, occurrence and distinction, small vocabulary, and similar syntactic behaviour all reduce the cost of our energy required to communicate with each other. The last characteristic demonstrates the flexibility of time expressions.

According to these five characteristics we propose two methods to recognize time expressions from unstructured text. Our first method is a type-based method termed SynTime ("Syn" stands for "syntactic"). SynTime defines three main token types, namely *time token*, *modifier*, and *numeral*, to group time-related token regular expressions. Time tokens include those words that explicitly express time information, such as time units (e.g., "year"). Modifiers modify time tokens and appear around them; for example, the two modifiers "several" and "ago" modify the time token "year" in the time expression "several years ago." Numerals include ordinals and numbers (except those tokens that are identified as YEAR, e.g., "2006"). From the raw text, SynTime firstly identifies time tokens, then recognizes modifiers and numerals, and finally recognize the full time expressions.

Naturally, SynTime is a rule-based tagger. The key difference between SynTime and other rule-based time expression taggers lies in the ways of defining token types and designing rules.



(a) Layout of SynTime. The layout consists of three levels: token level, type level, and rule level. Token types group the constituent words of time expressions. Heuristic rules work on token types in a heuristic manner and are independent of specific tokens and therefore SynTime is independent of specific domains, text types, and even languages.



(b) Layout of other rule-based time taggers. The layout mainly consists of two levels: token level and rule level. Deterministic rules work directly on tokens and phrases in a fixed manner and therefore other rule-based time taggers lack flexibility.

Figure 1.1: The key difference between SynTime and other rule-based time taggers

The definition of token types in SynTime is inspired by the part-of-speech (POS) of language, in which "linguists group some words of language into classes (sets) which show similar syntactic behaviour" [130]. SynTime defines token types for tokens according to their syntactic behaviours. Other rule-based taggers define token types for tokens based on their semantic meanings. For example, SUTime defines five semantic modifier types, such as frequency modifiers and approximate modifiers,[2] while SynTime defines five syntactic modifier types, such as modifiers that appear before time tokens and modifiers that appear after time tokens (see Section 4.1 for details). Accordingly, other rule-based taggers design deterministic rules working directly on tokens themselves. SynTime instead designs general rules working on token types rather than on tokens themselves. For example, our general rules do not work on the tokens "February" and "1989" but work on their token types "MONTH" and "YEAR." That is why we call SynTime a type-based method. More importantly, other rule-based taggers design rules in a fixed manner, including fixed length and fixed position. By contrast, SynTime designs general rules in a heuristic way based on the idea of boundary expansion. Therefore, SynTime

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

is much more flexible than other rule-based methods. Furthermore, these general heuristic rules are quite light-weight, which leads SynTime to run in real time.

Since our heuristic rules are designed to work on token types and are independent of specific tokens, SynTime is independent of specific domains, specific text types, and even specific languages that consists of specific tokens. In this dissertation, we test SynTime on specific domains and specific text types in English. Testing on other languages needs to construct a collection of token regular expressions in target languages under our defined token types or other token-type systems.

Our second method is a learning-based method termed TOMN. Specifically, TOMN defines a constituent-based tagging scheme termed TOMN scheme consisting of four tags, namely T, O, M, and N, indicating the constituent words of time expressions and corresponding the three main token types defined in SynTime, namely *time token*, *modifier*, *numeral*, and those words appearing *outside* time expressions.[3] In practice , TOMN models time expressions under a CRFs framework [98] with only a kind of TOMN pre-tag features and lemma features. During modeling and tagging, it assigns a word with one of the four TOMN tags.

The TOMN scheme can keep tag assignment consistent during training and therefore overcomes the problem of inconsistent tag assignment that is caused by conventional position-based tagging schemes.[4] The loose structure by which time expressions are formed exhibits in the following two aspects. Firstly, 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 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. The conventional tagging schemes like BILOU [170] are based on *the positions within a labeled chunk*, namely a unit-word chunk, and the beginning, inside, and last word of a multi-word chunk. Under the BILOU scheme, a word that appears in different positions within labeled time expressions is assigned with different tags; for example, the above time token "September" can be assigned with U,

[3] We use "TOMN" to denote our method for time expression recognition while use "TOMN scheme" to denote the constituent-based tagging scheme that TOMN defines to model time expressions.

[4] 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.

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

(a) Tag assignment under the 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 the 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 under our 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" and avoids potential tag balance.

Figure 1.2: Comparison of tag assignments under the BILOU scheme and our TOMN scheme. The BILOU scheme is based on the positions within labeled chunks, while our TOMN scheme is based on the constituent words of labeled chunks. Here, *inconsistent tag assignment* is defined as that during training, a word is assigned with different tags simply because this word appears in different positions within labeled chunks.[5]

B, L, or I (see Figure 1.2(a)). Such inconsistent tag assignment causes difficulty for statistical models to model time expressions. Firstly, inconsistent tag assignment reduces the predictive power of lexicon, and this contradicts the characteristic we find that time tokens can distinguish time expressions from common text. Secondly, inconsistent tag assignment might cause the problem of tag imbalance. By contrast, Our TOMN scheme is based on *the constituents of a labeled chunk* (i.e., time token, modifier, and numeral) and assigns the same constituent word with the same tag, regardless of its frequency and its positions within time expressions. Under our TOMN scheme, for example, the above time token "September" is consistently assigned with T (see Figure 1.2(b)). With such consistent tag assignment, our TOMN scheme protects the predictive power of time tokens and avoids the potential tag imbalance.

We evaluate the quality of SynTime and TOMN against four state-of-the-art methods (i.e., HeidelTime [199], SUTime [27], ClearTK [12], and UWTime [103]) on three diverse datasets (i.e., TE-3 [214], WikiWars [132], and Tweets).[6] HeidelTime and SUTime are rule-based methods while ClearTK and UWTime are learning-based methods. TE-3 and Tweets are

[5]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(s) while that unit should be consistently assigned with a same tag. The unit of interest can be a word, a phrase, a sentence, an article, a relation, a webpage, or a group of words as a whole, etc.

[6]The TE3-Silver dataset is only used in our analysis for the characteristics of time expressions; it is not used in our experiments because the labels of its time expressions are not ground-truth labels but instead are automatically generated by other time expression taggers.

comprehensive datasets while WikiWars is a domain-specific dataset about war. TE-3 and WikiWars are datasets in formal text while Tweets is in informal text. Experimental results demonstrate that SynTime and TOMN achieve comparable results on the WikiWars dataset, and significantly outperform these four state-of-the-art baselines on the TE-3 and Tweets datasets. More importantly, SynTime and TOMN achieve the best recalls on all the three datasets and exceptionally good results on the Tweets dataset. Experimental results also demonstrate the robustness of TOMN on cross-dataset experiments against the two learning-based baselines and demonstrate the advantage of our constituent-based tagging scheme against the conventional position-based tagging schemes.

1.2 Named Entity Analysis and Recognition

The two datasets we use to analyze named entities are CoNLL03 [178] and OntoNotes* (OntoNotes* is a derived version of the OntoNotes5 corpus [161, 162]; see Section 3.2.1 for details). From our analysis we find three common characteristics about named entities. Firstly, most named entities contain uncommon words, with more than 92.2% of named entities have at least one word hardly appearing 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. Thirdly, named entities are formed by loose structure, with more than 53.77% of distinct words appearing in different positions within named entities.

These three characteristics motivate us to design a CRFs-based learning method termed UGTO to recognize named entities from unstructured text. Specifically, UGTO defines a constituent-based tagging scheme termed UGTO scheme that consists of four tags: U, G, T, and O.[7] The UGTO scheme is designed to encode the constituent words of named entities. Specifically, U encodes *uncommon words* and entity-related tokens, such as "Boston" and "Africans." G encodes *generic modifiers* while T encodes *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 the ORG named entities "Boston University" and "Stanford University." O encodes those words

[7] Similar to the use of "TOMN" and "TOMN scheme," we use "UGTO" to denote our proposed method for named entity recognition while use "UGTO scheme" to denote the constituent-based tagging scheme that UGTO defines to model named entities.

that appear *outside* named entities. In modeling, UGTO assigns a word with a UGTO tag under a CRFs framework with only UGTO pre-tag features, a kind of word cluster features, and some basic lexical and POS features (see Section 6.4.1 for details).

UGTO extends the idea of TOMN from time expression modeling to named entity modeling. Like TOMN, UGTO overcomes the problem of inconsistent tag assignment and therefore fully leverages the information of uncommon words and proper nouns. The key difference between UGTO and TOMN lies in the difference between general named entities and time expressions. Firstly, time expressions contain only a small group of time-related words, which can be wholly collected (e.g., only 350 unique words appear in time expressions across the four analyzed datasets). By contrast, general named entities contain countless words, and it is difficult to collect all of them (e.g., 23,698 unique words appear in named entities across the two analyzed datasets). Secondly, POS tags cannot distinguish time expressions from common text and TOMN does not take into account any syntactic features. However, named entities are mainly made of proper nouns, which are a kind of syntactic features, and proper nouns are important information that is used in UGTO. In practice, UGTO derives two kinds of uncommon words from annotated training set and unannotated test set based on the idea that those words that hardly appear in the common text of the training set are likely to predict named entities (see Figure 6.1 for the detailed illustration of the idea).

We evaluate the quality of UGTO on two benchmark datasets (i.e., CoNLL03 [178] and OntoNotes* [161]) against two representative state-of-the-art baselines (StanfordNER [60] and LSTM-CRF [99]). CoNLL03 is a small dataset collected from news articles in formal text, while OntoNotes* is a large-scale datasets collected from diverse sources over a long period of time. StanfordNER is used as the representative of traditional hand-crafted-feature methods while LSTM-CRF is as the representative of recent auto-learned-feature methods. Experimental results demonstrate the effectiveness and efficiency of UGTO against these two representative baselines. Experimental results also demonstrate that traditional hand-crafted-feature methods can achieve state-of-the-art performance on named entity recognition, in comparison with the state-of-the-art auto-learned-feature method, and that joint modeling named entity recognition and classification does not improve the performance of named entity recognition, in both our model and these two representative baselines (see Section 6.5.3 for details).

1.3 Power-law Distributions in Length-Frequency of Entities

Estoup and Zipf find a very long time ago that the rank-frequency of words in natural languages follows a family of power-law distributions [54, 241, 242]. During his exploration, Zipf also finds that the meaning-frequency of words follows a family of power-law distributions as well [242]. The rank-frequency distribution of words is later credited as Zipf's law and provides a direction to understand the use of languages in our communicative system. Zipf's law has received tremendous attention of researchers from diverse fields (e.g., linguistics and statistics) for more than 80 years [157], and many researchers try to explain this linguistic phenomenon from different perspectives, such as random concatenative processes [41, 107], scale-invariance [30], optimization of entropy [123, 124], multiplicative stochastic processes [140], preferential reuse [189, 190, 212], symbolic descriptions of complex stochastic systems [42], semantic organization [76], and many others.

Besides the rank-frequency and meaning-frequency of words, researchers also explore to know whether power-law distributions appear in any other form of human languages? In the last two decades, the fields of computational linguistics and natural language processing have annotated numerous datasets that provide us opportunities to analyze another form of languages: entity. An entity is a real-world object, such as persons, locations, and organizations [34, 73, 178]. Through analyzing these annotated entities, we find that the length-frequency of entities follows a family of power-law distributions.

The concept of *entity* in this section broadly includes named entities [34, 73, 178], entity mentions [114, 161], time expressions [163, 164], aspect terms [115, 160], literary entities [9], informal entities [175], and domain-specific entities [68, 209] that have been well investigated in various areas related to computational linguistics and natural language processing. An instance of the entity concept reflects an object in reality, such as "China" and "United States." *Entity length* is defined by the number of words in an entity, denoted by l. Generally, there are two types of languages: *word-spaced languages* and *non-spaced languages*. Word-spaced languages are those languages that use spaces separating their words in their written systems (e.g., Arabic, English, and German), while non-spaced languages do not (e.g., Chinese and Japanese). For a word-spaced language, the entity length is calculated directly. For a non-spaced language, we firstly employ a segmentation tool to segment its entities, and then calculate the length of its

Table 1.1: Some examples of entities in word-spaced and non-spaced languages and their corresponding entity lengths (l). Symbols and punctuation are taken into account during the calculation of entity length. These entities of non-spaced languages are already segmented. The description in "[·]" indicates the translation.

Word-spaced Languages	
Entity	l
46,480	1
Chinese	1
Walter Cristofoletto	2
United Arab Emirates	3
10:00 p.m. on August 20 , 1940	7
human cytomegalovirus (HCMV) major immediate	7
Non-spaced Languages	
Entity	l
美国 [United States]	1
新加坡 [Singapore]	1
互联网 [Internet]	1
香港 国际 机场 [Hong Kong International Airport]	3
中华 人民 共和国 [People 's Republic of China]	4
悉尼 奥运会 主 体育场 [Sydney Olympic Main Stadium]	4

entities. Table 1.1 shows some examples of entities in these two types of languages and their corresponding entity lengths.

We discover that power-law distributions widely appear in the length-frequency of entities in different languages and different types of entities. Specifically, we analyze entities from seventeen languages: Arabic, Chinese, Dutch, English, Finnish, French, German, Italian, Japanese, Norwegisch, Polish, Portuguese, Russian, Spanish, Swahili, Swedish, and Turkish. The datasets that we use to analyze entities in different languages are collected from the HeiNER inventory [226] and the ACE04 corpus [51]. We also analyze different types of entities in English from the following ten datasets: ABSA [159, 160], ACE04 [51], BBN [223], Bioinformatics [44], CoNLL03 [178], LitBank [9], OntoNotes5 [161], TimeExp [132, 164, 214, 238, 240], Twitter [48, 196], and WikiAnchor [114] (see Section 7.1 for details of these datasets). Our analysis demonstrates that although these datasets vary in source, domain, text genre, generated time, corpus size, entity type, and annotation criterion, the length-frequency of their entities follows a family of power-law distributions, with a stable scaling property and well-defined means and finite variances [148] (see Section 7.2).

We explain this linguistic phenomenon of power-law distributions in the length-frequency of entities by the principle of least effort in communication [242] and the preference for short entities. To justify our explanation, we design a stochastic process, in which the preferential probabilities are derived from real-world datasets, that reproduces power-law distributions in the length-frequency of generated entities (see Section 7.3.2 for details).

1.4 Contributions

In this dissertation, together with my collaborators, I make the following contributions.

- We analyze four diverse datasets and summarize five common characteristics about time expressions. The first four characteristics provide evidence in terms of time expressions as part of language for the principle of least effort [242] and the last characteristic demonstrates the flexible structure of time expressions. Our analysis of time expressions can help explain many empirical observations reported in previous works about time expression recognition.
- We propose a type-based method termed SynTime to recognize time expressions from unstructured text by using syntactic token types and general heuristic rules. SynTime is independent of independent of specific domains, text types, and even languages. Furthermore, SynTime is light-weight and runs in real time.
- We identify a fundamental problem underlying in the conventional position-based tagging schemes: inconsistent tag assignment. To overcome this problem, we define a constituent-based tagging scheme to model time expressions. Our analysis of tagging schemes can help explain many empirical results and observations that are reported in previous works about the effectiveness of tagging schemes in named entity recognition. Our proposed method provides an idea to model target entities based on their constituents.
- We analyze two benchmark datasets and summarize three common characteristics about named entities. According to these characteristics, we propose a CRFs-based learning method with defining another constituent-based tagging scheme to modeling named entities. Surprisingly, experimental results demonstrate that joint modeling of named entity recognition and classification does not improve the performance of named entity recognition, in both our proposed model and two representative state-of-the-art methods.

- We discover that the length-frequency of entities in seventeen languages and different types of entities follows a family of power-law distributions, with stable scaling property and well-defined means and finite variances. We explain this linguistic phenomenon by the principle of least effort in communication and the preference for short entities, and justify our explanation by a stochastic process that reproduces power-law distributions in the length-frequency of generated entities.

1.5 Organization of this Dissertation

The structure of this dissertation is organized as follows.

In Chapter 1, we summarize the content of this dissertation and our contributions, including (1) the analysis of intrinsic characteristics of time expressions and named entities, (2) two methods for time expression recognition and one method for named entity recognition, and (3) our discovery of power-law distributions in the length-frequency of entities.

In Chapter 2, we overview the literature about time expression recognition and normalization, named entity recognition and classification, and power-law distributions in language.

In Chapter 3, we detail our analysis on four diverse datasets for intrinsic characteristics of time expressions and two benchmark datasets for intrinsic characteristics of named entities.

In Chapter 4, we detail our type-based time tagger, SynTime, which defines syntactic token types and design general heuristic rules to recognize time expressions from unstructured text, and the experiments that we conduct on three datasets to justify SynTime's effectiveness.

In Chapter 5, we detail our proposed CRFs-based learning time tagger, TOMN, which defines a constituent-based tagging scheme to model time expressions, and the experiments we conduct on three datasets to justify TOMN's effectiveness, efficiency, and robustness.

In Chapter 6, we detail our proposed CRFs-based learning method, UGTO, which defines another constituent-based tagging scheme with minimal features to model named entities, as well as the experiments we conduct on two benchmark datasets to justify UGTO's effectiveness.

In Chapter 7, we present our discovery of power-law distributions in length-frequency of entities as well as our explanation and justification for this linguistic phenomenon.

In Chapter 8, we draw a conclusion about this dissertation and outline some potential directions in future research.

Chapter 2

Related Works

The works that are related to this dissertation mainly include the research of time expression recognition and normalization, named entity recognition and classification, and power-law distributions in language.[1]

2.1 Time Expression Recognition and Normalization

The extensive studies of time expressions start from the sixth and seventh Message Understanding Conference (MUC-6 and MUC-7), in which Grishman & Sundheim [73] and Chinchor [34, 35] formally define the task of identifying time expressions from unstructured text, together with the tasks of identifying entity names and number expressions as well as other information extraction tasks. After MUC-6 and MUC-7, researchers from different fields (e.g., data mining, information retrieval, natural language processing, and related areas) have devoted tremendous effort to the analysis of time expressions [2, 23, 126, 227], specifying annotation standards for time expression [14, 55, 56, 57, 85, 100, 129, 163, 165], developing annotated corpora for time expression [85, 132, 164, 206], and organizing shared tasks to address the problems of recognizing and normalizing time expressions from unstructured text [13, 15, 16, 34, 35, 101, 146, 206, 214, 217, 219].

[1]Part of the content in this chapter has been published as Xiaoshi Zhong, Aixin Sun, and Erik Cambria. Time Expression Analysis and Recognition Using Syntactic Token Types and General Heuristic Rules. In *Proceedings of the 55th Annual Meetings of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 420-429, Vancouver, Canada, 2017 [240], Xiaoshi Zhong and Erik Cambria. Time Expression Recognition Using a Constituent-based Tagging Scheme. In *Proceedings of the 2018 World Wide Web Conference*, pages 983-992, Lyon, France, 2018 [238], and Xiaoshi Zhong, Erik Cambria, and Amir Hussain. Extracting Time Expressions and Named Entities with Constituent-based Tagging Schemes. In *Cognitive Computation*, pp. 1-19, 2020 [239]. Part of the content in this chapter is also under review as Xiaoshi Zhong, Erik Cambria, and Jagath C. Rajapakse. Power-law Distributions in Length-Frequency of Entities. Submitted to *Nature Communications*, 2020.

2.1.1 Language Factor

The majority of research in the time expression analysis is devoted to the study of English [13, 15, 16, 34, 56, 85, 95, 100, 101, 132, 146, 163, 164, 165, 214, 217, 219]. Besides English, Chinese is well studied and presented in English and Chinese literature [79, 80, 105, 228, 229, 230, 233]. Similarly, French, Italian, and Korean are strongly represented and boosted in series of works [8, 18, 25, 84, 86, 87, 102, 110, 125, 141, 216]. Many other languages receive attention as well: Arabic [21], Basque [3], Catalan [211], Croatian [191], Dutch [215], German [200], Portuguese [43, 46], Romanian [65], Spanish [181, 183, 203, 211], Swedish [10], Uyghur [142], Ukrainian [72], and Vietnamese [197]. Some works consider this problem in multilingual text [119, 147, 182, 197, 202, 203, 211, 225]. SynTime and TOMN focus on time expressions in English, and in the future, we plan to analyze time expressions in some other languages.

2.1.2 Domain and Textual Type Factor

The investigation of time expressions involves a variety of domains and textual types. The very first studies mainly focus the problem in formal text like news articles [19, 34, 164, 186]. Later on, these studies are gradually concerned with the problem in other domains and textual types. Mazur and Dale collect English articles from Wikipedia about famous wars and annotate the time expressions for domain-specific time expression analysis; this collected corpus is called WikiWars [132]. Similarly, Strotgen and Gertz develop the WikiWarsDE, which includes time expressions in the war domain collected from Wikipedia articles in German [200]. Strotgen and Gertz analyze time expressions in the texts from colloquial short message service (SMS) and scientific biomedical documents [201] while Degaetano-Ortlieb and Strotgen analyze time expressions in the scientific literature and their diachronic variation over a time span of about 350 years [47]. Tabassum et al. analyze time expressions in the tweets which are informal text [208]. Zhong et al. analyze time expressions across formal and informal text and comprehensive and specific domain text [238, 240]. A line of research have devoted tremendous effort on time expression recognition and normalization in the clinical domain [13, 15, 16, 53, 75, 78, 89, 101, 111, 135, 136, 176, 193, 205, 207, 210, 232], in which the progress in clinical domain is mainly among the i2b2 challenge and the series of clinical TempEval shared tasks.

In SynTime and TOMN, we analyze time expressions in comprehensive and specific domains as well as in the formal and informal text.

2.1.3 Time Expression Recognition

Although most approaches address the problem of time expression recognition together with time expression normalization as an end-to-end task, we discuss the two sub-tasks separately so as to better understand each of them. This section focuses on the time expression recognition and next section discuss the time expression normalization.

The methods for time expression recognition are mainly categorized into two kinds: rule-based methods and learning-based methods.

Rule-based Methods. Rule-based time taggers like TempEx, GUTime, HeidelTime, and SUTime mainly predefine a set of time-related words and regular expression patterns [27, 128, 199, 218]. HeidelTime hand-crafts rules with time resources like weekdays, seasons, and months, and leverages language clues like part-of-speech (POS) to identify time expression and then normalize them to the standard form in a pipeline UIMA (Unstructured Information Management Architecture[2]) [199]. SUTime [27] designs deterministic rules using a cascade finite automata [81] on regular expressions over tokens [29]. It firstly identifies individual words, then expands them to chunks, and finally to the full time expressions. Other rule-based taggers include FSS-TimEx [236], which uses finite-state rule cascades to recognize time expressions. These rule-based time taggers achieve very good performance in the TempEval shared tasks. Specifically, HeidelTime achieves the highest F_1 of 86% in TempEval-2 [219] and SUTime achieves the highest F_1 of 91.3% under the relaxed match in the TempEval-3 [214]. In the clinical evaluations (including the i2b2 challenge and clinical TempEval shared tasks) [13, 15, 205], the top systems develop corresponding rules based on either HeidelTime or SUTime to recognize the time expressions from clinical text.

Naturally, our SynTime is also a rule-based time expression tagger, while the key differences between SynTime and other rule-based taggers are that between the rules and the specific tokens SynTime introduces a layer of token type, and the rules that SynTime designs work on the token types, and are independent of specific tokens [240]. Moreover, the rules are designed in a heuristic way, leading SynTime to be much more flexible and expansible. As we will see, SynTime achieves much better results on various datasets in comparison with both rule-based taggers and learning-based taggers.

[2] http://uima.apache.org

Learning-based Methods. Machine learning-based methods mainly extract features from text and apply statistical models on these features for recognizing time expressions. Those features include character features (e.g., the first and last 3 characters of a word), word features (e.g., current, previous, and subsequent words), syntactic features (e.g., part-of-speech and noun phrase chunks), semantic features (e.g., lexical semantics and semantic role), and gazetteer features (e.g., matching in a dictionary) [12, 58, 59, 119]. Those statistical models include Markov logic network, logistic regression, support vector machines, maximum entropy, and conditional random fields [12, 58, 90, 119, 213]. These methods mainly leverage information from labeled data under supervised learning. Some of learning-based methods achieve good performance, and even the highest F_1 of 82.71% under strict match in TempEval-3 [12].

Outside the TempEval shared tasks, Angeli et al. leverage compositional grammar and employ an EM-style approach to learn a latent parser for time expression recognition [4]. In the method UWTime, Lee et al. employ a combinatory categorial grammar (CCG) [194] to define a set of lexicon with rules and use L1-regularization to learn from linguistic context for time expression recognition [103]. These two methods explicitly use linguistic information. In UWTime, especially, CCG could capture rich structure information of language, similar to the rule-based methods. Tabassum et al. focus on resolving the dates in tweets, and use distant supervision to recognize time expressions [208].

Unlike those methods that use standard features [12, 58, 59, 77, 90, 119, 213], TOMN derives only a kind of pre-tag features and lemma features according to the characteristics of time expressions, which can enhance the impact of significant features and reduce the impact of insignificant features [238]. Unlike those methods that use fixed structure information [4, 5, 103], TOMN uses loose structure information by grouping the constituent words of time expression under three main 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 tag assignment consistent.

2.1.4 Time Expression Normalization

Those methods that are developed for time expression normalization in the TempEval shared tasks and clinical evaluations (e.g., i2b2 challenge and clinical TempEval shared tasks) are mainly based on rules [12, 13, 15, 58, 119, 128, 199, 205, 213, 218]. Because these rule

systems share high similarity, Llorens et al. suggest to construct a large public knowledge base for the normalization task [118]. Some researchers treat the normalization problem as a learning task; Lee et al. [103] use AdaGrad algorithm and Tabassum et al. [208] use a log-linear algorithm to normalize time expressions. Recently, Berhard & Parker [14] and Laparra et al. [100] develop a semantically compositional annotation scheme to specify time expressions by which they can leverage machine learning techniques for this normalization task.

SynTime and TOMN focus on time expression recognition and leave time expression normalization to those highly similar rule-based methods or future work.

2.2 Named Entity Recognition and Classification

The research on named entity recognition and classification has a long history. Nadeau and Sekine [143] review its development of early years (from 1991 to 2006) in terms of languages (e.g., English, German, and Chinese) [31, 34, 73, 178, 221], text genres (e.g., scientific and journalistic) and domains (e.g., sports and business) [131, 138, 158], entity categories (e.g., PERSON, LOCATION, ORGANIZATION, and MISC) [34, 62, 73, 104, 178], learning methods (e.g., supervised, semi-supervised, and unsupervised learnings) [17, 22, 39, 144, 184], statistical learning techniques (e.g., hidden Markov models, maximum entropy models, and conditional random fields) [6, 17, 20, 133, 184], engineering features (e.g., word-level features, dictionary features, and document and corpus features) [17, 38, 39, 172, 188, 235], and shared task evaluations (e.g., ACE, MUC, and CoNLL) [34, 51, 73, 178].

Before the era of deep learning and neural networks, there are also works that consider several aspects of NERC, like leveraging unlabeled data for NERC [109], leveraging external knowledge for NERC [36, 91, 171], nested NERC [1, 61], and NERC in informal text [117, 175].

In the era of deep learning and neural networks, researchers employ neural networks and word embeddings to develop variants of models on the CoNLL03, ACE2004, and OntoNotes NERC dataset [36, 40, 49, 83, 99, 112, 116, 120, 121, 154, 156, 180, 204].

UGTO benefits some features (i.e., basic word and lemma features and general POS tags) from these traditional methods, and refines significant features (i.e., uncommon words and proper nouns) according to an in-depth analysis for the characteristics of named entities. Unlike these neural network-based methods that mainly compute semantic similarities among words,

UGTO focuses on the distinction between named entities and common text. And unlike most NERC methods that treat named entity recognition and classification as an end-to-end joint task, we focus on named entity recognition and demonstrates that joint modeling of named entity recognition and classification does not improve the performance of named entity recognition.

2.2.1 Named Entity Classification

A line of research is concerned with the problem of NEC (also known as entity typing), which assumes that named entities are already recognized from text. A variety of techniques have been developed for this problem [39, 63, 71, 114, 122, 145, 149, 174, 234]. These research leverage many features similar to the ones derived for the end-to-end NERC, such as bag of words, POS tags, and n-gram strings. A key difference between NEC and NERC is that researchers do not formulate NEC as a problem of sequence tagging but treat a whole named entity as a unit. We focus on NER and leave NEC to future work.

2.3 Power-law Distributions

Power-law distributions have been observed to appear in numerous natural and man-made systems [37, 69, 148, 173]. In this dissertation, however, we are mainly concerned with power-law distributions in human languages. Those works that are closely related to our work include power-law distributions in the rank-frequency and meaning-frequency of words as well as the distributions of word length and sentence length in a corpus.

2.3.1 Power-Law Distributions in Language

The most famous power-law distributions in language is the one in the rank-frequency of words. This linguistic phenomenon is originally discovered by Jean-Baptiste Estoup [54] and further explored and explained by George K. Zipf [241, 242]; it is later credited as Zipf's law.

Zipf's law [241, 242] reveals that the r-th most frequently occurring word in a corpus has the frequency defined by Equation (2.1).

$$f(r) \propto r^{-z} \tag{2.1}$$

where r denotes the frequency rank of a word and $f(r)$ denotes its frequency; the scaling exponent z is observed to be close to 1.

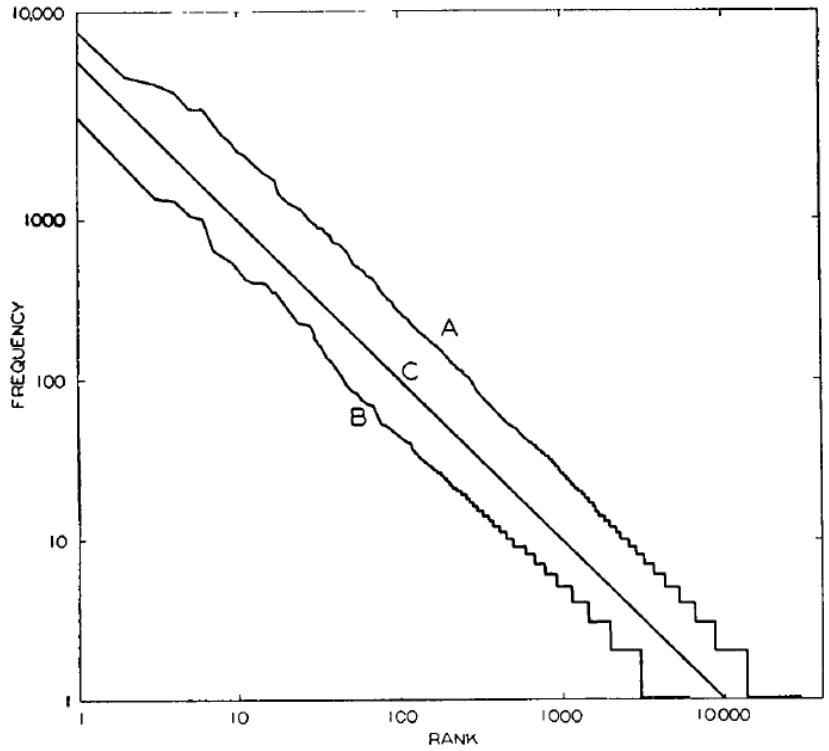


Figure 2.1: Rank-frequency distribution of words. (A) The James Joyce data; (B) the Eldridge data; (C) ideal curve with slope of negative unity. Adapted from Zipf's book [242].

Figure 2.1 is adapted from Zipf's book [242] and it shows a typical rank-frequency distribution of words in two datasets (i.e., the James Joyce data and the Eldridge data) and the ideal fitting curve. There are some variants of Zipf's law to view what is a word, among which one variant is "holophrases." Zipf treats these words connected with hyphens as a holophrase, such as "brother-in-law" and "hot-dog." The rank-frequency of words in a corpus with considering holophrases also follows a power-law distribution defined by Equation (2.1). The Zipf's law has been observed in many languages [42, 108, 157, 242].

During his exploration, Zipf finds that the meaning-frequency of words in a corpus also follows a family of power-law distributions, as defined by Equation (2.2).

$$f(m) \propto m^{-\beta} \tag{2.2}$$

where m denotes the number of the meanings of a word and $f(m)$ denotes its frequency; the scaling exponent β is observed to be approximate to 0.5.

Zipf's law has received tremendous attention from researchers (e.g., linguists and statisticians) for more than 80 years [157]. Many researchers try to explain this linguistic phenomenon from the perspectives of random concatenative processes [41, 107], scale-invariance [30], optimization of entropy [123, 124], multiplicative stochastic processes [140], preferential reuse [189, 190, 212], symbolic descriptions of complex stochastic systems [42], semantic organization [76], and many others.

2.3.2 Word Length and Sentence Length

According to a review article by Grotjahn & Altmann [74], Fucks first theoretically and experimentally demonstrates that the length-frequency of words in a corpus follows a family of Poisson distributions [66, 67], as defined by Equation (2.3).

$$p(l_s) = \frac{\lambda^{l_s} e^{-\lambda}}{l_s!} \tag{2.3}$$

where l_s denotes the number of *syllables* in a word, and $p(l_s)$ denotes its frequency in the corpus; the λ is the parameter of the curve.

The word length of a natural corpus has been observed to follow variants of Poisson distributions in more than 32 languages [11].

Williams [224] and Wake [220] observe that the length-frequency of sentences in a corpus follows a family of log-normal distributions, as defined by Equation (2.4).

$$p(l_w) = \frac{1}{l_w \sigma \sqrt{2\pi}} e^{-\frac{(\ln l_w - \mu)^2}{2\sigma^2}} \tag{2.4}$$

where l_w denotes the number of words in a sentence, and $p(l_w)$ denotes its frequency; μ and σ are parameters of the curve.

Gigurd et al. observe that the length-frequency of words as well as the one of sentences from English, Swedish, and German corpora follow a family of the variants of gamma distributions (in which word length is measured by the number of either words or phonemes or syllables in a word while sentence length is defined by the number of words in a sentence) [187].

2.3.3 Means and Variances of Power-law Distributions

Newman reviews the power-law distributions in numerous natural and man-made systems in the fields of biology, physics, earth and planetary sciences, economics and finance, computer

science, demography and social sciences [148]. He also analyzes some statistical properties of power-law distributions, such as means and variances.

For a general continuous variable x with a power-law distribution, it has a probability $p(x)dx$ of taking a value between x and $x + dx$, where

$$p(x) = Cx^{-\alpha} \tag{2.5}$$

with $\alpha > 0$, and C is a constant and is unimportant.

The n-th order moment of the variable is given by

$$E(x^n) = \int_{x_{min}}^{\infty} x^n p(x) dx = \frac{C}{n-\alpha} \left[x^{-\alpha+n}\right]_{x_{min}}^{\infty} \tag{2.6}$$

where x_{min} is the minimal value of x.

For the first order moment $E(x)$, namely the mean value, when $\alpha > 2$, it is defined. For the second order moment $E(x^2)$, which can be used to derive the variance ($Var(x) = E(x^2) - E(x)^2$), when $\alpha > 3$, it is defined and its value is finite.

2.3.4 Complementary to Rank-Frequency of Words

Zipf's law has been receiving tremendous attentions in statistical/quantitative linguistics for more than 80 years [24, 32, 42, 107, 108, 123, 124, 153, 157, 189, 190, 241, 242]. While these research narrow power-law distributions in the rank-frequency of words, we discover that power-law distributions also appear in another form of natural language: entity.

Like words and sentences, entities also play an important role in our communicative system. Our discovery of power-law distributions in the length-frequency of entities opens a door to understand our language use, complementary to power-law distributions in the rank-frequency and meaning-frequency of words.

Chapter 3

Data Analysis

In this chapter, we detail our analysis for the intrinsic characteristics of time expressions from four diverse datasets and for the ones of named entities from two benchmark datasets. According to the analysis we report five common characteristics about time expressions and two characteristics about named entities.[1]

3.1 Time Expression Analysis

3.1.1 Time Expression Datasets

We conduct an analysis on the following four diverse datasets: TimeBank, TE3-Silver, WikiWars, and Tweets. TimeBank [164] is a benchmark dataset in the series of the TempEval competitions [214, 217, 219], and it consists of 183 news articles. TE3-Silver is a large-scale dataset with 2,452 news articles that are collected from the Gigaword corpus [152]; its time expressions are automatically labeled by other three time taggers (i.e., TIPSem and TIPSem-B [119], and TRIOS [213]) and it is constructed as a silver dataset in the TempEval-3 competition [214]. The WikiWars dataset is constructed by collecting articles about some famous wars from Wikipedia [132]. Tweets is our manually labeled dataset that consists of 942 tweets, of which each contains one or more time expressions [240].

[1]The content in this chapter has been published as Xiaoshi Zhong, Aixin Sun, and Erik Cambria. Time Expression Analysis and Recognition Using Syntactic Token Types and General Heuristic Rules. In *Proceedings of the 55th Annual Meetings of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 420-429, Vancouver, Canada, 2017 [240], Xiaoshi Zhong and Erik Cambria. Time Expression Recognition Using a Constituent-based Tagging Scheme. In *Proceedings of the 2018 World Wide Web Conference*, pages 983-992, Lyon, France, 2018 [238], and Xiaoshi Zhong, Erik Cambria, and Amir Hussain. Extracting Time Expressions and Named Entities with Constituent-based Tagging Schemes. In *Cognitive Computation*, pp. 1-19, 2020 [239].

Table 3.1: Statistics of the four datasets. A tweet here is viewed as a document.

Dataset	#Documents	#Words	#Timexes
TimeBank	183	61,418	1,243
TE3-Silver	2,452	666,309	12,739
WikiWars	22	119,468	2,671
Tweets	942	18,199	1,127

Specifically, the Tweets dataset is constructed through the following procedure. We randomly sample 4000 tweets and apply SUTime on these tweets, among which 942 tweets contain at least one time expression that identified by SUTime. From the remaining 3,058 tweets, we randomly sample 500 and manually annotate them, finding that only 15 tweets contain time expressions. Therefore, we roughly consider that SUTime misses about 3% time expressions in tweets. Two annotators then manually annotate these 942 tweets with discussion to a final agreement according to the standards of TimeML and TimeBank. Finally, we obtain 1,127 manually labeled time expressions. For these 942 tweets, we randomly sample 200 tweets as the test set, and the remaining 742 as the training set.

Table 3.1 summarizes the statistics of these four datasets.

3.1.2 Time Expression Characteristics

Although these four datasets are diverse from each other in terms of sources, corpus sizes, text types, and domains, we will see that their time expressions demonstrate some similar characteristics. From our analysis, we find five such common characteristics about time expressions.

Characteristic 1 *Time expressions are very short, consisting of about 2 words on average.*

Figure 3.1 plots the distributions of the length of time expressions in the four analyzed datasets. Although these four datasets are quite different from each other in terms of sources (e.g., news articles, Wikipedia articles, and tweets) and corpus sizes (e.g., the numbers of words range from 18,199 to 666,309), the length of their time expressions follows a similar distribution. Most time expressions are very short, with more than 80% of time expressions containing no more than three words and more than 90% containing no more than four words. In particular, the percentages of one-word time expressions range from 36.23% in WikiWars through 40.31% in TimeBank and 53.85% in TE3-Silver to 62.91% in Tweets. This indicates

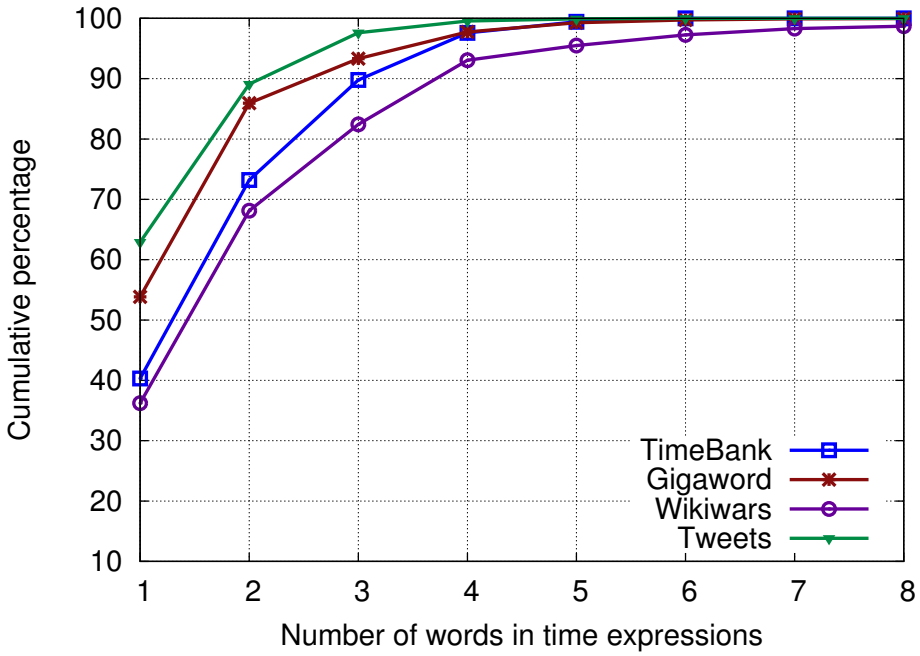


Figure 3.1: Length distributions of time expressions in the four analyzed datasets

Table 3.2: Average length of time expressions in these four datasets

Dataset	Average Length
TimeBank	2.00
TE3-Silver	1.70
WikiWars	2.38
Tweets	1.51

that in informal communication, people tend to use words in minimal length to express time information. Table 3.2 presents the average length of time expressions in each dataset. On average, a typical time expression contains about two words.

Characteristic 2 *Most time expressions contain time token(s). Time tokens can distinguish time expressions from common text while modifiers and numerals cannot.*

Table 3.3 presents the percentage of the three kinds of constituent words of time expressions that appear in time expressions (P_{timex}) and in common text (P_{text}). Here "common text" includes the whole text with time expressions excluded. P_{timex} is defined by Equation (3.1) while P_{text} is defined by Equation (3.2),

$$P_{timex}(T) = \frac{\#timex\ that\ contain\ T}{\#total\ timex} \tag{3.1}$$

Table 3.3: Percentage of the three kinds of constituent words of time expressions that appear in time expressions (P_{timex}) and in common text (P_{text})

Dataset	Time Token		Modifier		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
TE3-Silver	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

$$P_{text}(T) = \frac{\#\textit{tokens that are T}}{\#\textit{total tokens}} \tag{3.2}$$

where $T \in \{$*time token, modifier, numeral*$\}$.

The second column of Table 3.3 shows that most time expressions contain time tokens, with more than 91.8% of time expressions containing at least one time-related token. Some time expressions without time token depend on other time expressions. In the sequence "95 to 100 days," for example, the time expression "95" depends on the time expression "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 demonstrate 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.

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

Characteristic 3 *Only a small group of time words are used to express time information.*

From our analysis on the time expressions in these four datasets, we find that the group of words used to express time information is small.

Table 3.4 presents the number of distinct words and of distinct time tokens. "Distinct" here means ignoring the word variants and frequencies during counting. Words (or tokens) are manually normalized before counting and their variants are ignored. For example, "month,"

Table 3.4: Number of distinct words and distinct time tokens in time expressions

Dataset	No. of Words	No. of Time Tokens
TimeBank	130	64
TE3-Silver	214	80
WikiWars	224	74
Tweets	107	64

"months," and "mths" are treated the same as "month" and are counted only once; similarly, "year" and "5yrs" are counted as the same token "year." Numerals in the counting are ignored. As shown in Table 3.4, although these four datasets vary in sizes, domains, and text types, the numbers of their distinct time tokens are comparable and their sizes are small, with only about 70 (distinct time tokens). That means time expressions highly overlap at their time tokens within an individual dataset.

Across these four datasets, the number of distinct words is 350, which is about half of the simple summation, 675; the number of distinct time tokens in total is 123, less than half of the simple summation, 282. Among the 123 distinct time tokens, 45 appear in all these four datasets, and 101 appear in at least two datasets. This indicates that time tokens, which account for time expressions, are highly overlapped across the four datasets. In other words, time expressions highly overlap at their time tokens.

Characteristic 4 *POS tags cannot distinguish time expressions from common words, but within time expressions, POS tags can distinguish their constituents.*

Table 3.5 lists the top 10 most frequent POS tags that appear in time expressions, and their percentages over the corresponding tags in the whole text, defined by Equation 3.3.

$$Perc(t) = \frac{\textit{number of t in time expressions}}{\textit{number of t in the whole text}} \tag{3.3}$$

where t denotes a POS tag.

Note that the Tweets dataset includes only those 942 manually annotated tweets that contain at least one time expression; if taking into account those tweets that do not contain time expressions (which are 3,058 tweets; see Section 4.4.1), the $Perc$ of the POS tags in Tweets will be much lower. Among these 40 POS tags (10 $\times$ 4 datasets), 36 have the $Perc$ lower than 20%; other 4 POS tags are 3 CD and 1 RB. For each of the TimeBank, TE3-Silver, and

Table 3.5: Top 10 most frequent POS tags that appear in time expressions and their percentages over the corresponding tags in the whole text. $Freq$ denotes the number of times a POS tag appearing in time expressions while $Perc$ denotes the percentage of this POS tag in time expressions over the corresponding tag in the whole dataset.

TimeBank			**TE3-Silver**			**WikiWars**			**Tweets**		
Tag	$Freq$	$Perc$	**Tag**	$Freq$	$Perc$	**Tag**	$Freq$	$Perc$	**Tag**	$Freq$	$Perc$
NN	587	6.66	NNP	6902	8.77	CD	2113	67.85	NN	572	15.27
DT	396	7.16	CD	4582	22.11	NNP	1294	8.87	CD	323	55.40
CD	351	11.60	NN	3233	3.26	NN	783	5.70	RB	189	25.40
JJ	347	8.74	DT	2080	3.15	DT	582	4.67	DT	161	18.05
NNP	336	5.09	JJ	1940	3.82	IN	363	2.48	NNP	155	6.55
NNS	156	4.17	NNS	1356	3.19	JJ	328	3.82	JJ	118	12.38
RB	162	9.44	RB	597	3.52	RB	261	8.23	NNS	116	18.10
IN	76	1.13	IN	512	0.62	NNS	234	3.82	IN	20	1.24
,	20	0.61	,	175	0.56	,	171	2.80	JJR	10	17.86
CC	9	0.60	CC	69	0.38	VBG	28	1.50	VBP	9	2.72

WikiWars datasets, except the CD, all the POS tags have the $Perc$ less than 10%. This indicates that POS tags cannot provide enough information to distinguish time expressions from common words. However, the most common POS tags in time expressions are NN*, JJ, RB, CD, and DT. Within time expressions, time tokens usually have NN* and RB, modifiers have JJ and RB, and numerals have CD. This finding indicates that for time expressions, their similar constituent words behave in a similar syntactic way. When seeing this, I realize that this is exactly how linguists define part-of-speech for language; "linguists group some words of language into classes (sets) which show similar syntactic behaviour" [130]. **This is my eureka moment!** The definition of part-of-speech for language inspires us to define a syntactic type system for the time expression that is part of language (see Section 4.1 for our defined syntactic type system).

Characteristic 5 *Time expressions are formed by loose structure, with more than 53.5% of time tokens appearing in different positions within time expressions.*

We find that time expressions are formed by loose structure and the loose structure mainly exhibits in the following two aspects. Firstly, 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 forms "1 September 2006" by a numeral appearing before it and another time token appearing after it. Secondly, some

Table 3.6: Percentage of distinct time tokens and distinct 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
TE3-Silver	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

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 positions within time expressions, the time token "September" may appear as the (i) beginning or (ii) inside word of a time expression 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 a multi-word time expression when time expressions are modeled by the BILOU scheme.

Table 3.6 presents the percentages of distinct time tokens and distinct modifiers that appear in different positions within time expressions. "Different positions" here 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 distinct time tokens appear in different positions, and under the 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 the BIO scheme or the 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 Section 5.1 for details).

The first four characteristics are related to the principle of least effort [242]. That is, people tend to act with least effort so as to minimize the cost of energy at both individual and collective levels in all the human actions, including language use [242]. Time expressions are part of language and act as an interface of communication. Short expressions, occurrence and distinction, small vocabulary, and similar syntactic behaviour all reduce the cost of energy required for our humans to communicate with each other. The last characteristic demonstrates that the structure of time expressions is flexible.

To summarize: on average, a typical time expression contains two words, among which one is a time token and the other is a modifier or numeral, and the number of total distinct time tokens is small. Therefore, in order to recognize a time expression, we first recognize its time token, then recognize its modifier or numeral.

3.2 Named Entity Analysis

3.2.1 Named Entity Datasets

We conduct an analysis on the following two benchmark datasets for the characteristics of named entities: CoNLL03 and OntoNotes*. The original CoNLL03 and OntoNotes5 copora include the data in English and other languages for named entity analysis and other tasks, but here we focus our analysis on named entity analysis in the English data.

CoNLL03 is a small benchmark dataset that is derived from the Reuters RCV1 corpus, with 1,393 news articles collected between August 1996 and August 1997. This dataset contains 4 entity categories: PER, LOC, ORG, and MISC [178].

OntoNotes* is a dataset that is derived from the large-scale benchmark OntoNotes5 dataset [161]. OntoNotes5 is a portion of the OnteNotes 5.0 corpus for named entity analysis and consists of 3,370 articles that are collected from different sources (e.g., broadcast, newswire, weblogs, and telephone conversation) over a long period of time. It includes 18 entity categories.[2] Although the OntoNotes5 dataset is a benchmark dataset, we find that its annotation is far from perfect. For example, its guideline "OntoNotes Named Entity Guidelines (Version 14.0)" states that the ORDINAL includes all the ordinal numbers and the CARDINAL includes the whole numbers, fractions, and decimals, but we find in the common text 3,588 numeral words, which is 7.1% of the total numeral words. In addition, some sequences are annotated inconsistently. For the sequence "the Cold War," for example, in some cases the whole sequence is annotated as a named entity (i.e., "<ENAMEX>the Cold War</ENAMEX>," where "ENAMEX" is the annotation mark) while in some other cases only "Cold War" is annotated as a named entity (i.e., "the <ENAMEX>Cold War</ENAMEX>").

To get a high-quality dataset for named entity analysis, we derive a dataset termed OntoNotes* from the OntoNotes5 dataset by (1) removing those entity categories whose named entities are

[2]The 18 entity categories in the OntoNotes5 dataset are CARDINAL, DATE, EVENT, FAC, GPE, LANGUAGE, LAW, LOC, MONEY, NORP, ORDINAL, ORG, PERCENT, PERSON, PRODUCT, QUANTITY, TIME, and WORK_OF_ART.

Table 3.7: Statistics of the two datasets. "Whole" indicates the whole dataset.

Dataset	Portion	#Documents	#Words	#Entities	#Categories
CoNLL03	Training Set	946	203,621	23,499	4
	Development Set	216	51,362	5,942	
	Test Set	231	46,435	5,648	
	Whole	**1,393**	**301,418**	**35,089**	
OntoNotes*	Training Set	2,729	1,578,195	81,222	11
	Development Set	406	246,009	12,721	
	Test Set	235	155,330	7,537	
	Whole	**3,370**	**1,979,534**	**101,480**	

Table 3.8: Percentage of named entities that contain at least one word hardly appearing in the common text

	Whole	Training Set	Development Set	Test Set
CoNLL03	97.77	98.77	99.19	98.62
OntoNotes*	92.91	92.20	95.22	95.61

mainly composed of numbers and ordinals[3] and (2) moving all the "the" at the beginning of named entities and all the "'s" at the end of named entities outside their named entities (e.g., all the "<ENAMEX>the Cold War 's</ENAMEX>" are changed to "the <ENAMEX>Cold War</ENAMEX> 's").

When setting training, development, and test sets, we follow the setting by the CoNLL03 shared task [178] for the CoNLL03 dataset and follow the setting[4] by one of OntoNotes5's authors for our OntoNotes* dataset. Table 3.7 summarizes the statistics of these two datasets.

3.2.2 Named Entity Characteristics

Like the above four datasets used to analyze time expressions, these two benchmark datasets are different from each other in terms of corpus size, text genre, and entity categories, but we will see soon that their named entities demonstrate some similar characteristics.

Characteristic 6 *Most named entities contain uncommon word(s), with more than 92.2% of named entities having at least one word that hardly appears in common text.*

[3]The removed entity categories include CARDINAL, DATE, MONEY, ORDINAL, PERCENT, QUANTITY, and TIME.

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

Table 3.8 presents the percentages of named entities that contain at least one word hardly appearing in common text (case sensitive). Here "common text" includes the whole text with named entities excluded. The percentage is calculated within a set that contains named entities and common text, and the set can be a whole dataset (e.g., the CoNLL03 dataset) or only a splitting set (e.g., the training set of the CoNLL03 dataset). Within a set, for a word w, the rate of its occurrences in named entities over its occurrences in the whole text is defined by Equation (3.4):

$$r(w) = \frac{f_{entity}(w)}{f_{entity}(w) + f_{common}(w)} \tag{3.4}$$

where $f_{entity}(w)$ denotes the occurrences of w in named entities while $f_{common}(w)$ denotes the occurrences of w in common text. If $r(w)$ reaches a threshold R, then the word w is treated as hardly appearing in common text. For the CoNLL03 dataset and its splitting sets, R is set by 1, which means that the word does not appear in common text. For the OntoNotes* dataset and its splitting sets, R is set by 0.95, because its annotation is imperfect (as mentioned in Section 3.2.1): its common text contains some words that should be treated as named entities, such as "American."[5] We call such kind of words that mainly appear in named entities and hardly appear in common text *uncommon words*.

From Table 3.8 we can see that for a set, more than 92.2% of named entities contain at least one uncommon word. This phenomenon of uncommon words widely exists in the CoNLL03 and OntoNotes* datasets and their training sets, development sets, 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 these words of its test set that hardly appear in the common text of its training set tend to predict named entities.

Characteristic 7 *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.*

We find that named entities are mainly made up of proper nouns.[6] Table 3.9 lists the top 4 most frequent POS tags appearing in named entities and their percentages over the whole POS

[5]The threshold $R = 0.95$ for the OntoNotes* dataset is an empirical value. We think that the imperfect annotation should be controlled to an acceptable degree.

[6]If we take into account the original OntoNotes5 dataset, then these named entities are mainly made up of proper nouns and cardinal numbers.

Table 3.9: Top 4 most frequent POS tags in named entities and their percentage over the whole tags 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

tags in named entities (p_{entity}) and over the corresponding POS tags in the whole text (p_{whole}). p_{entity} is defined by Equation 3.5 and p_{whole} is defined by Equation 3.6.

$$p_{entity}(t) = \frac{f_{entity}(t)}{\sum_{t_i} f_{entity}(t_i)} \tag{3.5}$$

$$p_{whole}(t) = \frac{f_{entity}(t)}{f_{entity}(t) + f_{common}(t)} \tag{3.6}$$

where t denotes a POS tag, $f_{entity}(t)$ denotes the occurrences of the tag t in named entities while $f_{common}(t)$ denotes the occurrences of the tag t in common text.

From Table 3.9 we can see that the top 4 POS tags in both the CoNLL03 and OntoNotes* datasets 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.[7] Within named entities, these JJ words are mainly the nationality words, such as "American" and "Chinese."

Characteristic 8 *Named entities are formed by loose structure, with more than 53.77% of distinct words that appear in different positions within named entities.*

We find that named entities are also formed by loose structure, similar to time expressions (see Characteristic 5). Table 3.10 presents the percentages of distinct words that appear in

[7]The P_{text} of proper nouns does not reach 100% mainly because an individual dataset concerns certain types of named entities and partly because some NNP* words are incorrectly POS tagged, for example, "SURPRISE DEFEAT" is wrongly tagged as "NNP NNP," but it should be tagged as "JJ NN."

Table 3.10: Percentage of distinct words that appear in different positions within named entities

Dataset	BIO Scheme	BILOU Scheme
CoNLL03	53.77	59.14
OntoNotes*	57.13	79.67

different positions within time expression.[8] The definition of "different positions" is same as the one defined in Section 3.1.2. From Table 3.10 we can see that for each dataset, under the BIO scheme, more than 53.77% of distinct words appear in different positions, and under the BILOU scheme, more than 59.14% of distinct words appear in different positions. The appearance of words in different positions within named entities causes the position-based tagging scheme to suffer from the problem of inconsistent tag assignment, and we need to another appropriate tagging scheme (see Section 6.3 for details).

[8]We here do not report the percentages of different positions of different constituent words of named entities but simply report the ones of distinct words, due to two reasons: firstly, the vocabulary of named entities is large and it is difficult to collect all of them; secondly, the percentage of the different positions of distinct words is enough to reflect the loose structure of named entities.

Chapter 4

SynTime: Time Expression Recognition Using Syntactic Token Types and General Heuristic Rules

SynTime defines a syntactic token-type system for the constituent words of time expressions, and designs a small set of heuristic rules working on these token types. Figure 1.1(a) shows the layout of SynTime, which mainly consists of three levels: token level, type level, and rule level. At the token level, there lie specific tokens and token regular expressions. At the type level, token types group these tokens and token regular expressions. At the rule level, heuristic rules work on token types and are independent of specific tokens. For example, heuristic rules do not work on the tokens "1989" and "February," but work on their token types "YEAR" and "MONTH." In other words, our heuristic rules are designed in a general manner. For this reason, our token types and heuristic rules are independent of specific domains, specific text types, and even specific languages that consist of specific tokens. In this dissertation, we test SynTime on specific domains (i.e., general domain and war domain) and specific text types (i.e., formal text and informal text) in English. Testing on other languages needs to construct a set of token regular expressions in the target languages under our defined token-type system or another defined token-type system.[1]

Figure 4.1 displays the overview of SynTime in practice. As shown on the left-hand side, SynTime is initialized with token regular expressions. After initialization, SynTime can be directly applied on text to recognize time expressions. On the other hand, SynTime can be

[1]The content in this chapter has been published as Xiaoshi Zhong, Aixin Sun, and Erik Cambria. Time Expression Analysis and Recognition Using Syntactic Token Types and General Heuristic Rules. In *Proceedings of the 55th Annual Meetings of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 420-429, Vancouver, Canada, 2017 [240].

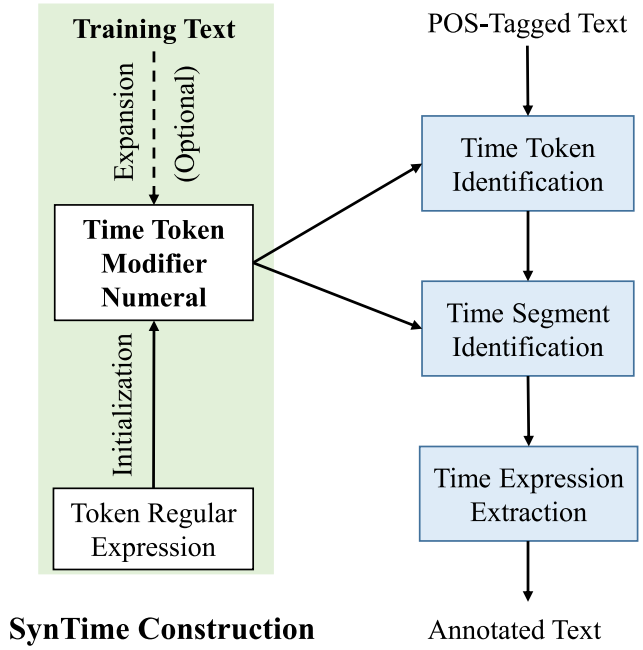


Figure 4.1: Overview of SynTime in practice. The left-hand side shows the SynTime construction, with an initialization using token regular expressions and an optional expansion using the training text. The right-hand side shows the three main steps of how SynTime recognizes time expressions.

easily expanded by simply adding time-related token regular expressions derived from training text under these defined token types. The expansion enables SynTime to recognize time expressions from the text in different domains and different textual types.

As shown on the right-hand side of Figure 4.1, SynTime recognizes time expressions from unstructured text through three main steps. In the first step, SynTime identifies time tokens from the POS-tagged raw text. Around these identified time tokens, in the second step, SynTime searches for modifiers and numerals to form time segments. In the last step, SynTime transforms time segments to time expressions.

4.1 SynTime Construction

We define a syntactic token-type system for the constituent words of time expressions, specifically, 15 token types are defined for time tokens, 5 token types are for modifiers, and 1 token type for numerals. These token types are described below and are summarized in Table 4.1.

Table 4.1: SynTime defines 15 token types for time tokens, 5 token types for modifiers, and 1 token type for numerals. The last column indicates the number of distinct tokens that are grouped under the token type, without counting token variants. "-" indicates that the token type involves changing digits and cannot be counted.

Token Type	Description	Examples	No. of Tokens
	Time Token		
DECADE	decade instances	1910s, 1940s, fifties	-
YEAR	year instances	1970, 1989, 2006	-
SEASON	season instances	Summer, Winter	5
MONTH	month instances	February, September	12
WEEK	day of the week	Monday, Friday	7
DATE	date instances	2016-09-07, 9/2006	-
TIME	time instances	03:45:32, 20:43	-
DAY_TIME	time within a day	morning, afternoon	27
TIMELINE	relative to today	yesterday, tomorrow	12
HOLIDAY	holiday instances	Christmas	20
PERIOD	period instances	daily	9
DURATION	duration instances	5-year	-
TIME_UNIT	time units	year(s)	15
TIME_ZONE	time zones	GMT, UTC	6
ERA	era AD and BC	AD, BC	2
	Modifier		
PREFIX	modifiers appear before time tokens	the, about	48
SUFFIX	modifiers appear after time tokens	ago, old	2
LINKAGE	link two time tokens	and, or, to, -	4
IN_ARTICLE	indefinite articles	a, an	2
COMMA	comma	,	1
	Numeral		
NUMERAL	numbers, ordinals	20, third	-

Token types to tokens is like POS tags to words. For example, "February" has a POS tag of NNP and a token type of MONTH.

Time Token. We define 15 token types for time tokens and use the names for token types similar to the Joda-Time classes[2]: DECADE (-), YEAR (-), SEASON (5), MONTH (12), WEEK (7), DATE (-), TIME (-), DAY_TIME (27), TIMELINE (12), HOLIDAY (20), PERIOD (9), DURATION (-), TIME_UNIT (15), TIME_ZONE (6), and ERA (2). The number in "(·)" represents the number of distinct tokens that are grouped under this token type. "-" indicates that this

[2] http://www.joda.org/joda-time/

token type involves changing digits and cannot be counted.

Modifier. We define 3 token types for modifiers according to their possible positions relative to time tokens. Those modifiers that appear before time tokens are defined as PREFIX (48), while those modifiers that appear after time tokens are defined as SUFFIX (2). LINKAGE (4) links two time tokens. Besides, we define two special token types for modifiers, namely, COMMA (1) for the comma "," and IN_ARTICLE (2) for the two indefinite articles "a" and "an."

TimeML [163] and TimeBank [164] do not treat most prepositions (e.g., "on'' and "at") as part of time expressions. SynTime follows the standards of TimeML and TimeBank and therefore does not group those prepositions under our defined token types.

Numeral. Numbers and ordinals in time expressions can be a time token, such as the "10" in "October 10, 2016," or a modifier, such as the "10" in "10 days." We define the token type NUMERAL (-) to group ordinals and numbers.

SynTime Initialization. SynTime is initiated by importing token regular expressions from SUTime,[3] which is a state-of-the-art rule-based tagger that achieves the highest recall in TempEval-3 [27, 28]. Specifically, we collect from SUTime only its tokens and token regular expressions, and discard its other rules of recognizing full time expressions.

4.2 Time Expression Recognition

SynTime designs a small set of heuristic rules working on these defined token types to recognize time expressions. This recognition process mainly includes three steps: (1) time token identification, (2) time segment identification, and (3) time expression extraction.

4.2.1 Time Token Identification

Identifying time tokens is simple and straightforward, through matching the words in raw text with the token regular expressions grouped in SynTime. Some words might cause ambiguity. For example, the word "May" can be a modal verb, or a noun indicating the fifth month of a year. To filter out these ambiguous words, we employ the information of POS tags, which are obtained by using Stanford POS Tagger.[4] The strategy of using POS tags to identify the instances of defined token types is based on Characteristic 4 that is illustrated in Section 3.1.2.

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

[4] http://nlp.stanford.edu/software/tagger.shtml

In this step, besides the time tokens are identified and assigned with their token types, the modifier and numeral words are also identified and assigned with their token types, if these words are matched with any of the modifier and numeral regular expressions. In the next two steps, SynTime will no longer work on specific tokens, but works on token types.

4.2.2 Time Segment Identification

The task of time segment identification is to search the surroundings of each identified time token for modifiers and numerals, and then gather the time token with its modifiers and numerals to form a time segment. The searching for modifiers and numerals is conducted under some simple heuristic rules in which the key idea is to expand the boundaries of time tokens.

At first, each time token is treated as a time segment. If it is either a PERIOD or a DURATION, then there is no need to further search. Otherwise, search its left-hand side and its right-hand side for modifiers and numerals. For the left-hand side searching, if encounter a PREFIX or a NUMERAL or an IN_ARTICLE, then continue searching. For the right-hand side searching, if encounter a SUFFIX or a NUMERAL, then continue searching. Both the left- and right-hand side searchings stop when encountering a COMMA or a LINKAGE or a non-modifier or non-numeral word (i.e., a word that is neither identified as a time token nor a modifier nor a numeral). The left-hand side searching for a time token does not exceed its previous time token; the right-hand side searching does not exceed its subsequent time token. A time segment consists of exactly one time token and zero or some modifiers or numerals.

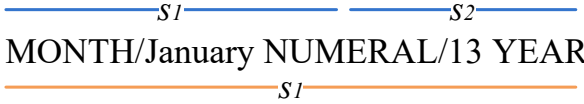
A special kind of time segments does not contain any time token; instead, they depend on other time segments appearing nearby them. For example, the sequence "8 to 20 days" is assigned with the token types "NUMERAL LINKAGE NUMERAL TIME_UNIT," in which "LINKAGE/to NUMERAL/20 TIME_UNIT/days" is identified as a time segment while "NUMERAL/8 LINKAGE/to" is identified as a dependent time segment without any time token (see Figure 4.2(e)).

4.2.3 Time Expression Extraction

The task of time expression extraction is to extract time expressions from the identified time segments, in which the key step is to determine whether to merge two adjacent or overlapping time segments into a new time segment.

(a) A stand-alone time segment is extracted as a time expression

(b) Merge two adjacent time segments into a new time segment



(c) Merge overlapping time segments into a new time segment

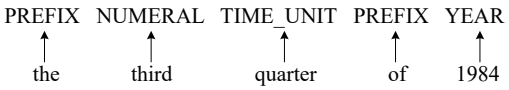
(d) Merge two overlapping time segments into a new time segment

(e) A time segment and a dependent time segment

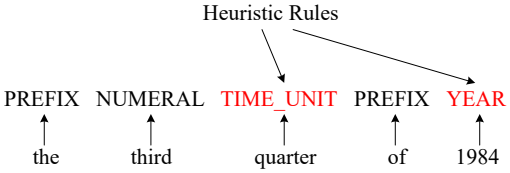
Figure 4.2: Examples of time segments and time expressions. The labels s_1 and s_2 indicate time segments, while the label e_1 indicates time expressions.

We scan the time segments in a sentence from the beginning to the end. A stand-alone time segment is extracted as a time expression (see Figure 4.2(a)). The focus is to deal with two or more time segments that are adjacent or overlapping. If two time segments s_1 and s_2 are adjacent, then merge them to form a new time segment s_1 (see Figure 4.2(b)). Consider the case that s_1 and s_2 overlap at a shared boundary. According to our strategy of time segment identification, the shared boundary can be a modifier or a numeral. If the shared boundary is neither a COMMA nor a LINKAGE, then merge s_1 and s_2 (see Figure 4.2(c)). If the shared boundary is a LINKAGE, then extract s_1 as a time expression and continue scanning. When the shared boundary is a COMMA, merge s_1 and s_2 only if the COMMA's previous token and next token simultaneously satisfy the following three conditions: (1) the previous token is a time token or a NUMERAL, (2) the next token is a time token, and (3) the token types of the previous token and the next token are not the same (see Figure 4.2(d)).

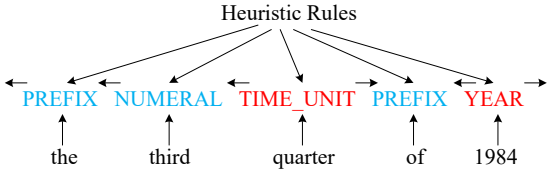
(a) At the beginning, it is a sequence of specific tokens



(b) Firstly, assign these tokens with token types by simply looking up them at the token regular expressions

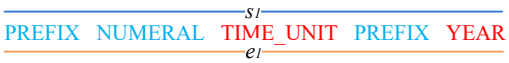


(c) Then, identify time tokens through the records of their token types when assigning token types to tokens

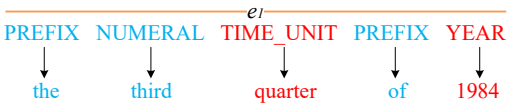


(d) After that, search the surroundings of time tokens for modifiers and numerals to form time segments

(e) Merge two adjacent time segments into a new time segment



(f) Extract a stand-alone time segment a time expression. This time expression is now at the type level



(g) Export a sequence of specific tokens from the sequence of token types as a time expression

(h) Finally, we get the recognized time expression at the token level

Figure 4.3: Key steps of how SynTime recognizes a sequence as a time expression

Although Figure 4.2 shows these examples as the identified token types together with their specific tokens, we should note that heuristic rules only work on these token types and are independent of specific tokens. After the step of time expression extraction, a time expression is exported from a sequences of token types (which is at the type level) as a sequence of specific tokens (which is at the token level) (see Figure 4.3(f), 4.3(g), and 4.3(h)). Figure 4.3 shows eight key steps to demonstrate how SynTime recognizes the sequence "the third quart of 1984" as a time expressions in practice.

4.3 SynTime Expansion

SynTime can be expanded by simply adding new keywords or new token regular expressions under our defined token types without changing any rule. The expansion requires these added keywords and token regular expressions to be annotated manually. We apply the initial SynTime on the manually annotated time expressions from the training text and list those words that are not covered. Whether an uncovered word will be added to SynTime is manually determined. The rule for determination is that the added words can not cause ambiguity and should be generic. The WikiWars dataset contains a few examples like this: "The time Arnold reached Quebec City." Words in this example are extremely descriptive, and we do not collect them. On the other hand, tweets contain many informal variants and abbreviations in time expressions; for example, "2day" and "tday" are two popular spellings of "today." Such kind of informal variants and abbreviations are collected.

According to our analysis described in Characteristic 3, not many words are used to express time information, therefore, the manual addition of keywords will not cost too much effort. In addition, we find that even in tweets people tend to use formal words. In a set of Twitter word clusters that are trained from 56 million English tweets,[5] the most frequent used words are those formal words, and their frequencies are much higher than the ones of informal words. For example, in the cluster of "today,"[6] the most frequent word is the formal one "today," which occurs 1,220,829 times, while the second most frequent one "2day" occurs only 34,827 times. The low rate of informal words (e.g., about only 4% informal words in the "today" cluster) suggests that even in an informal environment, the manual addition of keywords costs little.

4.4 Experiments

We evaluate the quality of SynTime against four state-of-the-art baselines (i.e., HeidelTime, SUTime, ClearTK-TimeML (short as "ClearTK" for convenience), and UWTime) on three datasets (i.e., TE-3, WikiWars, and Tweets). WikiWars is a domain-specific dataset about famous wars. TE-3 and WikiWars are the two datasets in formal text while the Tweets dataset is in informal text. In experiments, we implement SynTime in two versions: SynTime-I

[5] http://www.cs.cmu.edu/~ark/TweetNLP/cluster_viewer.html

[6] http://www.cs.cmu.edu/~ark/TweetNLP/paths/01111110010.html

and SynTime-E. SynTime-I is the initial version, while SynTime-E is an expanded version of SynTime-I by adding keywords under our defined token types from the training text.

4.4.1 Experimental Setup

Datasets. We use the following three datasets in our experiments: TE-3, WikiWars, and Tweets. The TE-3 dataset uses the TimeBank corpus as its training set and the TE3-Platinum corpus as its test set. TimeBank consists of 183 news articles and TE3-Platinum consists of 20 new articles; they are comprehensive corpora in formal text and are described in the TempEval-3 competition [214]. Although these two corpora are used in the same competition, they are collected independently. WikiWars is a domain-specific dataset in formal text, consisting of 22 English Wikipedia articles about famous wars [132]. Tweets is our manually labeled dataset that are collected from Twitter [240]. The three datasets are detailed in Section 3.1.1.

Compared Methods. We compare SynTime with the following state-of-the-art methods: HeidelTime [199], SUTime [27], ClearTK [12] and UWTime [103]. HeidelTime and SUTime are rule-based methods and use predefined deterministic rules and achieve the best results in the relaxed match, while ClearTK [12] uses a CRFs framework with the BIO scheme and achieves the best result in the strict match in the TempEval-3 competition [214]. UWTime uses combinatory categorial grammar (CCG) to predefine linguistic structure for time expressions and achieves better results than HeidelTime on the TE-3 and WikiWars datasets [103]. When testing HeidelTime on the Tweets dataset, we use its Colloquial setting which is designed for informal text. When training ClearTK and UWTime on the Tweets dataset, we try the following two settings: (1) training it on only the training set of Tweets, (2) training it on the TimeBank dataset and the Tweets training set together. The second setting achieves slightly better results and we report the results of this setting.

Evaluation Metrics. We follow the TempEval-3 competition and use its evaluation toolkit[7] to report results in terms of *Strict Match* and *Relaxed Match* [214] under the three standard metrics:$Precision$ ($Pr.$), $Recall$ ($Re.$), and F_1. Strict match means exact match between the recognized time expressions and the ground-truth time expressions while relaxed match means that there exist certain overlap between the recognized ones and the ground-truth ones. $Pr.$,

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

$Re.$ and f_1 are defined by Equations (4.1), (4.2), and (4.3), respectively.

$$Pr. = \frac{TP}{TP + FP} \tag{4.1}$$

$$Re. = \frac{TP}{TP + FN} \tag{4.2}$$

$$F_1 = \frac{2 \times Pr. \times Re.}{Pr. + Re.} \tag{4.3}$$

where TP (true-positive) denotes the number of time expressions that are recognized by the model and simultaneously appear the ground-truth, FP (false-positive) denotes the number of time expressions that are recognized by the model but do not appear in the ground-truth, while FN (false-negative) denotes the number of time expressions appearing in the ground-truth but are not recognized by the model.

4.4.2 Experimental Results

Table 5.2 presents the overall performance of SynTime and the four baselines on the three datasets. Among the total 18 measures, SynTime-I and SynTime-E achieve 11 best results and 12 second best results. Except the strict match on the WikiWars dataset, both SynTime-I and SynTime-E achieve the F_1 above 91%. For the relaxed match on all the three datasets, SynTime-I and SynTime-E achieve the recalls above 92%. The high recalls are consistent with Characteristic 2 that more than 91.81% of time expressions contain at least one time token (see Table 3.2). This indicates that SynTime covers most of time tokens. On the Tweets dataset, SynTime-I and SynTime-E achieve exceptionally good performance; their F_1 reaches 91.74% with an absolute 11.37% improvement in the strict match and 95.87% with an absolute 6.33% improvement in the relaxed match. The reasons are that in the informal environment people tend to use time expressions in their minimum length (62.91% one-word time expressions in Tweets; see Figure 3.1), the size of time-related keywords is small (only 64 distinct time tokens; see Table 3.4), and even in tweets people tend to use formal words (see Section 4.3 for our finding about informal variants and abbreviations from a set of Twitter word clusters). For the precision, SynTime-I and SynTime-E achieve comparable results with the baselines in the strict match and performs slightly poorer in the relaxed match.

Next we discuss the comparison between the initial version SynTime-I and the four compared methods as well as the comparison between the expanded version SynTime-E and SynTime-I.

Table 4.2: Overall performance of SynTime and the four baselines on the three datasets. Within each metric, the best result is highlighted in boldface while the second best is underlined. Some results are reported directly from their original papers indicated by the references.

Dataset	Method	Strict Match			Relaxed Match		
		$Pr.$	$Re.$	F_1	$Pr.$	$Re.$	F_1
TimeBank	HeidelTime [203]	83.85	78.99	81.34	93.08	87.68	90.30
	SUTime [28]	78.72	80.43	79.57	89.36	91.30	90.32
	ClearTK[12]	85.90	79.70	82.70	93.75	86.96	90.23
	UWTime [103]	86.10	80.40	83.10	**94.60**	88.40	91.40
	SynTime-I	91.43	92.75	92.09	94.29	**95.65**	**94.96**
	SynTime-E	**91.49**	**93.48**	**92.47**	93.62	**95.65**	94.62
WikiWars	HeidelTime[198]	**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
	ClearTK	87.69	80.28	**83.82**	96.80	90.54	**93.56**
	UWTime [103]	**87.70**	78.80	83.00	**97.60**	87.60	92.30
	SynTime-I	80.00	80.22	80.11	92.16	92.41	92.29
	SynTime-E	79.18	**83.47**	81.27	90.49	**95.39**	92.88
Tweets	HeidelTime	**89.58**	72.88	80.37	95.83	77.97	85.98
	SUTime	76.03	77.97	76.99	88.43	90.68	89.54
	ClearTK	86.83	75.11	80.54	96.59	83.54	89.59
	UWTime	88.54	72.03	79.44	**96.88**	78.81	86.92
	SynTime-I	89.52	94.07	91.74	93.55	98.31	95.87
	SynTime-E	89.20	**94.49**	**91.77**	93.20	**98.78**	**95.88**

SynTime-I vs. Compared Methods. On the TimeBank dataset, SynTime-I achieves the F_1 of 92.09% in the strict match and the one of 94.96% in the relaxed match. On the Tweets dataset, SynTime-I achieves the F_1 of 91.74% and 95.87%, respectively. It outperforms all the baseline methods. The reason is that for the two rule-based time taggers, their rules are designed in a fixed way that lacks flexibility. For example, SUTime can recognize the time expression "1 year" but not the one "year 1." For the two learning-based baseline, some of their features actually hurt the modeling. Time expressions involve quite many changing digits which by themselves affect the pattern recognition and modeling learning. For example, it is difficult to build a connection between the two time expressions "May 22, 1986" and "February 01, 1989" at the word level or the character level. One suggestion is to consider a type-based learning method that can use the type information. For example, the above two time expressions refer to the same pattern of "MONTH NUMERAL COMMA YEAR" at the level of token types. Part-of-speech (POS) is a kind of type information; the above two time expressions refer to the same

Table 4.3: Number of time tokens and modifiers added for expansion

Dataset	No. of Time Tokens	No. of Modifiers
TimeBank	3	5
WikiWars	16	21
Tweets	3	2

pattern of "NNP CD , CD." According to our analysis, however, POS tags cannot distinguish time expressions from common words (see Characteristic 4). Features need carefully designing. On the WikiWars dataset, SynTime-I achieves competitive results in both matches. The reason is that time expressions in the WikiWars dataset include many prepositions and quite a few descriptive time expressions. SynTime cannot fully recognize these kinds of time expressions because it follows the standards of TimeML and TimeBank.

SynTime-E vs. SynTime-I. Table 4.3 lists the number of time tokens and modifiers that are added to the SynTime-I so as to get the SynTime-E. On the TimeBank and Tweets datasets, only a few tokens are added, the corresponding results are affected slightly. This confirms that the number of time tokens is small, and that SynTime-I covers most time tokens. On the WikiWars dataset, because much more tokens are added, SynTime-E performs much better than SynTime-I, especially in the recall. SynTime-E improves the recall by absolute 3.25% in the strict match and by absolute 2.98% in the relaxed match. This indicates that with more words added from specific domains (e.g., the WikiWars dataset about war), SynTime can significantly improve the performance.

4.5 Limitations

There are two possible limitations in SynTime. Firstly, SynTime assumes that all the time expressions appearing in text are correct. In daily life, however, people might write an invalid time expression (e.g., "31 February 2008") and SynTime cannot exclude such invalid time expressions but instead recognizes them as valid ones. Secondly, SynTime assumes that words are tokenized and POS tagged correctly. In reality, however, the tokenized and tagged words are not that perfect, due to the limitation of the used tool. For example, Stanford POS Tagger assigns VBD to the word "sat" in the sequence "friday or sat" while the word should be tagged as NNP. These incorrect tokenized tokens and POS tags affect the final performance.

Chapter 5

TOMN: Time Expression Recognition with A Constituent-based Tagging Scheme

TOMN defines a constituent-based tagging scheme to model time expressions under a framework of conditional random fields (CRFs). Figure 5.1 displays the overview of TOMN that mainly includes three parts: TOMN scheme, TmnRegex, and time expression recognition. The TOMN scheme consists of four tags. TmnRegex is a set of regular expressions about time-related tokens. Time expressions are modeled under a CRFs framework with the help of TmnRegex and the TOMN scheme as well as minimal features derived from context according to their characteristics described in Section 3.1.2.[1]

5.1 TOMN Scheme

Characteristic 5 states that time expressions are formed by loose structure and suggests us to explore an appropriate tagging scheme to model time expressions. We therefore define a constituent-based tagging scheme termed TOMN scheme with four tags: T, O, M, and N; they indicate the constituent words of time expressions, namely *time tokens*, modifiers, *numerals*, and the words appearing *outside* time expressions.

Conventional tagging schemes like the BIO scheme[2] [179] and the BILOU scheme[3] [170] are based on *the positions within labeled chunks*. BIO indicates the beginning, inside, and outside words of a chunk; BILOU indicates a unit-word chunk, and the beginning, inside, last

[1]The content in this chapter has been published as Xiaoshi Zhong and Erik Cambria. Time Expression Recognition Using a Constituent-based Tagging Scheme. In *Proceedings of the 2018 World Wide Web Conference*, pages 983-992, Lyon, France, 2018 [238].

[2]The BIO scheme denotes the standard IOB2 scheme described in [179].

[3]The BILOU scheme is also widely known as the IOBES scheme and the BIOES scheme.

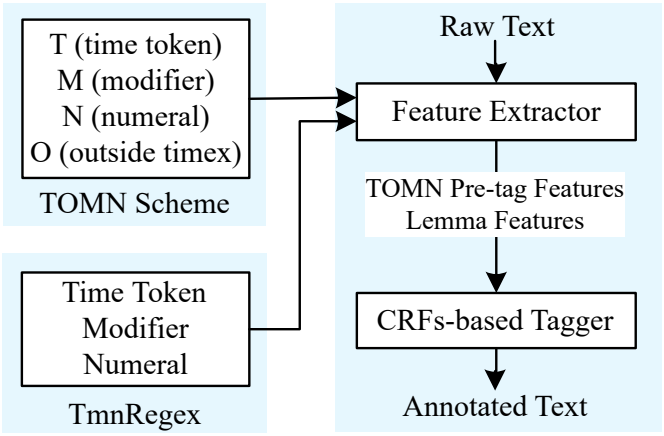


Figure 5.1: Overview of TOMN. Top-left side shows the TOMN scheme, consisting of four tags. Bottom-left side is the TmnRegex, a set of regular expressions for time-related words. Right-hand side shows the time expression modeling, with TmnRegex and TOMN scheme.

words of a multi-word chunk. By contrast, our TOMN scheme is based on *the constituents of labeled chunks*, indicating the constituent words of time expressions. Next, we use the BILOU scheme as the representative of these conventional position-based tagging schemes for analysis.

Using the BILOU scheme for time expression modeling leads to the problem of inconsistent tag assignment.[4] Characteristic 5 demonstrates that time expressions are formed by loose structure which exhibits in the two aspects of loose collocations and exchangeable order. Under the BILOU scheme, both loose collocations and exchangeable order lead to 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 1.2(a)). These four "September" have the same word (i.e., the word itself) and express the same meaning (i.e., the ninth month of a year), 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 that is assigned with different tags causes confusion for statistical models to model the word. If a word is assigned with different tags in an equal number, then the word itself cannot provide

[4] A typical supervised-learning procedure involves tag assignment in two stages: feature extraction during the training stage and sequence tagging during the test stage. We focus on the training stage to analyze the impact of tag assignment in different types of tagging schemes.

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 2 which states 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 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 text with 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 collected from text with the style of "September 1, 2006" in which "September" should be predicted as B.

Our TOMN scheme instead 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 Section 5.2) and the same word plays the 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, our 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 1.2(b)). We can see that these four "September" are consistently assigned with the same tag of "T" and statistical models need only to model them as "T," without any confusion. With consistent tag assignment, our TOMN scheme protects the predictive power of time tokens and avoids the potential tag imbalance.

In addition, our TOMN scheme models a word by fewer tags than the BILOU scheme. The BILOU scheme typically models a time token by four tags (i.e., U, B, L, or I) and models a modifier or numeral by five tags (i.e., U, B, L, I, or O), while our TOMN scheme models a time token by only one tag (i.e., T) and models a modifier or numeral by two tags (i.e., M or N if the modifier or numeral appears inside time expressions and O if it appears outside time expressions). Compared with the BILOU scheme, our TOMN scheme reduces the computational complexity for training a model.

5.2 TmnRegex

Characteristic 3 indicates that only a small group of words that are used in time expressions. TOMN employs three time-related token types, namely time token, modifier, and numeral, to group those words. The three token types are the same as the ones defined in SynTime [240] and correspond to three of the above four tags (i.e., T, M, and N) defined in the TOMN scheme.

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 time expression "the last month." Numerals include ordinals and numbers, except those that are recognized as year (e.g., "2006"). Token types are defined on top of specific tokens themselves and are not necessarily relevant to their context. For example, "2006" alone expresses time information, so it is treated as a time token; on the other hand, although the "1" in the time expression "1 September 2006'' implies the day, itself alone does not express time information, so it is treated as a numeral.

These three token types with those words they group constitute a set of token regular expressions, which is denoted by TmnRegex. TmnRegex is constructed by importing token regular expressions for its time token, modifier, and numeral from the state-of-the-art rule-based time tagger SUTime.[5] Like SynTime, TmnRegex collects from SUTime only the regular expressions at the level of tokens and discards its regular expressions for the whole time expressions. In summary, TmnRegex contains only 115 distinct time tokens, 57 distinct modifiers, and 58 numerals, without counting those words with changing digits.

5.3 Time Expression Recognition

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

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

5.3.1 Feature Extraction

The features we extract for time expression modeling include two kinds: TOMN pre-tag features and lemma features. During the feature extraction, we use w_i to denote the i-th word in the text.

TOMN Pre-tag Features. Characteristic 2 states 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 our consideration, they are treated as pre-tag features under our 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 pre-tagged by N; other common words are pre-tagged by O. The assignment of pre-tags for words is conducted by simply looking up them at the token regular expressions grouped in TmnRegex.

The last four columns of Table 3.3 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 those words that are pre-tagged as modifiers and numerals (i.e., those words with pre-tags of M or N; they are simply denoted by M/N in the remaining of this paragraph) to record whether or not 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 the time expression "last two months," the modifier "last" (M) together with the modifier "two" (N) modifies the time token "months" (T). This checking is a loop searching relying on the identified time tokens. For each identified time token (i.e., those words with a pre-tag of T), we search its left-hand side without exceeding the previous time token and search its right-hand side without exceeding the next time token. When searching a side of a time token, if encounter a M/N, then record this M/N and continue searching; if encounter a word that is not a 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 while those M/N that do not modify any time token will not recorded. For example, the modifier "two" (M) in the time expression "two months" is recorded because it modifies the time token "months" (T); by contrast, in the sequence "two apples," the modifier "two" (N) will not recorded because it does not modify any time token but only modifies the common word "apples." The checking result is treated as a feature for modeling.

For the TOMN pre-tag features, we extract them in a 5-word window of the current word w_i for 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 only consider whether the current word w_i is recorded or not.

Table 5.1: Extracted features for word w_i in named entity modeling

1.	TOMN pre-tag features 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}
2.	If w_i is a M or N, then check whether or not it directly or indirectly modifies any time token
3.	Lemma features in a 5-word window of w_i

In the training phase, we consider the TOMN pre-tag features for only those words appearing in labeled time expressions. In the test phase, we extract the TOMN pre-tag features for all the words in the whole text.

Lemma Features. The lemma features include the word shape in a 5-word window of w_i, 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 its lemma is set by its token type. For example, the lemma of "20:16" is set by TIME. We use the following five special token types as lemma for those 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 by the same lemma TIME at the type level.

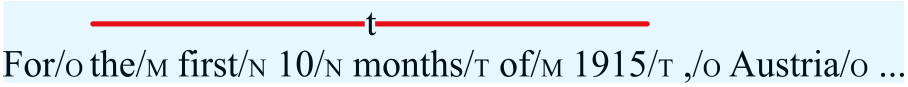
The lemma features are extracted for all the words in the whole text in both the training phase and the test phase.

We do not consider the features of characters and word variants because they cannot help build connections among time expressions but hurt the modeling learning and pattern recognition. For example, "Sept." is an abbreviation of "September" and both of them express the same meaning but computer does not treat them as the same thing.

We also do not consider the POS features and other syntactic features. Characteristic 4 indicates that POS tags cannot distinguish time expressions from common text, and our experiments confirm that adding POS tags as features does not improve the performance. On the other hand, Characteristic 5 shows that time expressions are formed by loose structure, which together with Characteristic 4 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 learning method, which already considers the dependency, to distinguish time expressions from common text. We therefore do not use those syntactic features in our model.

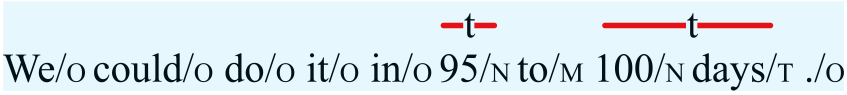
Table 5.1 summarizes the features that are extracted for the word w_i for time expression modeling. Typically up to 11 features are extracted for a word.

(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 5.2: Examples of time expression extraction. The label t indicates time expressions.

Feature Values. For the TOMN pre-tag features, we extract them as separate features with binary values. The theory of scales of measurement suggests that non-ordinal attributes should be transformed onto separate dimensions [195].The TOMN pre-tag features and the checking features are non-ordinal, therefore, they are extracted as separate features. For the lemma features, we follow their traditional use to incorporate multiple values under a feature.

5.3.2 Model Learning and Tagging

TOMN models time expressions under a CRFs framework [98] with the extracted features described above. In implementation, we use Stanford Tagger[6] to obtain the lemma features and use CRFSuite[7] with the default setting for model learning and sequence tagging. During sequence tagging, one 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 pre-tag features as well as in sequence tagging as the labeling tags.

Time Expression Extraction. After sequence tagging, those T, M, and N words (i.e., non-O words) that appear together are extracted as a time expression. See Figure 5.2(a) and 5.2(b).

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

A special kind of modifiers (i.e., the linkers "to," "-," "or," and "and") separates those non-O words into two or more parallel time expressions. See Figure 5.2(c) and 5.2(d).

5.4 Experiments

We conduct experiments to evaluate the quality of TOMN on three datasets (i.e., TE-3, WikiWars, and Tweets) in comparison with five state-of-the-art methods (i.e., HeidelTime, SUTime, SynTime, ClearTK and UWTime).

5.4.1 Experimental Setup

Datasets. We use the same three datasets in our experiments as the ones used in SynTime. These three datasets are detailed in Section 3.1.1 and 4.4.1.

Baseline Methods. We compare TOMN with the five state-of-the-art methods: HeidelTime [199], SUTime [27], ClearTK [12], UWTime [103], and SynTime [240]. The first four methods are described in Section 4.4.1 while SynTime is our first method proposed for time expression recognition. In implementation, 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 results under *Strict Match* and *Relaxed Match* in the three standard metrics: $Precision$ ($Pr.$), $Recall$ ($Re.$), and F_1. They are the same as the metrics described in Section 4.4.1 and defined by Equations (4.1), (4.2), and (4.3).

5.4.2 Experimental Results

Table 5.2 reports the overall performance of TOMN and the five compared methods on the three experimental datasets. Among the total 18 measures, TOMN achieves 13 best or second best results. It performs better than SynTime which achieves 10 best or second best results, and much better than other four baselines which achieve at most 4 best or second best results. For each measure, TOMN achieves either the best result or a comparable result with the best result. Especially for the F_1, TOMN performs the best in the strict F_1 on the Tweets dataset and in the relaxed F_1 on the WikiWars dataset. For other F_1, TOMN achieves the comparable results

Table 5.2: Overall performance of TOMN and the five baselines on the three experimental datasets. Within each metric, the best result is highlighted in boldface while the second best is underlined. Some results are reported directly from their publicly available sources.

Dataset	Method	Strict Match			Relaxed Match		
		$Pr.$	$Re.$	F_1	$Pr.$	$Re.$	F_1
TE-3	HeidelTime[203]	83.85	78.99	81.34	93.08	87.68	90.30
	SUTime[28]	78.72	80.43	79.57	89.36	91.30	90.32
	SynTime[240]	91.43	**92.75**	**92.09**	94.29	**95.65**	**94.96**
	ClearTK[12]	85.90	79.70	82.70	93.75	86.96	90.23
	UWTime[103]	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[198]	**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[240]	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[103]	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[240]	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

compared to the corresponding best results; most of the differences between their performance are less than 0.5%.

5.4.3 TOMN vs. Baseline Methods

We further compare TOMN with the rule-based baselines and the learning-based baselines.

TOMN vs. Rule-based Baselines. On the TE-3 and Tweets datasets, TOMN achieves comparable results with SynTime. On the WikiWars dataset, TOMN achieves the F_1 with absolute 2.0% $\sim$ 2.3% increase in comparison with SynTime. This indicates that compared with SynTime, TOMN is equally effective on comprehensive data and more effective on domain-specific data. The reason is that the heuristic rules of SynTime are greedy for recalls at the cost of precisions, and such cost is expensive when it comes to domain-specific data. TOMN instead leverages statistical information from the whole corpus, which might miss some rare

time expressions but helps recognize time expressions more precisely; especially in domain-specific data, the statistical information significantly improves the precisions at little cost of recalls. For HeidelTime and SUTime, except the strict F_1 on the WikiWars dataset, TOMN outperforms the two baselines on all the three datasets, with up to absolute 15.3% increase in recalls and up to absolute 12.0% increase in F_1. The reason is that these deterministic rules of HeidelTime and SUTime are designed in fixed manners that lack flexibility [240].

TOMN vs. Learning-based Baselines. Except the strict F_1 on the WikiWars dataset, TOMN outperforms ClearTK and UWTime on all the three datasets in all the recalls and all the F_1. Especially on the TE-3 and Tweets datasets, TOMN improves the recalls by at least absolute 9.8% in the strict match and at least absolute 5.1% in the relaxed match, and improves the F_1 by at least absolute 8.5% in the strict match and at least absolute 3.1% in the relaxed match. The reasons are that (1) the fixed linguistic structure that are predefined in UWTime cannot fully capture the loose structure of time expressions, (2) the BIO scheme used in ClearTK suffers from the problem of inconsistent tag assignment and reduces the predictive power of time tokens, and (3) some of their features (e.g., POS tags and syntactic dependency features) actually hurt the modeling learning and pattern recognition. For the strict F_1 on the WikiWars dataset, TOMN performs slightly poorer than these two learning-based methods, because TOMN uses the same token regular expressions as SynTime and follows TimeBank and SynTime to exclude most prepositions (except "of") from time expressions while some time expressions in the WikiWars dataset include those prepositions.

5.4.4 Cross-dataset Performance

We conduct a series of cross-dataset experiments to evaluate the robustness of TOMN in comparison with the two learning-based methods that require training. In cross-dataset experiments, a method is trained on the training set of one dataset and then tested on the test sets of other datasets. Since these three datasets (i.e., TE-3, WikiWars, and Tweets) used in our experiments are quite diverse, the cross-dataset experiments can evaluate the robustness of a learning-based method. Table 5.3 presents the cross-dataset performance on the test set of the TE-3 dataset; Table 5.4 presents the performance on the test set of WikiWars; Table 5.5 on the test set of Tweets. For a convenient comparison, Table 5.3, 5.4, and 5.5 also present the performance on the single-dataset experiments. "Single-dataset" here means that the training set and the test set

Table 5.3: Cross-dataset performance on the test set of TE-3. "Training" indicates the dataset whose training set is used for training. Colored 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 5.4: 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

belong to the same dataset. The results of the single-dataset experiments are reported directly from Table 5.2, and they are indicated by the colored background in Table 5.3, 5.4, and 5.5.

On the test set of TE-3, TOMN achieves at least 84.0% in the strict F_1 and at least 93.4% in the relaxed F_1 (see the rows of WikiWars and Tweets in Table 5.3). On the test set of Tweets, TOMN achieves at least 85.5% in the strict F_1 and at least 94.3% in the relaxed F_1 (see the rows of TE-3 and WikiWars in Table 5.5). It significantly outperforms ClearTK and UWTime. On the test set of WikiWars, TOMN achieves comparable results with ClearTK and UWTime in the relaxed match but performs poorer than UWTime in the strict match. Especially when trained on the training set of Tweets, TOMN achieves only 63.0% in the strict F_1, which is absolute 7.5% lower than the one of UWTime (see the rows of TE-3 and Tweets in Table 5.4). Tweets contains

Table 5.5: 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**

many short time expressions (62.9% one-word time expressions; see Figure 3.1) 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, when TOMN is trained on the training set of Tweets, it cannot fully recognize the long and descriptive time expressions in the test of WikiWars. UWTime instead predefines some linguistic structure, which contributes significantly to the exact recognition of those long and descriptive time expressions.

Let us look at the single-dataset and cross-dataset performance in the relaxed match. TOMN achieves similar performance, regardless of which dataset it is trained on. Specifically, in the relaxed F_1, TOMN achieves about 93.9% on the test of TE-3, about 91.6% on the test set of WikiWars, and about 94.8% on the one of Tweets. By contrast, ClearTK and UWTime perform relatively well on the single-dataset experiments but much worse on the cross-dataset experiments. Especially on the test of Tweets, their relaxed F_1 drops from at least 87.0% when they are trained on the training set of Tweets to at most 80.3% when they are trained on the training sets of other datasets. This demonstrates that TOMN is much more robust than ClearTK and UWTime.

The robustness of TOMN can be explained by Characteristics 2 and 3. Characteristic 2 indicates that time tokens are capable of predicting time expressions and Characteristic 3 indicates that time expressions highly overlap at their time tokens within an individual dataset and across different datasets. That means, the time tokens from one dataset can help recognize

Table 5.6: Performance of controlled experiments for the impact of factors. "BIO" denotes the systems that replace the TOMN labeling tags by the BIO tags while "BILOU" denotes the systems that replace by the BILOU tags. "$trad$" indicates the traditional strategy for time expression extraction while "$nono$" indicates the non-O strategy. "$-$" indicates that this kind of features that are removed from TOMN. "PreTag" denotes the TOMN pre-tag features while "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**

the time tokens from other datasets. Therefore, in terms of the relaxed match, the cross-dataset performance should be comparable to the single-dataset performance.

5.4.5 Factor Analysis

We conduct controlled experiments to analyze the impact of the TOMN scheme as labeling tags as well as the impact of the features that are used in TOMN. The experimental results are presented in Table 5.6.

Impact of the 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 the TOMN scheme to the BIO scheme to get a BIO system and change to the 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.[8]

The tag assignment of the 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 the BIO scheme while it is assigned with U under the BILOU scheme. When extracting time expressions from a 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 the BILOU scheme is extracted as a time expression. The other strategy follows the one used for TOMN in which those consecutive non-O words are extracted as a time expression (see Section 5.3.2). The traditional strategy is denoted by "$trad$" while the non-O strategy is denoted by "$nono$." The results of the BIO and BILOU systems are reported as respective "BIO" and "BILOU" in Table 5.6. 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 as follows. Time expressions on average contain about two words (see Characteristic 1); in that case, the BILOU scheme is reduced approximately to the BLOU scheme and the BIO scheme is changed approximately to the BLO scheme. Between the BLOU scheme and the BLO scheme there is only slight difference; under the impact of inconsistent tag assignment and TOMN pre-tag features, this slight difference affects slightly to the performance. In what follows we do not distinguish the BILOU scheme from the BIO scheme and do not distinguish the non-O strategy from the traditional strategy; the four methods of BIO$_{trad}$, BIO$_{nono}$, BILOU$_{trad}$, and BILOU$_{nono}$ are simply represented by "BILOU."

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

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

On the Tweets dataset, TOMN and BILOU achieve similar performance; the difference between their performance in most measures is less than 1%. The reason is that 62.9% of time expressions in the Tweets dataset are one-word time expressions (see Figure 3.1) and 96.0% of time expressions contain time tokens (see Characteristic 1), and they together indicates that these one-word time expressions contain only time tokens. In that case, the TOMN scheme is reduced approximately to the TO scheme and the BILOU scheme is reduced approximately to the UO scheme. Then the UO scheme becomes a constituent-based tagging scheme in which U models time tokens. It is equivalent to the TO scheme. (In that case, the BIO scheme is reduced approximately to the BO scheme in which B models time tokens. Then the BO scheme is equivalent to the TO scheme as well as the UO scheme.)

Impact of the TOMN Pre-tag Features. To analyze the impact of the TOMN pre-tag features, we remove them from TOMN. After they are removed, although most of the precisions increase and even reach the highest scores, all the recalls and F_1 drop dramatically, with absolute 10.4% $\sim$ 32.9% decreases in recalls and absolute 4.8% $\sim$ 19.1% decreases in F_1. That means the TOMN pre-tag features significantly improve the performance and confirms the predictive power of time tokens. The results also validate that pre-tagging features is a good way to use the information of those lexicon. And our constituent-based TOMN scheme keeps the pre-tagging assignment consistent, just like the way that it keeps the labeling tag assignment consistent.

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

5.4.6 Computational Efficiency

We briefly discuss the computational efficiency of TOMN in comparison with the five state-of-the-art baselines. HeidelTime, SUTime, SynTime, ClearTK, and TOMN are implemented by the Java language, while UWTime is implemented by Python. For the rule-based methods, HeidelTime and SUTime run nearly in real time, and SynTime runs in real time. Table 5.7

Table 5.7: Running time that TOMN and the two learning-based baselines cost to complete a whole process, including both training and test (unit: seconds)

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

presents the running time that TOMN and the learning-based baselines cost to complete a whole process (including both training and test) on the three datasets on a Mac OS laptop (1.4GHz Processor and 8GB Memory). In practice, UWTime implements both time expression recognition and normalization, while ClearTK and TOMN implement only the time expression recognition. Different programming languages and different concerning tasks might be factors that affect the computational efficiency, however, from Table 5.7 we still can see that TOMN is more efficient than ClearTK and UWTime. Considering only the test, TOMN runs in real time.

5.5 Discussion

The analysis of time expressions can explain many empirical observations that are reported in other works about time expression recognition. For example, UzZaman et al. report that using an extra large-scale of dataset does not improve the performance of time expression recognition [214]; Bethard reports that using the TimeBank dataset alone performs better than using the TimeBank and AQUAINT datasets together on time expression recognition [12]; Filannino et al. report that features of gazetteers, shallow parsing, and propositional noun phrases do not contribute a significant improvement on time expression recognition [58]. These observations can be explained by the characteristics illustrated in Section 3.1.2. Characteristics 2, 3, 4, and 5 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 many empirical observations that are reported in other works about the impact of the BIO (or IOB2) and BILOU (or IOBES) schemes in named entity recognition and classification (NERC). Ratinov and Roth report that the BILOU

scheme outperforms the BIO scheme on the MUC-7 and CoNLL03 NERC datasets [170]; Dai et al. report that the IOBES scheme performs better than the IOB2 scheme in drug name recognition [45]. When looking at their results, however, we find that those improvements are rather slight, most of them are less than 1%; in some cases, the BIO scheme performs better than the BILOU scheme. Lample et al. confirm that they do not observe the significant improvement of the IOBES scheme over the IOB2 scheme on the CoNLL03 NERC dataset [99]. These observations can be explained by our analysis of tagging schemes in Section 5.1 and 5.4.5. Basically, the BIO and BILOU schemes are based on *the positions within labeled chunks* and implicitly assume that target entities should be formed by a 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 the CoNLL03 (English NERC) dataset [178] as an example. We find that for each of the CoNLL03's 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; 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, which is 53.5% (see Characteristic 5); the percentage 93.7% is similar to the one of time expressions that contain time tokens, which is 91.8% (see Characteristic 2); the length distribution is similar to the one of time expressions in the Tweets dataset (see Characteristic 1). That means named entities demonstrate some common characteristics similar to time expressions. 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 are approximately 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 categories: PER, LOC, ORG, and MISC), we find that some named entities are annotated with different entity categories. In its 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 NERC systems achieve the F_1 at around 94.5% on the development set and around 91.5% on the test set [36, 99, 120, 121, 154, 170, 231], and why more than 10 years' effort improves the F_1 by only 0.8% on the development set (from 2003's 93.9% [64] to current 94.7% [121]) and by only 2.9% on the test set (from 2003's 88.7% [64] to current 91.6% [36]). 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 the 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.

Chapter 6

UGTO: Named Entity Recognition with Uncommon Words and Proper Nouns

Characteristics 6 and 7 suggest that for a dataset, those words of its development set and test set that hardly appear in the common text of its training set tend to predict named entities, and those words are mainly proper nouns. This is our main idea for named entity recognition. Figure 6.1 visualizes this idea with a simple example: in the unannotated test set, those words like "Boston" and "Reuters" that hardly appear in the common text of the annotated training set tend to predict named entities. Such words are also called *uncommon words* and they include two kinds: the first kind of uncommon words appears in the named entities of the training set (e.g., "Boston" and "Africans") while the second kind does not (e.g., "Reuters"). The remaining of this chapter illustrates how we develop our idea in UGTO.[1]

UGTO models named entities under a CRFs framework and follows a typical CRFs procedure. Figure 6.2 shows the overview of UGTO in practice. It mainly includes four components: (1) uncommon word induction, (2) word lexicon, (3) UGTO scheme, and (4) named entity modeling, with the help of uncommon words, word lexicon, and the UGTO scheme.

6.1 Uncommon Word Induction

For each dataset, we induce two kinds of uncommon words from the *annotated training set* and the *unannotated test set*.

[1]The content in this chapter has been published as Xiaoshi Zhong, Erik Cambria, and Amir Hussain. Extracting Time Expressions and Named Entities with Constituent-based Tagging Schemes. In *Cognitive Computation*, pp. 1-19, 2020 [239].

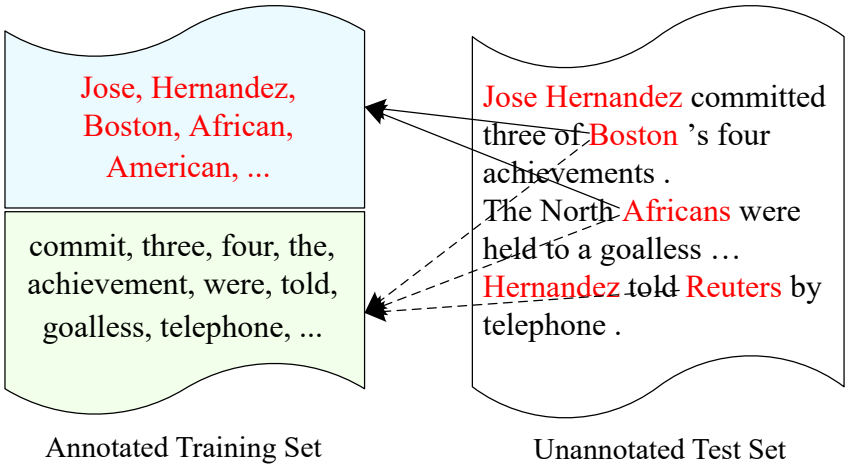


Figure 6.1: Main idea: those words (red font) of unannotated test set that hardly appear in the annotated the common text of the training set (bottom-left) are likely to predict named entities. Such words include two kinds: the first kind (e.g., "Boston") appears in the annotated named entities of the training set (top-left) while the second kind (e.g., "Reuters") does not. The training set is highlighted by colored background that means annotated. The test set instead is unannotated. Solid arrow denotes appearing in the named entities of the training set while dashed arrow denotes hardly appearing in the common text of the training set.

The first kind of uncommon words is induced from the annotated training set. At first, there is an empty list L. For each word w in the named entities of the training set, we calculate its rate ($r(w)$) of hardly appearing in the common text of the training set by Equation (3.4). If $r(w)$ reaches a threshold R, then we add w to L. Like the setting in Section 3.2.2, R is set to 1 for the CoNLL03 dataset and to 0.95 for the OntoNotes* dataset.

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 recognize out-of-vocabulary named entities. This kind of uncommon words can be viewed as the information derived from 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.

6.2 Word Lexicon

Word lexicon includes two kinds of entity-related words: entity tokens and modifiers. Entity tokens are collected from external sources: some entity tokens are from the entity list provided

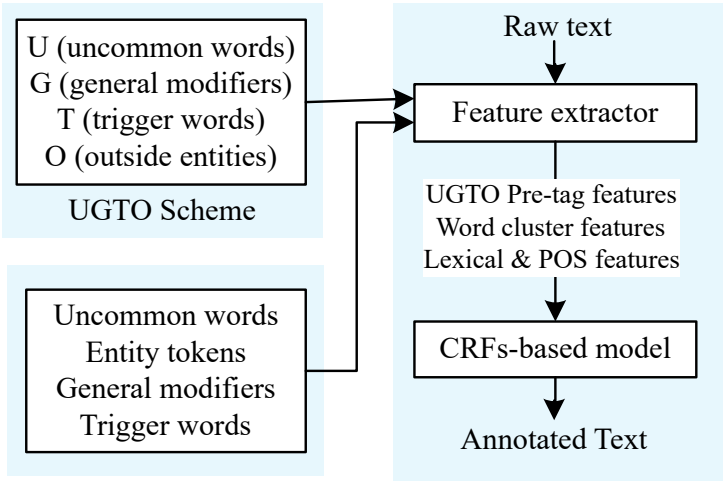


Figure 6.2: Overview of UGTO in practice. The top-left side shows the UGTO scheme that consists of four tags. The bottom-left side are the uncommon words and word lexicon. The right-hand side shows named entity modeling, with the help of the UGTO scheme and uncommon words and word lexicon.

by the CoNLL03 shared task [178] and some are from Wikipedia.[2] Modifiers are collected from the training set according to the annotation guideline of the dataset; they include two kinds: generic modifiers and trigger words. Generic modifiers can modify several categories of entity tokens, such as "of" and "and," while trigger words modify a specific categories of entity tokens, such as "Mr." modifying PER entity tokens and "Inc" modifying ORG entity tokens.

For these entity tokens, we put all of them together, without using their entity categories (e.g., PER, LOC, and ORG), so as to remove the impact of semantic information carried in their entity categories. For the trigger words, we separate 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 use lexicon in word sequences [91, 171], we use lexicon in words. For example, we do not use "Boston University" but use "Boston" and "University." This strategy leads our model to a high coverage and more efficient. For example, with n distinct words, our model can identify up to n one-word sequences, n^2 two-word sequences, n^3 three-word sequences, etc. Furthermore, during feature extraction, our model needs only to

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

Table 6.1: Statistics of word lexicon

Word Lexicon	Number
Entity Token	9,658
Generic Modifier	17
PER Trigger Word	31
Other Trigger Word	116

scan the text word by word, avoiding the difficulty of choosing which algorithm for sequence matching and its computational cost.

Table 6.1 summarizes the statistics of the word lexicon. Note that these word lexicon is collected with only a little effort.

6.3 UGTO Scheme

Characteristic 8 states that named entities are formed by loose structure and suggests to explore another appropriate tagging scheme. We then design another constituent-based tagging scheme termed UGTO scheme to encode uncommon words and word lexicon for named entity modeling. The UGTO scheme consists of four tags: U, G, T, and O; they indicate and encode the constituent words of named entities. Specifically, U encodes uncommon words and entity tokens. G encodes generic modifiers while T encodes trigger words. O encodes those words outside named entities.

6.4 Named Entity Modeling

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

6.4.1 Feature Extraction

The features we extract for named entity modeling include three kinds: UGTO pre-tag features, word cluster features, and basic lexical & POS features. During feature extraction, the i-th word in text is denoted by w_i.

UGTO Pre-tag Features. The UGTO pre-tag features are designed to encode the information of uncommon words and word lexicon under our UGTO scheme. Specifically, a word is encoded by U if it satisfies two conditions: (1) it appears in the list L induced in Section

Table 6.2: Extracted features for the word w_i for named entity modeling

1	UGTO pre-tag features in a 5-word window of w_i, namely pre-tags 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 set of 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, its POS tag

6.1 (i.e., the first kind of uncommon words) or does not appear in the common text of the training set (i.e., the second kind of uncommon words[3]); (2) it has a POS tag of NNP* or is matched by an entity token 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 the generic modifiers. A word is encoded by TP if it is matched by any of the PER trigger words. A word is encoded by T if it is matched by other trigger words.

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

Word Cluster Features. Previous works have demonstrated that word clusters are useful for many information extraction tasks [109, 134]. We follow these works to derive the prefix paths of 4, 8, and 12 bits from a set of hierarchical word clusters as features for a word. In practice, we use the publicly available word clusters "bllip-clusters"[4] for the CoNLL03 dataset and use the one[5] trained by the OntoNotes 5.0 corpus [161] for the OntoNotes* dataset.

Lexical & POS Features. The lexical & POS features are widely used for named entity modeling and we extract 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; (3) its POS tag.

Feature Values. Similar to the setting of feature values for TOMN described in Section 5.3.1, the UGTO pre-tag features and word cluster features are treated as separate features with binary values, while the basic lexical and POS features are incorporated with multiple values under a features.

Table 6.2 summarizes the features extracted for w_i for named entity modeling. For the UGTO pre-tag features and lexical & POS features, we extract them in a 5-word window of

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

[4]http://people.csail.mit.edu/maestro/papers/bllip-clusters.gz

[5]https://drive.google.com/file/d/0B2ke42d0kYFfN1ZSVExLNlYwX1E/view

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 current w_i.

6.4.2 Model Learning and Sequence Tagging

UGTO models named entities with the above extracted features under a CRFs framework. Similar to TOMN described in Chapter 5, in experiments, UGTO uses Stanford Tagger[6] to obtain the information of word lemma and POS tags and uses a Java version of CRFSuite[7] with its default parameters as the CRFs framework for model learning and sequence tagging. Note that the UGTO scheme is used in two different phases with two different functional uses: (1) during feature extraction, it is used as pre-tags to encode uncommon words and word lexicon; (2) during model learning and sequence tagging, it is used as labeling tags.

After model learning and sequence tagging, we extract named entities from these tagged sequences. For those models that exclude entity categories from labeling tags, the U, G, and T words (i.e., non-O words) that appear together form a named entity (see Figure 6.3(a), 6.3(b), and 6.3(c)). For those models that incorporate entity categories into labeling tags, the consecutive non-O words that are tagged with the same entity category together form a named entity (see Figure 6.3(d), 6.3(e), and 6.3(f)).

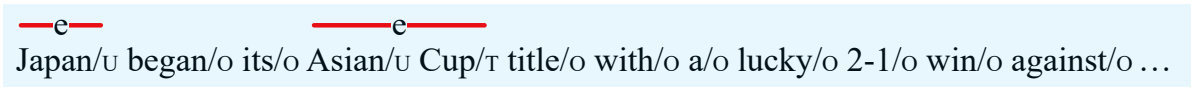
6.5 Experiments

6.5.1 Experimental Setup

Datasets. The two benchmark datasets we use for the experiments of named entity recognition are CoNLL03 [178] and OntoNotes* [161]. They are detailed in Section 3.2.1.

Compared Methods. The compared methods include two representative state-of-the-art methods: StanfordNER [60] and LSTM-CRF [99]. StanfordNER derives hand-crafted features under CRFs with the BIO scheme. LSTM-CRF derives automatic features learned by long short-term memory networks (LSTMs) [82] under CRFs with the IOBES scheme. We use StanfordNER as the representative of those traditional hand-crafted-feature methods and LSTM-CRF as the representative of those auto-learned-feature methods.

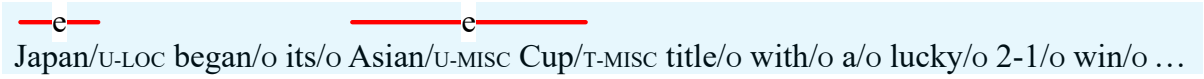
[6] http://nlp.stanford.edu/software/tagger.shtml
[7] http://www.chokkan.org/software/crfsuite/



(a) Consecutive non-O words together form a named entity

(b) Consecutive non-O words together form a named entity

(c) Consecutive non-O words together form a named entity



(d) Consecutive words that are tagged with the same entity type form a named entity

(e) Consecutive words that are tagged with the same entity type form a named entity

(f) Consecutive words that are tagged with the same entity type form a named entity

Figure 6.3: Examples of named entities extracted from tagged sequences. The label e indicates named entities. The first three examples (i.e., 6.3(a), 6.3(b), and 6.3(c)) demonstrate the extraction in the models that exclude entity categories from labeling tags during model learning and sequence tagging, while the last three (i.e., 6.3(d), 6.3(e), and 6.3(f)) demonstrate the extraction in the models that incorporate entity categories into labeling tags.

Evaluation Metrics. We use the evaluation toolkit of the CoNLL03 shared task [178] to report results under the three standard metrics: $Precision$ ($Pr.$), $Recall$ ($Re.$), and F_1.

These three evaluation metrics are similar to the ones defined by Equations (4.1), (4.2), and (4.3), respectively, except the meanings of TP, FP, and FN. In named entity recognition, TP (true-positive) denotes the number of named entities that are recognized by the model and simultaneously appear the ground-truth, FP (false-positive) denotes the number of named entities that are recognized by the model but do not appear in the ground-truth, while FN (false-negative) denotes the number of named entities appearing in the ground-truth but are not recognized by the model.

6.5.2 Experimental Design

We design two kinds of experiments to evaluate UGTO against the two representative 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.*

Experiment 1 is a pure task of named entity recognition, in which a model excludes entity categories from labeling tags in the whole process. Designing this experiment is to test the quality of UGTO against the two baselines. The labeling tags of UGTO include {U, G, T, O}; the ones of StanfordNER include {B, I, O}; the ones of LSTM-CRF include {I, O, B, E, S}.

Experiment 2 is a joint talk of named entity recognition and classification, in which a model incorporates entity categories into labeling tags during modeling and tagging. Designing this experiment is to answer the question: *whether does named entity classification enhance named entity recognition during modeling?* In this experiment, the labeling tags of a model include the combinations of basic tags and entity categories. Specifically, the labeling tags of UGTO on the CoNLL03 dataset include thirteen tags { U-PER, U-LOC, U-ORG, U-MISC, G-PER, G-LOC, G-ORG, G-MISC, T-PER, T-LOC, T-ORG, T-MISC, O }. Similarly, the labeling tags for StanfordNER and LSTM-CRF on the CoNLL03 dataset include the combinations of their basic tags and entity categories, such as B-PER, B-LOC, B-ORG, B-MISC, I-LOC, and O.

We are mainly concerned with and report only the performance of named entity recognition. For Experiment 2, after named entities are extracted from tagged sequences, we convert them to the CoNLL-style format and remove their entity categories so as to report the performance of named entity recognition. We do the same conversion for both UGTO and the two baselines.

6.5.3 Experimental Results

Table 6.3 reports the overall performance of UGTO and the two baselines in named entity recognition on the two datasets.[8]

UGTO$_{w/o}$ vs. Baselines in Experiment 1: UGTO$_{w/o}$ outperforms StanfordNER$_{w/o}$ and LSTM-CRF$_{w/o}$ on both the CoNLL03 and OntoNotes* datasets in recalls and F_1. Specifically, UGTO$_{w/o}$ reduces 3.86%$\sim$14.00% of errors in F_1. Compared with StanfordNER$_{w/o}$ which mainly treats the named entities of the training set as a kind of dictionary, UGTO$_{w/o}$ explicitly

[8]Note that Table 6.3 and 6.4 report only the performance on named entity recognition, without named entity classification.

Table 6.3: Named entity recognition performance of UGTO and baselines. "w/o" indicates Experiment 1 and "$w/type$" indicates Experiment 2. † indicates that the improvement of our result over the best one of baselines is statistically significant ($p < 0.05$ under t-test).

Dataset	Method	Development 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**†

takes into account both the named entities and common text of the 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 the literature, LSTM-CRF significantly outperforms StanfordNER on the joint talk of named entity recognition and classification [99], however, it performs comparably with or worse than UGTO$_{w/o}$ and StanfordNER$_{w/o}$ on the pure named entity recognition. This indicates that simple hand-crafted-feature methods can achieve state-of-the-art performance on named entity recognition.

Experiment 2 vs. Experiment 1: For each of UGTO and the two baselines, we compare its performance in Experiment 2 with its performance in Experiment 1. On both the CoNLL03 and OntoNotes* datasets, UGTO$_{w/type}$ and UGTO$_{w/o}$ perform similarly and comparably; StanfordNER$_{w/type}$ and StanfordNER$_{w/o}$ perform similarly and comparably; LSTM-CRF$_{w/type}$ and LSTM-CRF$_{w/o}$ also perform similarly and comparably. That means that the joint task of named entity recognition and classification does not improve the performance of named entity recognition, in both our model and the two state-of-the-art methods.

6.5.4 Factor Analysis in Experiment 1

We conduct controlled experiments to analyze the impact of UGTO labeling tags and features that are used in UGTO. Their results are reported in Table 6.4.

Table 6.4: 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	Development 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

Impact of UGTO Labeling Tags: To analyze the impact of the UGTO labeling tags, we replace them by the BIO tags (as well as the IOBES tags) and keep other factors unchanged. The BIO and IOBES schemes achieve similar results and we report the results of the BIO scheme as a representative. UGTO$_{w/o}$ performs better than BIO, because the BIO and BILOU schemes suffer from the problem of inconsistent tag assignment, while the UGTO scheme overcomes this problem [238].

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

Impact of Word Cluster Features: Word cluster features are helpful in UGTO (about 0.45% improvement) but are not significant as their impact in some other works [109, 134, 151, 170]. The reason is that the UGTO pre-tag features play a similar role as word clusters in improving the coverage and connecting words at an abstraction level.

6.6 Error Analysis

There is a limitation in UGTO: when extracting named entities from tagged sequence, UGTO would wrongly treat several consecutive entities as a named entity. Comparing the two examples in Figure 6.3(c) and 6.3(f), for example, UGTO wrongly extracts the two named entities "Australian" and "Tom Moody" as a named entity "Australian Tom Moody."

Chapter 7

Power-law Distributions in Length-Frequency of Entities

When analyzing time expressions and named entities, we find that their lengths follow a family of power-law distributions. Furthermore, we discover that power-law distributions widely appear in the length-frequency of entities in seventeen languages and different types of entities.[1]

7.1 Real-world Datasets

The real-world datasets we use to analyze the length-frequency of entities include the following two types: (1) entities in different languages and (2) different types of entities. These datasets contain annotated entities and we directly collect their annotated entities for analysis, including both their training sets and test sets.

7.1.1 Entities in Different Languages

This type of datasets includes entities in seventeen languages: Arabic, Chinese, Dutch, English, Finnish, French, German, Italian, Japanese, Norwegian, Polish, Portuguese, Russian, Spanish, Swahili, Swedish, and Turkish. They are collected from the ACE04 corpus [51] and the HeiNER inventory [226]. Specifically, the Arabic entities are collected from the Arabic subset of the ACE04 corpus and other sixteen are from the HeiNER inventory.

The ACE04 corpus is developed for several linguistic tasks in multiple languages, in this section, we are mainly concerned with entity analysis and therefore collect its Arabic entities for

[1]The content in this chapter is under review as Xiaoshi Zhong, Erik Cambria, and Jagath C. Rajapakse. Power-law Distributions in Length-Frequency of Entities. Submitted to *Nature Communications*.

the analysis in different languages. The HeiNER inventory contains entities that are collected from Wikipedia articles. It has two versions: complete version and disambiguation version. The complete version keeps all the entities, while the disambiguation version removes some entities so as to disambiguate those different entities that have the same meaning. Since our goal is to analyze the length-frequency distributions of real-world entities and the complete version can better reflect their use in reality, we use the complete version in our analysis.

Among the seventeen languages, Chinese and Japanese are those languages that do not use spaces separating their words, i.e., *non-spaced languages*, while other fifteen are those languages that use spaces separating their words, i.e., *word-spaced languages*. For the word-spaced languages, the length of their entities can be calculated directly. For the two non-spaced languages, we firstly employ word segmentation tools to segment their entities and then calculate the entity length. Specifically, we employ the Stanford CoreNLP, [2] a widely used tool for processing language text, to segment the Chinese entities, and employ the Japanese tokenizer Nagisa [3] to segment the Japanese entities.

Table 7.1 summarizes the statistics of these entities in seventeen languages. The length of an entity is defined by the number of its words, denoted by l. From the table we can see that the sizes of these entities are diverse, ranging from 30,742 (Swahili) and 44,284 (Arabic) to 9,755,151 (German) and 43,652,029 (English). The maximal l of these entities ranges from 10 (Swahili) to 41 (Arabic) and even 66 (Japanese). However, the average length of their entities is comparable; on average, an entity contains about 2 words.

7.1.2 Different Types of Entities

This type of datasets broadly includes those datasets that are developed for the analyses of named entities [178], entity mentions (including anchor text) [114], time expressions [164], aspect terms [160], literary entities [9], informal entities [175], and domain-specific entities [209] that have been well investigated in various areas related to computational linguistics and natural language processing.[4] Because English is the most studied language in computational linguistics

[2] https://github.com/xszhong/CoreNLP

[3] https://github.com/taishi-i/nagisa

[4] In this paper, we use "different types of entities" or "entity types" to represent named entities, entity mentions, time expressions, aspect terms, etc., while use "different categories of entities" or "entity categories" to represent the predefined labels (e.g., PERSON, LOCATION, and ORGANIZATION) that are assigned to specific entities. In our analysis of entity length, we are concerned with "different types of entities" and do not care about "entity category."

Table 7.1: Statistics of entities in seventeen languages. Entity length l is defined by the number of words in an entity.

Language	**No. of Entities**	**Max** l	**Average** l
Arabic	44,284	41	2.15
Chinese	1,959,726	19	1.45
Dutch	3,818,743	27	2.20
English	43,652,029	39	2.53
Finnish	1,379,140	38	1.72
French	7,900,615	35	2.53
German	9,755,151	25	1.97
Italian	4,966,688	25	2.04
Japanese	6,904,116	66	2.56
Norwegisch	1,353,658	27	1.81
Polish	3,968,006	27	2.00
Portuguese	2,896,665	29	2.22
Russian	2,145,902	27	2.34
Spanish	4,241,548	33	2.16
Swahili	30,742	10	1.93
Swedish	1,919,149	22	1.78
Turkish	633,935	25	1.92

and natural language processing, we analyze these broad entities in English. We try to find as many datasets as we could access that are developed for entity analysis, and finally collect ten datasets about entities. The ten datasets are briefly described below.

ABSA contains two corpora that are used in SemEval-2014 [160] and SemEval-2015 [159] for aspect-based sentiment analysis. We consider and collect their aspect terms.

ACE04 [51] is a benchmark dataset used for the 2004 Automatic Content Extraction (ACE) technology evaluation. It consists of various types of data collected from different sources (e.g., newswire and broadcast news) for the analyses of entities and relations in three languages: Arabic, Chinese, and English. Its Arabic entities are used for the analysis of entities in different languages, as described in Section 7.1.1. Its English entities are used for the analysis of entities in different types of entities.

BBN [223] consists of Wall Street Journal articles for pronoun co-reference and entity analysis. It includes 12 named entity categories (e.g., PERSON and ORGANIZATION), nine nominal entity categories (e.g., ANIMAL and PLANT), and seven numeric categories (DATE and TIME). We collect and analyze all of its entities, without considering the entity categories.

Bioinformatics contains fourteen corpora that are developed for the analysis of biomedical entities: AnatEM [166], BC2GM [192], BC5CDR [222], BioNLP09 [92], BioNLP11 [169], BioNLP13CG [168], BioNLP13GE [94], BioNLP13PC [150], CHEMDNER [96], CRAFT [7], ExPTM [167], JNLPBA [93], LINNAEUS [70], NCBI-Disease [52]. Crichton et al. collect these fourteen corpora and we can get the dataset from their paper [44].

CoNLL03 [178] is a benchmark dataset with 1,393 news articles derived from the Reuters RCV1 Corpus, which is collected between the period of August 1996 and August 1997. It includes four entity categories, but we simply collect its entities and ignore the categories for length-frequency analysis.

LitBank [9] is a dataset collected from 100 different English-language literary articles across over a long period of time and it is annotated according to ACE04 entity categories for the analysis of literary entities.

OntoNotes5 [161] is a large-scale of dataset collected from different sources (e.g., newswire and web data) over a long period of time for the comprehensive analyses of syntax, co-reference, proposition, word sense, and named entities in three languages (i.e., English, Chinese, and Arabic). Here we are concerned with its entities in English. The set of English entities in the OntoNotes5 dataset contains 3,637 articles and includes 18 entity categories (e.g., PERSON, LOCATION, GPE, and NORP). Similarly, we just collect its entities and ignore the entity categories for the length-frequency analysis.

TimeExp consists of three corpora that are developed for time expression analysis: TempEval-3 (including TimeBank [164], TE3-Silver, AQUAINT, and Platinum corpus [214]), WikiWars [132], and Tweets [238, 240].

Twitter consists of two corpora: WNUT16 [196] and Broad Twitter Corpus [48]. These two corpora are collected from Twitter for the analysis of entities in informal text.

WikiAnchor [114] treats the anchor text (i.e., the text in the hyperlinks) from Wikipedia (20110513 version) as entity mentions.

Table 7.2 summarizes the statistics of these entities in each dataset. For each of these datasets that include several corpora (i.e., Twitter, ABSA, TimeExp, and Bioinformatics), we simply merge all the entities from the whole corpora. Note again that we collect these entities and ignore their entity categories for length-frequency analysis. From the table we can see that the size of entities and the maximal l of entities in these datasets are diverse dramatically, while

Table 7.2: Statistics of different types of entities

Dataset	Entity Type	No. of Entities	Max l	Average l
ABSA	aspect terms	9,979	21	1.45
ACE04	named entities	29,949	57	2.43
BBN	named entities	98,427	15	1.26
Bioinformatics	biomedical entities	450,729	86	1.80
CoNLL03	named entities	35,087	14	1.45
LitBank	literary entities	13,912	129	2.93
OntoNotes5	named entities	155,413	28	1.85
TimeExp	time expressions	18,484	22	1.80
Twitter	informal entities	20,515	14	1.39
WikiAnchor	anchor text	2,690,849	49	2.10

their average l are comparable, ranging around 2 (words). The length-frequency distributions of these entities are analyzed in Section 7.2.2 and plotted in Figure 7.2.[5]

7.2 Power-law Distributions in Length-Frequency of Entities

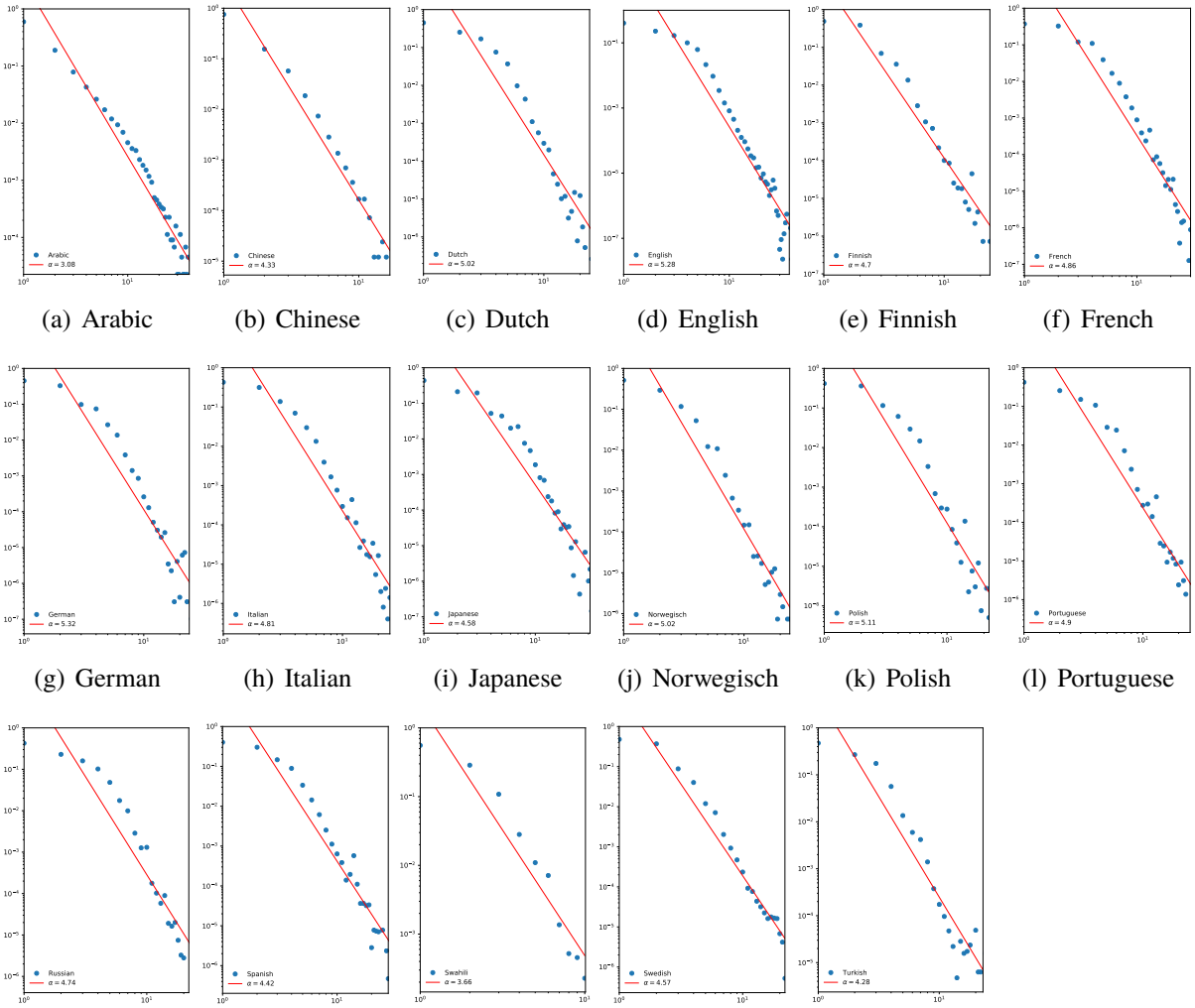
This section displays the power-law distributions we discover in the length-frequency of entities in different languages and different types of English entities. As described in Section 7.1, although these datasets are dramatically different form each other in terms of language, source, domain, text genre, generated time, corpus size, entity category, and annotation criterion, the length-frequency of their entities follows a similar power-law distribution, as defined by Equation (7.1).

$$p(l) = Kl^{-\alpha} \tag{7.1}$$

where l denotes the number of words in an entity and $p(l)$ denotes the percentage of the l-length entities over the whole entities; the constant K is unimportant and the scaling exponent α is of interest. Equation (7.1) can be written as Equation (7.2).

$$p(l) \propto l^{-\alpha} \tag{7.2}$$

[5]In our analysis, we do not care about the categories of entities. Or, we demonstrate that although these entities are assigned with different entity categories, the length-frequency of these entities follows a similar power-law distribution.

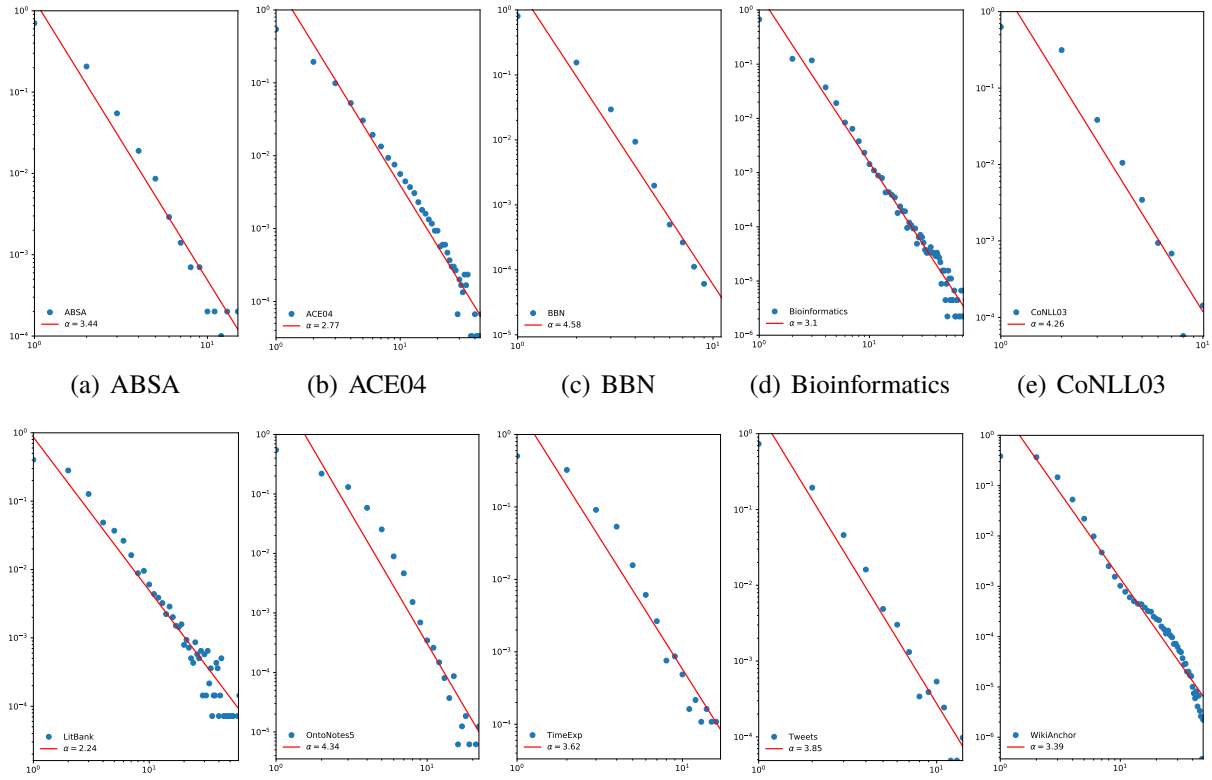


(a) Arabic (b) Chinese (c) Dutch (d) English (e) Finnish (f) French

(g) German (h) Italian (i) Japanese (j) Norwegisch (k) Polish (l) Portuguese

(m) Russian (n) Spanish (o) Swahili (p) Swedish (q) Turkish

Figure 7.1: Power-law distributions in the length-frequency of entities in different languages. The horizontal axis indicates the entity length (l), while the vertical axis indicates the percentage ($p(l)$). (a) Arabic (α = 3.08), (b) Chinese (α = 4.33), (c) Dutch (α = 5.02), (d) English (α = 5.28), (e) Finnish (α = 4.7), (f) French (α = 4.86), (g) German (α = 5.32), (h) Italian (α = 4.81), (i) Japanese (α = 4.58), (j) Norwegisch (α = 5.02), (k) Polish (α = 5.11), (l) Portuguese (α = 4.9), (m) Russian (α = 4.74), (n) Spanish (α = 4.42), (o) Swahili (α = 3.66), (p) Swedish (α = 4.57), and (q) Turkish (α = 4.28). The scaling exponent α ranges from 3.08 to 5.32, indicating that these power-law distributions possess a relatively stable scaling property. All the α are greater than 3, indicating that all these power-law distributions have well-defined means and finite variances.



(a) ABSA (b) ACE04 (c) BBN (d) Bioinformatics (e) CoNLL03

(f) LitBank (g) OntoNotes5 (h) TimeExp (i) Twitter (j) WikiAnchor

Figure 7.2: Power-law distributions in the length-frequency of entities in different types of entities. (a) ABSA (α=3.44), (b) ACE04 (α = 2.77), (c) BBN (α = 4.58), (d) Bioinformatics (α = 3.10), (e) CoNLL03 (α = 4.26), (f) LitBank (α = 2.24), (g) OntoNotes5 (α = 4.34), (h) TimeExp (α = 3.62), (i) Twitter (α = 3.85), (j) WikiAnchor (α = 3.44). All the α are greater than 2, indicating that all these power-law distributions have defined means. Except ACE04, all the α are greater than 3, indicating finite variances in their corresponding distributions.

7.2.1 Power-law Distributions in Length-Frequency of Entities in Different Languages

Figure 7.1 plots the length distributions of entities in the seventeen analyzed languages in a log-log scale. We can see that the length-frequency of entities in these datasets follows a similar power-law distribution. (For the curve fitting, we use the least square algorithm to estimate the parameter i.e., α.) Specifically, the power laws fit the length-frequency distribution of the Arabic entities where the scaling exponent $\alpha = 3.08$, fit the one of the Chinese entities where $alpha = 4.33$, fit the one of the Dutch entities where $\alpha = 5.02$, fit the one of English where $\alpha = 5.28$, fit the one of Finnish where $\alpha = 4.7$, fit the one of French where $\alpha = 4.86$, fit the one

of German where $\alpha = 5.32$, fit the one of Italian where $\alpha = 4.81$, fit the one of Japanese where $\alpha = 4.58$, fit the one of Norwegisch where $\alpha = 5.02$, fit the one of Polish where $\alpha = 5.11$, fit the one of Portuguese where $\alpha = 4.9$, fit the one of Russian where $\alpha = 4.74$, fit the one of Spanish where $\alpha = 4.42$, fit the one of Swahili where $\alpha = 3.66$, fit the one of Swedish where $\alpha = 4.57$, and fit the one of Turkish where $\alpha = 4.28$.

Although the sizes of the entities in these languages range from 30,742 (Swahili) to 43,652,029 (English), their scaling exponent α range only from 3.08 to 5.32. This indicates that these power-law distributions have a relatively stable scaling property. In addition, all the α are greater than 3, indicating that the power-law distributions in all the seventeen languages have well-defined means and finite variances [148] (see Section 2.3.3).

7.2.2 Power-law Distributions in Length-Frequency of Entities in Different Types of Entities

Figure 7.2 plots the entity length distributions in different types of entities in the log-log scale. We can see again that the length-frequency of entities in these ten datasets follows a family of power-law distributions. Specifically, the power laws fit the length-frequency distribution of ABSA's entities where $\alpha = 3.44$, fit the one of ACE04's where $\alpha = 2.77$, fit the one of BBN's where $\alpha = 4.58$, fit the one of Bioinformatics's where $\alpha = 3.10$, fit the one of CoNLL03's where $\alpha = 4.26$, fit the one of LitBank's where $\alpha = 2.24$, fit the one of OntoNotes5's where $\alpha = 4.34$, fit the one of TimeExp's where $\alpha = 3.62$, fit the one of Twitter's where $\alpha = 3.85$, and fit the one of WikiAnchor's where $\alpha = 3.44$.

Although the ten datasets contain different entity types (see Table 7.2) and entity categories (e.g., PERSON and LOCATION) differing from each other in many aspects (e.g., domain and corpus size), the length-frequency of their entities follows the same family of power-law distributions, where their scaling exponents α range from 2.24 to 4.58. This indicates again that these power-law distributions possess the stable scaling property and defined-mean property.

7.3 Explanation and Justification

We propose an explanation for this linguistic phenomenon of power-law distributions in the length-frequency of entities, and then design a stochastic process to simulate the generation of entities so as to justify our explanation.

7.3.1 Explanation

The phenomenon of power-law distributions in the length-frequency of entities can be explained by the principle of least effort in communication and the preference for short entities. That is, whenever we need an entity to express our idea, assuming that we are able to make our idea understood, we would prefer to use a short one; a short entity can reduce both the speaker's and the listener's effort to communicate with each other. For example, to refer to the country of China, most of us would prefer to use "China" (length $l = 1$) or "P.R.C." ($l = 1$) rather than its full name "People's Republic of China" ($l = 4$); similarly, to refer to the country of the United States, most of us would prefer to use "United States" ($l = 2$) or "U.S." ($l = 1$) or "U.S.A." ($l = 1$) rather than its full name "United States of America" ($l = 4$). (We would use the formal name of an entity for the first time, and its short form for the rest communication.) In this sense, the length-frequency distribution of entities indicates the probability of the number of words we prefer to use in an entity. Our analysis from twenty-seven datasets suggests that, on average, the probability of our preference for a one-word entity is about 0.517 and the one for a two-word entity is about 0.276. The preference for short entities leads the length-frequency of entities to follow a family of power-law distributions.

7.3.2 Justification

To justify our explanation, we design the following stochastic process to generate entities: there is a *constant* probability (p_l) that generates an l-word entity, where the value of p_l is set by the weighted average of the percentages of l-word entities from a set of datasets, as defined by Equation (7.3).

$$p_l = \sum_{i=1}^{n} w_i p_l^i \tag{7.3}$$

where n is the number of the used datasets, p_l^i is the percentage of l-word entities in the i-th dataset, and w_i is the weight of the i-th dataset. In our experiments, we set $w_i = \frac{1}{n}$, which indicates the arithmetic average.[6] (There are other ways to set the value of w_i, for example, we can set w_i according to the number of entities in the i-th dataset. But the setting based on the number of entities will overlook those datasets that contain only a few entities, because

[6]Note that p_l in Equation (7.3) represents the preferential probability that generates an l-word entity, while $p(l)$ in Equation (7.1) represents the percentage of l-word entities over the whole entities.

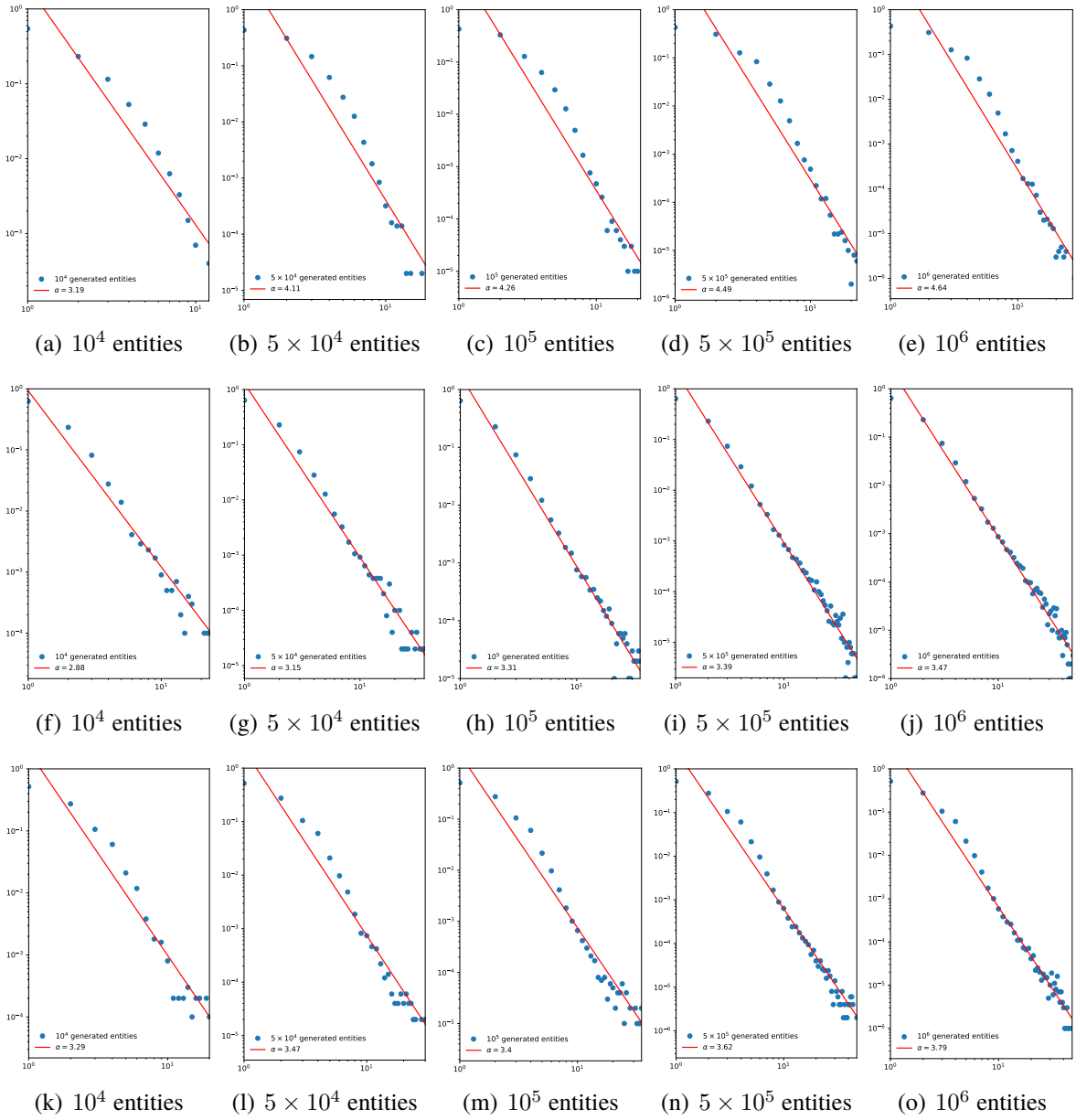
these analyzed datasets are quite diverse in terms of the number of their entities (e.g., English (43,652,029) vs. Swahili (30,742) and WikiAnchor (2,690,849) vs. ABSA (9,979)). Since our goal is to justify the validity of our explanation, we simply set the w_i as the arithmetic average.)

By using this stochastic process, we generate three series of entities using the preferential probabilities (p_l) that are derived from these real-world datasets described in Section 7.1. The first series of entities is generated by using the p_l derived from the above seventeen datasets in different languages (see Table 7.1); the second series is generated by using the p_l derived from the above ten datasets about different types of entities (see Table 7.2); the third series is generated by using the whole twenty-seven datasets. Each series contains five groups of generated entities in the following sizes: 10^4, 5×10^4, 10^5, 5×10^5, and 10^6. For each group of generated entities, the entities are generated one by one, namely, an entity is generated each time.[7] The length-frequency distributions of these generated entities are plotted in Figure 7.3. Specifically, the length-frequency distributions of the first series of generated entities are plotted in Figure 7.3(a)$\sim$7.3(e); the ones of the second series of generated entities are plotted in Figure 7.3(f)$\sim$7.3(j); the ones of the third series are in Figure 7.3(k)$\sim$7.3(o).

We can see that the length-frequency of all the three series of generated entities follows a similar family of power-law distributions, as defined by Equation (7.1), similar to the ones of real-world entities plotted in Figure 7.1 and 7.2. For each series of generated entities, their length-frequency distributions are similar to the corresponding ones of real-worlds entities that are used to derived the preferential probabilities. Within each series of generated entities, although the sizes of the five groups are significantly different, ranging from 10^4 to 10^6, their scaling exponent α ranges only from 2.88 to 4.64. This indicates that these power-law distributions in the length-frequency of generated entities possess the stable scaling property, similar to the ones of real-world entities. All the α are greater than 2, indicating well-defined means in all these length-frequency distributions.

Overall, the length-frequency distributions of all the three series of generated entities justify that power-law distributions in length-frequency of entities can be reproduced by a stochastic process with assigned preferential probabilities, and therefore justify our explanation.

[7] The cases that multiple entities (e.g., 2, 3, and 5) are generated each time also leads to similar length-frequency distributions.



(a) 10^4 entities (b) 5×10^4 entities (c) 10^5 entities (d) 5×10^5 entities (e) 10^6 entities

(f) 10^4 entities (g) 5×10^4 entities (h) 10^5 entities (i) 5×10^5 entities (j) 10^6 entities

(k) 10^4 entities (l) 5×10^4 entities (m) 10^5 entities (n) 5×10^5 entities (o) 10^6 entities

Figure 7.3: Power-law distributions in the length-frequency of generated entities. The first series of generated entities is simulated by our designed stochastic process using the preferential probabilities (p_l) derived from the seventeen datasets of different languages; it includes: (a) 10^4 entities (α = 3.19), (b) 5×10^4 entities (α = 4.11), (c) 10^5entities (α = 4.26), (d) 5×10^5 entities (α = 4.49), and (e) 10^6 entities (α = 4.64). The second series of generated is simulated by using the probabilities derived from the ten datasets of different types of entities; it includes: (f) 10^4 entities (α = 2.88), (g) 5×10^4 entities (α = 3.15), (h) 10^5 entities (α = 3.31), (i) 5×10^5 entities (α = 3.39), and (j) 10^6 entities (α = 3.47). The third series of generated entities is simulated by using the preferential probabilities derived from the whole twenty-seven datasets; it includes: (k) 10^4 entities (α = 3.29), (l) 5×10^4 entities (α = 3.47), (m) 10^5 entities (α = 3.4), (n) 5×10^5 entities (α = 3.62), and (o) 10^6 entities (α = 3.79).

7.4 Discussion

Power-law distributions in the rank-frequency of words in a corpus have attracted tremendous attention from researches like linguists and statisticians to understand the use of languages in our communicative system [24, 32, 42, 107, 108, 123, 124, 153, 157, 189, 190, 241, 242]. Our discovery that the length-frequency of entities follows a family of power-law distributions contributes a complementary consideration for the understanding of language use. In what follows we discuss the relation between the power-law distributions in the length-frequency of entities and the ones in the rank-frequency of words.

Both power-law distributions in the length-frequency of entities and in the rank-frequency distribution of words can be explained under Zipf's general principle of least effort, but there are also some differences between these two distributions in terms of their contents and contexts.

First of all, the numbers of data points are different. In the rank-frequency distribution of words, an r-rank word together with its frequency appears as a data point, while in the length-frequency distribution of entities, all the l-word entities composite a data point. So the number of data points in the rank-frequency distribution of words is as large as the size of vocabulary in a corpus, while the one in the length-frequency distribution of entities is generally less than 100, according to our analysis that in most datasets, the maximal length of entities contains less than 100 words (see Table 7.1 and 7.2).

Secondly, their scaling exponents are different. The scaling exponents in the rank-frequency distribution of words are observed to approximate to 1, indicating that these power-law distributions do not have defined means nor finite variances. By contrast, the scaling exponents in the length-frequency distribution of entities are greater than 2, indicating that all these power-law distribution have well-defined means; in most datasets, the scaling exponents are greater than 3, indicating finite variances in these power-law distributions. Moreover, since the maximal length of entities is generally less than 100 or finite, the number of data points in the length-frequency of entities is finite. In real-world entities, therefore, the means and variances of their length-frequency distributions are well-defined and finite.

Thirdly, the mechanisms of word generation and entity generation are different. In the rank-frequency distribution of words, a widely appreciated model that is used to generate words is the one developed by George Udny Yule [212] and Herbert Alexander Simon [189, 190].

In the Yule-Simon model, the preferential probability that generates a word depends on the existing words; during the generation process, the preferential probabilities are always *changed*. By contrast, in the length-frequency distribution of entities, the preferential probability that generates an l-word entity does not depends on existing entities; during the generation process, the preferential probabilities are *constant*. The constant preferential probabilities lead the process of entity generation to be much simpler and easier to control.

Chapter 8

Conclusion and Future Work

8.1 Conclusion of this Dissertation

We conduct an in-depth analysis on four diverse datasets for the intrinsic characteristics of time expressions and summarize five such common characteristics. The first four characteristics provide evidence in terms of time expressions for Zipf's principle of least effort [242] and the last one demonstrate the flexibility of time expressions. According to these characteristics, we propose two methods to recognize time expressions from unstructured text, including a type-based method termed SynTime and a learning-based method termed TOMN. SynTime is inspired by the part-of-speech of language and defines a syntactic token type system for the constituent words of time expressions, and designs a small set of general heuristic rules to recognize time expressions based on the idea of boundary expansion. Since these heuristic rules are only relevant to token types and are independent of specific tokens, SynTime is independent of specific domains, specific text types and even specific languages that consists of specific tokens. TOMN is a CRFs-based learning method with a defined constituent-based tagging scheme to model time expressions. The constituent-based tagging scheme overcomes the problem of inconsistent tag assignment that is caused by the conventional position-based tagging schemes. Experimental results on three diverse datasets demonstrate the effectiveness, efficiency, and robustness of SynTime and TOMN compared with state-of-the-art baselines, including rule-based time taggers and learning-based time taggers. Moreover, our analyses of time expressions and tagging schemes help explain many empirical results and observations that are reported in previous works about time expression recognition and tagging schemes in named entity recognition.

Similar to our analysis on time expressions, we also analyze two benchmark datasets for the intrinsic characteristics of named entities and summarize such three common characteristics. These three characteristics motivate us to design a learning-based method termed UGTO to model named entities, with another defined constituent-based tagging scheme and only a kind pre-tag features, word cluster features, and some basic lexical and POS features. Experiments on two benchmark datasets demonstrate the effectiveness of UGTO against two representative state-of-the-art methods. Experimental results also demonstrate that joint modeling of named entity recognition and classification does not improve the performance of named entity recognition, in both our model and the two representative models.

When analyzing time expressions and named entities, we discover that their length-frequency follows a family of power-law distributions, with stable scaling property and well-defined means and finite variances. Furthermore, we find that power-law distributions widely appear in the length-frequency of entities in different languages and different types of their entities. We explain this linguistic phenomenon by Zipf's principle of least effort in communication [242] and the preference for short entities, and justify our explanation by reproducing power-law distributions in the length-frequency of entities with a stochastic process that is assigned with preferential probabilities derived from real-word datasets.

8.2 Future Work

In the future, we plan to conduct research on the following directions. Firstly, we plan to complete the task of time expression normalization so as to develop SynTime as an end-to-end tool for publicly use. Secondly, we plan to test our syntactic token types and general heuristic rules on other languages, especially on those low-resource languages with little available labeled data, such as Finnish and Hindi. We can leverage translation tools to translate the labeled English text to target languages and then further manually correct these automatically labeled data to reduce the effort but simultaneously obtain high-quality labeled data. Thirdly, we plan to address the limitations existed in our constituent-based tagging schemes, and then apply them to model other entities in other languages. Fourthly, we plan to further investigate the power-law distributions in the length-frequency of entities in more languages and more types of entities, as well as the relationship between the power-law distributions in the length-frequency of entities and the ones in the rank-frequency of words.

References

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

[2] O. Alonso, J. Strotgen, R. Baeza-Yates, and M. Gertz. Temporal information retrieval: Challenges and opportunities. In *Proceedings of 1st International Temporal Web Analytics Workshop*, pages 1–8, 2011.

[3] B. Altuna, M. J. Aranzabe, and A. D. de Ilarraza. Eusheideltime: Time expression extraction and normalisation for basque. *Procesamiento del Lenguaje Natural*, (59):15–22, 2017.

[4] G. Angeli, C. D. Manning, and D. Jurafsky. 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*, pages 446–455, 2012.

[5] G. Angeli and J. Uszkoreit. Language-independent discriminative parsing of temporal expressions. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pages 83–92, 2013.

[6] M. Asahara and Y. Matsumoto. Japanese named entity extraction with redundant morphological analysis. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, pages 8–15, 2003.

[7] M. Bada, M. Echert, D. Evans, K. Garcia, K. Shipley, D. Sitnikov, W. A. Baumgartner, K. B. Cohen, K. Verspoor, J. A. Blake, and L. Hunter. Concept annoation in the craft corpus. *BMC Bioinformatics*, 13(161):1–20, 2012.

[8] J. A. Baldwin. Learning temporal annotation of french news. Master's thesis, Graduate School of Arts and Sciences, Georgetown University, 2002.

[9] D. Bamman, S. Popat, and S. Shen. An annotated dataset of literary entities. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2138–2144, 2019.

[10] A. Berglund. Extracting temporal information and ordering events for swedish. Master's thesis, 2004.

[11] K.-H. Best. Word length in old icelandic songs and prose texts. *Journal of Quantitative Linguistics*, 3(2):97–105, 1996.

[12] S. Bethard. Cleartk-timeml: A minimalist approach to tempeval 2013. In *Proceedings of the 7th International Workshop on Semantic Evaluation*, pages 10–14, 2013.

[13] S. Bethard, L. Derczynski, G. Savova, J. Pustejovsky, and M. Verhagen. Semeval-2015 task 6: Clinical tempeval. In *Proceedings of the 9th International Workshop on Semantic Evaluation*, pages 806–814, 2015.

[14] S. Bethard and J. L. Parker. A semantically compositional annotation scheme for time normalization. In *Proceedings of the 2016 Conference on Language Resources and Evaluation*, pages 3779–3786, 2016.

[15] S. Bethard, G. Savova, W.-T. Chen, L. Derczynski, J. Pustejovsky, and M. Verhagen. Semeval-2016 task 12: Clinical tempeval. In *Proceedings of the 10th International Workshop on Semantic Evaluation*, pages 1052–1062, 2016.

[16] S. Bethard, G. Savova, M. Palmer, and J. Pustejovsky. Semeval-2017 task 12: Clinical tempeval. In *Proceedings of the 11th International Workshop on Semantic Evaluation*, pages 565–572, 2017.

[17] D. M. Bikel, S. L. Miller, R. M. Schwartz, and R. M. Weischedel. Nymble: A high-performance learning name-finder. In *Proceedings of the fifth Conference on Applied Natural Language Processing*, pages 194–201, 1997.

[18] A. Bittar, P. Amsili, P. Denis, and L. Danlos. French timebank: An iso-timeml annotated reference corpus. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, pages 130–134, 2011.

[19] B. Boguraev, J. Pustejovsky, R. Ando, and M. Verhagen. Timebank evolution as a community resource for timeml parsing. *Language Resources and Evaluation*, 41(1):91–115, 2007.

[20] A. Borthwick, J. Sterling, E. Agichtein, and R. Grishman. Nyu: Description of the mene named entity system as used in muc-7. In *Proceedings of the 7th Message Understanding Conference*, 1998.

[21] T. Boudaa, M. E. Marouani, and N. Enneya. Arabic temporal expression tagging and normalization. In *Proceedings of International Conference on Big Data, Cloud and Applications*, pages 546–557, 2018.

[22] S. Brin. Extracting patterns and relations from the world wide web. In *Proceedings of the International Workshop on The World Wide Web and Databases*, pages 172–183, 1998.

[23] R. Campos, G. Dias, A. M. Jorge, and A. Jatowt. Survey of temporal information retrieval and related applications. *ACM Computing Surveys*, 47(2):15:1–41, 2014.

[24] J. B. Carroll. On sampling from a lognormal model of word frequency distribution. *Computational Analysis of Present-Day American English*, pages 406–424, 1967.

[25] T. Caselli, V. B. Lenzi, R. Sprugnoli, E. Pianta, and I. Prodanof. Annotating events, temporal expressions and relations in italian: the it-timeml experience for the ita-timebank. In *Proceedings of the 5th Linguistic Annotation Workshop*, pages 143–151, 2011.

[26] N. Chambers, S. Wang, and D. Jurafsky. Classifying temporal relations between events. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, pages 173–176, 2007.

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

[28] A. X. Chang and C. D. Manning. Sutime: Evaluation in tempeval-3. In *Proceedings of the Second Joint Conference on Lexical and Computational Semantics (SEM)*, pages 78–82, 2013.

[29] A. X. Chang and C. D. Manning. Tokensregex: Defining cascaded regular expressions over tokens. Technical report, Department of Computer Science, Stanford University, 2014.

[30] N. Chater and G. D. Brown. Scale-invariance as a unifying psychological principle. *Cognition*, 69(3):B17–B24, 1999.

[31] H.-H. Chen and J.-C. Lee. Identification and classification of proper nouns in chinese texts. In *Proceedings of the 16th Conference on Computational Linguistics*, pages 222–229, 1996.

[32] Y.-S. Chen. Zipf's law in natural languages, programming languages, and command languages: the simon-yule approach. *International Journal of Systems Science*, 22(11):2299–2312, 1991.

[33] N. A. Chinchor. Muc-7 named entity task definition. In *Proceedings of the 7th Message Understanding Conference*, volume 29, 1997.

[34] N. A. Chinchor. Muc-7 named entity task definition. In *Proceedings of the 7th Message Understanding Conference*, volume 29, 1998.

[35] N. A. Chinchor. Overview of muc-7/met-2. In *Proceedings of the 7th Message Understanding Conference*, 1998.

[36] J. P. Chiu and E. Nichols. Named entity recognition with bidirectional lstm-cnns. *Transactions of the Association for Computational Linguistics*, 4:357–370, 2016.

[37] A. Clauset, C. R. Shalizi, and M. E. J. Newman. Power-law distributions in empirical data. *SIAM Review*, 51(4):661–703, 2009.

[38] M. Collins. Ranking algorithms for named-entity extraction: Boosting and the voted perceptron. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 489–496, 2002.

[39] M. Collins and Y. Singer. 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*, 1999.

[40] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. P. Kuksa. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12:2493–2537, 2011.

[41] B. Conrad and M. Mitzenmacher. Power laws for monkeys typing randomly: the case of unequal probabilities. *IEEE Transactions on Information Theory*, 50(7):1403–1414, 2004.

[42] B. Corominas-Murtra and R. V. Solé. Universality of zipf's law. *Physical Review E*, 82(1):011102, 2010.

[43] F. Costa and A. Branco. Timebankpt: A timeml annotated corpus of portuguese. In *Proceedings of the 8th International Conference on Language Resources and Evaluation*, pages 3727–3734, 2012.

[44] G. Crichton, S. Pyysalo, B. Chiu, and A. Korhonen. A neural network multi-task learning approach to biomedical named entity recognition. *BMC Bioinformatics*, 18(1):368–371, 2017.

[45] H.-J. Dai, P.-T. Lai, Y.-C. Chang, and R. T.-H. Tsai. Enhancing of chemical compound and drug name recognition using representative tag scheme and fine-grained tokenization. *Journal of Cheminformatics*, 7.S1(S14):1–10, 2015.

[46] R. F. de Azevedo, M. R. d. S. R. Joao Pedro Santos Rodrigues, and C. M. C. Moro. Temporal tagging of noisy clinical texts in brazilian. In *Proceedings of International Conference on Computational Processing of the Portuguese Language*, pages 231–241, 2018.

[47] S. Degaetano-Ortlieb and J. Strötgen. Diachronic variation of temporal expressions in scientific writing through the lens of relative entropy. In *Proceedings of International Conference of the German Society for Computational Linguistics and Language Technology*, pages 259–275, 2017.

[48] L. Derczynski, K. Bontcheva, and I. Roberts. Broad twitter corpus: A diverse named entity recognition resource. In *Proceedings of the 26th International Conference on Computational Linguistics*, pages 1169–1179, 2016.

[49] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, volume 1, pages 4171–4186, Minneapolis, Minnesota, USA, 2019. Association for Computational Linguistics.

[50] Q. X. Do, W. Lu, and D. Roth. 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*, pages 677–687, 2012.

[51] G. Doddington, A. Mitchell, M. Przybocki, L. Ramshaw, S. Strassel, and R. Weischedel. The automatic content extraction (ace) program tasks, data, and evaluation. In *Proceedings of the 2004 Conference on Language Resources and Evaluation*, pages 1–4, 2004.

[52] R. I. Dogan, R. Leaman, and Z. Lu. Ncbi disease corpus: A resource for disease name recognition and concept normalization. *Journal of Biomedical Informatics*, 47:1–10, 2014.

[53] J. D'Souza and V. Ng. Classifying temporal relations in clinical data: A hybrid, knowledge-rich approach. *Journal of Biomedical Informatics*, 46:S29–S39, 2013.

[54] J. B. Estoup. Gammes stenographiques. In *Institut Stenographique de France, Paris*, 1916.

[55] L. Ferro. Tides instruction manual for the annotation of temporal expressions. Technical report, MITRE, 2001.

[56] L. Ferro, L. Gerber, I. Mani, B. Sundheim, and G. Wilson. Tides 2005 standard for the annotation of temporal expressions. Technical report, MITRE, 2005.

[57] L. Ferro, I. Mani, B. Sundheim, and G. Wilson. Tides temporal annotation guidelines - version 1.0.2. Technical report, MITRE, 2001.

[58] M. Filannino, G. Brown, and G. Nenadic. Mantime: Temporal expression identification and normalization in the tempeval-3 challenge. In *Proceedings of the 7th International Workshop on Semantic Evaluation*, pages 53–57, 2013.

[59] M. Filannino and G. Nenadic. Temporal expression extraction with extensive feature type selection and a posteriori label adjustment. *Data & Knowledge Engineering*, 100:19–23, 2015.

[60] J. R. Finkel, T. Grenager, and C. Manning. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43nd Annual Meeting of the Association for Computational Linguistics*, pages 363–370, 2005.

[61] J. R. Finkel and C. Manning. Nested named entity recognition. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 141–150, 2009.

[62] M. Fleischman. Automated subcategorization of named entities. In *Proceedings of the Student Research Workshop and Tutorial Abstracts, ACL (Companion Volume)*, pages 25–30, 2001.

[63] M. Fleischman and E. Hovy. Fine grained classification of named entities. In *Proceedings of the 19th international conference on Computational linguistics*, pages 1–7, 2002.

[64] R. Florian, A. Ittycheriah, H. Jing, and T. Zhang. Named entity recognition through classifier combination. In *Proceedings of the 7th Conference on Natural Language Learning*, pages 168–171, 2003.

[65] C. Forascu and D. Tufis. Romanian timebank: An annotated parallel corpus for temporal information. In *Proceedings of the 8th International Conference on Language Resources and Evaluation*, pages 3762–3766, 2012.

[66] W. Fucks. Theorie der wortbildung. *Mathematisch-Physikalische Semesterberichte*, 4:195–212, 1955.

[67] W. Fucks. Die mathematischen gesetze der bildung von sprachelementen aus ihren bestandteilen. *Nachrichtentechnische Fachberichte*, 3:7–21, 1956.

[68] K. Fukuda, T. Tsunoda, A. Tamura, and T. Takagi. Toward information extraction: Identifying protein names from biological papes. In *Proceedings of the Pacific Symposium on Biocomputing*, pages 707–718, 1998.

[69] X. Gabaix. Power laws in economics and finance. *Annual Revew of Economics*, 1(1):255–294, 2009.

[70] M. Gerner, G. Nenadic, and C. M. Bergman. Linnaeus: A species name identification system for biomedical literature. *BMC Bioinformatics*, 11(85):1–17, 2010.

[71] C. Giuliano. Fine-grained classification of named entities exploiting latent semantic kernels. In *CoNLL*, 2009.

[72] N. Grabar and T. Hamon. Automatic detection of temporal information in ukrainian general-language texts. Technical report, CNRS University Lille and LIMSI University Paris-Saclay, 2018.

[73] R. Grishman and B. Sundheim. Message understanding conference - 6: A brief history. In *Proceedings of the 16th International Conference on Computational Linguistics*, 1996.

[74] R. Grotjahn and G. Altmann. Modelling the distribution of word length: Some methodological problems. *Contributions to Quatitative Linguistics*, pages 141–153, 1993.

[75] C. Grouin, N. Grabar, T. Hamon, S. Rosset, X. Tannier, and P. Zweigenbaum. Eventual situations for timeline extraction from clinical reports. *Journal of the American Medical Informatics Association*, 20:820–827, 2013.

[76] P. Guiraud. The semic matrices of meaning. *Information (International Social Science Council)*, 7(2):131–139, 1968.

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

[78] T. Hao, X. Pan, Z. Gu, Y. Qu, and H. Weng. A pattern learning-based method for temporal expression extraction and normalization from multi-lingual heterogeneous clinical texts. *BMC Medical Informatics and Decision Making*, 18:16–25, 2018.

[79] R. He, B. Qin, T. Liu, Y. Pan, and S. Li. A novel heuristic error-driven learning for recognizing chinese time expression. *Journal of Chinese Language and Computing*, 18(4):139–159, 2008.

[80] R.-F. He, B. Qin, T. Liu, Y.-Q. Pan, and S. Li. Recognizing the extent of chinese time expressions based on the dependency parsing and error-driven learning. *Journal of Chinese Information Processing*, 21(5):36–40, 2007.

[81] J. R. Hobbs, D. E. Appelt, J. Bear, D. Israel, M. Kameyama, M. Stickel, and M. Tyson. Fastus: A cascaded finite-state transducer for extracting information from natrual-language text. In *Finite State Devices for Natural Language Processing*, pages 383–406, 1997.

[82] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9:1735–1780, 1997.

[83] Z. Huang, W. Xu, and K. Yu. Bidirectional lstm-crf models for sequence tagging. In *https://arxiv.org/abs/1508.01991v1*, 2015.

[84] S. Im, H. You, H. Jang, S. Nam, and H. Shin. Ktimeml: Specification of temporal and event expressions in korean text. In *Proceedings of the 7th Workshop on Asian Language Resources, ACL-IJCNLP 2009*, pages 115–122, 2009.

[85] W. F. S. IV, S. Bethard, S. Finan, M. Palmer, S. Pradhan, P. C. de Groen, B. Erickson, T. Miller, C. Lin, G. Savova, and J. Pustejovsky. Temporal annotation in the clinical domain. *Transactions of the Association for Computational Linguistics*, 2:143–154, 2014.

[86] S. B. Jang, J. Baldwin, and I. Mani. Automatic timex2 tagging of korean news. *ACM Transactions on Asian Language Information Processing*, 3(1):51–65, 2004.

[87] Y.-S. Jeong, W.-T. Joo, H.-W. Do, C.-G. Lim, K.-S. Choi, and H.-J. Choi. Korean timeml and korean timebank. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation*, pages 356–359, 2016.

[88] H. Ji and R. Grishman. Knowledge base population: Successful approaches and challenges. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, pages 1148–1158, 2011.

[89] P. Jindal and D. Roth. Extraction of events and temporal expressions from clinical narratives. *Journal of Biomedical Informatics*, 46:S13–S19, 2013.

[90] H. Jung and A. Stent. Att1: Temporal annotation using big windows and rich syntactic and semantic features. In *Proceedings of the 7th International Workshop on Semantic Evaluation*, pages 20–24, 2013.

[91] J. Kazama and K. Torisawa. 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*, pages 698–707, 2007.

[92] J.-D. Kim, T. Ohta, and J. Tsujii. Corpus annotation for mining biomedical events from literature. *BMC Bioinformatics*, 9(10):1–25, 2008.

[93] J.-D. Kim, T. Ohta, Y. Tsuruoka, and Y. Tateisi. Introduction to the bio-entity recognition task at jnlpba. In *Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and its Applications*, pages 70–75, 2004.

[94] J.-D. Kim, Y. Wang, and Y. Yasunori. The genia event extraction shared task, 2013 edition - overview. In *Proceedings of the 2013 Workshop on Biomedical Natural Language Processing*, pages 8–15, 2013.

[95] O. Kolomiyets and M.-F. Moens. Meeting tempeval-2: Shallow approach for temporal tagger. In *Proceedings of the NAACL HLT Workshop on Semantic Evaluation: Recent Achievements and Future Directions*, pages 52–57, 2009.

[96] M. Krallinger, F. Leitner, O. Rabal, M. Vazquez, J. Oyarzabal, and A. Valencia. Chemdner: The drugs and chemical names extraction challenge. *Journal of Cheminformatics*, 7(Suppl 1):S1, 2015.

[97] M. Krallinger, F. Leitner, O. Rabal, M. Vazquez, J. Oyarzabal, and A. Valencia. Overview of the chemical compound and drug name recognition (chemdner) task. In *BioCreative Challenge Evaluation Workshop*, volume 2, pages 2–33, 2015.

[98] J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the 18th International Conference on Machine Learning*, pages 281–289, 2001.

[99] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, and C. Dyer. Neural architecture for named entity recognition. In *Proceedings of the 15th Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 260–270, 2016.

[100] E. Laparra, D. Xu, and S. Bethard. From characters to time intervals: New paradigms for evaluation and neural parsing of time normalizations. *Transactions of the Association for Computational Linguistics*, 6:343–356, 2018.

[101] E. Laparra, D. Xu, S. Bethard, A. S. Elsayed, and M. Palmer. Semeval 2018 task 6: Parsing time normalizations. In *Proceedings of the 12th International Workshop on Semantic Evaluation*, pages 88–96, 2018.

[102] A. Lavelli, B. Magnini, M. Negri, E. Pianta, M. Speranza, and R. Sprugnoli. Italian content annotation bank (i-cab): Temporal expressions (v. 1.0). Technical report, ITC-irst, Centro per la Ricerca Scientifica e Technologica Povo, 2005.

[103] K. Lee, Y. Artzi, J. Dodge, and L. Zettlemoyer. Context-dependent semantic parsing for time expressions. In *Proceedings of the 52th Annual Meeting of the Association for Computational Linguistics*, pages 1437–1447, 2014.

[104] S. Lee and G. G. Lee. Heuristic methods for reducing errors of geographic named entities learned by bootstrapping. In *Proceedings of the International Conference on Natural Language Processing*, pages 658–669, 2005.

[105] H. Li, J. Strötgen, J. Zell, and M. Gertz. Chinese temporal tagging with heideltime. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 133–137, 2014.

[106] J. Li and C. Cardie. Timeline generation: Tracking individuals on twitter. In *Proceedings of the 23rd International Conference on World Wide Web*, pages 643–652, 2014.

[107] W. Li. Random texts exhibit zipf's-law-like word frequency. *IEEE Transactions on Information Theory*, 38(6):1842–1845, 1992.

[108] W. Li. Zipf's law everywhere. *Glottometrics*, 5:14–21, 2002.

[109] P. Liang. Semi-supervised learning for natural language. Master's thesis, Massachusetts Institute of Technology, 2005.

[110] C.-G. Lim and H.-J. Choi. Efficient temporal information extraction from korean documents. In *Proceedings of IEEE 18th International Conference on Mobile Data Management*, page 366370, 2017.

[111] Y.-K. Lin, H. Chen, and R. A. Brown. Medtime: A temporal information extraction system for clinical narratives. *Journal of Biomedical Informatics*, 46:S20–S28, 2013.

[112] W. Ling, C. Dyer, A. W. Black, I. Trancoso, R. Fermandez, S. Amir, L. Marujo, and T. Luis. 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*, pages 1520–1530, 2015.

[113] X. Ling, S. Singh, and D. S. Weld. Design challenges for entity linking. *Transactions of the Association for Computational Linguistics*, 3:315–328, 2015.

[114] X. Ling and D. S. Weld. Fine-grained entity recognition. In *Proceedings of the Twenty-Sixth Conference on Artificial Intelligence*, 2012.

[115] B. Liu. *Sentiment Analysis and Opinion Mining*. Morgan & Claypool Publishers, 2012.

[116] L. Liu, J. Shang, X. Ren, F. F. Xu, H. Gui, J. Peng, and J. Han. Empower sequence labeling with task-aware neural language model. In *Proceedings of the 32nd AAAI Conference on Artifical Intelligence*, 2018.

[117] X. Liu, S. Zhang, F. Wei, and M. Zhou. Recognizing named entities in tweets. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, pages 359–367, 2011.

[118] H. Llorens, L. Derczynski, R. Gaizauskas, and E. Saquete. Timen: An open temporal expression normalisation resource. In *Proceedings of the 8th International Conference on Language Resources and Evaluation*, pages 3044–3051, 2012.

[119] H. Llorens, E. Saquete, and B. Navarro. Tipsem (english and spanish): Evaluating crfs and semantic roles in tempeval-2. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 284–291, 2010.

[120] G. Luo, X. Huang, C.-Y. Lin, and Z. Nie. Joint named entity recognition and disambiguation. In *Proceedings of the 2005 Conference on Empirical Methods in Natural Language Processing*, pages 879–888, 2015.

[121] X. Ma and E. Hovy. 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)*, pages 1064–1074, 2016.

[122] Y. Ma, E. Cambria, and S. Gao. Label embedding for zero-shot fine-grained named entity typing. In *Proceedings of the 26th International Conference on Computational Linguistics*, pages 171–180, 2016.

[123] B. Mandelbrot. An information theory of the statistical structure of language. *Communication Theory*, 84:486–502, 1953.

[124] B. Mandelbrot. On the theory of word frequencies and on related markovian models of discourse. *Structure of Language and its Mathematical Aspects*, 12:190–219, 1961.

[125] G. Manfredi, J. Strötgen, J. Zell, and M. Gertz. Heideltime at eventi: Tuning italian resources and addressing timeml's empty tags. In *Proceedings of the 4th International Workshop EVALITA*, pages 39–43, 2014.

[126] I. Mani. Recent developments in temporal information. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing*, pages 45–60, 2003.

[127] I. Mani, M. Verhagen, B. Wellner, C. M. Lee, and J. Pustejovsky. 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*, pages 753–760, 2006.

[128] I. Mani and G. Wilson. Robust temporal processing of news. In *Proceedings of the 38th annual meeting on Association for Computational Linguistics.*, pages 69–76, 2000.

[129] I. Mani, G. Wilson, L. Ferro, and B. Sundheim. Guidelines for annotation temporal information. In *Proceedings of the first International Conference on Human Language Technology Research*, pages 1–3, 2001.

[130] C. Manning and H. Schutze. *Foundations of Statistical Natural Language Processing*. Cambride: MIT Press, 1999.

[131] D. Maynard, V. Tablan, C. Ursu, H. Cunningham, and Y. Wilks. Named entity recognition from diverse text types. In *Proceedings of 2001 Recent Advances in Natural Language Processing Conference*, pages 257–274, 2001.

[132] P. Mazur and R. Dale. Wikiwars: A new corpus for research on temporal expressions. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 913–922, 2010.

[133] A. McCallum and W. Li. 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*, 2003.

[134] S. Miller, J. Guinness, and A. Zamanian. 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.

[135] T. Miller, S. Bethard, D. Dligach, C. Lin, and G. Savova. Extracting time expressions from clinical text. In *Proceedings of the 2015 Workshop on Biomedical Natural Language Processing*, pages 81–91, 2015.

[136] T. Miller, S. Bethard, D. Dligach, S. Pradhan, C. Lin, and G. Savova. Discovering narrative containers in clinical text. In *Proceedings of the 2013 Workshop on Biomedical Natural Language Processing*, pages 18–26, 2013.

[137] A.-L. Minard, M. Speranza, E. Agirre, I. Aldabe, M. van Erp, B. Magnini, G. Rigau, and R. Urizar. Semeval-2015 task 4: Timeline: Cross-document event ordering. In *9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 778–786, 2015.

[138] E. Minkov, R. C. Wang, and W. W. Cohen. Extracting personal names from email: Applying named entity recognition to informal text. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 443–450, 2005.

[139] M. Mintz, S. Bills, R. Snow, and D. Jurafsky. Distant supervision for relation extraction without labeled data. In *Proceedings of the 47th Annual Meeting of the ACL and the 4th IJCNLP of the AFNLP*, pages 1003–1011, Singapore, 2009.

[140] M. Mitzenmacher. A brief history of generative models for power law and lognormal distributions. *Internet mathematics*, 1(2):226–251, 2004.

[141] V. Moriceau and X. Tannier. French resources for extraction and normalizationg of temporal expressions with heideltime. In *Proceedings of the 9th International Conference on Language Resources and Evaluation*, pages 3239–3243, 2014.

[142] A. Murat, A. Yusup, Z. Iskandar, A. Yusup, and Y. Abaydulla. Applying lexical semantics to automatic extraction of temporal expressions in uyghur. *Journal of Information Processing Systems*, 14(4), 2018.

[143] D. Nadeau and S. Sekine. A survey of named entity recognition and classification. *Lingvisticae Investigationes*, 30(1):3–26, 2007.

[144] D. Nadeau, P. D. Turney, and S. Matwin. Unsupervised named-entity recognition: Generating gazetteers and resolving ambiguity. In *Proceedings of the Conference of the Canadian society for computational studies of intelligence*, pages 266–277, 2006.

[145] N. Nakashole, T. Tylenda, and G. Weikum. Fine-grained semantic typing of emerging entities. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pages 1488–1497, 2013.

[146] M. Negri and L. Marseglia. Recognition and normalization of time expressions: Itc-irst at tern 2004. Technical report, ITC-irst, 2004.

[147] M. Negri, E. Saquete, P. Martinez-Barco, and R. Munoz. Evaluating knowledge-based approaches to the multilingual extension of a temporal expression normalizer. In *Proceedings of the Workshop on Annotating and Reasoning about Time and Events*, pages 30–37, 2006.

[148] M. E. Newman. Power laws, pareto distributions and zipf's law. *Contemporary physics*, 46(5):323–351, 2005.

[149] C. Niu, W. Li, J. Ding, and R. K. Srihari. A bootstrapping approach to named entity classification using successive learners. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*, pages 335–342, 2003.

[150] T. Ohta, S. Pyysalo, R. Rak, A. Rowley, H.-W. Chun, S.-J. Jung, C.-H. Jeong, S.-P. Choi, J. Tsujii, and S. Ananiadou. Overview of the pathway curation (pc) task of bionlp shared task 2013. In *Proceedings of the 2013 Workshop on Biomedical Natural Language Processing*, pages 67–75, 2013.

[151] O. Owoputi, B. O'Connor, C. Dyer, K. Gimpel, N. Schneider, and N. A. Smith. Improved part-of-speech tagging for online conversational text with word clusters. In *Proceedings of NAACL-HLT 2013*, pages 380–390, 2013.

[152] R. Parker, D. Graff, J. Kong, K. Chen, and K. Maeda. Engilish gigaword fifth edition. 2011.

[153] A. Parker-Rhodes and T. Joyce. A theory of word-frequency distribution. *Nature*, 178:1308, 1956.

[154] A. Passos, V. Kumar, and A. McCallum. Lexicon infused phrase embeddings for named entity resolution. In *Proceedings of the 8th Conference on Computational Language Learning*, pages 78–86, 2014.

[155] J. Patrick and M. Li. High accuracy information extraction of medication information from clinical notes: 2009 i2b2 medication extraction challenge. *Journal of the American Medical Informatics Association*, 17(5):524–527, 2010.

[156] M. E. Peters, W. Ammar, C. Bhagavatula, and R. Power. Semi-supervised suquence tagging with bidirectional language models. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, pages 1756–1765, 2017.

[157] S. T. Piantadosi. Zipf's word frequency law in natural language: A critical review and future directions. *Psychonomic bulletin & review*, 21(5):1112–1130, 2014.

[158] T. Poibeau and L. Kosseim. Proper name extraction from non-journalistic texts. *Language and Computers*, 37:144–157, 2001.

[159] M. Pontiki, D. Galanis, H. Papageorgiou, S. Manandhar, and I. Androutsopoulos. Semeval-2015 task 12: Aspect based sentiment analysis. In *Proceedings of the 9th International Workshop on Sementic Evaluation*, pages 486–495, 2015.

[160] M. Pontiki, D. Galanis, J. Pavlopoulos, H. Papageorgiou, I. Androutsopoulos, and S. Manandhar. Semeval-2014 task 4: Aspect based sentiment analysis. In *Proceedings of the 8th International Workshop on Semantic Evaluation*, pages 27–35, 2014.

[161] S. Pradhan, A. Moschitti, N. Xue, H. T. Ng, A. Bjorkelund, O. Uryupina, Y. Zhang, and Z. Zhong. Towards robust linguistic analysis using ontonotes. In *Proceedings of the 7th Conference on Computational Natural Language Learning*, pages 143–152, 2013.

[162] S. S. Pradhan, E. Hovy, M. Marcus, M. Palmer, L. Ramshaw, and R. Weischedel. Ontonotes: A unified relational semantic representation. In *Proceedings of the 2007 IEEE International Conference on Semantic Computing*, pages 517–526, 2007.

[163] J. Pustejovsky, J. Castano, R. Ingria, R. Sauri, R. Gaizauskas, A. Setzer, G. Katz, and D. Radev. Timeml: Robust specification of event and temporal expressions in text. *New Directions in Question Answering*, 3:28–34, 2003.

[164] J. Pustejovsky, P. Hanks, R. Sauri, A. See, R. Gaizauskas, A. Setzer, B. Sundheim, D. Radev, D. Day, L. Ferro, and M. Lazo. The timebank corpus. *Corpus Linguistics*, 2003:647–656, 2003.

[165] J. Pustejovsky, K. Lee, H. Bunt, and L. Romary. Iso-timeml: An international standard for semantic annotation. In *Proceedings of the Seventh conference on International Language Resouces and Evaluation (LREC'10)*, pages 394–397, 2010.

[166] S. Pyysalo and S. Ananiadou. Anatomical entity mention recognition at literature scale. *Bioinformatics*, 30(6):868–875, 2014.

[167] S. Pyysalo, T. Ohta, M. Miwa, and J. Tsujii. Towards exhausive protein modification event extraction. In *Proceedings of the 2011 Workshop on Biomedical Natural Language Processing*, pages 114–123, 2011.

[168] S. Pyysalo, T. Ohta, R. Rak, A. Rowley, H.-W. Chun, S.-J. Jung, S.-P. Choi, J. Tsujii, and S. Ananiadou. Overview of the cancer genetics and pathway curation tasks of bionlp shared task 2013. *BMC Bioinformatics*, 16(Suppl 10):S2, 2015.

[169] S. Pyysalo, T. Ohta, R. Rak, D. Sullivan, C. Mao, C. Wang, B. Sobral, J. Tsujii, and S. Ananiadou. Overview of the id, epi and rel tasks of bionlp shared task 2011. *BMC Bioinformatics*, 13(Suppl 11):S2, 2012.

[170] L. Ratinov and D. Roth. Design challenges and misconceptions in named entity recognition. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning*, pages 147–155, 2009.

[171] L. Ratinov and D. Roth. Design challenges and misconceptions in named entity recognition. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning*, pages 147–155, 2009.

[172] Y. Ravin and N. Wacholder. Extracting names from natural-language text. Technical report, IBM Research Division, 1997.

[173] W. J. Reed. The pareto, zipf and other power laws. *Economics letters*, 74(1):15–19, 2001.

[174] X. Ren, W. He, M. Qu, L. Huang, H. Ji, and J. Han. Afet: Automatic fine-grained entity typing by hierarchical partial-label embedding. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1369–1378, 2016.

[175] A. Ritter, S. Clark, Mausam, and O. Etzioni. Named entity recognition in tweets: An experimental study. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1524–1534, 2011.

[176] K. Roberts, B. Rink, and S. M. Harabagiu. A flexible framework for recognizing events, temporal expressions, and temporal relations in clinical text. *Journal of the American Medical Informatics Association*, 20(5):867–875, 2013.

[177] E. F. T. K. Sang. Introduction to the conll-2002 shared task: Language-independent named entity recognition. In *Proceedings of the 6th Conference on Natural Language Learning*, 2002.

[178] E. F. T. K. Sang and F. D. Meulder. Introduction to the conll-2003 shared task: Language-independent named entity recognition. In *Proceedings of the 7th Conference on Natural Language Learning*, pages 142–147, 2003.

[179] E. F. T. K. Sang and J. Veenstra. Representing text chunks. In *Proceedings of the ninth Conference on European Chapter of the Association for Computational Linguistics*, pages 173–179, 1999.

[180] C. N. D. Santos and V. Guimaraes. Boosting named entity recognition with neural character embeddings. In *Proceedings of the 5th Named Entities Workshop*, pages 25–33, 2015.

[181] E. Saquete, P. Martinez-Barco, and R. Munoz. Recognizing and tagging temporal expressions in spanish. In *Proceedings of LREC Workshop on Annotation Standards for Temporal Information in Natural Language*, pages 44–51, 2002.

[182] E. Saquete, P. Martinez-Barco, and R. Munoz. Evaluation of the automatic multilinguality for time expression resolution. In *Proceedings of the 15th International Workshop on Database and Expert Systems Applications*, pages 25–30, 2004.

[183] R. Sauri, E. Saquete, and J. Pustejovsky. Annotating time expressions in spanish. timeml annotation guidelines (version tempeval-2010). Technical report, Barcelona Media - Innovation Center, 2010.

[184] S. Sekine. Nyu: Description of the japanese ne system used for met-2. In *Proceedings of the 7th Message Understanding Conference*, 1998.

[185] S. Sekine and E. Ranchhod. *Named entities: recognition, classification and use*, volume 19. 2009.

[186] A. Setzer and R. Gaizauskas. Annotating events and temporal information in newswire texts. In *Proceedings of the Second International Conference on Language Resources and Evaluation*, pages 1287–1294, 2000.

[187] B. Sigurd, M. Eeg-Olofsson, and J. van de Weijer. Word length, sentence length and frequency - zipf revisited. *Studia Linguistica*, 58(1):37–52, 2004.

[188] J. F. D. Silva, Z. Kozareva, and J. G. P. Lopes. Cluster analysis and classification of named entities. In *Proceedings of the 2004 Conference on Language Resources and Evaluation*, 2004.

[189] H. A. Simon. On a class of skew distribution functions. *Biometrika*, 42(3/4):425–440, 1955.

[190] H. A. Simon. Some further notes on a class of skew distribution functions. *Information and Control*, 3(1):80–88, 1960.

[191] L. Skukan, G. Glavas, and J. Snajder. Heideltime.hr: Extracting and normalizing temporal expressions in croatian. In *Proceedings of the 9th Slovenian Language Technologies Conference*, pages 99–103, 2014.

[192] L. Smith, L. K. Tanabe, R. J. nee Ando, C.-J. Kuo, I.-F. Chung, C.-N. Hsu, Y.-S. Lin, R. Klinger, C. M. Friedrich, K. Ganchev, M. Torii, H. Liu, B. Haddow, C. A. Struble, R. J. Povinelli, A. Vlachos, W. A. B. Jr, L. Hunter, B. Carpenter, R. T.-H. Tsai, H.-J. Dai, F. Liu, Y. Chen, C. Sun, S. Katrenko, P. Adriaans, C. Blaschke, R. Torres, M. Neves, P. Nakov, A. Divoli, M. Mana-Lopez, J. Mata, and W. J. Wilbur. Overview of biocreative ii gene mention recogntion. *Genome Biology*, 9(2):S2, 2008.

[193] S. Sohn, K. B. Wagholikar, D. Li, S. R. Jonnalagadda, C. Tao, R. K. Elayavilli, and H. Liu. Comprehensive temporal information detection from clinical text: Medical events, time, and tlink identification. *Journal of the American Medical Informatics Association*, 20(5):836–842, 2013.

[194] M. Steedman. *Surface Structure and Interpretation*. The MIT Press, 1996.

[195] S. S. Stevens. On the theory of scales of measurement. *Science*, 103(2684):677–680, 1946.

[196] B. Strauss, B. E. Toma, A. Ritter, M.-C. de Marneffe, and W. Xu. Results of the wnut16 named entity recognition shared task. In *Proceedings of the 2nd Workshop on Noisy User-generated Text*, pages 138–144, 2016.

[197] J. Strötgen, A. Armiti, T. V. Canh, J. Zell, and M. Gertz. Time for more languages: Temporal tagging of arabic, italian, spanish, and vietnamese. *ACM Transactions on Asian Language Information Processing*, 13(1):1–21, 2014.

[198] J. Strötgen, T. Bogel, J. Zell, A. Armiti, T. V. Canh, and M. Gertz. Extending heideltime for temporal expressions referring to historic dates. In *Proceedings of the 9th International Conference on Language Resources and Evaluation*, pages 2390–2397, 2014.

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

[200] J. Strötgen and M. Gertz. Wikiwarsde: A german corpus of narratives annotated with temporal expressions. In *Proceedings of German Society for Computational Linguistics and Language Technology*, pages 129–134, 2011.

[201] J. Strötgen and M. Gertz. Temporal tagging on different domains: Challenges, strategies, and gold standards. In *Proceedings of 8th International Conference on Language Resources and Evaluation*, pages 3746–3753, 2012.

[202] J. Strotgen and M. Gertz. Multilingual and cross-domain temporal tagging. *Language Resources and Evaluation*, 47(2):269–198, 2013.

[203] J. Strötgen, J. Zell, and M. Gertz. Heideltime: Tuning english and developing spanish resources. In *Proceedings of the Second Joint Conference on Lexical and Computational Semantics (SEM)*, pages 15–19, 2013.

[204] E. Strubell, P. Verga, D. Belanger, and A. McCallum. Fast and accurate entity recognition with iterated dilated convolutions. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2670–2680, 2017.

[205] W. Sun, A. Rumshisky, and O. Uzuner. Annotating temporal information in clinical narratives. *Journal of Biomedical Informatics*, 46:S5–S12, 2013.

[206] W. Sun, A. Rumshisky, and O. Uzuner. Evaluating temporal relations in clinical text: 2012 i2b2 challenge. *Journal of the American Medical Informatics Association*, 20:806–813, 2013.

[207] W. Sun, A. Rumshisky, and O. Uzuner. Normalization of relative and incomplete temporal expressions in clinical narratives. *Journal of the American Medical Informatics Association*, 22(5):1001–1008, 2015.

[208] J. Tabassum, A. Ritter, and W. Xu. Tweetime: A minimally supervised method for recognizing and normalizing time expressions in twitter. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 307–318, 2016.

[209] K. Takeuchi and N. Collier. Bio-medical entity extraction using support vector machines. *Artificial Intelligence In Medicine*, 33(2):125–137, 2005.

[210] B. Tang, Y. Wu, M. Jiang, Y. Chen, J. C. Denny, and H. Xu. A hybrid system for temporal information extraction from clinical text. *Journal of the American Medical Informatics Association*, 20:828–835, 2013.

[211] M. Taule, T. Marti, and M. Recasens. Ancora: Multilevel annotated corpora for catalan and spanish. In *Proceedings of the 6th International Conference on Language Resources and Evaluation*, 2008.

[212] G. Udny Yule. A mathematical theory of evolution, based on the conclusions of dr. jc willis, frs. *Philosophical Transactions of the Royal Society of London Series B*, 213:21–87, 1925.

[213] N. UzZaman and J. F. Allen. Trips and trios system for tempeval-2: Extracting temporal information from text. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 276–283, 2010.

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

[215] M. van de Camp and H. Christiansen. Resolving relative time expressions in dutch text with constraint handling rules. In *Proceedings of the 7th International Workshop on Constraint Solving and Language Processing*, pages 74–85, 2012.

[216] N. Vazov. A system for extraction of temporal expressions from french texts based on syntactic and semantic constraints. In *Proceedings of the Workshop on Temporal and Spatial Information Processing*, pages 14–21, 2001.

[217] M. Verhagen, R. Gaizauskas, F. Schilder, M. Hepple, G. Katz, and J. Pustejovsky. Semeval-2007 task 15: Tempeval temporal relation identification. In *Proceedings of the 4th International Workshop on Semantic Evaluation*, pages 75–80, 2007.

[218] M. Verhagen, I. Mani, R. Sauri, R. Knippen, S. B. Jang, J. Littman, A. Rumshisky, J. Phillips, and J. Pustejovsky. Automating temporal annotation with tarqi. In *Proceedings of the ACL Interactive Poster and Demonstration Sessions.*, pages 81–84, 2005.

[219] M. Verhagen, R. Sauri, T. Caselli, and J. Pustejovsky. Semeval-2010 task 13: Tempeval-2. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 57–62, 2010.

[220] W. C. Wake. Sentence-length distributions of greek authors. *Journal of the Royal Statistical Society: Series A (General)*, 120(3):331–346, 1957.

[221] L.-J. Wang, W.-C. Li, and C.-H. Chang. Recognizing unregistered names for mandarin word identification. In *Proceedings of the 14th Conference on Computational Linguistics*, volume 4, pages 1239–1243, 1992.

[222] C.-H. Wei, Y. Peng, R. Leaman, A. P. Davis, C. J. Mattingly, J. Li, T. C. Wiegers, and Z. Lu. Overview of the biocreative v chemical disease relation (cdr) task. In *Proceedings of the 5th BioCreative Challenge Evaluation Workshop*, pages 154–166, 2015.

[223] R. Weischedel and A. Brunstein. Bbn pronoun coreference and entity type corpus. *Linguistic Data Consortium*, 112, 2005.

[224] C. B. Williams. A note on the statistical analysis of sentence-length as a criterion of literary style. *Biometrika*, 31(3/4):356–361, 1940.

[225] G. Wilson, I. Mani, B. Sundheim, and L. Ferro. A multilingual approach to annotating and extracting temporal information. In *Proceedings of the Workshop on Temporal and Spatial Information Processing*, page 12, 2001.

[226] C. S. Wolodja Wentland, Johannes Knopp and M. Hartung. Building a multilingual lexical resource for named entity disambiguation, translation and transliteration. In *Proceedings of the Sixth International Language Resources and Evaluation*, Marrakech, Morocco, 2008.

[227] K.-F. Wong, Y. Xia, W. Li, and C. Yuan. An overview of temporal information extraction. *International Journal of Computer Processing of Oriental Languages*, 18(2):137–152, 2005.

[228] M. Wu, W. Li, Q. Chen, and Q. Lu. Normalizing chinese temporal expressions with multi-label classification. In *Proceedings of the 2th International Conference on Natural Language Processing and Knowledge Engineering*, pages 318–323, 2005.

[229] M. Wu, W. Li, Q. Lu, and B. Li. Ctemp: A chinese temporal parser for extracting and normalizing temporal information. In *Proceedings of the Second International Joint Conference on Natural Language Processing*, pages 694–706, 2005.

[230] T. Wu, Y. Zhou, X. Huang, and L. Wu. Chinese time expression recognition based on automatically generated basic-time-unit rules. *Journal of Chinese Information Processing*, 24(4):3–10, 2010.

[231] M. Xu, H. Jiang, and S. Watcharawittayakul. A local detection approach for named entity recognition and mention detection. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, pages 1237–1247, 2017.

[232] Y. Xu, Y. Wang, T. Liu, J. Tsujii, and E. I.-C. Chang. An end-to-end system to identify temporal relation in discharge summaries: 2012 i2b2 challenge. *Journal of the American Medical Informatics Association*, 20:849–858, 2013.

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

[234] D. Yogatama, D. Gillick, and N. Lazic. Embedding methods for fine grained entity type classification. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, volume 2, pages 291–296, 2015.

[235] S. Yu, S. Bai, and P. S. Wu. Description of the kent ridge digital labs system used for muc-7. In *Proceedings of the 7th Message Understanding Conference*, 1998.

[236] V. Zavarella and H. Tanev. Fss-timex for tempeval-3: Extracting temporal information from text. In *Proceedings of the 7th International Workshop on Semantic Evaluation*, pages 58–63, 2013.

[237] D. Zelenko, C. Aone, and A. Richardella. Kernel methods for relation extraction. *Journal of Machine Learning Research*, 3:1083–1106, 2003.

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

[239] X. Zhong, E. Cambria, and A. Hussain. Extracting time expressions and named entities with constituent-based tagging schemes. *Cognitive Computation*, pages 1–19, 2020.

[240] X. Zhong, A. Sun, and E. Cambria. Time expression analysis and recognition using syntactic token types and general heuristic rules. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, volume 1, pages 420–429, Vancouver, Canada, 2017.

[241] G. Zipf. *The Psychobiology of Language*. London: Routledge, 1936.

[242] G. Zipf. *Human Behavior and the Principle of Least Effort: An Introduction to Human Ecology*. Addison-Wesley Press, Inc., 1949.