

STUDY ON FEATURE GENERALIZATION
FOR NAMED ENTITY RECOGNITION
(固有表現抽出のための素性の一般化の研究)

by

Han-Cheol CHO

趙 漢哲

A Dissertation

Presented to
the Graduate School of the University of Tokyo
on December 13th, 2013
in Partial Fulfillment of the Requirements
for the Degree of Doctor of Information Science and Technology
in Computer Science

Thesis Supervisor: Reiji SUDA 須田 礼仁

Professor

ABSTRACT

Information extraction (IE) is one of the important research fields in the natural language processing (NLP) area. It aims to extract useful information from various types of text such as web blogs, e-mails, newswire articles, and research papers. Extracted information can be directly consumed by human users, or can be integrated into knowledge bases that are easily accessible by machine. For extracting information from text, it is essential to recognize smallest information units since they participate in the construction of more complex knowledge and eventually the whole picture of the text. These units are called named entities and the task of extracting named entities is named entity recognition (NER).

While many solutions have been proposed from rule-based to statistical approaches, current state-of-the-art systems are mostly based on supervised learning techniques that use manually annotated data for training. However, preparing annotated data for a target domain is time-consuming and costly work; as a result, the amount of training data is often very limited. In previous studies, the data sparseness problem, which mainly results from the small size of training data, has been considered as a major obstacle in supervised learning approaches.

In this thesis, we tackle this problem in two perspectives. First, we propose a feature generation method that incorporates multiple segment representations (SRs) such as *IOB2* and *IOBES* into a single model. This method enables a model to exploit the features capturing the characteristics of words that often appear at specific positions, while alleviating the negative effect of these features due to their low frequency. Second, we propose the use of a context gazetteer, a list of contexts with which entities can co-occur, as new non-local context features. Unlike previous studies, we build a context gazetteer from an encyclopedic database because it allows us to use rich and sophisticated context patterns.

To investigate the effect of the proposed feature generation method, we applied it to the BioCreative 2 gene mention recognition (BC2GMR) task and the CoNLL 2003 NER (CoNLL2003) shared task. In case of traditional NER models using only one SR, a model using a more complex SR achieves higher precision than those using less complex

SRs, whereas recall starts to drop when SR becomes too complex. On the other hand, the models using multiple SRs improve both precision and recall as more and more SRs are incorporated. To evaluate the effectiveness of a context gazetteer, we applied the context gazetteer built from the EntrezGene database to the BC2GMR task. The results improve both precision and recall, and the major improvement comes from recall. We analyze the results to show that the context gazetteer built from a large amount of unlabeled data can provide useful context features that are not easily obtainable from a small amount of manually annotated data or human curated resources.

ABSTRACT (in Japanese)

情報抽出は、Web や学術文書などの構造化されていない文書から有用な情報を取り出し、人間と機械の双方がアクセス可能な知識ベースを構築する事を目的とする、自然言語処理における重要な研究分野の一つである。この情報抽出の技術を構成する基盤技術の一つとして、固有表現抽出がある。これは、情報抽出の前処理として、抽出対象となる固有表現を認識し、予め規定しておいた意味クラスに分類するタスクである。固有表現認識は、基礎的な処理である一方、非常に重要な処理でもある。何故なら、複雑で、込み入った情報も、元を正せば、基本単位の組み合わせによって構成されるからである。

固有表現認識は、Message Understanding Conference (MUC) プロジェクトを発端に、20年以上の歴史があり、これまでルールベースの手法から統計的な手法まで、実に様々な手法が提案されてきた。現在の最新のシステムは、その殆どが、人間がタグ付けした正解データに基づく教師付き学習の手法によって構築されている。しかし、人手でタグ付けしたデータを準備するには膨大な時間と費用を要するため、学習データの量が限られていることが多い。教師付きの機械学習手法でこのように少量の学習データを用いる場合、実際のシステムの実行時に、学習データに出現しなかった事例が多く現れることが大きな問題となる（疎データ問題）。この問題に対して、これまで、単語の表層形の代わりに品詞やチャンクラベルのようなより一般化された情報を学習に用いたり、学習時データに出現しない単語を被覆するために外部の辞書情報を用いたり、大量のラベルなし文書内の統計情報を用いて単語をクラスタリングした結果を利用するなど、複数の側面から解決が試みられてきた。

この論文では、我々が新たに提案する二種類の特徴を利用することによって、固有表現抽出における疎データ問題を解消する。第一に、我々は複数の異なる境界ラベル集合を一つの統計モデルに統合するための特徴生成手法を提案する。固有表現抽出のラベルは、一般的に系列ラベリングの問題として定式化される。各ラベルは、ある単語が固有表現中のどの位置に現れるかを示す境界情報と固有表現の意味クラスを表す情報の二つで構成されている。境界情報を表すラベル集合には、複数のバリエーションがあり、細かい分類の境界ラベルを利用すれば、ある特定の位置に出現する単語の特徴などの有用な情報を捉える事ができるが、学習データの量が足りない場合には過学習となる恐れがある。我々の提案する手法では、細かい粒度の境界ラベル集合と一般化された粗い粒度の境界ラベル集合を統合的に利用することで、過学習を避けながらも、情報量が高い特徴の恩恵を受けるこ

とが出来る。第二に、我々は、大規模なデータベースを利用して獲得した、固有表現の手がかり表現を利用する。一般的に、固有表現は手がかり表現と共に出現する場合が多い。しかし、人手で作られた少量のデータだけでは、そのような重要な手がかり表現を網羅的に抽出することは難しい。我々は、これらの手がかり表現が固有表現との係り受け関係を持つフレーズとして現れることに着目し、大規模なラベルなし文書データや辞書情報を利用し、手がかり表現を自動的に獲得する手法を提案する。

我々の提案手法を検証するために、固有表現抽出の具体例として BioCreative 2 の遺伝子名認識タスクと CoNLL 2003 固有表現抽出タスクを例に、実験を行った。一つの境界ラベル集合のみを用いた場合には、より複雑なラベル集合が、粒度の粗いラベル集合より高い適合率を示す一方で、ラベル集合が複雑すぎる場合には、再現率を落としてしまうが、我々の提案する複数のラベル集合を同時に用いる手法では、適合率、再現率の双方で性能の向上が見られた。また、EntrezGene データベースから手がかり表現を抽出し、適用する実験においても、適合率、再現率双方で性能の向上が見られた。特に、疎データ問題のために、これまでの手法では抽出不可能だった複雑な手がかり表現を獲得したことで、再現率の向上が大きく見られた。実際に、本手法で得られた手がかり表現を分析したところ、少量のデータからは容易には抽出が難しい複雑かつ有用な手がかり表現が抽出されていることを確認した。

Acknowledgement

This thesis might not have been written without the help from a number of people. I would like to express my deepest gratitude to them.

First and foremost, I am greatly indebted to professor Jun'ichi Tsujii, my academic advisor at the Tsujii laboratory. He accepted me as a member of his laboratory where many of the brilliant researchers and students were eager to come across the world. Moreover, his unceasing passion for research and an engaging personality was an excellent example to me of how to be a successful researcher in academic fields.

Associate professor Jin-Dong Kim helped me a lot when I was preparing to go study abroad. He introduced me to professor Tsujii and gave me an opportunity to present my research at the Tsujii laboratory. My graduate school life at the University of Tokyo, therefore, would not be possible without his help.

Dr. Naoaki Okazaki, who now works at Tohoku University as an associate professor, supervised my doctoral research and gave me a lot of invaluable advices and comments. When the Tsujii laboratory was closed due to the retirement of professor Tsujii, he kindly asked me to come to the Inui&Okazaki laboratory at Tohoku University and continued to guide my research. Moreover, I received many advices when I had troubles in my personal life. In every way, I owe him beyond description.

Professor Kentaro Inui at Tohoku University willingly accepted me to his laboratory as a special research student when the Tsujii laboratory was closed. He is not only an insightful researcher but also a very energetic person having close

relationships with his students and other researchers in diverse fields. In spite of his busy schedule, he always prepared events for his laboratory members. I participated in these events and spent a wonderful time.

Professor Suda at the University of Tokyo took care of my graduate school life after professor Tsujii got retired. I appreciate his help and support.

I would like to express my gratitude to all members of Tsujii laboratory at the University of Tokyo and Inui&Okazaki laboratory at Tohoku University. At the Tsujii laboratory, I spent a wonderful time with Dr. Makoto Miwa, Dr. Tam Wailok, Dr. Tadayoshi Hara, Dr. Yuichiroh Matsubayashi, Dr. Jun Hatori, Dr. Tomoko Ohta, Dr. Sampo Pyysalo, Dr. Rune Saetre, Dr. Takuya Matsuzaki, Dr. Yusuke Miyao, Dr. Pontus Stenetorp, Mr. Iwasawa, Ms. Noriko Katsu, Ms. Minako Ito and all the other members. At the Inui&Okazaki laboratory, I had great experiences with Dr. Yotaro Watanabe, Dr. Eric Nichols, Dr. Junta Mizuno, Ms. Tomoko Yamaki, and all the other members.

I also would like to express my appreciation to my family. As always, their support and encouragement gave me the greatest power in the pursuit of the final goal in my graduate study. And I met Mio, my love, at the hardest time of my life, but she always stayed with me and gave me the strength to overcome it. It is one of the most precious gifts that I received during my life.

Contents

1	Introduction	1
1.1	Named Entity Recognition	2
1.2	Living in the Age of Information Overload	3
1.3	The Overview of This Thesis	8
2	Background	11
2.1	Historical Background	12
2.1.1	NER in the Newswire Domain	12
2.1.2	NER in the Biomedical Domain	14
2.2	Methodological Background	18
2.2.1	Dictionary-based Approach	18
2.2.2	Rule-based Approach	21
2.2.3	Machine Learning Approach	23
2.3	Features	31
2.3.1	Local Features	31
2.3.2	Global Features	33
2.3.3	Features from External Resources	34
2.4	Evaluation	36
2.5	The Data-sparseness Problem and Feature Generalization	38
2.5.1	NER as the Mixture of the Segmentation and Classification Tasks	39
2.5.2	Generalization of Combinatorial Syntactic Structure Features	40
3	Named Entity Recognition with Multiple Segment Representa- tions	43

3.1	Introduction	43
3.2	Segment Representations	45
3.2.1	Segment Representations in Various NLP tasks	45
3.2.2	Relation among Segment Representations	47
3.3	The Effects of Different Segment Representations on NER	52
3.3.1	Evaluation based on Standard Performance Measures	54
3.3.2	Evaluation based on the Difference of Tagging Results	54
3.4	The Proposed Method	55
3.4.1	The Mapping Relation of Segment Representations	56
3.4.2	A Modified Linear Chain CRFs Model for Multiple Segment Representations	57
3.4.3	Boosting up Tagging Speed	58
3.5	Experiments	59
3.5.1	NER in the Biomedical Domain	59
3.5.2	NER in the General Domain	71
3.6	Summary	74
4	Inducing Context Gazetteers from Encyclopedic Databases for Named Entity Recognition	77
4.1	Introduction	77
4.2	Related Work	80
4.3	Building a Context Gazetteer	81
4.4	Evaluation	84
4.4.1	Data	85
4.4.2	Machine Learning and Featurization	88
4.4.3	Experiment Results	91
4.4.4	Result Analysis	95
4.5	Summary	97
5	Conclusion	99
5.1	Contribution of this Thesis	99
5.2	Future Work	100

List of Figures

1.1	An example of NER in the newswire domain.	3
1.2	An example of NER in the biomedical domain.	3
1.3	The web search results with the query, “Who is the current CEO of Apple.”	4
1.4	The content of the third ranked webpage that mentions Tim Cook is the CEO of Apple.	5
1.5	The web search results with the query, “Who are the founders of Google.”	7
1.6	The NER result on the text, “Google was founded by Larry Page and Sergey Brin.”	8
2.1	Priorities among four entity types in the Fisher et al. [40]’s system.	20
2.2	The system architecture of the company name extraction system by Rau [101].	22
2.3	NER formulated as a sequence labeling task.	26
2.4	The system architecture of the semi-supervised NER system by Nadeau [87].	29
2.5	Generating character n-grams ($n = 3$) from the word <i>Microsoft</i>	33
2.6	An example of NER errors.	37
3.1	Positional uncertainty of entity words in the training data of the GENETAG corpus.	49
3.2	The hierarchical relation among the seven SRs.	51
3.3	Gene and alternative gene annotations in the BC2GMR training data.	60

3.4	The effect of the proposed method on precision based on the training data size.	67
3.5	The effect of the proposed method on recall based on the training data size.	68
3.6	The effect of the proposed method on F1-score based on the training data size.	69
3.7	Positional uncertainty of entity words in the training data of the CoNLL NER corpus.	72
4.1	An example of syntactically constrained non-local contexts.	79
4.2	An example context of the length 3.	82
4.3	Building a context gazetteer from an encyclopedic database.	83
4.4	Examples of high confidence extracted context patterns.	87
4.5	The feature weights of quantized binary features.	93
4.6	Three sentences excepted from the test data.	96

List of Tables

2.1	Scientific events that involve NER in the newswire domain and their detailed information.	13
2.2	Scientific events that involve NER in the biomedical domain and their detailed information.	15
2.3	The detailed information on the corpora for biomedical IE tasks in the perspective of NER.	16
2.4	The detailed information on the corpora for biomedical IE tasks in the perspective of NER (continued).	17
2.5	Exemplary local features.	32
2.6	Exemplary Global features.	34
2.7	Exemplary external resource features.	35
2.8	Types of errors in NER by Nadeau [87].	38
3.1	Definition of SRs for NER, WS, and SP.	45
3.2	A sample text annotated with various SRs.	47
3.3	Samples of entity words having low positional uncertainty.	50
3.4	Samples of entity words having high positional uncertainty.	50
3.5	Mapping segment labels of the <i>BIES</i> SR to those of the simpler six SRs.	51
3.6	The performance of the seven models on the BC2GMR task.	54
3.7	The comparison of tagging results between the <i>IO</i> and <i>BIES</i> models.	55
3.8	Main and additional SRs used for four groups.	56
3.9	Features for the biomedical NER.	61
3.10	Explanation of symbols used for features.	62
3.11	The performance on the BC2GMR task.	63

3.12	The estimated p values between the proposed and conventional models.	64
3.13	The tagging results of two conventional models (<i>BIES</i> and <i>IO</i>) and a proposed model (<i>BIES&IO</i>).	65
3.14	The performance comparison to the other systems based on the official evaluation.	70
3.15	The distribution of segment labels for each entity type on the CoNLL NER data.	73
3.16	The performance on the CoNLL NER data.	73
4.1	Features used for experiments.	89
4.2	Explanation of symbols used for features.	90
4.3	Performance evaluation using six types of context pattern featurization methods. The upword and downward arrows indicate the change of performance compared to the baseline model.	92
4.4	Performance evaluation using entity and context gazetteers.	94

Chapter 1

Introduction

As the proverb says, *knowledge is power*. This old saying implies that valuable knowledge is rare and hard to obtain. Furthermore, the rapid growth of information in modern society [78] makes people increasingly difficult to acquire relevant information to their needs. Consequently, there is a pressing need for an effective means of finding necessary data from a vast amount of information.

In this thesis, we describe our study on named entity recognition (NER), which aims to recognize important entities, such as people, organizations, and locations, that are mentioned in text. NER is a fundamental task that plays an important role in many fields of study, such as question answering (Q/A) and information extraction (IE), that can improve the accessibility to information. We address one of its most important issues, the data-sparseness problem, from the viewpoint of feature generalization and present two novel methods to alleviate this problem in Chapter 3 and 4 respectively.

Before proceeding to the main chapters, we explain NER in Section 1.1, describe how it can help people to efficiently acquire information in Section 1.2, and show the overview of this thesis in Section 1.3.

1.1 Named Entity Recognition

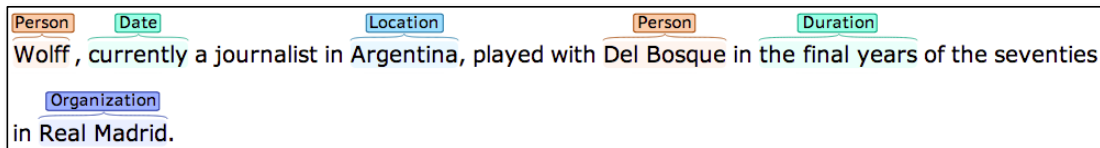
Named entity recognition (NER) is the task that identifies mentions of entities in text and classifies them into one of pre-defined entity types. To avoid confusion though this thesis, we will explain three terminologies used in this definition: entity, entity mention, and entity type.

An *entity* is an object that exists in the world; for example, every individual is a distinctive entity. The word “named” in the expression “named entity recognition” is used to restrict entities to rigid designators, which refer to the same things in all possible worlds in which that objects exist and never designate anything else, as defined by Kripke [65]. In reality, however, named entities often include temporal and numerical expressions, which may not be rigid designators depending on the context in which they appear. For instance, April 2013 is a rigid designator, whereas April is not.

An *entity mention* is the realization of an entity. For example, “William Henry Bill Gates III,” “Bill Gates,” and “Gates” are entity mentions that can refer to the person who is best known as the co-founder of Microsoft Corporation. Notice that an entity can be mentioned in many ways and NER usually does not try to figure out if two or more entity mentions indicate the same entity or not. This problem has been considered as a separate task called co-reference resolution [16, 117, 119]. In addition, an *entity word* indicates a word that is a part of an entity mention.

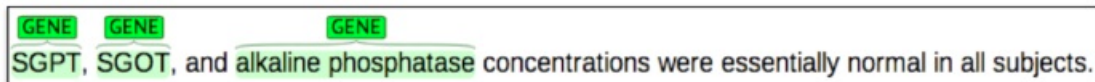
Lastly, each entity belongs to one of pre-defined *entity types*. For instance, Bill Gates is an entity of type *Person*. In the newswire domain, person, organization, and location have been considered as the most important types and time, date, and monetary amount are also frequently targeted [88]. On the other hand, biomedical substances such as genes, proteins, and chemicals are regarded as important entity types in the biomedical domain [22].

Figure 1.1 shows an example of an NER task in the newswire domain using an



Person Date Location Person Duration
Wolff, currently a journalist in Argentina, played with Del Bosque in the final years of the seventies
Organization
in Real Madrid.

Figure 1.1: An example of NER in the newswire domain.



GENE GENE GENE
SGPT, SGOT, and alkaline phosphatase concentrations were essentially normal in all subjects.

Figure 1.2: An example of NER in the biomedical domain.

example text presented in the CoNLL 2002 shared task home page¹. We applied the Stanford Named Entity Recognizer and discovered five types of entities; each of them is person (“Wolf” and “Del Bosque”), date (“currently”), location (“Argentina”), duration (“the final years”), and organization (“Real Madrid”). Notice that three entity mentions (“Del Bosque,” “the finally years,” and “Real Madrid”) consist of multiple words and two temporal expressions (“currently” and “the final years”) are not rigid designators.

In addition, Figure 1.2 shows an example of a biomedical NER task that aims to identify the mentions of gene names. The text in this figure is excerpted from the GENETAG [124] corpus and three gene names (“SGPT,” “SGOT,” and “alkaline phosphatase”) are shown in green color.

1.2 Living in the Age of Information Overload

The Internet has revolutionized the way people access information. Reading news, buying products, and communicating with others on-line have become very natural to all of us. These activities have not only increased the consumption of information, but also accelerated the production of information by individuals and organizations. For example, publishers now sell digitalized books in addition to

¹URL: <http://www.cnts.ua.ac.be/conll2002/ner/>

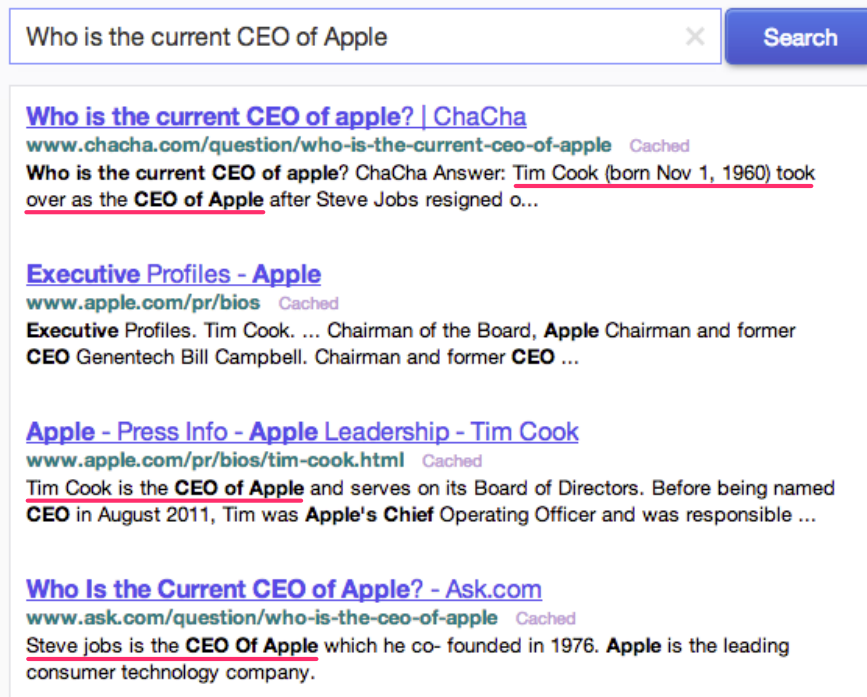


Figure 1.3: The web search results with the query, “Who is the current CEO of Apple.” Relevant information to the query is underlined in red color.

their paper editions, consumers write reviews on the products they purchased, and people tweet and post to their blogs. Everyday, an enormous amount of information is being produced. This phenomenon can lead to the information overload [47] that causes difficulty in understanding an issue or making decisions. Therefore, an effective means of discovering necessary information is essential to make the best use of such a huge amount of information available in the modern information age.

One of the most popular solutions for finding information in the Web is to use web search engines such as *Google* and *Yahoo*. Suppose that we want to know the name of the current CEO of Apple, which is a quite simple question. For a query like “Who is the current CEO of Apple,” a search engine will return a ranked list of documents based on the relevance to the query as shown in Figure 1.3. Yahoo web search engine is used for this example. In this result, the snippets of the first and third ranked documents mention that the CEO of Apple is *Tim*



Figure 1.4: The content of the third ranked webpage that mentions Tim Cook is the CEO of Apple, which is underlined in red color.

Cook. The second one also provides the same information as its content is about the executive profiles of Apple including the CEO, Tim Cook, and vice presidents, although its snippet does not correctly reflect it. However, the snippet of the fourth ranked webpage says that *Steve jobs* is the CEO of Apple. This kind of situation, where search results are contrary to each other, forces us to read the content of these documents for figuring out which one is the correct information. For example, the third ranked document is the press information of Apple's CEO as shown in Figure 1.4. This page provides decisive information on the query since it is an official webpage of Apple. On the other hand, the information in the fourth ranked webpage is from a question answering (Q&A) focused web search engine, *Ask.com*, and naturally less reliable than the official press information. Examining the content of this webpage reveals that the original answer mentioning that Steve Jobs is the CEO of Apple is outdated because there is an additional comment pointing out that Time Cook is the current CEO.

There is no doubt that information retrieval (IR) technologies, such as web search engines, are indispensable tools. However, their goal of finding relevant documents to a few query words is becoming less satisfactory because people still have to spend a non-negligible amount of time on reviewing search results for

various reasons. For example, search results can be contradictory to each other, as mentioned above, or information can be scattered around multiple documents. To lift a burden from people and improve the accessibility to information, we need more advanced technologies that can deal with complex information needs and pinpoint necessary information at the level of smaller units, such as paragraphs, sentences, or even exact answers to a query rather than documents.

These problems have been tackled in various fields of study such as question answering (Q/A) and information extraction (IE). Q/A aims to directly answer a question instead of showing a list of relevant documents. For example, a Q/A system will answer the factoid question², “Who is the president of the United States,” with the exact name of the U.S. president, “Barack Obama.” On the other hand, IE is the task that transforms unstructured and semi-structured data (e.g., plain text and web pages) into structured information (e.g., entities and their relations). Information represented in structured form is much easier to exploit in a systematic way than unstructured or semi-structured data. It allows us to deal with complex queries that involve semantic constraints because these constraints can be evaluated on structured information. Google web search engine implemented these kinds of features for simple factoid questions. Figure 1.5 shows the Google search results with the must-be-answered query, “Who are the founders of Google.” The answer to the query, “Larry Page” and “Sergey Brin,” is directly shown at the top of the page. Compared to the traditional IR search results, it is very simple and efficient.

Although it is difficult to figure out exactly how Google makes it work, we assume that it includes three fundamental modules as most Q/A systems do. These modules recognize the type of a question, retrieve documents relevant to the query and its type, and extract (or generate) the answer in serial order [49]. In this procedure, it is crucial to narrow down documents for processing since Q/A sys-

²A factoid question is a fact-based question that can be answered shortly (e.g., with a few word) [31]

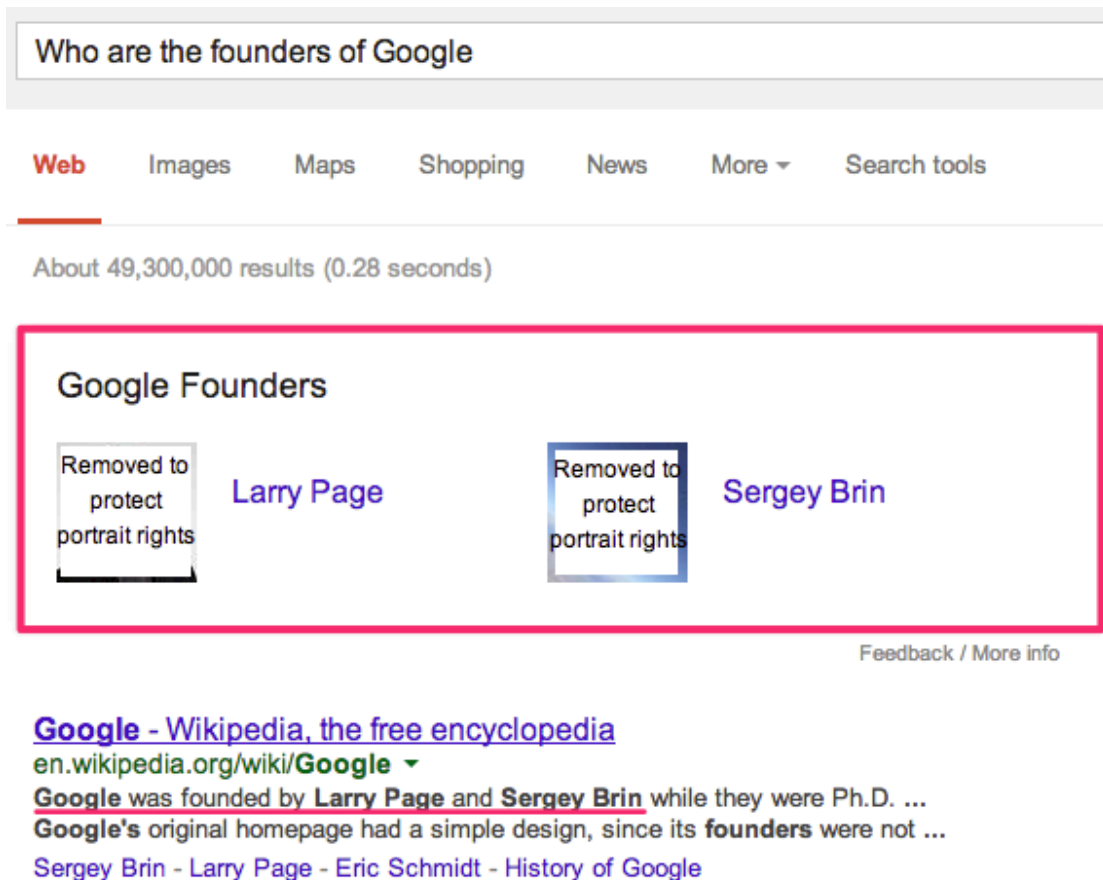


Figure 1.5: The web search results with the query, “Who are the founders of Google.” The names of two founders, Larry Page and Sergey Brin, are shown in the red box.

tems, in general, utilize a variety of NLP techniques, which are computationally very expensive. Named entity recognition (NER) plays an important role for this purpose. Suppose that a Q/A system figured out the type of the question mentioned above simply by recognizing the interrogative pronoun *Who*. Then, NER recognizes entity mentions in both the query and documents. For instance, the Figure 1.6 shows the the snippet of the first ranked document in Figure 1.5, which is marked with the red underline. It also shows three entity mentions identified in the text by using the Stanford NER system³, “Google” as the organization type

³We used the Stanford Named Entity Recognizer [39] for the identification of entity mentions

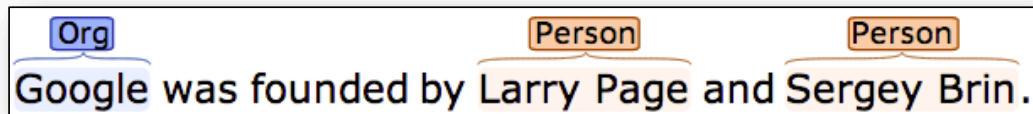


Figure 1.6: The NER result on the text, “Google was founded by Larry Page and Sergey Brin.” Three entity mentions, *Google* as the organization type and *Larry Page* and *Sergey Brin* as the person type, are recognized.

and “Larry Page” and “Sergey Brin” as the person type. Now, the Q/A system can exclude the documents that do not mention any names of *people*; moreover, it can further reduce the number of documents by filtering out the documents that do not refer to the *organization Google*.

In this thesis, we focus on NER because it plays a crucial role in various areas that improve the accessibility to information. Furthermore, most information processing systems such as Q/A and IE are cascaded systems that use the output of one sub-system as the input of another sub-system. Since cascaded systems frequently suffer from errors propagated from earlier stages and NER is a very beginning step, the first sub-system in many cases, improving its performance will have most significant impact on overall performance.

1.3 The Overview of This Thesis

In this chapter, we explained NER and described the necessity of various information technologies for people living in the modern information society. The following chapters are organized as follows:

Chapter 2 begins with the summarization of the historical background over the last two decades. Especially, we focus on diverse scientific events that involved or dedicated to NER research because the advancement in NER research has been

and the BRAT rapid annotation toolkit [118] for visualization. URLs: <http://nlp.stanford.edu/software/corenlp.shtml> and <http://brat.nlplab.org/>.

mostly driven by community-wide competitions. Then, we explain rule-based, dictionary-based, and statistical approaches to NER tasks and the changes of mainstream approach in the late 1990s. The third section describes various features utilized for NER in previous studies and the fourth section introduces evaluation matrices. Finally, the last section explains two challenging issues that we will tackle in this thesis.

Chapter 3 deals with the issue of segment representations (SRs), such as the *IOB2* and *IOBES* notations, for recognizing multi-token named entities. In previous studies, the choice of a better SR has been regarded as a secondary thing to do compared with designing sophisticated features that encode textual characteristics of named entities. While examining the effects of different SRs on NER tasks, however, we noticed that incorporating multiple SRs into a single NER model could alleviate a harmful effect resulting from the data-sparseness problem. We present a novel feature generation method [19] that incorporates multiple segment representations (SRs) into a single NER model. Evaluation results show that new NER models utilizing the proposed method consistently outperform conventional NER models.

Chapter 4 describes the brittleness of state-of-the-art NER systems that heavily depend on local contexts and introduces related studies that exploit sentence-level and document-level non-local features to overcome this problem. Then, we propose the use of a context gazetteer [18], a list of contexts with which entity mentions can co-occur, as a new sentence-level non-local context feature resource. The main difference of our approach from previous work is that it automatically creates a context gazetteer from an encyclopedic database. Therefore, a generated gazetteer has rich and sophisticated context patterns. Experiment results show that an NER model utilizing a context gazetteer achieves higher precision and recall over a strong baseline model.

Chapter 5 summarizes our studies that tried to alleviate one of the most challenging problems in NER research and the research direction for future work.

Chapter 2

Background

For extracting information from text, it is essential to recognize smallest information units that participate in the construction of more complex knowledge and eventually the whole picture of the text. NER takes the responsibility of this task in most information processing systems.

NER has been studied for a long time as a fundamental research topic in the information extraction (IE) and natural language processing (NLP) areas. Especially, much of the advancement in NER has been driven by task specific competitions. In Section 2.1, we explain the historical background of NER with influential conferences and workshops that have made a great impact on this field. While many studies were based on rule-based and dictionary-based approaches in the early days of NER research, most of current NER systems use a machine learning approach. Section 2.2 describes the change of methodological approaches to NER over the last two decades. In Section 2.3, we summarize various features proposed in previous studies since they are one of the most important factors that significantly affect performance of NER systems. In Section 2.4, we explain evaluation criteria used for scoring NER systems. It is important to choose an appropriate evaluation scheme depending on the purpose of NER. If a NER system is a pre-processing component of a bigger system, a strict evaluation scheme will be better than a relaxed one. On the other hand, a relaxed evaluation scheme will be

sufficient if a user directly consumes the output of a NER system. Lastly, Section 2.5 briefly introduces two research issues that we have tackled in this thesis.

2.1 Historical Background

In this section, we explain the historical background over the last two decades by introducing influential scientific events that spurred research in the NER field. We summarized this information in two well-studied domains separately, the newswire and biomedical areas, because these domains have different challenging issues while sharing some similar problems.

2.1.1 NER in the Newswire Domain

A pioneering research [101] in this domain can be traced back to early 1990s. In this study, the author describes a system to extract company names from financial news stories. Research in the NER field started to accelerate in 1996 with the sixth Message Understanding Conference¹ (MUC-6) [46] that involved NER as a separate task for the first time because of its importance in IE.

Numerous scientific events, which are dedicated to or involve NER, have followed MUC-6. The National Institute of Standards and Technology (NIST) has supported many of these events including the subsequent MUC-7 and its multilingual portion known as the Multilingual Entity Task 2 (MET-2) [16] in 1997, the Broadcast News Recognition Evaluation (HUB-4)² [53] in 1998, and the Automatic Content Extraction (ACE) Evaluation³ [33] from 1999 to 2008. The Conference on Natural Language Learning (CoNLL)⁴ also held two influential shared tasks for NER [125, 126] from 2002 to 2003. The datasets released in these shared tasks are regarded as de facto standards for measuring performance of a NER system in

¹http://www.itl.nist.gov/iaui/894.02/related_projects/muc/index.html

²<http://www.itl.nist.gov/iad/mig/tests/bnr/1998/>

³<http://www.itl.nist.gov/iad/mig/tests/ace/>

⁴<http://ifarm.nl/signll/conll/>

Name	Year	Language	Entity Types
MUC-6	1996	en	PER, ORG, LOC, Date, Time, Money, Percentage
MUC-7 & MET-2	1997	en, zh, ja, es	Same to MUC-6
HUB-4	1998	en, zh, es	Same to MUC-6
CoNLL	2002	es, nl	PER, LOC, ORG, MISC
	2003	en, de	Same to CoNLL 2002
ACE	2000	en	PER, ORG, LOC, GPE, FAC
	2000	en, zh	Same to ACE 1
	2003	en, zh, ar	Same to ACE 1
	2004	en, zh, ar	PER, ORG, LOC, GPE, FAC, VEH, WEA
	2005-2007	en, zh, ar	PER, ORG, LOC, GPE, FAC, VEH, WEA, and sub-types
	2008	en, zh, ar	PER, ORG, LOC, GPE, FAC, and sub-types

Table 2.1: Scientific events that involve NER in the newswire domain and their detailed information. In the Language column, ar, zh, nl, en, de, ja, and es refer to Arabic, Chinese, Dutch, English, German, Japanese, and Spanish languages following the ISO 639-1 standard. In the Entity types column, PER, ORG, LOC, GPE, FAC, VEH, WEA, and MISC stand for person, organization, location, geopolitical entity, facility, vehicle, weapons, and miscellaneous names.

the newswire domain.

While most of these competitions targeted English text as a source of information, there was an effort to overcome this limitation. For example, MET2 as a part of MUC-7 introduced Chinese, Japanese, and Spanish for the NER task. HUB-4 also used not only English but also Chinese and Spanish for IE. ACE evaluations first used only English text, but gradually expanded its target languages to include Chinese and Arabic. The Information Retrieval and Extraction Exercise (IREX) [110] and HAREM [108], which stands for a NER evaluation in Portuguese, tar-

geted Japanese and Portuguese respectively. The information on these events are summarized in Table 2.1.

2.1.2 NER in the Biomedical Domain

Published articles are a valuable source of information. In the biomedical domain, however, an enormous quantity of published articles hinder researchers from finding important information relevant to their research even within their own field of study. The statistics of MEDLINE⁵, which is the most influential bibliographic database in the life science area, shows that the database contains over 19 million references in over 6,000 international journals, and the number is continuously growing by adding nearly 600,000 references every year [22].

The biomedical community extensively uses IE technologies to reduce human efforts in the search of information. NER, as one of frequently used IE technologies, can expedite the curation of terminology databases by filtering out unnecessary parts of text while recommending important regions [1]. Moreover, NER is necessary for more complex IE stages such as relation extraction and event extraction.

In late 1990s, Fukuda et al. [43] proposed a rule-based system that identifies protein names from biological papers. The ABGene⁶ [122, 123] was one of the earliest publicly available NER systems in the biomedical domain. It works on top of an extended Brill POS tagger [11] and uses manually created post-processing rules to recognize gene and protein names.

From 2000, community-wide efforts for biomedical IE research came into action in the form of competitions as summarized in Table 2.2. The JNLPBA shared task⁷ [59] is one of the earliest competitions in this domain. The shared task uses a subset of the GENIA corpus [58] and aims to recognize five biomedical substances related to transcription factors in human blood cells. The Critical Assessment

⁵MEDLINE is the U.S. National Library of Medicine[®]'s (NLM[®]) premier bibliographic database for life sciences with a concentration on biomedicine.

⁶<ftp://ftp.ncbi.nlm.nih.gov/pub/tanabe/AbGene>

⁷<http://www.nactem.ac.uk/genia/shared-tasks/bionlp-jnlpba-shared-task-2004>

Name	Detailed Information	
JNLPBA	Year	2004
	Genre	Transcription factors in human blood cells
	Size	2000/404 abstracts for training/testing
	Entity types	Protein, DNA, RNA, Cell-line, Cell-type
BioCreAtIvE	Year	2005, 2007
	Genre	Unrestricted
	Size	10,000/5,000 sentences for 2005
		15,000/5,000 sentences for 2007
Entity types	Gene	
CALBC	Year	2009, 2010
	Genre	Immunology
	Size	50,000/100,000 abstracts for 2009
		100,000/up to 850,000 abstracts for 2010
Entity types	CHED, PRGE, DISO, SPE	

Table 2.2: Scientific events that involve NER in the biomedical domain and their detailed information. In the Entity types column, CHED, PRGE, DISO, and SPE refer to chemical entities and drugs, genes and proteins, diseases and disorders, and species respectively.

of Information Extraction in Biology (BioCreAtIvE) challenges⁸ [50, 115], on the other hand, involve only genes and proteins (as a single category) in its NER task and the topics are not restricted to specific biomedical fields. The Collaborative Annotation of a Large Biomedical Corpus (CALBC) competitions⁹ [103] use four categories (chemical entities and drugs, genes and proteins, diseases and disorders, and species) on immunology. However, CALBC is different from other competitions in two ways. First, the annotated corpus has been created automatically, which is called the silver standard corpus. Second, the corpus consists of 50,000 Medline abstracts for training and 100,000 documents for testing. This is almost

⁸<http://www.biocreative.org/>

⁹<http://www.ebi.ac.uk/Rebholz-srv/CALBC/>

Corpus Name	Detailed Information	
GENIA	Year	2003
	Size	2,000 abstracts
	Entity types	47 types of biological entities on transcription factors in human cells [58]
GENETAG*	Year	2005
	Size	20,000 sentences
	Entity types	Gene/Protein
AIMed	Year	2005
	Size	225 abstracts
	Entity types	Protein
BioInfer	Year	2007
	Size	1,100 sentences
	Entity types	35 types similar to GENIA [98]
PennBioIE CYP	Year	2008
	Size	1,100 abstracts
	Entity types	5 types of 3 categories on the inhibition of cytochrome P450 enzymes [69]
PennBioIE Oncology	Year	2008
	Size	1,414 abstracts
	Entity types	24 types of 5 categories on cancer, concentrating on molecular genetics [69]

Table 2.3: The detailed information on the corpora for biomedical IE tasks in the perspective of NER. The corpora marked with the asterisk are designed for NER, while the others are for different IE tasks.

ten times larger than the other corpora used in the JNLPBA and BioCreAtIvE competitions.

In addition to these competitions, there have been many attempts to create corpora for biomedical IE. Some of them are designed solely for NER, whereas the others involve named entity annotations for more complex IE tasks such as relation extraction and event extraction, which can be used for NER. Table 2.3

Corpus Name	Detailed Information	
SCAI*	Year	2008
	Size	100 abstracts
	Entity types	6 types of chemical entities [63]
AZDC*	Year	2009
	Size	2,783 sentences (793 abstracts)
	Entity types	Disease
LINNAEUS*	Year	2010
	Size	100 full text articles
	Entity types	Species
AnEM*	Year	2012
	Size	200 abstracts and 300 sections
	Entity types	11 types of anatomical entities [94]
CellFinder*	Year	2012
	Size	2,100 sentences (10 full text articles)
	Entity types	Anatomical part, Gene/protein, Species, Cell component, Cell types, Cell line
NCBI Disease*	Year	2012
	Size	2,783 sentences (793 abstracts)
	Entity types	Specific disease, Disease class, Modifier, Composite mention

Table 2.4: The detailed information on the corpora for biomedical IE tasks in the perspective of NER. The corpora marked with the asterisk are designed for NER, while the others are for different IE tasks (continued).

and 2.3 give the detailed information on these corpora in the perspective of NER.

The diversity of entity types is a distinctive characteristic of biomedical NER and IE tasks. For example, the original GENIA corpus [58] involves 47 types of biological entities in the sub-domain based on three MeSH¹⁰ terms, transcription factor, human, and blood cell. The PennBioIE [69] CYP and Oncology corpora

¹⁰The **Medical Subject Headings** (MeSH) is the National Library of Medicine’s controlled vocabulary thesaurus.

have 5 and 24 types of biomedical named entities respectively. In case of the AnEM [94] corpus, 11 anatomical entity types are annotated.

Looking at the entity types of these corpora, we can also notice what kinds of entities are considered more important than others. Genes and proteins, whether they are treated separately or not, have been the most important entity types in various biomedical IE tasks such as NER [58, 69, 124], relation extraction [13, 58, 98], and event extraction [58, 91, 92, 93, 99]. During last few years, however, other types are getting the attention from the community too. For instance, AZDC [72] and NCI disease [34] are annotated corpora for disease names, SCAI corpus [63] for six types of chemical compound names, LINNAEUS [45] for species names, CellFinder [90] for six types of cell names, and AnEM [94] for 11 types of anatomical entities.

More information on the history and the advancement of biomedical IE technologies can be found in the survey articles by Cohen and Hersh [22], Nadeau and Sekine [88], and Simpson and Demner-Fushman [114].

2.2 Methodological Background

NER has been studied for a long time and various approaches have been proposed. Most of these approaches can be divided into three categories: dictionary-based, rule-based, and machine learning based approaches. This section presents the survey of previous studies based on these categories and describes exemplary methods. Research based on a hybrid approach will be explained in the section that deals with its primary method.

2.2.1 Dictionary-based Approach

Dictionary-based NER uses dictionaries of target entity types (e.g., dictionaries of the names of people, companies, locations, etc.) and identifies the occurrences of the dictionary entries (e.g., Bill Gates, Facebook, Madison square, etc.) in text

[40, 45, 64, 89]. This approach, while looks very straightforward at first glance, has two difficulties due to the productivity and ambiguity of natural language. First, entities can be referred to in various ways. For instance, Thomas Alval Edison, Thomas Edison, and Edison can be used to mention a famous American inventor. Unfortunately, it is not possible to create a comprehensive dictionary, which enumerates all of these variations, in most cases. Second, even the same entity mention can designate multiple entities. For example, “Washington” is the name of the first president of the U.S. as well as the name of a state in the U.S. [28]. Therefore, a NER system that relies on a dictionary has to deal with these problems.

To address the first issue, Krauthammer et al. [64] employed an approximate string matching technique. Their biomedical NER system uses the Basic Local Alignment Search Tool (BLAST) [3], which is a popular DNA and protein sequence comparison tool, for identifying not only gene and protein names but also their spelling variations. Navarro et al. [89] proposed a NER system, *Matchsimile*, which recognizes person and company names. In addition to an approximate string matching technique, it also utilizes a set of personal names formation rules such as combination, abbreviation, ordering, omission, and insertion of words. A species name recognition system, LINNAEUS [45], tackled this problem by using a set of regular expressions generated from a dictionary.

The second problem, the ambiguity of entity types, is a characteristic of natural language. Ordinary language is inherently ambiguous because generating unambiguous expressions is often very costly and some ambiguities can be easily resolved by resorting to supplementary information such as linguistic and communicative context [42]. Fisher et al. [40] tried to solve this problem by assuming the priorities between entity types. Their NER system, which participated in the MUC-6 competition, is organized in a serial architecture so that the predictions made at the earlier stages (the types of higher priorities) cannot be violated by the outputs at the latter stages (the types of lower priorities). Figure 2.1 shows the priorities

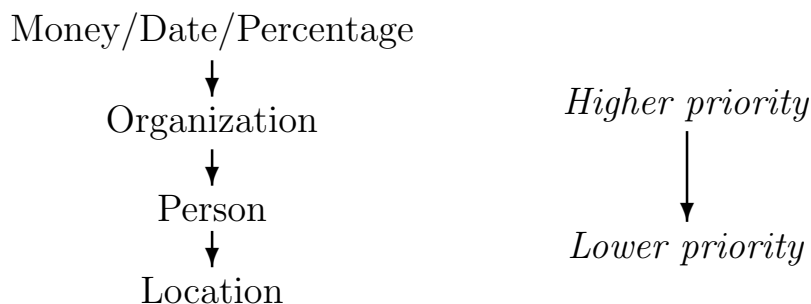


Figure 2.1: Priorities among four entity types in the Fisher et al. [40]’s system.

among four entity types: numeric expression (money, date, and percentage), organization, person, and location. While Fisher et al. [40] gives priorities to entity types, LINNAEUS [45] used contextual information in text for disambiguating the types of recognized species names. For a species name that has more than two candidate types and where one of the possible types is mentioned explicitly within text, LINNAEUS uses this type for all occurrences of the species name. It also disambiguates acronyms by detecting acronym definitions in the form of “*species (acronym)*” where *species* is in the dictionary and *acronym* is a sequence of capital letters, digits, or hyphens.

An important advantage of dictionary-based NER is its inherent ability for semantic disambiguation, which is also called entity normalization in the biomedical domain [21, 77]. Dictionary-based NER, as its name means, identifies entity mentions that match to dictionary entries. As a result, each entity mention recognized in text retains the information about the relation between the entity mention and the matched dictionary entry. Using this information, we can find out a group of entity mentions that indicates the same entity regardless of their surface forms. In addition, it is also possible to distinguish two entity mentions of the same surface form into different entities. Other approaches usually employ an independent step for this process after NER.

2.2.2 Rule-based Approach

Rule-based NER systems rely on hand-crafted rules for identifying entity mentions. These rules can be structural, contextual, or lexical patterns [67]. For example, the following list shows two rules of recognizing corporation names and person names:

- $\langle \textit{proper noun} \rangle^+ \langle \textit{corporate designator} \rangle \rightarrow \langle \textit{corporation name} \rangle$
- $\langle \textit{capitalized last name} \rangle, \langle \textit{capitalized first name} \rangle \rightarrow \langle \textit{person name} \rangle$

The first rule detects company names that consist of one or more proper nouns followed by a corporate designator such as “Microsoft Corporation” and “Ford Motor Company.” The second rule recognizes person names written in order of family name, comma, and given name.

This approach has advantages compared to a dictionary-based approach. First, it does not require a large dictionary¹¹. Preparing dictionaries that have enough coverage on target entity types is often too costly, especially for resource-poor entities and languages [37, 104]. In such a situation, a rule-based approach is a way to go. Second, rule-based NER systems can handle unknown entities better than dictionary-based systems. In professional areas, for example, terminologies (entities) often follow domain-specific nomenclatures. A rule-based system can easily recognize terminologies that follow systematic naming conventions, whether they are already known or not, by using a small number of rules. On the other hand, a dictionary-based system needs a list of these terminologies, which can be hundreds of thousands entries.

In 1991, Rau [101] proposed a rule-based company name extraction system, which is often cited as the root of NER research. This system exploits various linguistic cues (rules) for identifying company names. In Figure 2.2, starting at the left top with mixed case input, the system recognizes company names by looking

¹¹Although rule-based systems can use extraction rules exclusively, utilizing a small dictionary in addition to the rules usually helps to improve the results.

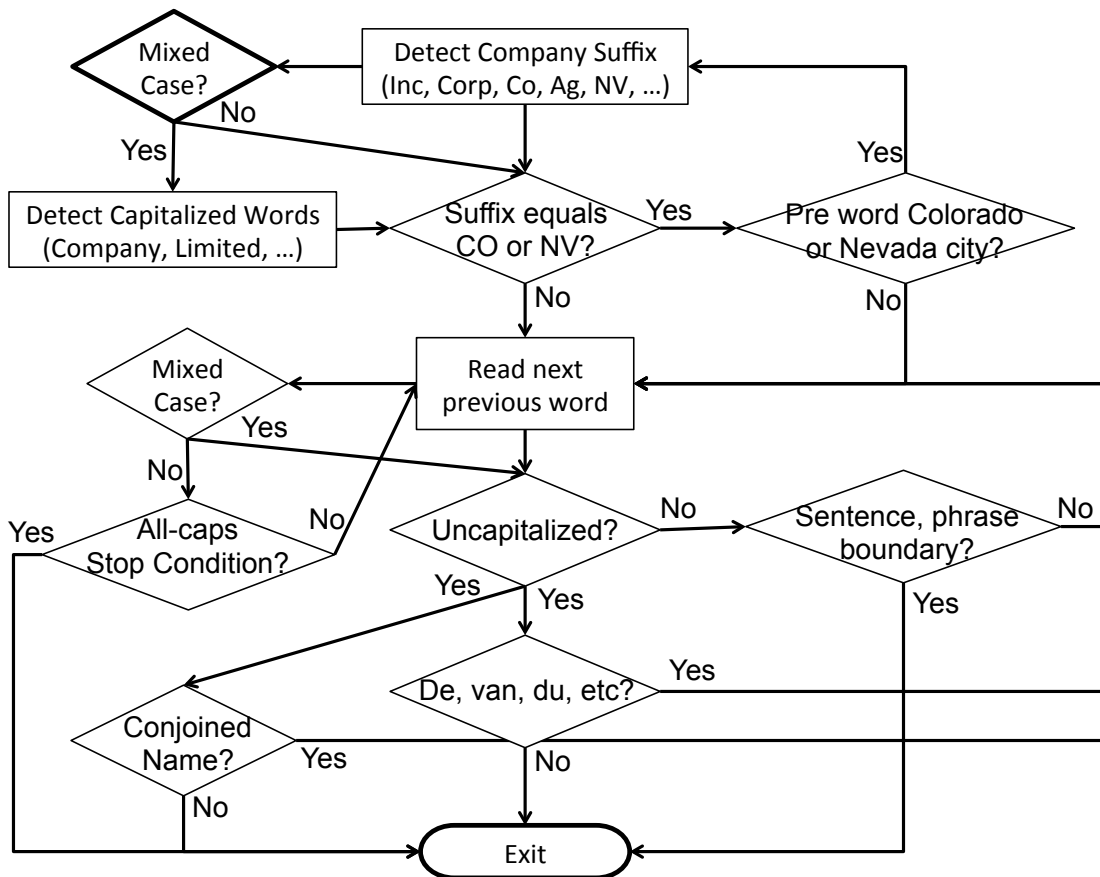


Figure 2.2: The system architecture of the company name extraction system by Rau [101].

backward from a company name indicator (e.g., Incorporated, Corporation, etc.) to the first non-capitalized word. However, this strategy fails if input text consists of only upper case letters or a company name contains a conjunction. To deal with the first issue, the author developed complex stop conditions based on a stopword list, company name length restriction, and syntactic analysis results. For the second problem, it uses three heuristics to determine if a conjunction belongs to a company name.

Beginning with the MUC-6 and MUC-7 conferences [17, 46], many NER systems started to appear such as FASTUS [4], LaSIE [44], UMass System [40], and

NetOwl [67]. Similar to Rau [101]’s system, they use a set of manually curated rules carefully designed for a target domain. In the biomedical domain, many NER systems have been developed for identifying the names of various biomedical substances. For example, PROPER [43], to the best of our knowledge, is the first biomedical NER system that identifies protein names. ABGene [122] is one of the earliest publicly available gene name recognition systems and contributed to the creation of the GENETAG corpus [124]. LINNAEUS [45], which is released in 2010, detects species names in biomedical text.

A rule-based approach, however, has drawbacks too. First, for designing rules of high precision and broad coverage, domain experts must consider all the ways in which the target information is expressed and think of their plausible variations [4]. This process takes a significant amount of time and often needs many iterations of trial and error for improving its performance. Second, extraction rules created from a corpus are very domain dependent in terms of entity types, textual genres, languages, and so on. Considering that domain adaptation is one of the biggest needs in NER, this is a serious disadvantage of a rule-based approach.

Eventually, these difficulties triggered the migration of the mainstream strategy for NER from a rule-based approach to a machine learning approach in the late 1990s.

2.2.3 Machine Learning Approach

Dictionary-based and rule-based approaches were two dominant paradigms in the early days. However, most current state-of-the-art NER systems employ machine learning techniques as their core component. This trend stands out when we compare the ratio of the rule-based systems that participated in the MUC-7 competition and the CoNLL shared task that held in 1997 and 2003 respectively. In MUC-7, five out of six systems used rule-based approaches, whereas the sixteen systems in the CoNLL shared task were based on supervised learning techniques [87].

A machine learning approach is superior to rule-based and dictionary-based approaches in many aspects. It can resolve the ambiguity of entity types by exploiting contextual information (e.g., words that frequently co-occur with entity mentions) better than a dictionary-based approach. Compared to a rule-based approach, annotating training data for supervised machine learning (or preparing a few seed samples for semi-supervised machine learning) is much simpler than devising complex rules of high precision and coverage. This characteristic implies that domain adaptation of a machine learning based NER system is relatively easier than that of a rule-based system. Above all, however, the most important advantage is that an underlying machine learning algorithm automatically disambiguates entity mentions. It is not necessary to consider how to resolve the ambiguity of entity mentions and their types or how to apply rules when different orders of applying these rules lead to inconsistent results. Consequently, a machine learning approach can greatly reduce human involvement and cost in developing a NER system.

Researchers have applied various machine learning techniques to NER and proved their effectiveness. Most of these techniques can be categorized into three types: supervised learning, semi-supervised learning, and unsupervised learning approaches. A supervised learning approach trains a model from labeled data and uses it to predict the labels of new input data. On the other hand, a semi-supervised learning approach uses a small number of seed data and iteratively increases training data by annotating unlabeled text automatically. Lastly, an unsupervised approach typically uses clustering techniques. It splits input data instances into groups based on the similarity of instances. In this section, we describe these approaches with representative previous studies.

Supervised Learning Approach

A supervised learning approach is the current dominant technique for solving NER problems. In this approach, NER is mostly formalized as a sequence labeling task

in which each word of input text is represented with one of pre-defined labels. A supervised NER system trains a model using a large amount of training data labeled by human annotators; and then, it uses the trained model for identifying entity mentions in new input text.

Previous studies have applied various machine learning techniques to NER tasks such as Hidden Markov Models (HMMs) [8], Decision Trees [111], Maximum Entropy (ME) [10], Neural Networks [14], Support Vector Machines (SVMs) [5], and Conditional Random Fields (CRFs) [82]. Especially, CRFs [38, 39, 55, 66, 71, 80, 82, 112, 134] and SVMs [52, 57, 81, 85, 121, 133] have received a great attention because of their exceptional performance in NER tasks.

A supervised learning approach mostly formulates NER as a sequence labeling task. The goal of this task is to find the most likely label sequence for an input sentence. Figure 2.3 shows an example of NER as a sequence labeling task with an input text, “Wolff, currently is a journalist in Argentina, played with Del Bosque in the final years of the seventies in Real Madrid.” In this figure, each token¹² can take one of pre-defined labels such as the *O*, *B-PER*, and *I-PER* labels. Every label, except the *O* label, consists of two parts: a segment label, which indicates the position of a token within an entity mention, and a class label, which indicate the type of an entity mention to which a token belongs. If a token has the *B-PER* label like “Wolf” in the above example, the token is a part of the entity mention of type *Person* and it appears at the beginning of the entity mention. The output will be the most likely label sequence as shown with red arrows.

In a supervised learning approach, two important factors have to be considered for developing a high performance NER system. The first one is how to overcome the problems that result from the limited size of available training data; and, the second one is how to exploit rich and sophisticated features. Most NER systems use external resources such as gazetteers [55] to deal with unknown words that do not

¹²A token mostly corresponds to a word. However, it can be larger or smaller units than a word depending on a tokenization scheme.

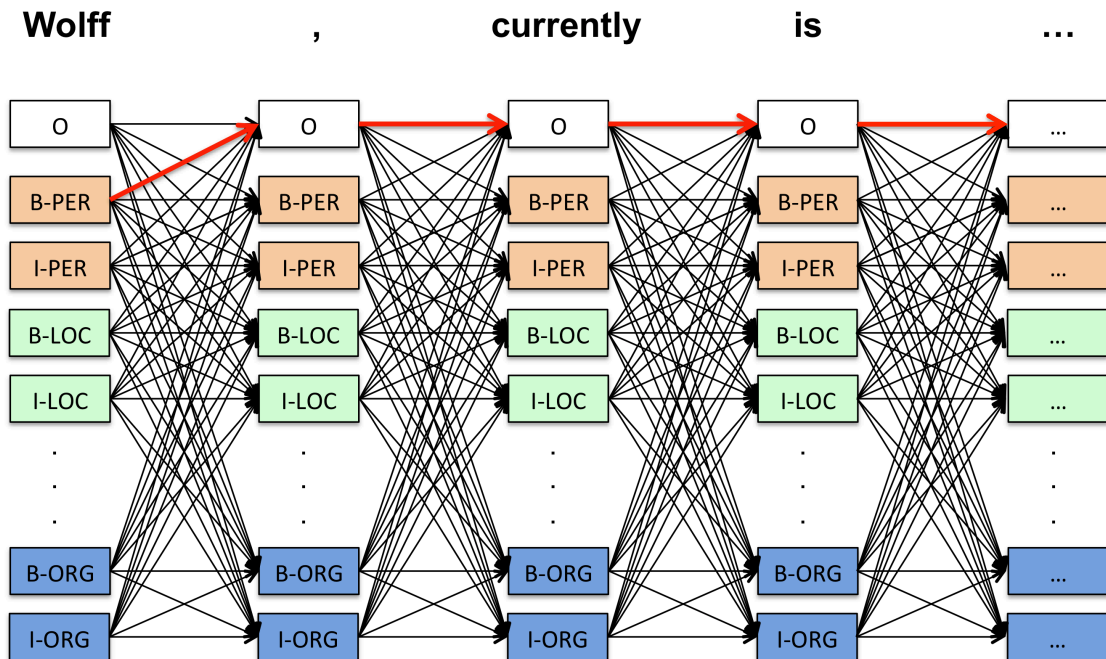


Figure 2.3: NER formulated as a sequence labeling task. The *PER*, the *LOC*, the *ORG*, and the *O* designate entity types such as *Person*, *Location*, *Organization*, and *Outside-of-entity*. The prefixes, *B-* and *I-*, are segment labels that indicate the position of a token within an entity mention such as *Beginning* and *Inside*.

appear in training data. Combining the outputs of multiple classifiers [41] is also effective because different machine learning methods have different generalization power. The second issue has been addressed by using non-local contexts such as context aggregation [15], two-stage prediction aggregation [66] and deep syntactic information [38, 116, 130].

Semi-Supervised Learning Approach

While a supervised learning approach has various advantages compared to dictionary-based and rule-based approaches, it still needs a large amount of manually annotated training data to achieve high performance. Since preparing training data is often a costly and time-consuming task, a semi-supervised learning approach addresses this problem by exploiting a large amount of unlabeled data in addition

to a small number of labeled data. Most semi-supervised systems use a *bootstrapping* method that starts a learning process with a small number of seed data; for example, several entity mentions for each entity type. A bootstrapping method consists of two-step processes. First, it extracts contextual patterns in which entity mentions appear as if a dictionary-based system does. Second, it identifies new entity mentions that co-occur with the extracted contextual patterns at the first step in the same way that a rule-based system does. By iteratively applying this procedure, it is able to obtain a large quantity of entity mentions and contextual patterns for each entity type.

A bootstrapping method has been used in many previous studies. Collins and Singer [25] adopted this method for named entity classification that categorizes an entity mention into one of pre-defined entity types. Their method begins with a rather strong assumption that either an internal feature of an entity mention (e.g., an entity mention begins with “Mr.”) or an external characteristic of an entity mention (e.g., the head word of its appositive modifier is “president”) can sufficiently determine its entity type. Then, it extracts candidate entity mentions and contextual patterns from the parsed New York Times text. A candidate entity mention is a sequence of consecutive proper nouns, or a noun phrase that appears in one of two syntactic structures, apposition and prepositional phrase; and, a candidate contextual pattern is the head word of an appositive modifier to a candidate entity mention, or a preposition of a candidate entity mention together with the noun it modifies. In the following examples, excerpted from their article, **Maury Cooper** and **Georgia** are used as candidate entity mentions and president and plant in as candidate contextual patterns.

- ..., says **Maury Cooper**, a vice president at S.&P.
- ... fraud related to work on a federally funded sewage plant in **Georgia**

These syntactic constraints make candidate entity mentions and contextual patterns have the same labels in most cases. Then, it gives an entity type to each

of candidate entity mentions and contextual patterns by using a bootstrapping algorithm and a small number of seed rules. Lastly, they evaluated their system on the randomly selected 1,000 candidate entity mentions. Riloff and Jones [105] proposed the mutual bootstrapping method that uses only a small number of seed entity mentions as initial input. Their method searches unlabeled text for seed entity mentions and extracts contextual patterns surrounding recognized entity mentions. Then, it uses the extracted contextual patterns to identify new entity mentions in text and iterates these two processes. Unlike the previous work [25], they do not constrain contextual patterns to specific syntactic structures. Cucchiarrelli and Velardi [27] adopted the mutual bootstrapping method [105]; however, they extract contextual patterns from specific syntactic relations (e.g., subject-object) to reduce noise during the bootstrapping process. Pasca et al. [96] utilize the distributional similarity measure [76] to increase the variety of contextual patterns by substituting a part of these patterns with synonyms. For instance, the contextual pattern “X was born in November” will be diversified into new patterns by replacing the word *November* with the other words such as January, February, and March. Nadeau [87] presents two important methods for a semi-supervised learning approach: one is to remove noise during the bootstrapping process and the other one is to identify unambiguous entity mentions for entity type classification.

Figure 2.4 shows the system architecture of a representative semi-supervised NER system proposed by Nadeau [87]. The input of this system is a handful of seed data (e.g., Montreal, Boston, Paris, and Sydney for City) as shown at the top of this figure. In the middle part, which is a group of semi-supervised learners, the list creator produces a dictionary of entity mentions for each entity type by using a bootstrapping method. The noise filter verifies entity mentions recognized by the list creator and discards them if they do not share lexical similarity with other entity mentions of the same type or do not appear in multiple distinct documents. This step is one of the most important processes in a semi-supervised approach since noise in the early iteration of the bootstrapping process significantly deter-

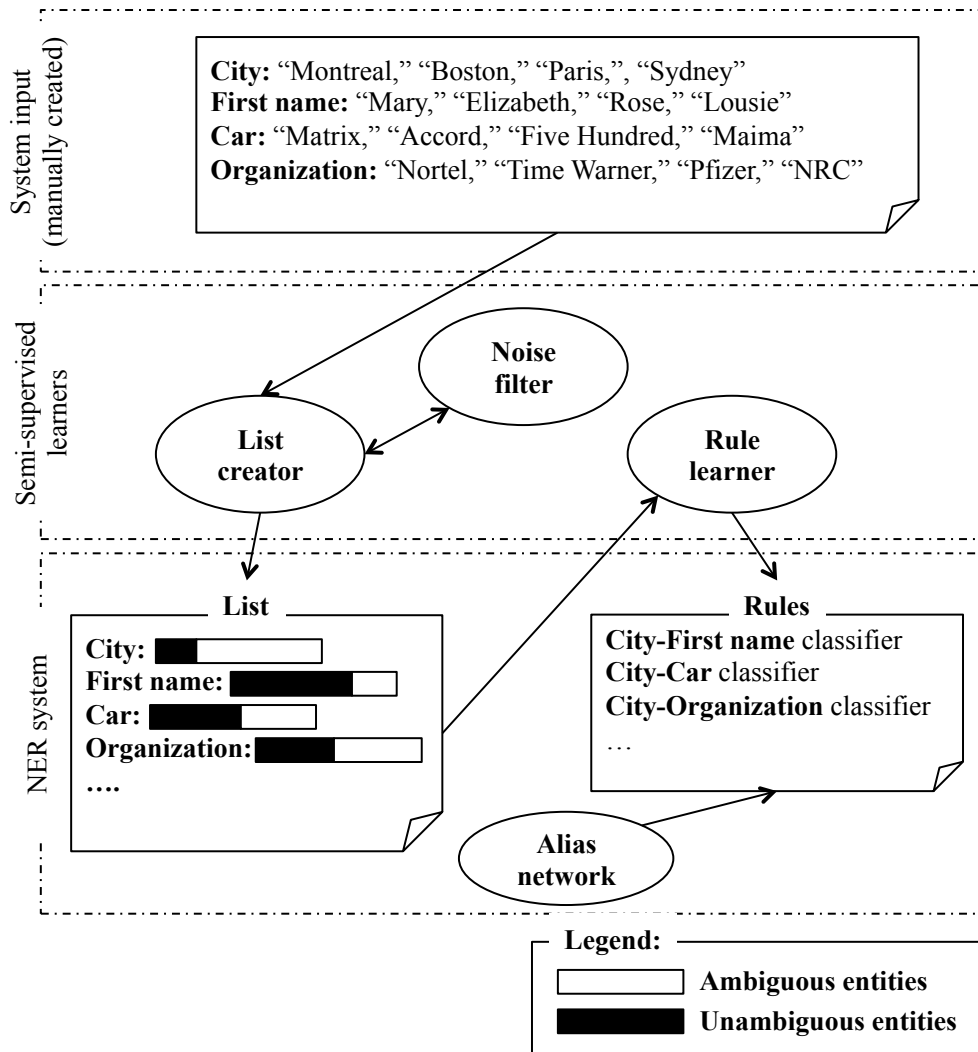


Figure 2.4: The system architecture of the semi-supervised NER system by Nadeau [87].

orates the quality of results. Then, the rule learner trains entity type classifiers by using only unambiguous entity mentions. The bottom part, the NER system, uses the lists and rules generated by three semi-supervised learners for identifying entity mentions and classifying their types.

The greatest advantage of a semi-supervised learning approach is the minimization of human involvement in the preparation of training data since manual

annotation by human experts is very expensive and often takes years of work. This feature also makes domain adaptation very easy. For example, the semi-supervised NER system explained in Figure 2.4 recognizes 100 entity types, whereas supervised NER systems usually identify less than 10 entity types. In addition, it is also known that learning many entity types simultaneously reduces the chance of misclassification of entity mentions [25, 135]. The most difficult problem in this approach is that it inherently suffers from semantic drift that introduces noise into extracted entity mentions and contextual patterns during the bootstrapping process. Performance of semi-supervised NER systems usually degrades if it iterates the bootstrapping process too many times.

Unsupervised Learning Approach

An unsupervised learning approach does not use any manually labeled data unlike previous two learning approaches. It typically utilizes a clustering method that gathers entity mentions sharing internal or external features. Each group of entity mentions does not have a specific type, but it can be inferred from lexical resources, such as WordNet, or unlabeled text, such as news articles.

Previous studies frequently utilize idiomatic expressions that appear with an entity mention and its type. Evans and Street [36] assume that a capitalized word sequence is an entity mention and classify it using the hyponym/hypernym identification method of Hearst [48]. For a capitalized word sequence X , for instance, they search for documents that have the expression “such as X ” and use the noun that precedes this phrase as the entity type of X . Etzioni et al. [35] also exploit a number of automatically generated expressions. For instance, they use phrases such as “ X is a city,” “ X and other towns,” “cities X ,” “cities such as X ,” and “cities including X ” to recognize city type entity mentions. On the other hand, entity types can be inferred by using lexical resources. For example, the hypernyms of an entity mention in WordNet [2] or the category name of an entity mention in Wikipedia [62] can be used as its entity type.

An unsupervised learning approach is a very challenging task in various aspects. However, it is able to discover novel information since it does not limit entity types from the beginning.

2.3 Features

A NER system exploits various information to identify entities mentioned in text. To recognize names of people, for instance, it may examine if a current word is capitalized, if it follows a title, such as Mr., Mrs., and Dr., and so on. Each of this information is called a *feature*. In this section, we introduce a variety of features that have been proposed and utilized in previous studies.

2.3.1 Local Features

Local features come from a word that is to be classified (hereafter, *the focus word*) or its neighboring words within a context window, which are usually two (or three) words to the left and right from the focus word. Most NER systems exploit various kinds of local features since they are very effective while easy to obtain.

Table 2.5 summarizes local features commonly used in NER systems. *Word* features are the most basic features that involve word unigrams and word bigrams. Lower case unigrams and bigrams are also frequently used for case-insensitive match. *Word-internal* features usually indicate specific properties of a word; for example, capitalized words (e.g., Jobs, Apple), upper case words (e.g., HP, IKEA, UPS), and mixed case words (e.g., eBay, McDonalds) will trigger this type of features. *Word-shape* features, which were first introduced by Collins [24], transform a token into a pattern that represents its shape. For instance, a word shape pattern may map all upper case letters into “A,” lower case characters into “a,” punctuations into “-,” and numbers into “0” as shown in the following examples.

- A company name: AT&T → AA-A

Feature Class	Examples
Word	Word unigram, word bigram, lower case word unigram & bigram
Word-internal	A word is capitalized, a word consists of only upper case letters, a word has mixed-case letters, a word has internal punctuation(s)
Word-shape	A word shape pattern, a summarized word shape pattern
Character	Character n-grams (e.g., bigram, trigram, four-gram)
Morphology	Stem, lemma, prefix, suffix
Part-of-Speech	Noun, verb, number, foreign word

Table 2.5: Exemplary local features.

- A telephone number: 03-3908-1111 \rightarrow 00-0000-0000

In case of a summarized word shape pattern, repeated characters will be condensed into a single character as shown below.

- AT&T \rightarrow A-A
- 03-3908-1111 \rightarrow 0-0-0

These features are especially useful when recognizing patternized entity mentions such as phone numbers and date expressions. *Character* features [97] involve character n-grams of a word. As shown in Figure 2.5, a special character can be attached to a word to distinguish n-grams at the begin and the end of the word from the others while generating character n-grams. In the *Morphology* feature class, suffix features exploit the information about common endings of words. For example, IT company names often end with “tech,” “ex,” and “soft” [7]. Prefix,

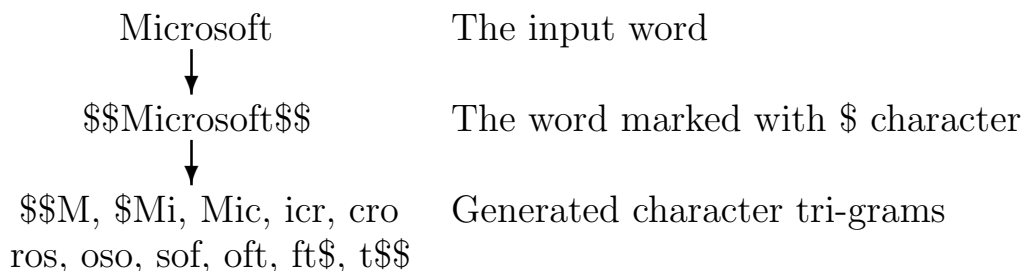


Figure 2.5: Generating character n-grams ($n = 3$) from the word *Microsoft*. \$ is a special character that denotes the begin and the end of a word.

stem, and lemma features are intended to utilize the information about the roots of words. While obtaining prefix (or stem) features is a relatively simple process, lemma features need a more sophisticated method since a lemma can be completely different from its inflected form (e.g., good vs. better) and it is also dependent on the context (meet as a verb or meeting as a noun vs. meeting). *Part-of-Speech* (POS) features represent syntactic roles of words. Considering that named entities are mostly noun phrases (or a part of noun phrases), these features can effectively narrow down the number of candidate entity mentions.

2.3.2 Global Features

As its name means, global features come from relatively further places than local features. In this section, we introduce three types of global features: sentence level, document level, and corpus level global features. Sentence level global features aim to utilize syntactically or semantically related words that appear at distant positions from the focus word. Document level features, on the other hand, mostly focus on the label consistency of multiple entity mentions of the same entity. Corpus level features usually exploit word and phrase frequency in a labeled (or unlabeled) corpus.

Table 2.6 summarizes some of these global features. Settles [112] and Finkel

Feature Class	Examples
Sentence-level global feature	The head word, the governor of the head word, dependency path of length n ($2 < n < 5$)
Document level global feature	All features of the same token within a document, predicted labels of the same tokens (or entity mentions) by a NER system
Corpus level global feature	Word and phrase frequency, co-occurrence, meta-information (e.g., tables, lists, etc.)

Table 2.6: Exemplary Global features.

et al. [38] utilized sentence-level global features for NER and Smith and Wilbur [116] used it for named entity classification. Especially, Smith and Wilbur [116] evaluated the effect of various parsers, constituency parsers and dependency parsers, in the biomedical domain.

Chieu and Ng [15] addressed the label inconsistency problem where multiple mentions of the same entity have different labels. They aggregate the features of the same token in a document and decide their label at the same time. This approach, however, can lead to excessive number of features. Krishnan and Manning [66] dealt with this issue by using the output (predicted labels) of the first NER system as additional features for the second NER system.

Statistics gathered from a large corpus such as word and phrase frequency is also often used as corpus level global features. Da Silva et al. [30] used corpus statistics to filter out recognized entity mentions that involve long lower case words.

2.3.3 Features from External Resources

External resources are very important assets for NER since they provide a large amount of entity mentions including those not appeared in training data. Table

Feature Class	Examples
Entity dictionary	Company name, person name, music title, country name, location names, gene names, disease names, chemical names
General dictionary	Common noun, stop word, common abbreviations
Entity cue dictionary	Common beginning and ending words of company name, person name, street name, and etc.

Table 2.7: Exemplary external resource features.

2.7 summarized some of these features.

A straightforward technique is to identify entity mentions in text by using a dictionary of target entity mentions and use the results as features. Features can be boolean values if a word appears as a part of an entity mention in a dictionary, or nominal values that combines the word and its entity type recognized by a dictionary (e.g., “Bill-Celebrity,” “Gates-Celebrity”, “Apple-Company”).

General dictionaries can be utilized to detect and remove general words that are identified as entity mentions. For instance, Mikheev [83] used a general dictionary to deal with entity mentions that appeared at ambiguous position (e.g., at the beginning of a sentence). In addition, NER often use stop word lists to remove frequently occurring noisy entity mentions.

We can also use entity cue dictionaries consisting of words that frequently appear as a part of entity mentions of specific entity types. For example, it will be much easier to recognize company names such as “General Electronics,” “American Airlines,” and “Micron Technologies” with a company name cue dictionary that includes “General,” “American,” and “Technologies” [44, 101]. Other related studies also utilize entity cue dictionaries of person names [102, 131] and place names [131] based on the ideas that organization names often follow their founders’ name

(e.g., Bill & Melinda Gates Foundation, Bell laboratories, Toyota) and the name of their birthplace (e.g., Massachusetts Institute of Technology, Hitachi).

In most cases, the use of external knowledge improves NER performance even with a simple dictionary look-up technique that searches for an dictionary entry that exactly matches a candidate entity mention. However, the coverage of dictionaries can be improved by allowing negligible noise between a dictionary entry and a candidate entity mention. For example, a candidate entity mention can be made of stems or lemmas instead of words [20]. In this case, the word “technologies” will successfully match to “technology.” Moreover, approximate string matching techniques based on edit-distance or character n-gram cosine similarity [23, 86] can further improve the coverage of dictionaries.

2.4 Evaluation

Objective assessment of a NER system is essential for the advancement of NER research. For this purpose, it is necessary to use an evaluation method that can assess various aspects of a NER system. In this section, we explain what kinds of errors that a NER system can make, how to measure performance based on precision/recall/F1-score, and why boundary errors are less critical than type errors in some applications.

Suppose that there is a hypothetical NER system that annotates an example sentence¹³ as shown in Figure 2.6. The system outputs are shown with the labels that begin with *System* and the human annotations are denoted by the labels with *Correct*. The system completely misses one entity mention and commits boundary and type errors on four entity mentions, whereas it correctly identifies the last entity mention. These errors exhibit five different types of errors¹⁴ as explained in Table 2.8.

¹³This example is excerpted from the thesis of Nadeau [87].

¹⁴The original source of this classification is from an informal article, <http://nlpers.blogspot.jp/2006/08/doing-named-entity-recognition-dont.html>.

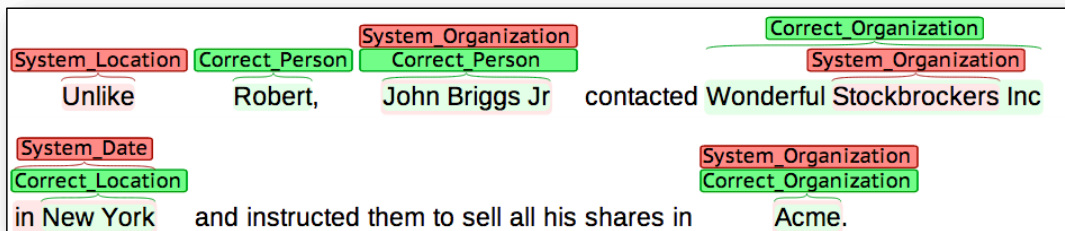


Figure 2.6: An example of NER errors.

An evaluation measure based on micro-averaged¹⁵ precision/recall/F1-score has been most widely used in NER. For the example in Figure 2.6, these scores will be calculated as follows:

$$Precision = \frac{\text{The number of correct system outputs}}{\text{The number of all system outputs}} = \frac{1}{5} = 0.2, \quad (2.1)$$

$$Recall = \frac{\text{The number of correct system outputs}}{\text{The number of all human annotations}} = \frac{1}{5} = 0.2, \quad (2.2)$$

$$F1 - score = \frac{2 \times Precision \times Recall}{Precision + Recall} = \frac{0.08}{0.4} = 0.2. \quad (2.3)$$

Although this evaluation measure can assess how precisely and completely a NER system can identify entity mentions, it does not distinguish different types of errors. It is an obvious weakness considering the fact that some types of errors can be less critical than the others depending on the purpose of a NER system. For example, the JNLPBA shared task [59] and the BioCreAtIvE gene mention recognition task [115] evaluate the performance of a NER system based on the relaxed matching criterion in addition to the exact matching criterion. More specifically, the former checks only the left boundary (and the right boundary) of recognized

¹⁵Micro-average method calculates the precision and recall of a system without distinguishing the types of outputs. On the other hand, macro-average method computes the precision and recall on the different types of output and takes the average of them.

System output	Correct output	The type of an error
⟨Location⟩ Unlike ⟨/Location⟩	Unlike	A complete error - False positive
Robert	⟨Person⟩ Robert ⟨/Person⟩	A complete error - False negative
⟨Organization⟩ John Briggs Jr ⟨/Organization⟩	⟨Person⟩ John Briggs Jr ⟨/Person⟩	A partial error - Entity type error
⟨Organization⟩ Stockbrokers ⟨/Organization⟩	⟨Organization⟩ Wonderful Stockbrokers Inc ⟨/Organization⟩	A partial error - Entity boundary error
⟨Date⟩ in New York ⟨/Date⟩	⟨Location⟩ New York ⟨/Location⟩	A partial error - Entity type and boundary error

Table 2.8: Types of errors in NER by Nadeau [87].

entity mentions. The latter allows an entity mention to have multiple boundaries if they are semantically equivalent. From this, we can infer that the relaxed measures in these competitions assume that entity boundary errors are less problematic than the other errors. Furthermore, MUC [16] and ACE [33] conferences use their own evaluation measures that take account of different types of errors in NER.

2.5 The Data-sparseness Problem and Feature Generalization

A supervised learning approach has been adopted in most recent state-of-the-art NER systems because of its exceptional performance compared to other methods. However, this approach often suffers from the data-sparseness problem that

results from the lack of sufficient training data and the use of combinatoric features that are very sparse in this training data. This section briefly introduces the motivation of our studies that addresses the data-sparseness problem in the view point of feature generalization. These studies will be presented in Chapter 3 and 4 respectively.

2.5.1 NER as the Mixture of the Segmentation and Classification Tasks

A supervised learning-based NER is mostly formulated as a sequence labeling task. However, it actually consists of two sub-tasks: the segmentation task and the classification task. The segmentation task is necessary since an entity mention comprises one or more words. On the other hand, the classification task is to identify the entity type of a recognized entity mention.

Previous studies deal with these sub-tasks in three ways. The first and the most popular approach is to integrate these two tasks by augmenting type labels with a set of segment labels. For example, instead of using the label *Person*, we use the *B-Person* and *I-Person* labels that denote not only the entity type of a word but also its position within an entity mention. This approach is very popular because of its simplicity and comparable performance to the other methods. However, it is difficult to evaluate the effect of different segment representations (SRs) independently. Furthermore, it makes a feature¹⁶ space very sparse; and, machine learning methods often do not generalize well in such a sparse feature space.

The second approach is to use a pipe-lined system [61, 73, 74]. This approach first identifies entity mentions without entity type information and then classifies the entity types of the recognized entity mentions. While it allows us to choose the best SR that maximizes the performance of the segmentation task [132], the errors at this step can propagate to the classification step and result in incorrect entity mentions.

¹⁶In the context of machine learning, a feature is the combination of a label and a textual cue, which is often called a feature in informal situation.

The last approach is to use a Semi-Markov model [109] that labels a sequence (segment) of words, not word by word. Compared to previous two approaches, a semi-Markov model is very powerful since it is able to exploit non-Markovian features within a segment. The biggest problem of this approach is the computational complexity of inference, which is proportional to the maximum length of a segment.

In Chapter 3, we propose a new feature generation method that incorporates multiple SRs into a traditional Markov model. The proposed method has three important advantages compared to previous approaches. First, it allows a model to better capture the characteristics of segments that cannot be represented by a single SR. Second, when the size of a training data is small, it prevents complex SRs from degrading performance by generalized SRs. Third, after training, the size (and also the tagging speed) of a model using the proposed method can be reduced so that it is equivalent to that of a conventional model.

2.5.2 Generalization of Combinatorial Syntactic Structure Features

In a supervised learning approach, a set of features plays a crucial role for obtaining high performance. It is well known that a NER model achieves relatively high performance even with only simple local features. Its performance can be further improved with the use of non-local features.

Unlike occurrence-based non-local features, such as context aggregation [15] and two-stage prediction aggregation [66], syntactic structure features [38, 116, 130] have not made much contribution to the improvement of NER performance. Based on our experiments and analyses, we assume that there are two important problems in the modelling of syntactic structure features. First, combinatorial syntactic features are very sparse. For example, a dependency path feature of length 3 is much more sparse than a word tri-gram feature, which is used in most NER systems, since it includes not only three words but also their relations. Previous studies tried to solve this problem by generalizing these features. For

instance, Finkel et al. [38] designed this type of features that consist of words at the end of dependency paths (the head and governor of a noun phrase) and Smith and Wilbur [116] used features that comprise words at the both end of dependency paths and the relations between them. Second, even we use generalized features as mentioned above, a manually labeled training data is too small to extract enough amount of syntactic structure features. These features occurring in a test data are mostly unseen features in a training data. Therefore, it is essential to utilize a large amount of unlabeled data to overcome this problem.

In Chapter 4, we propose to use a new type of resource, a context gazetteer, that is a list of contexts co-occurring with entity mentions and present a method to create it from an encyclopedic database. A context gazetteer consists of dependency paths of variable lengths to capture more syntactically meaningful contexts than traditional linear contexts. Moreover, each context is assigned with confidence value to reflect how they are likely to appear with entity mentions. A context gazetteer can provide rich and sophisticated context patterns because it is built from a huge amount of highly precise and automatically labeled data.

Chapter 3

Named Entity Recognition with Multiple Segment Representations

3.1 Introduction

Named Entity Recognition (NER) aims to identify meaningful segments in text and categorize them into pre-defined semantic classes such as people, locations, and organizations. This is an important task because its performance directly affects the quality of many succeeding natural language processing (NLP) applications such as information extraction, question answering, and machine translation. NER has been mostly formalized as a sequence labeling task that performs the recognition of segments and the classification of their semantic classes simultaneously by assigning a label to each token of an input text.

While many researchers have focused on developing features that capture textual cues of entity mentions, there are only a few studies [71, 100] that examined the effects of different segment representations (SRs) such as the *IOB2* and the *IOBES* notations. This issue has been extensively discussed for a different NLP task, word segmentation (WS). In this task, complex SRs consisting of four to six segment labels have been proposed based on linguistic intuitions [132] and statistical evidence from corpora [136] and shown to be more effective than the simple

BI SR¹. However, complex SRs are not always beneficial, especially when the size of training data is small, since they can result in undesirably sparse feature space. In NER, the data-sparseness problem is an important issue because only a small portion of training data are entity mentions. Therefore, the use of a complex SR, which may better explain the characteristics of target segments than a simple SR, may not be much effective or even can bring performance degradation.

In this chapter, we present a feature generation method that creates an expanded feature space with multiple SRs. The expanded feature space allows a model to exploit highly discriminative features of complex SRs while alleviating the data-sparseness problem by incorporating features of simple SRs. Furthermore, our method incorporates different SRs as feature functions of Conditional Random Fields (CRFs) so that we can use the well-established procedure for training. We also show that the size of a new model using the proposed method can be reduced as small as that of the conventional model using only the most complex SR after training process. It is very advantageous since the tagging speed of the new model is also equivalent to that of the conventional model. The proposed method is evaluated on the two NER tasks: the BioCreative 2 gene mention recognition task [115] and the CoNLL 2003 NER shared task [126]. The experimental results demonstrate that the proposed method contributes to the improvement of NER performance.

The next section investigates several SRs developed for various NLP tasks and explains a hierarchical relation among them that is the key concept to our proposed method. In Section 3.3, we shows the effect of different SRs on NER and analyze the results in two ways. This analysis motivates the necessity of using multiple SRs for NER. Section 3.4 describes the proposed feature generation method that creates an expanded feature space with multiple SRs. We also show how to speed up the tagging speed of a model using the proposed method. In Section 3.5, we present the experimental results and the detailed analysis. Finally, Section 3.6

¹The *BI* SR identifies characters at the **B**eginning and **I**nside of words.

Task	SR type	Segment Labels	Examples
NER	<i>IOB2</i>	<i>B, I, O</i>	B, BI, BII, ..., O
	<i>IOBES</i>	<i>S, B, I, E, O</i>	S, BE, BIE, BIIE, ..., O
SP	<i>IOB2</i>	<i>B, I, O</i>	B, BI, BII, ..., O
	<i>IOE2</i>	<i>I, E, O</i>	E, IE, IIE, ..., O
	<i>IOB1</i>	<i>B*, I, O</i>	I, II, ..., B*, B*I, B*II, ..., O
	<i>IOE1</i>	<i>I, E*, O</i>	I, II, ..., E*, IE*, IIE*, ..., O
	<i>IOBES</i>	<i>S, B, I, E, O</i>	S, BE, BIE, BIIE, ..., O
WS	<i>BI</i>	<i>B, I</i>	B, BI, BII, ...
	<i>BIS</i>	<i>S, B, I</i>	S, BI, BII, ...
	<i>BIES</i>	<i>S, B, I, E</i>	S, BE, BIE, BIIE, ...
	<i>BB₂IES</i>	<i>S, B, B₂, I, E</i>	S, BE, BB ₂ E, BB ₂ IE, ...
	<i>BB₂B₃IES</i>	<i>S, B, B₂, B₃, I, E</i>	S, BE, BB ₂ E, BB ₂ B ₃ E, BB ₂ B ₃ IE, ...

Table 3.1: Definition of SRs for NER, WS, and SP.

summarizes the contribution of our research and future work.

3.2 Segment Representations

SRs are necessary for sequence labeling tasks that involve segmentation as a sub-task. This section introduces SRs used in various NLP tasks and presents a hierarchical relation among these SRs that will become the basis of our proposed method.

3.2.1 Segment Representations in Various NLP tasks

Several SRs have been developed for and adopted to various NLP tasks such as NER [100], WS [132, 136], and shallow parsing (SP) [68, 107]. Table 3.1 presents the definition of these SRs. Each SR in the *SR type* column consists of segment labels in the *Segment Labels* column. The *Examples* column presents a few example

label sequences of entity mentions, chunks, and words with respect to the target tasks. We would like to note that the *O* label of the SRs in the NER and the SP tasks denotes a token that does not belong to any target segments. In WS, however, the *O* label is not necessary because every character of an input sentence is a part of a word.

In NER, the *IOB2* and the *IOBES* SRs have been used most frequently. The *IOB2* SR distinguishes tokens at the **B**eginning, the **I**nside, and the **O**utside of entity mentions. On the other hand, the *IOBES* SR identifies tokens at the **B**eginning, the **I**nside, and the **E**nd of multi-token entity mentions, tokens of **S**ingle token entity mentions, and tokens of the **O**utside of entity mentions. In SP, the *IOB2* and the *IOBES* SRs work in the same manner as in NER. The *IOE2* SR uses the *E* label to differentiate the end tokens of chunks instead of the *B* label of the *IOB2* SR. The *IOB1* and the *IOE1* SRs are basically equivalent to the *IO* SR that uses the *I* label to denote tokens of chunks and the *O* label to indicate tokens outside chunks. However, the *IO* SR can not distinguish the boundary of two consecutive chunks of a same type. To overcome this problem, the *IOB1* SR assigns *B** label to the token at the beginning of the second chunk, whereas the *IOE1* SR gives the *E** label to the token at the end of the first chunk. Lastly, in WS, the *BI* SR identifies the beginning and the inside of words, the *BIS* SR deals with single character words separately by assigning the *S* label to these words and the *BIES* SR uses the *E* label for the end characters of words. In addition, the *BB₂IES* assigns the *B₂* label to the second characters of words consisting of more than two characters, whereas the *BB₂B₃IES* gives the *B₂* and the *B₃* labels to the second and third characters of words comprised of more than three characters.

Table 3.2 shows a sample text annotated with the seven SRs which will be used in this work. In addition to the *IOB2* and the *IOBES* SRs that have been commonly used in NER, we also use the *IOE2* SR to investigate whether it is better to distinguish the beginning or the end of entity mentions. The *IO* SR is adopted as the simplest SR that actually does not perform any segmentation. Because

Text	IO	IOB2	IOE2	IOBES	BI	IE	BIES
Gamma	I-gene	B-gene	I-gene	B-gene	B-gene	I-gene	B-gene
glutamyl	I-gene	I-gene	I-gene	I-gene	I-gene	I-gene	I-gene
transpep- tidase	I-gene	I-gene	E-gene	E-gene	I-gene	E-gene	E-gene
(O	O	O	O	B-O	E-O	S-O
GGTP	I-gene	B-gene	E-gene	S-gene	B-gene	E-gene	S-gene
)	O	O	O	O	B-O	I-O	B-O
activity	O	O	O	O	I-O	I-O	I-O
in	O	O	O	O	I-O	I-O	I-O
the	O	O	O	O	I-O	I-O	I-O
...

Table 3.2: A sample text annotated with various SRs. (NEs are in bold face font.)

two entity mentions are not likely to appear consecutively, we can recognize entity mentions as a sequence of tokens that have a same label. The *BI*, the *IE*, and the *BIES* SRs, to the best of our knowledge, were proposed for WS and have not been used for NER. We applied these SR to NER by regarding the *O* label as a semantic class and augmenting it with the remaining segment labels. This application is based on the observation that tokens appearing around entity mentions are not random words. In this example, for instance, the left round bracket appears between the full name of a gene and its abbreviation and the right round bracket occurs after the abbreviated gene mention. Therefore, it is worth differentiating these tokens from the others by assigning separate labels.

3.2.2 Relation among Segment Representations

Conceptually, only two segment labels are necessary (e.g. *B-gene* and *I-gene* for gene mentions) to distinguish segment boundaries unambiguously. However,

many words tend to appear at specific positions, not random places. For example, the names of location often end with the words such as “Street,” “Road,” and “Avenue” and the names of companies are frequently followed by the phrases such as “Corporation” and “Co., Ltd.” Therefore, complex SRs that can capture these characteristics of target segments are able to create a more informative feature space than simple SRs. Xue [132] articulated that choosing a suitable SR is a task-specific problem that depends on the characteristics of segments and the size of available training data.

As a measure of analyzing the positional tendency of entity words, we used information entropy. Assuming that an entity word appears at one of four relative positions following the *IOBES* SR, its information entropy (hereafter, positional uncertainty) is calculated as follows:

$$H(w) = - \sum_{p \in \{B, I, E, S\}} \frac{C(w^p)}{C(w)} \log_2 \frac{C(w^p)}{C(w)} \quad (3.1)$$

where w is an entity word, w^p is the entity word at the position p , $C(w)$ is the frequency of the entity word at any positions in labeled data, and $C(ew^p)$ is the frequency of the entity word at the position p . The positional uncertainty of an entity word ranges from 0.0 to 2.0.

Figure 3.1 shows the positional uncertainty of entity words in the training data of the GENETAG corpus, which is used for the BioCreative 2 gene mention recognition task [115]. The data has only one entity type, gene. To estimate the positional uncertainty reliably, we used entity words that appear more than or equal to 5 times, which are 1,133 unique entity words in total. There are two noticeable points in this result. First, the peak value appears at the center of the graph and starts to drop as the positional uncertainty increases (or decreases). It indicates that the majority of entity words do not appear at random positions, but have mild tendency to positions. Second, there is the second peak value at the

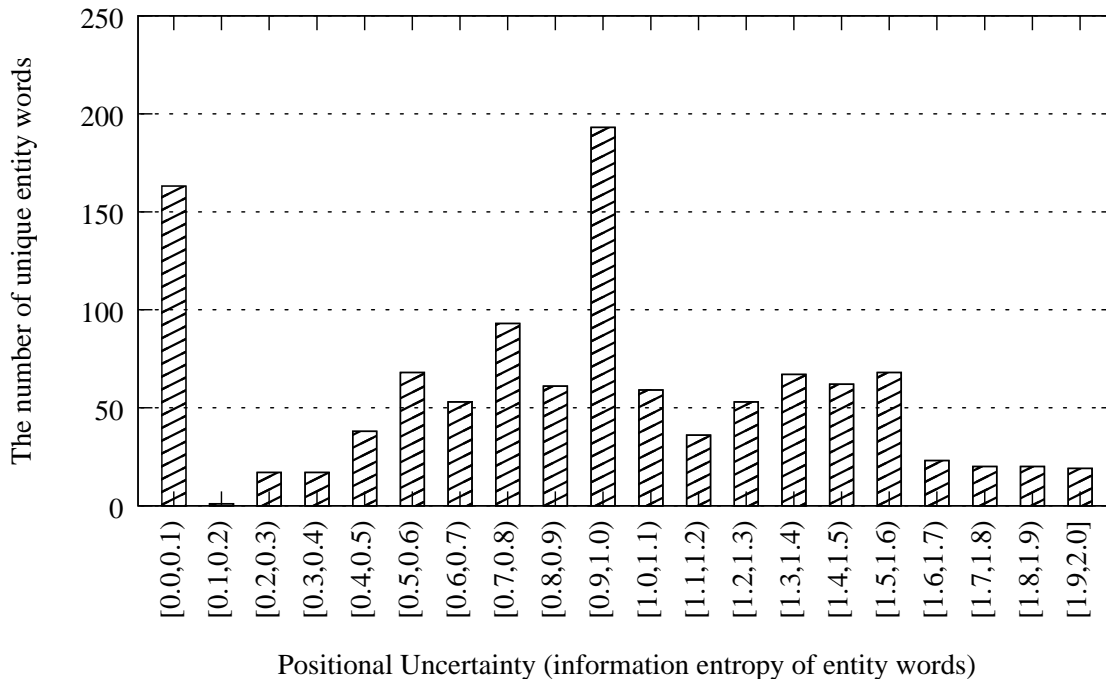


Figure 3.1: Positional uncertainty of entity words in the training data of the GENETAG corpus.

leftmost position. It means that about 170 entity words have very low positional uncertainty ranging from 0.0 to 0.1. Many of them are abbreviations and acronyms, which are mostly single token entity mentions. In addition, the words that indicate the semantic class of entity mentions such as “adrenoceptor,” “globulin,” and “aminotransferase” frequently appear at the end of entity mentions. For example, the entity word, globulin, appears 23 times in the training data. Almost all of them (22/23) appear at the end of entity mentions and the only one of them occurs as a single word entity mention. Table 3.3 shows some of actual examples in which italicized expressions are entity mentions and bold-faced expressions are the entity word, globulin. However, the number of entity words of high positional uncertainty is not negligible. These include “actin,” “Jun,” “collagen,” “erbA,” and “telomerase.” For instance, actin appears 52 times in the training data and occurs

	“... were given <i>varicella-zoster immune globulin</i> , or ...”
E (22/23)	“... between <i>sex-hormone-binding globulin</i> capacity ...”
	“... spinal fluid <i>gamma globulin</i> elevations, and ...”
S (1/23)	“Serum levels of albumin, <i>globulin</i> , and coagulation ...”

Table 3.3: Samples of entity words having low positional uncertainty.

B (10/52)	“... of the second <i>actin-binding domain</i> that can ...”
I (10/52)	“..., including the <i>yeast actin-associated protein Abp1p</i> .”
E (12/52)	“... and <i>beta-actin</i> (-3400 to +912) promoters but ...”
S (20/52)	“... the organization of the <i>actin</i> /myosin cytoskeleton, ...”

Table 3.4: Samples of entity words having high positional uncertainty.

at various positions within entity mentions. Table 3.4 shows examples in which the entity word, *actin*, appears. Considering the result of these analyses, using only one SR does not seem to fully utilize the positional characteristics of entity words.

Segment labels of a complex SR often denote more specific positions than those of a simple SR. Although every pair of any SRs can be inter-convertible if enough context information (segment labels of neighboring tokens) is provided, some of them are *deterministically* mappable by looking at only current labels. For example, to convert the *IOBES* SR to the *IOB2* SR, we can simply map the *B* and the *S* labels of the *IOBES* SR to the *B* label of the *IOB2* SR, the *I* and the *E* labels to the *I* label. Figure 3.2 shows the hierarchical relation among the seven SRs used in the previous example in Table 3.2. In this figure, a complex SR can be deterministically mapped to a simple SR if they are connected by directed arrow(s). Table 3.5 shows how to map the segment labels of the *BIES* SR to those of simpler six SRs.

The existing sequence labeling framework using the Viterbi algorithm assumes

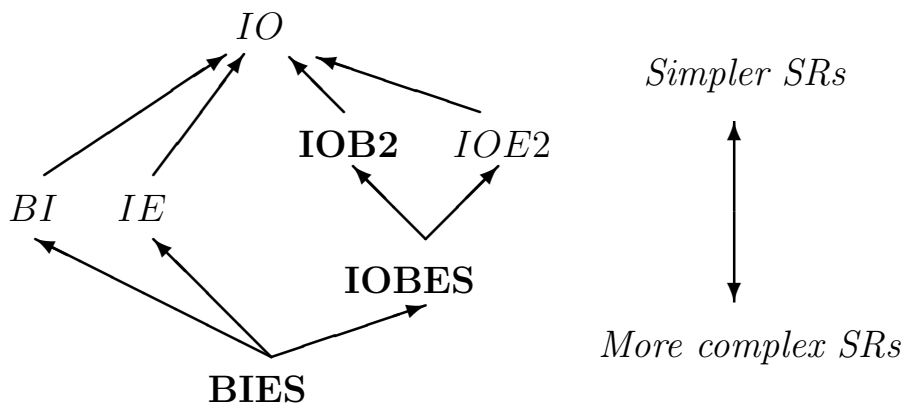


Figure 3.2: The hierarchical relation among the seven SRs.

	Segment				Non-segment			
BIES	S	B	I	E	S	B	I	E
↓								
BI	B	B	I	I	B	B	I	I
IE	E	I	I	E	E	I	I	E
IOBES	S	B	I	E	O	O	O	O
IOB2	B	B	I	I	O	O	O	O
IOE2	E	I	I	E	O	O	O	O
IO	I	I	I	I	O	O	O	O

Table 3.5: Mapping segment labels of the *BIES* SR to those of the simpler six SRs. *Non-segment* is a sequence of tokens tagged with the *O* label.

the Markov property for computational tractability. Therefore, it is impossible to use arbitrary context information for mapping segment labels of one SR to those of another SR. However, we can avoid this problem by considering only a subset of SRs that can be deterministically mapped from one SR to another SR as shown in Figure 3.2. For example, when we use the *IOBES* SR, we can utilize the features created from not only this SR but also the other SRs which can be deterministically mapped from it (e.g. *IOB2*, *IOE2*, and *IO*).

3.3 The Effects of Different Segment Representations on NER

To investigate the effects of different SRs on NER, we performed a preliminary experiment on the BioCreative 2 gene mention recognition (BC2GMR) task [115]. For the experiment, we trained seven models with seven different SRs (*IO*, *IOB2*, *IOE2*, *BI*, *IE*, *IOBES*, and *BIES*), but with the same textual cues². Among these SRs, the *BI*, the *IE*, and the *BIES* SRs were originally designed for the WS task and do not use the *O* label. We assumed a sequence of continuous *O* labeled tokens as a kind of special entity mentions, namely *O*-class entity mentions, and gave them separate *O* labels to apply these SRs to the NER tasks. For example, the *BI* SR uses the *B-O* and *I-O* labels instead of the *O* label.

For machine learning, we implemented a linear-chain CRFs with the L-BFGS algorithm³. Lafferty et al. [70] defines a linear chain CRFs as a distribution:

$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \exp \sum_{t=1}^T \sum_{k=1}^K \lambda_k f_k(y_{t-1}, y_t, \mathbf{x}) \quad (3.2)$$

where $\mathbf{x} = \langle x_1, x_2, \dots, x_T \rangle$ is an input token sequence, $\mathbf{y} = \langle y_1, y_2, \dots, y_T \rangle$ is an output label sequence for \mathbf{x} , $Z(\mathbf{x})$ is a normalization factor over all label sequences, T is the length of the input and output sequences, K is the number of features, f_k is a feature, and λ_k is a feature weight for the f_k .

In a linear-chain CRFs, f_k is either a transition feature or a state feature. For example, a transition feature⁴ f_i , which represents the transition from the *B-gene*

²These textual cues are often called features. However, we use the term *feature* to indicate the combination between a textual cue and a label.

³<http://www.chokkan.org/software/liblbfgs/>

⁴A transition feature is a combination of previous and current labels. An input token sequence is not used for transition features in the current implementation.

label to the *E-gene* label of the *IOBES* SR, can be defined as

$$f_i(y_{t-1}, y_t, \mathbf{x}) = \begin{cases} 1 & ((y_{t-1} = \mathbf{B-gene}) \wedge (y_t = \mathbf{E-gene})) \\ 0 & (otherwise) \end{cases} \quad (3.3)$$

and a state feature⁵ f_j , which indicates that the current state is *E-gene* and its corresponding input token is “protein,” can be defined as

$$f_j(y_{t-1}, y_t, \mathbf{x}) = \begin{cases} 1 & ((y_t = \mathbf{E-gene}) \wedge x_t = (\mathbf{“protein”})) \\ 0 & (otherwise). \end{cases} \quad (3.4)$$

Training a linear chain CRFs model is equivalent to find a set of feature weights which maximize a model log-likelihood for a given training data. However, it is often necessary to use *regularization* to avoid overfitting. We use the following model log-likelihood formula [120]. The last term is for regularization.

$$l(\theta) = \sum_{i=1}^N \sum_{t=1}^T \sum_{k=1}^K \lambda_k f_k(y_{t-1}^{(i)}, y_t^{(i)}, \mathbf{x}^{(i)}) - \sum_{i=1}^N \log Z(\mathbf{x}^{(i)}) - C \sum_{k=1}^K \lambda_k^2 \quad (3.5)$$

The parameter C determines the strength of regularization and it can be chosen by using development data. A smaller C value will result in a model that fits training data better than a bigger C value, while it is more likely to be overfitting. In the preliminary experiment, we reserved the last 10% of the original training data as the development data for tuning the C value. We examined ten C values⁶ for each model and used the best performing C value for evaluation on the test data.

⁵A state feature is a combination of a current label and a textual cue created from a sequence of input tokens within a context window.

⁶These C values are 2^{-5} , 2^{-4} , 2^{-3} , 2^{-2} , 2^{-1} , 2^0 , 2^1 , 2^2 , 2^3 , and 2^4 .

Model	#labels	Precision	Recall	F1-score
IO	2	77.67 (88.13)	70.10 (81.39)	73.69 (84.63)
IOB2	3	78.60 (88.73)	72.12 (83.07)	75.22 (85.81)
IOE2	3	78.64 (88.79)	72.56 (83.48)	75.48 (86.05)
BI	4	79.31 (89.64)	72.04 (83.10)	75.50 (86.25)
IE	4	79.15 (89.12)	71.54 (82.15)	75.15 (85.49)
IOBES	5	79.59 (89.83)	72.58 (83.53)	75.93 (86.56)
BIES	8	80.70 (90.58)	72.58 (83.26)	76.42 (86.77)

Table 3.6: The performance of the seven models on the BC2GMR task.

We used features generated from input tokens, lemmas, POS-tags, chunk-tags, and gazetteer matching results. The detailed explanation of the feature set is in Section 3.5.

3.3.1 Evaluation based on Standard Performance Measures

The seven models are evaluated in standard performance measures: precision, recall, and F1-score. As shown in Table 3.6, precision tends to improve as the number of labels increases. On the other hand, recall does not exhibit such a clear tendency where the *IOE2* and *IOBES* models achieve the higher recall than other models. If we follow the conventional approach, the *BIES* SR, which has not been used for NER, will be most suitable for this corpus.

3.3.2 Evaluation based on the Difference of Tagging Results

Although the evaluation in standard performance measures demonstrated that the *BIES* SR is most suitable for this corpus, we found that the tagging results of these seven models are quite varied. Table 3.7 shows how the tagging results change when the SR alters from the simplest one (*IO*) to the most complex one (*BIES*) in terms of true positive (TP), false negative (FN), true negative (TN),

from IO	# of instances	to BIES	# of instances
TP	4438	TP	4139
		FN	299
FN	1893	TP	456
		FN	1437
TN	-	TN	-
		FP	397
FP	1276	TN	574
		FP	702

Table 3.7: The comparison of tagging results between the *IO* and *BIES* models.

and false positive (FP). Since the *BIES* model clearly outperforms the *IO* model, we anticipate that the *BIES* model will produce more correct tagging results. The *BIES* model actually corrects 456 false negatives and 574 false positives of the *IO* model. However, surprisingly, it introduces new 299 false negatives and 397 false positives which are non-negligible amount of errors.

This analysis suggests that different SRs produce feature spaces which can be complementary to each other; and, incorporating multiple SRs into a model is highly likely to improve its recognition performance. In the following section, we explain how to integrate multiple SRs into a CRF-based NER model.

3.4 The Proposed Method

In this section, we present a feature generation method which incorporates multiple SRs into a single CRF-based NER model. An expanded feature space created with the proposed method allows a model to exploit both high discriminative power of complex SRs and robustness of simple SRs against the data sparseness problem.

In Section 3.4.1, we explain the mapping relation of the SRs, and design four groups of SRs for the proposed method. Section 3.4.2 describes a modified lin-

Group	Main SR	Additional SR
<i>IOB2+</i>	<i>IOB2</i>	<i>IO</i>
<i>IOBES+</i>	<i>IOBES</i>	<i>IOB2, IOE2, IO</i>
<i>BIES+</i>	<i>BIES</i>	<i>BI, EI, IOBES, IOB2, IOE2, IO</i>
<i>BIES&IO</i>	<i>BIES</i>	<i>IO</i>

Table 3.8: Main and additional SRs used for four groups.

ear chain CRFs model which can automatically generate and evaluate features of multiple SRs. In Section 3.4.3, we show that a simple model computation after training makes the tagging speed of a proposed model using multiple SRs as fast as the conventional model using the most fine-grained SR of the proposed model.

3.4.1 The Mapping Relation of Segment Representations

In Section 3.2.2, we presented a hierarchical relation among seven SRs that can be deterministically mappable and explained how to exploit multiple SRs without violating the Markov property. We call the most complex SR among all SRs used for a model as a *main SR*, and the other SRs as *additional SRs*. A conventional NER model can be interpreted as a model using only a main SR. For the experiment, we selected two most popular SRs, *IOB2* and *IOBES*, and the most complex one, *BIES*, as the main SRs. As additional SRs, we basically use all deterministically mappable SRs to show the maximum effect of the proposed method. Three groups of SRs are shown in Table 3.8 and their names are marked with ‘+’ symbol. In addition, we trained a model using only the *BIES* and the *IO* SRs, which are the most complex and the simplest SRs. This will minimize the increase of the total number of features, while making the model exploit complementary feature information of SRs in very different types of SRs.

3.4.2 A Modified Linear Chain CRFs Model for Multiple Segment Representations

In Section 3.3, we briefly introduced a linear chain CRFs. To enable a model to use features generated from multiple SRs, we define two new terminologies: Γ as a set of SRs and F^γ as a set of features generated with the SR γ . Then, we modify the original probability distribution as

$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \exp \sum_{t=1}^T \sum_{\gamma \in \Gamma} \sum_{k=1}^{|F^\gamma|} \lambda_k^\gamma f_k^\gamma(y_{t-1}, y_t, \mathbf{x}) \quad (3.6)$$

where f_k^γ is the k -th feature generated with the γ SR and λ_k^γ is the feature weight for this feature. This modified CRFs model can exploit features generated from multiple SRs.

However, we need to remind that a label sequence \mathbf{y} belongs to the main SR. Therefore, it cannot directly evaluate the features of additional SRs. For example, a model, which uses the *IOBES* as its main SR and the *IOB2* as its additional SR, may have a transition feature $f_i^{IOB2} \in F^{IOB2}$ as below. (To avoid confusions, we explicitly use the name of the SR as superscript to which a label belongs.)

$$f_i^{IOB2}(y_{t-1}^{IOBES}, y_t^{IOBES}, \mathbf{x}) = \begin{cases} 1 & ((y_{t-1}^{IOBES} = \mathbf{B-gene}^{IOB2}) \\ & \wedge (y_t^{IOBES} = \mathbf{I-gene}^{IOB2})) \\ 0 & (otherwise) \end{cases} \quad (3.7)$$

This feature cannot be directly evaluated because the input argument labels (y_{t-1} and y_t) are of the main SR (*IOBES*) while the feature is of an additional SR (*IOB2*).

To solve this problem, we define a label conversion function, $g^\gamma(y)$ which converts a label of the main SR to the corresponding label of an additional SR γ .

Then the transition feature above can be re-defined as

$$f_i^{IOB2}(g^{IOB2}(y_{t-1}^{IOBES}), g^{IOB2}(y_t^{IOBES}), \mathbf{x}) = \begin{cases} 1 & ((y_{t-1}^{IOB2} = \mathbf{B-gene}^{IOB2}) \\ & \wedge (y_t^{IOB2} = \mathbf{I-gene}^{IOB2})) \\ 0 & (\textit{otherwise}). \end{cases} \quad (3.8)$$

The same modification applies to state features. For example, a state feature $f_j^{IOB2} \in F^{IOB2}$ can be re-defined as

$$f_j^{IOB2}(g^{IOB2}(y_{t-1}^{IOBES}), g^{IOB2}(y_t^{IOBES}), \mathbf{x}) = \begin{cases} 1 & ((x_t = \mathbf{“protein”}) \\ & \wedge (y_t^{IOB2} = \mathbf{I-gene}^{IOB2})) \\ 0 & (\textit{otherwise}). \end{cases} \quad (3.9)$$

For $g^\gamma(y)$, we use a *deterministic* conversion function that works as explained in Section 3.4.1. This mapping function allows us to use well-established algorithms for training a model.

3.4.3 Boosting up Tagging Speed

A models using the proposed method generates more features and it inevitably slows down training speed. However, we can speed up the tagging speed of this model as fast as the model using only the main SR. The proposed method uses a deterministic label mapping function. It means that we know what kinds of features of additional SRs are going to be triggered for every feature of the main SR. The model can work as if it uses only the main SR by calculating the sum of feature weights that always appear together in advance and using it as the new weight of a main SR feature.

Equation 3.10 shows how to calculate the sum of feature weights for the main

SR, BIES, and the additional SRs, IOBES, BI, IE, IOB2, IOE2, and IO SRs.

$$\bar{w}(f_i^{\text{BIES}}) = w(f_i^{\text{BIES}}) + w(f_j^{\text{IOBES}}) + w(f_k^{\text{BI}}) + \dots + w(f_o^{\text{IO}}) \quad (3.10)$$

where j, k, \dots, o are the feature indices of the additional SRs that correspond to the feature index i of the main SR. The size and tagging speed of the resulting model is identical to the model actually trained with the main SR only.

3.5 Experiments

The proposed method is evaluated on two NER tasks in different domains: the BioCreative 2 gene mention recognition (BC2GMR) task [115] and the CoNLL 2003 NER shared task [126].

We added a necessary functionality⁷ into our implementation of a linear-chain CRFs so that it produces features with a given set of SRs as shown in Table 3.8. For machine learning, the L-BFGS algorithm is chosen. The training process terminates if the variance of the model likelihood of the latest twenty models is smaller than 0.0001 or if it reaches the maximum number of iterations, 2,000.

3.5.1 NER in the Biomedical Domain

The GENETAG corpus used in the BC2GMR task consists of single entity type, *Gene*. For one entity type, however, it provides two types of annotations: one that has main gene mentions and the other one has the alternative gene mentions. Figure 3.3 shows an example of these gene and alternative gene annotations. Each (main) gene mention in the main annotation may have alternative gene mentions that are semantically equivalent but have different textual spans. The official

⁷While this functionality is not difficult to implement, we found that incorporating it into a publicly available CRF toolkit, CRFSuite [95], is not a simple task because of its optimized code for speed.

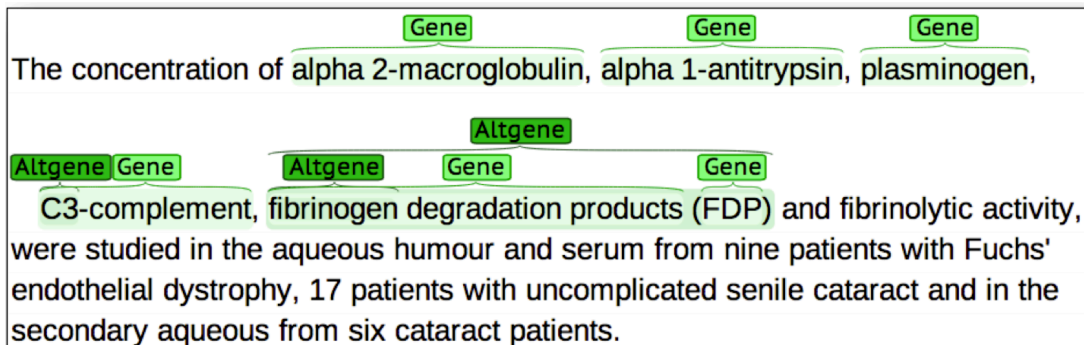


Figure 3.3: Gene and alternative gene annotations in the BC2GMR training data.

evaluation scheme considers a recognized gene mention as true positive if it has the same entity type and textual span to one of the gene (or alternative gene) mentions in the annotation. Therefore, one can say that the official evaluation using the both annotations is based on a relaxed-match criterion. The reason of this evaluation scheme is that the main goal of this task is to assist human database curators so that finding exact entity mention boundaries is not crucial as long as they are semantically correct. In general, however, the detection of correct entity boundaries is an important sub-task of NER and the relaxed-match results can underestimate it. Therefore, we use the strict-match results for comparing the models while providing both the strict-match and relaxed-match (official) results.

To prepare the experiment, we performed the following pre-processing. First, the corpus is tokenized based on the same tokenization method in the previous work [71]. Although this tokenization method produces more tokens than the Penn Treebank tokenization⁸, the output is very consistent: that is, no entity mentions begin or end in the middle of a token. Second, the tokenized texts are fed into the GENIA tagger [128] to obtain lemmatization, POS-tagging, and shallow parsing information. Lastly, we applied two gazetteers compiled from the EntrezGene [79] and the Meta-thesaurus of the Unified Medical Language Systems (UMLS) [9].

⁸<http://www.cis.upenn.edu/~treebank/tokenization.html>

Class	Description
Token	$\{w_{t-2}, \dots, w_{t+2}\} \wedge y_t, \{w_{t-2,t-1}, \dots, w_{t+1,t+2}\} \wedge y_t,$ $\{\bar{w}_{t-2}, \dots, \bar{w}_{t+2}\} \wedge y_t, \{\bar{w}_{t-2,t-1}, \dots, \bar{w}_{t+1,t+2}\} \wedge y_t,$
Lemma	$\{l_{t-2}, \dots, l_{t+2}\} \wedge y_t, \{l_{t-2,t-1}, \dots, l_{t+1,t+2}\} \wedge y_t,$ $\{\bar{l}_{t-2}, \dots, \bar{l}_{t+2}\} \wedge y_t, \{\bar{l}_{t-2,t-1}, \dots, \bar{l}_{t+1,t+2}\} \wedge y_t$
POS	$\{p_{t-2}, \dots, p_{t+2}\} \wedge y_t, \{p_{t-2,t-1}, \dots, p_{t+1,t+2}\} \wedge y_t,$
Lemma & POS	$\{l_{t-2}p_{t-2}, \dots, l_{t+2}p_{t+2}\} \wedge y_t,$ $\{l_{t-2,t-1}p_{t-2,t-1}, \dots, l_{t+1,t+2}p_{t+1,t+2}\} \wedge y_t$
Chunk	$\{c_t, w_{t_last}, \bar{w}_{t_last}, the_{lhs}\} \wedge y_t$
Character	Character 2,3,4-grams of w_t
Orthography	All capitalized, all numbers, contain Greek letters, ... (Detailed explanation of the orthographical features can be found in the related work [73])
Gazetteer	$\{g_{t-2}, \dots, g_{t+2}\} \wedge y_t, \{g_{t-2,t-1}, \dots, g_{t+1,t+2}\} \wedge y_t,$ $\{g_{t-2}l_{t-2}, \dots, g_{t+2}l_{t+2}\} \wedge y_t,$ $\{g_{t-2,t-1}l_{t-2,t-1}, \dots, g_{t+1,t+2}l_{t+1,t+2}\} \wedge y_t$

Table 3.9: Features for the biomedical NER.

Features are extracted from tokens, lemmas, POS-tags, chunk-tags, and gazetteer matching results. The feature set for our biomedical NER system is listed in Table 3.9 and the symbols used for the features are explained in Table 3.10. Most of these features are common for biomedical NER tasks [71, 73, 88], while chunk features and several orthographic features are newly added. The L2-regularization parameter (C) is optimized by using the first 90% of the original training data as the training data and the rest 10% as the development data. Ten C values⁹ are tested on the development data and the best-performing one is chosen for each model.

Table 3.11 summarizes the experimental results of seven models using a single

⁹These C values are $2^{-5}, 2^{-4}, 2^{-3}, 2^{-2}, 2^{-1}, 2^0, 2^1, 2^2, 2^3$ and 2^4 .

Symbol	Description
w_t	a t -th word
\bar{w}_t	a normalized t -th word. If w_t contains numbers, continuous numeric parts are conflated into a single zero (e.g. “p53” to “p0”). If w_t is a non-alphanumeric character, it becomes an under-bar symbol (e.g. “_” to “_”).
l_t	a t -th lemma
\bar{l}_t	a normalized t -th lemma
p_t	a t -th POS-tag
c_t	the chunk type of w_t
w_{t_last}	the last word of a current chunk
\bar{w}_{t_last}	the normalized last word of a current chunk
the_{ths}	if ‘the’ exists from the beginning of a current chunk to w_{t-1}
g_t	Gazetteer label for the t -th word

Table 3.10: Explanation of symbols used for features (see Table 3.9).

SR (the conventional models) and four models using multiple SRs (the proposed models) based on the strict-match and the relaxed-match (in a pair of parentheses). Conventional models tend to improve precision as the granularity of SR increases compared to the baseline model¹⁰ (BM). The best baseline model (best BM) records the highest precision that is notably higher than that of the BM. However, recall does not exhibit such an obvious tendency. For example, the recall of the best BM is almost identical to that of the *IOE2* and the *IOBES* models.

Proposed models improve both precision and recall as the granularity of main SR increases. In addition, every proposed model outperforms the conventional models that employ one of the SRs used by the proposed model. The best proposed model (best PM) achieves higher recall (1.22%) and comparable precision (-0.09%) to the best BM. The improvement of recall is an important merit of the

¹⁰The baseline model uses the most popular SR, *IOB2*.

Model	Precision	Recall	F1-score	AFI	#feat
IO	77.67 (88.13)	70.10 (81.39)	73.69 (84.63)	17.00	4.2
IOB2 (BM)	78.60 (88.73)	72.12 (83.07)	75.22 (85.81)	16.38	6.4
IOE2	78.64 (88.79)	72.56 (83.48)	75.48 (86.05)	16.29	6.4
BI	79.31 (89.64)	72.04 (83.10)	75.50 (86.25)	15.06	8.5
IE	79.15 (89.12)	71.54 (82.15)	75.15 (85.49)	15.02	8.5
IOBES	79.59 (89.83)	72.58 (83.53)	75.93 (86.56)	15.68	10.6
BIES (best BM)	80.70 (90.58)	72.58 (83.26)	76.42 (86.77)	13.44	16.9
IOB2+	78.56 (88.51)	72.39 (83.21)	75.35 (85.78)	16.69	10.9
IOBES+	79.93 (89.88)	72.86 (83.65)	76.24 (86.66)	16.33	27.5
BIES+ (best PM)	80.61 (90.18)	73.80 (84.17)	77.05 (87.08)	15.60	61.4
BIES&IO	80.40 (90.00)	73.54 (84.00)	76.82 (86.90)	15.01	21.2

Table 3.11: The performance on the BC2GMR task. AFI stands for the average number of feature instances per feature in the training data. #feat means the number of unique features (million).

proposed method because NER models frequently suffer from low recall due to an asymmetric label distribution where the *O* labels dominate the other labels [54] in training data. Considering that the only difference of the proposed models from the conventional ones is a set of SRs for feature generation, we can conclude that the proposed method effectively remedies the data sparseness problem of using fine-grained SR while takes advantage of its high discriminative power. This conclusion is also supported by the relation between the average number of feature instances per feature (AFI) and the number of features (#feat). For example, the best PM has about 20% higher AFI (15.60) than the best BM (13.44), whereas it has almost four times more features than the best BM.

To verify whether these improvements are meaningful, we performed the sta-

	IOB2+	IOBES+	BIES+	BIES&IO
IO	0.0000	0.0000	0.0000	0.0000
IOB2	0.2174	0.0001	0.0000	-
IOE2	-	0.0075	0.0000	-
BI	-	-	0.0000	-
IE	-	-	0.0000	-
IOBES	-	0.0970	0.0000	-
BIES	-	-	0.0039	0.0219

Table 3.12: The estimated p values between the proposed models and the conventional models. p values lower than 0.05 are in boldface.

tistical significance test using the bootstrap re-sampling method [115], which is commonly used for NER. Table 3.12 presents the estimated p values for the proposed models (the top row) against the conventional models (the leftmost column). In most cases, the proposed models have the p values lower than 0.05. Comparing a proposed model and its counterpart model, which uses the main (most fine-grained) SR of the proposed model, the p value decreases as the proposed model integrates more SRs of different granularity. As a result, the *BIES+* model has the p value lower than 0.05 whereas the *IOB2+* and the *IOBES+* do not. Interestingly, the *BIES&IO* model also rejects the null hypothesis against the best BM given the threshold p value 0.05. Considering that both the *BIES&IO* and the *IOB2+* models use only two SRs, integrating SRs of very different granularity is more effective than that of similar granularity.

We also show how the tagging results change when the proposed method is applied. For the sake of analysis, we use two conventional models, *BIES* and *IO*, and the proposed model, *BIES&IO*, that utilizes the SRs of the *IO* and *BIES* models. In Table 3.13, the tagging results of the two conventional models are divided into two groups depending on whether they make the same predictions or not. Then, we investigated what kinds of predictions the *BIES&IO* model makes.

1. Agreed		
BIES vs. IO	BIES&IO	
TP vs. TP (4139)	TP:99.42% (4115)	FN:0.58% (24)
TN vs. TN (-)	TN:-% (-)	FP: -% (65)
FP vs. FP (702)	FP:96.58% (678)	TN:3.42% (24)
FN vs. FN (1437)	FN:95.96% (1379)	TP:4.04% (58)

2. Disagreed		
BIES vs. IO	BIES&IO	
TP vs. FN (456)	TP:91.23% (416)	FN:8.77% (40)
TN vs. FP (574)	TN:88.50% (508)	FP:11.50% (66)
FP vs. TN (397)	FP:82.12% (326)	TN:17.88% (71)
FN vs. TP (299)	FN:77.59% (232)	TP:22.41% (67)

Table 3.13: The tagging results of two conventional models (*BIES* and *IO*) and a proposed model (*BIES&IO*). The number of entity mentions is shown in parenthesis.

The upper table titled with “Agreed” shows the tagging results of the *BIES&IO* model when the *IO* and *BIES* models make the same predictions. In most cases, the *BIES&IO* model makes the same predictions with the conventional models ($\geq 96\%$). In the lower table titled with “Disagreed”, the two conventional models make different predictions and only one of them is correct. We can see that the tagging results of the *BIES&IO* model tend to follow the results of the *BIES* model (from about 78% to 91%). However, the *BIES&IO* model makes less predictions same to the *BIES* model when it makes wrong predictions (from about 90% to 80%), even though the *BIES* model clearly outperforms the *IO* model by 2.73 points in F1-score.

We present several gene mentions that are correctly recognized obviously by the help of the proposed method. For example, *BIES&IO* model correctly recog-

nized a gene mention *mouse and human HPRT genes*, whereas the *BIES* model recognized only a part of it, *human HPRT genes*. Both words, *mouse* and *human*, mostly appear at the beginning of a gene mention (94 vs. 25 times in the training data), whereas rarely in the middle of a gene mention (7 vs. 3 times). The *BIES* model is likely to give the *B* label to *human* because it occurs almost four times more than *mouse* in the training data. On the other hand, the *IO* model, which correctly recognized this gene mention, does not experience this problem because it can give the same *I* label to these words. We think that the *BIES&IO* model successfully recognized this gene mention because it could exploit the features generated with the *IO* SR. There are similar cases where the *BIES&IO* and *IO* models correctly recognized gene mentions such as *serum insulin* and *type I and II collagen*, while the *BIES* model recognized only the last word, *insulin* and *collagen*. These last words often appear as gene mentions by themselves (33 among 44 times for *insulin* and 8 among 16 times for *collagen*). Therefore, the *BIES* model is likely to give the *S* label for these words.

However, incorporating the features of the *IO* model can cause difficulties in finding correct entity boundaries. For example, the *BIES* model correctly recognized gene mentions such as *Oshox1*, *phP1* and *Pms-*, whereas the *BIES&IO* and *IO* models recognized incorrect textual spans as *upstream Oshox1 binding sites*, *phP1 mutation* and *Pms*.

Next, we examined the effect of the proposed method based on the size of available training data. Models are trained on the first 10%, 20%, 40%, and 100% of the original training data that is 15,000 sentences in total. Regularization parameters are tuned by using the last 10% of the original training data as the development data. For the models using 100% of the original training data, they are first trained on the first 90% portion for parameter tuning and the final models are trained on the full training data.

Figure 3.4 shows the precision of the three proposed models (*IOB2+*, *IOBES+*, and *BIES+*) and their counterpart model (*IOB2*, *IOBES*, and *BIES*). The pre-

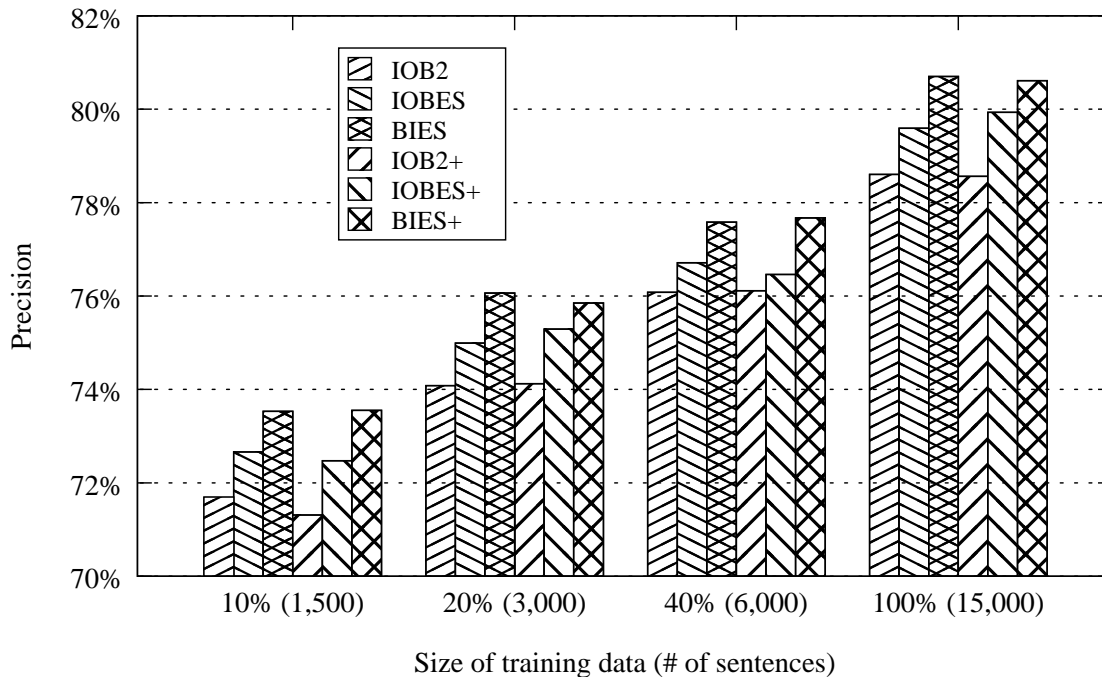


Figure 3.4: The effect of the proposed method on precision based on the training data size.

cision of a proposed model is almost identical to that of its counterpart model at each point. In addition, the models using more fine-grained SRs achieve higher precision than the models using coarse-grained ones regardless of application of the proposed method. This result shows that precision is mostly determined by the granularity of the most fine-grained SR employed by a model.

However, fine-grained SRs can cause negative impact on recall. In Figure 3.5, for instance, the *BIES* model achieves the lowest recall when the size of training data is 10% and 20% of the original training data. The low recall of the *BIES* model at the beginning is due to the insufficient training data considering that it achieves similar or higher recall than the other two conventional models as the size of training data reaches 40%. On the contrary, the proposed model, *BIES+*, achieves comparable recall to the best performing model, *IOBES+*, from the beginning. This result indicates that the proposed method can alleviate

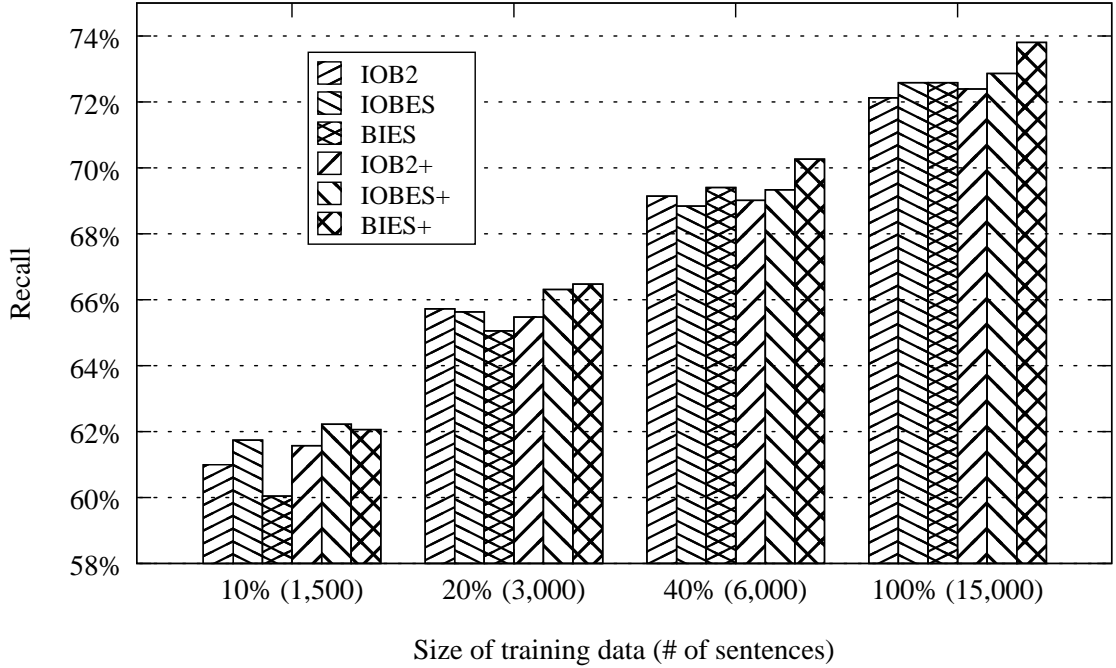


Figure 3.5: The effect of the proposed method on recall based on the training data size.

the performance degradation that results from the use of a fine-grained SR when the size of training data is small by utilizing the features of coarse-grained SRs. Moreover, as the size of training data increases, the *BIES+* model outperforms all other models since the model can effectively deal with entity words of different positional uncertainty by using the features of SRs of different granularities.

One of the most important advantages of the proposed method is the consistent performance improvement over conventional models. As shown in Figure 3.6, three new models (*BIES+*, *IOBES+*, and *IOB2+*) using the proposed method achieve consistently higher F1-score than their counter-part conventional models (*BIES*, *IOBES*, and *IOB2*). Even the *BIES+* model does not exhibit performance degradation when the size of training data is just 10% (1,500 sentences) of the original training data.

In Table 3.14, we compare the best proposed model (best PM) to the systems

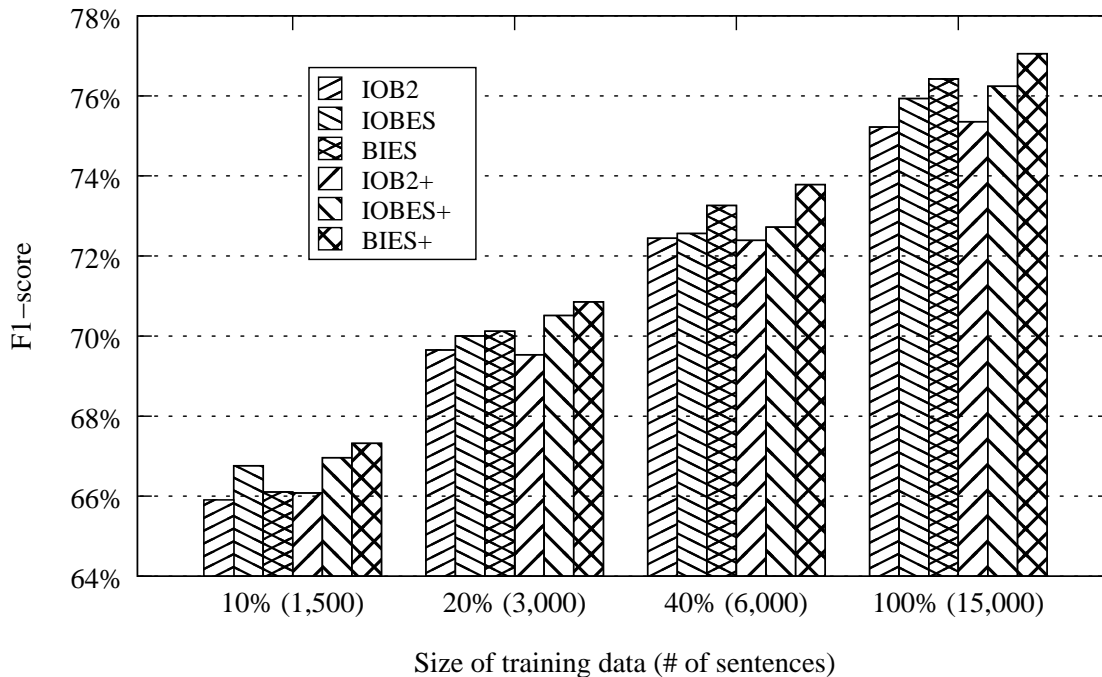


Figure 3.6: The effect of the proposed method on F1-score based on the training data size.

participated in the BC3GMR competition. The comparison is just for reference since BC2 systems exploit various techniques and external resources such as model ensemble, post-processing, abbreviation detection and resolution, semi-supervised learning, gazetteers, and unlabeled data. This information is summarized in the last column of Table 3.14. The best PM is also compared with BANNER¹¹ [71], a publicly available system for biomedical NER tasks, and two state-of-the-art systems [51, 75]. It is placed between the 1st and 2nd ranked BioCreative 2 systems. The overview paper of BioCreative 2 competition states that a difference of 1.23 or more in F1-score is statistically significant ($p < 0.05$). Therefore, we can conclude that our system rivals to the top performing system in the BioCreative 2 competition. Two recently proposed state-of-the-art systems [51, 75] achieve higher

¹¹<http://cbioc.eas.asu.edu/banner/>

Systems	Precision	Recall	F1-score	Add. tech.
Li et al. [75]	90.52	87.63	89.05	E, G, U
Hsu et al. [51]	88.95	87.65	88.30	E, G
BC2-1st	88.48	85.97	87.21	G, P, S
BIES+ (best PM)	90.18	84.17	87.08	G
BC2-2nd	89.30	84.49	86.83	E, G, P
BIES (best BM)	90.58	83.26	86.77	G
BC2-3rd	84.93	88.28	86.57	E
BC2-4th	87.27	85.41	86.33	E, P
BC2-5th	85.77	86.80	86.28	G
BC2-6th	82.71	89.32	85.89	G, P
IOB2 (BM)	88.73	83.07	85.81	G
BANNER	87.18	82.78	84.93	A, P
BC2-7th	86.97	82.55	84.70	A, G

Table 3.14: The performance comparison to the other systems based on the official evaluation. BC2-x means a system participated in the BC2GMR competition and ranked at the x-th position. Add. tech. column shows additional techniques used for these systems, A: Abbreviation resolution, E: Ensemble classifier, G: Gazetteer, P: Post-processing, S: Semi-supervised method and U: Unlabeled data .

performance than the best PM. They obtain such a high performance by combining the results of multiple NER models. The best component NER model in each state-of-the-art system achieves 86.20 and 87.12 in F1-score respectively. Therefore, we can say that the best PM achieves the state-of-the-art performance as a single NER model. In addition, there is a possibility that even better performance can be obtained by integrating the best PM into these systems.

While the proposed method produces a more desirable feature space for a model and improves its performance, the increase of the number of features inevitably slows down training speed. The last column in Table 3.11 shows the number of features for each model that is proportional to the training speed. The most

complex model, *BIES+*, uses more than 60 million features; and the training speed is almost ten times slower than the *IOB2* baseline model. As a simple speed up technique, the *BIES&IO* model is trained with only two SRs, *BIES* and *IO*. Surprisingly, this model achieves comparable performance to the *BIES+* model with a relatively small increase of training time. Therefore, the *BIES&IO* model would be a good alternative to the conventional models when the training speed is important.

3.5.2 NER in the General Domain

The proposed method is also evaluated on the CoNLL 2003 NER shared task data which is a general domain NER corpus. Features used in the study [55] are adopted in this experiment. We used the POS and the chunking information originally provided in the CoNLL training data. However, gazetteers are not employed to observe the effects of our proposed method in isolation.

Figure 3.7 shows the positional uncertainty of entity words in the training data of the CoNLL 2003 NER shared task corpus [126]. The data has four entity types: person (PER), location (LOC), organization (ORG), and miscellaneous (MISC). Entity words appearing more than or equal to 5 times are used for estimating the positional uncertainty reliably. Compared to the GENETAG corpus [115], the CoNLL corpus shows only one peak point of very low positional uncertainty. It is also much higher than the other entity words of different positional uncertainty. The biggest reason of this result is that most location and company names consist of a single word. Table 3.15 shows the segment label distribution on each entity type of the CoNLL NER training data. The only exception is person names since the source of the CoNLL NER data is a collection of news wire articles from the Reuters Corpus and these articles use full names first when they mention a specific person such as politicians and celebrities. From the second occurrence, however, last names are frequently used. The following list presents a few examples of single word entity mentions.

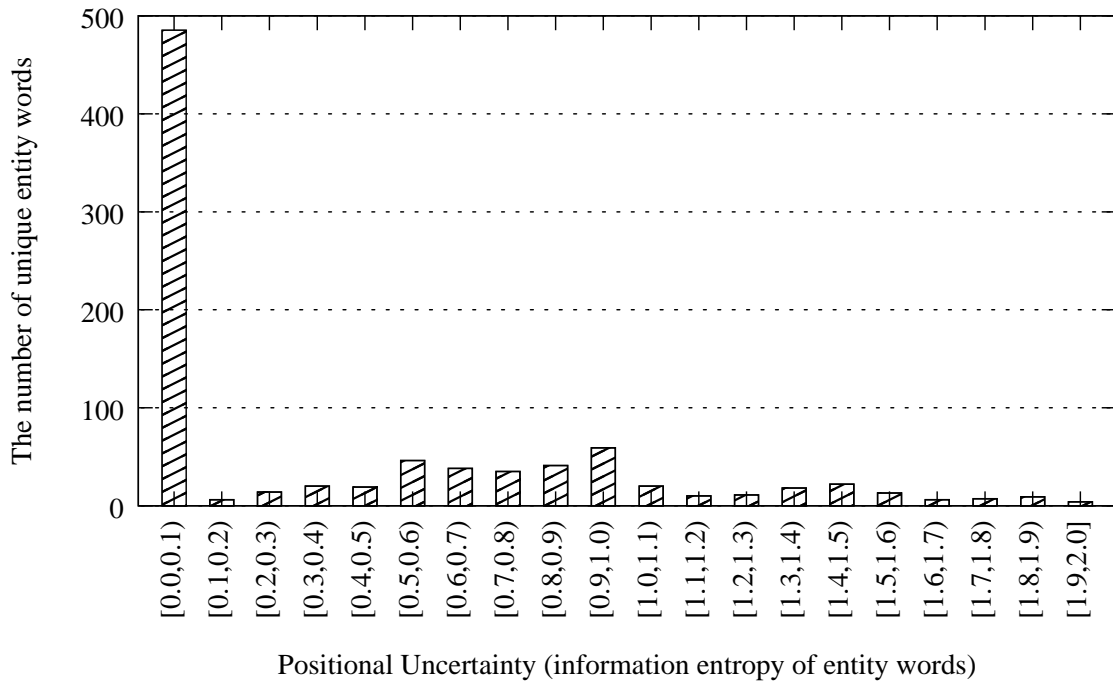


Figure 3.7: Positional uncertainty of entity words in the training data of the CoNLL NER corpus.

PER: Arafat, Fishler, Hendrix, Saddam, Skandalidis

LOC: Beijing, Britain, Europe, Florida, France, Germany, and London, and Taiwan

ORG: Blackburn, Durham, EU, NFU, Reuters, Sussex, Xinhua

MISC: Africans, British, BSE, GMT, Greek, Iraqi, Russian

And the following list shows a few examples of multi-word entity mentions.

PER: Jimi Hendrix, John Lloyd Jone, Loyola de Palacio, Nikolas van des Pas

LOC: Abu Dhabi, Golan Heights Israel, Middle East, Mount Lebanon, United States, West Bank

ORG: Association for Relations Across the Taiwan Straits, BBC Radio, Reuters Television, Welsh National Farmers ' Union

	B	I	E	S
PER	38.50%	2.18%	38.50%	20.82%
LOC	12.53%	1.40%	12.53%	73.53%
ORG	24.79%	12.16%	24.79%	38.26%
MISC	18.22%	6.42%	18.22%	57.13%

Table 3.15: The distribution of segment labels for each entity type on the CoNLL NER data.

Model	Precision	Recall	F1-score	AFI	#feat
IO	83.50	82.14	82.81	28.88	3.10 M
IOB2 (BM)	83.91	82.61	83.25	27.84	5.57 M
IOE2	83.85	82.38	83.11	27.79	5.57 M
IOBES	83.75	82.56	83.15	26.79	10.52 M
BI	83.73	82.56	83.14	26.01	6.19 M
IE (best BM)	83.77	82.86	83.31	25.46	6.19 M
BIES	83.45	82.67	83.06	23.02	12.38 M
IOB2+	84.30	82.99	83.64	28.35	8.67 M
IOBES+	84.34	83.18	83.76	27.75	24.76 M
BIES+ (best PM)	84.35	83.50	83.92	26.41	49.52 M
BIES&IO	83.93	83.07	83.50	25.60	15.47 M

Table 3.16: The performance on the CoNLL NER data.

MISC: Ai n’ t no telling [music title], Bovine Spongiform Encephalopathy [mad cow disease]

Table 3.16 shows the experimental results. The *IE* model achieves the best F1-score in this task. However, the difference compared to other models is not so significant, except the *IO* model. In addition, as a SR becomes more fine-grained, the overall performance begins to decrease as shown with the *IOB2*, *IOBES*, and

BIES models. This result is contrary to the analysis of positional uncertainty in Figure 3.7 because the majority of entity words in this corpus tend to appear at specific positions. The size of the training data could be a reason since the number of entity mentions is relatively smaller than that of the GENETAG corpus. For example, entity mentions of the *MISC* class only appear 3,438 times, whereas the training data of the GENETAG corpus has almost 18,000 entity mentions of the single class, *gene*. In addition, the average number of feature instances per feature (AFI) in the training data drops steeply as the granularity of a SR increases as shown in the fifth column.

When the proposed method is applied, the performance of the proposed models (*IOB2+*, *IOBES+*, *BIES+*, and *BIES&IO*) consistently improves. Especially, the *BIES+* model achieves the best performance for the test data while its corresponding baseline model *BIES* records the worst. Since the results are very similar to that of the previous experiment, we omit the detailed analysis on this task.

3.6 Summary

In this chapter, we presented a feature generation method for incorporating multiple SRs into a single CRFs model. Our method creates a more desirable feature space; therefore, a model can exploit both features of fine-grained SRs which provide high discriminative power and features of coarse-grained SRs which alleviate the problems that can be caused by the data-sparseness. Furthermore, we explained how a model computation after training can make the tagging speed of a model using the proposed method as fast as a model using a single SR.

The proposed method is evaluated on two NER tasks of biomedical and general domain corpora. The results demonstrated that our motivation of using multiple SRs is beneficial to better NER performance. In biomedical domain NER task, our NER system without any post-processing techniques has reached to the per-

formance of the top system which exploit abundant of external resources and post-processing techniques. In addition, we provided the results of the statistical significance test to show that the improvement is not by chance, and the detailed performance analysis to explain the effects of using multiple SRs for NER. Lastly, the evaluation on CoNLL NER corpus is also provided to show the domain independence of our proposed method.

Although many researches say that statistical NER systems have reached the plateau of performance, we think that still there is a room for meaningful improvement. Our method suggested one of such ways that use multiple perspectives for a problem. In addition, the proposed method is applicable to any segmentation tasks such as shallow parsing and word segmentation. We expect that the proposed method is also beneficial to these tasks too because the proposed model using multiple SRs exhibited better performance than the best conventional model.

Chapter 4

Inducing Context Gazetteers from Encyclopedic Databases for Named Entity Recognition

4.1 Introduction

Named entity recognition (NER) is a task that recognizes entity mentions of interest in text. Entity types vary depending on the target domains. In the news domain, for example, the names of people, locations and organizations are the most common entity types [16, 126], whereas the names of genes and gene products are the most important types in the biomedical domain [60, 115]. In fact, NER has been regarded as a fundamental sub-task in many natural language processing (NLP) applications such as information extraction, question and answering, and machine translation.

NER has been tackled in various ways from rule-based to statistical approaches. Most current approaches formalize this problem as a sequence labeling task and employ machine learning (ML) techniques, such as Conditional Random Fields (CRF) [82] and Support Vector Machines (SVMs) [5], as their core component. However, the success of ML-based approaches heavily depends on the availability of training corpus similar to other NLP tasks [6]. Previous studies tried to solve

this problem in two ways: by automatically (or semi-automatically) increasing the amount of training data and by utilizing features that generalize well to cover unseen examples. The first approach is generally referred to as semi-supervised ML approach that involves various techniques such as bootstrapping and active learning [27, 87, 96, 105, 113, 129]; and, the second one is called feature engineering [7, 24, 38, 97, 116]. These two approaches have their own merits and demerits. For instance, semi-supervised approach utilizes unlabeled data that is far larger than labeled data. However, this process inevitably introduces noisy data into training corpus due to the annotation error and the semantic drift [29, 129]. On the other hand, feature engineering can improve a model by utilizing generalized features and existing training data. A problem of this approach is that generalized features are not always discriminative enough to allow accurate prediction.

In this study, we present the idea of a new resource, context gazetteer, which takes advantage of the previous two approaches and a method to automatically create it from a certain type of unlabeled data, encyclopedic database. A context gazetteer consists of partial dependency paths of variable length that frequently co-occur with entity mentions. In the viewpoint of feature engineering, these sophisticated contexts are relatively unambiguous than traditional linear contexts such as word uni-grams and bi-grams because they are syntactically constrained. Confidence values assigned to the contexts also allow a model to take into account the different predictive power of different contexts. A model exploits these contexts by generalizing them in the form of a gazetteer since most of them do not appear in training data. In the viewpoint of semi-supervised approach, the proposed method also automatically annotates unlabeled data and extracts contexts from it. Since it is relatively easy to obtain a large quantity of unlabeled data compared to labeled data, we can harvest rich and sophisticated context patterns that can help to recognize both unknown entity mentions, which do not appear in training data, and out-of-vocabulary (OOV) entity mentions, which are not registered in traditional gazetteers. The proposed method is based on a single pass algorithm, which

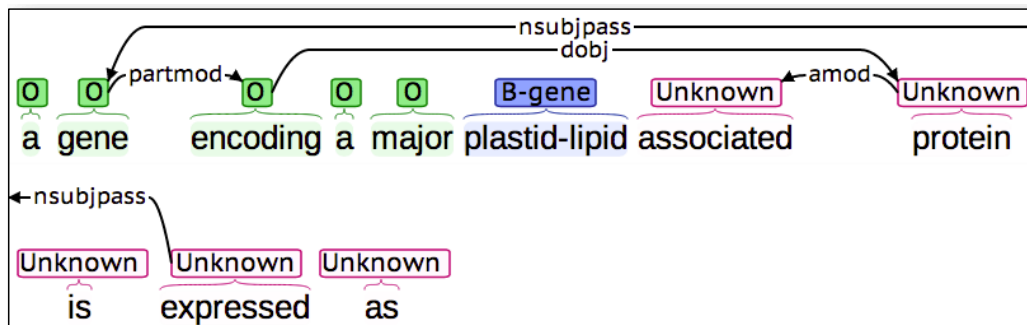


Figure 4.1: An example of syntactically constrained contexts of the word, *associated*. In the text, *plastid-lipid associated protein* is a gene name in which the first word is labeled with the *B-gene*. The dependency label *amod* stands for adjectival modifier, *dobj* for direct object, *partmod* for participial modifier and *nsubjpass* for the passive nominal subject.

performs entity annotation and context extraction processes only once. Therefore, it can avoid the problem of noisy annotation due to the incomplete annotation and the semantic drift.

Figure 4.1 shows an example of syntactically constrained non-local contexts that can help to recognize entity mentions. Presuming that we shall determine the label of the word *associated*, the direct and indirect head words of the word *associated*, such as *protein*, *encoding*, *gene*, and *expressed*, can be used as informative features for prediction. Compared traditional local context features, which are extracted from a small linear window, these contexts can cover much broader areas within an input sentence.

In experiment, we build a context gazetteer of gene names and apply it for a biomedical named entity recognition task. It is particularly interesting that top-ranked entries in the context gazetteer appear in various forms. As expected, there are many predicate–argument structure style contexts using domain specific verbal (and nominal) predicates such as “express,” “inhibit,” and “promote.” Moreover, abbreviation, apposition, and conjunction dependencies are frequently included as a part of highly confident context patterns. These contexts can be interpreted as

fragments of domain knowledge that appear in stereotypical syntactic structures in text. The context gazetteer boosted both the precision from 79.00 to 79.26 and the recall from 71.99 to 72.78. As a consequence, the overall F1-score is improved from 75.33 to 75.88.

The remainder of this study is organized as follows. In Section 4.2, we explain related studies to our work. Section 4.3 describes the proposed method for creating a context gazetteer from an encyclopedic database. For evaluating the usefulness of the new resource, we build a context gazetteer of gene names from the EntrezGene database [79] and apply it to the BioCreative 2 gene name recognition task [115] in Section 4.4. During the evaluation, we demonstrate what kinds of context patterns are harvested and how different featurization methods affect recognition performance. We also manually examine the results to analyze the effect of using the context gazetteer. Section 4.5 summarizes the contributions of this work, and explains the future work for generalizing learned contexts.

4.2 Related Work

This section presents a summary of three types of related studies of sentence level non-local features, gazetteer induction and semi-supervised learning.

Sentence level non-local features usually depend on a deep parsing technique. For example, a previous work [38] used the Stanford dependency parser [32] to exploit features such as the head and governor of the noun phrases in a biomedical NER task. A more recent work [116] evaluated the effect of seven different parsers in feature generation for finding base noun phrases including gene names. However, they extract contexts only from training data, whereas we use a large amount of automatically annotated data. As a result, our approach is likely to provide richer and more sophisticated context patterns than their methods.

Gazetteers are invaluable resources for NER tasks, especially for dealing with unknown words that do not appear in training data. They might have the same

semantic categories to target entity classes [41], or related classes that are often more fine-grained sub-classes of the target entity classes [55, 100]. Word clusters are also useful resources for NER similar to gazetteers. In a related study [84], the Brown clustering algorithm [12] were applied to NER successfully. A more recent work [56] used the dependency relations between verbs and multiword nouns for clustering multiword expressions. However, to the best of our knowledge, all of the related work that we have surveyed produce entity gazetteers (clusters).

The most similar concept to the contexts in this research can be found in the studies related to semi-supervised learning approach. For instance, a bootstrapping method [106] extracts context patterns from unlabeled data by using a small set of seed words (entity mentions in case of NER) for a target class. In turn, it extracts new entity mentions by using the extracted context patterns, and repeats this process. However, the quality of context patterns (and also entity mentions) degrades as iteration goes on because it inevitably suffers from semantic drift. In contrast, our method induces a large number of highly precise contexts without a repetitive process by exploiting an encyclopedic database. This approach has become more realistic lately because of many publicly available resources such as Wikipedia¹ and domain-specific databases [79].

4.3 Building a Context Gazetteer

A context gazetteer is a confidence assigned list of dependency paths (hereinafter, contexts) of variable length that can co-occur with target entity mentions. Figure 4.2 portrays an exemplary context of length 3. It is a high confidence context in the context gazetteer of gene names that will be used in the experiment section. It means that a word X surrounded by the context consisting of the head word *expression*, a dependent *cells* and a grand-dependent *cancer* with the corresponding dependency relations *prep_of*, *prep_in* and *nn* is likely to be an entity word,

¹<http://www.wikipedia.org/>

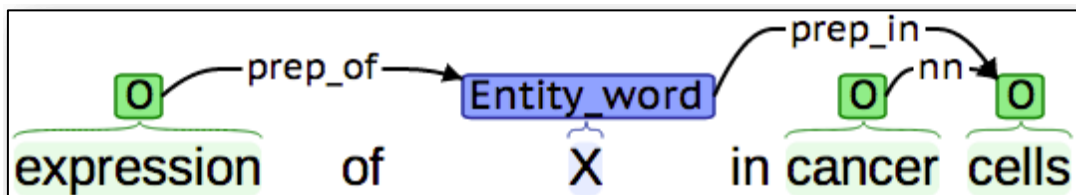


Figure 4.2: An example context of the length 3 in which X is a slot for an entity word. (*pref_of* stands for prepositional modifier of, *pref_in* for prepositional modifier in and *nn* for noun compound modifier.)

which is a part of a target entity mention. This context can help to recognize the headword of an underlined gene name in a sentence, “The *expression* of FasL in gastric *cancer cells* and of Fas in apoptotic TIL was also detected in vivo.”

A useful context gazetteer should have rich and sophisticated contexts that are specific to target semantic classes. For the first requirement, we extract contexts from a large amount of automatically labeled data rather than a few manually annotated data. To satisfy the second requirement, confidence values are assigned to the extracted contexts. Figure 4.3 is the flowchart for the context gazetteer generation. Each step is explained in detail in the following.

Step 1.

An encyclopedic database consists of domain specific entity mentions (shown as entity in the figure) and their descriptions (shown as text in the figure). For each entity mention, we label every occurrence of it in the description by using the exact string matching algorithm. The primary reason for using an encyclopedic database rather than the list of target entity mentions and some free text is to remove the ambiguity of the semantic categories of target entity mentions appearing in free text [129]. For example, presuming that we are going to generate labeled data with the names of people by using some free text (e.g. newspapers) and a list of the names of people automatically, the process would invariably create very noisy data because human names are often used as the names of companies (e.g.,

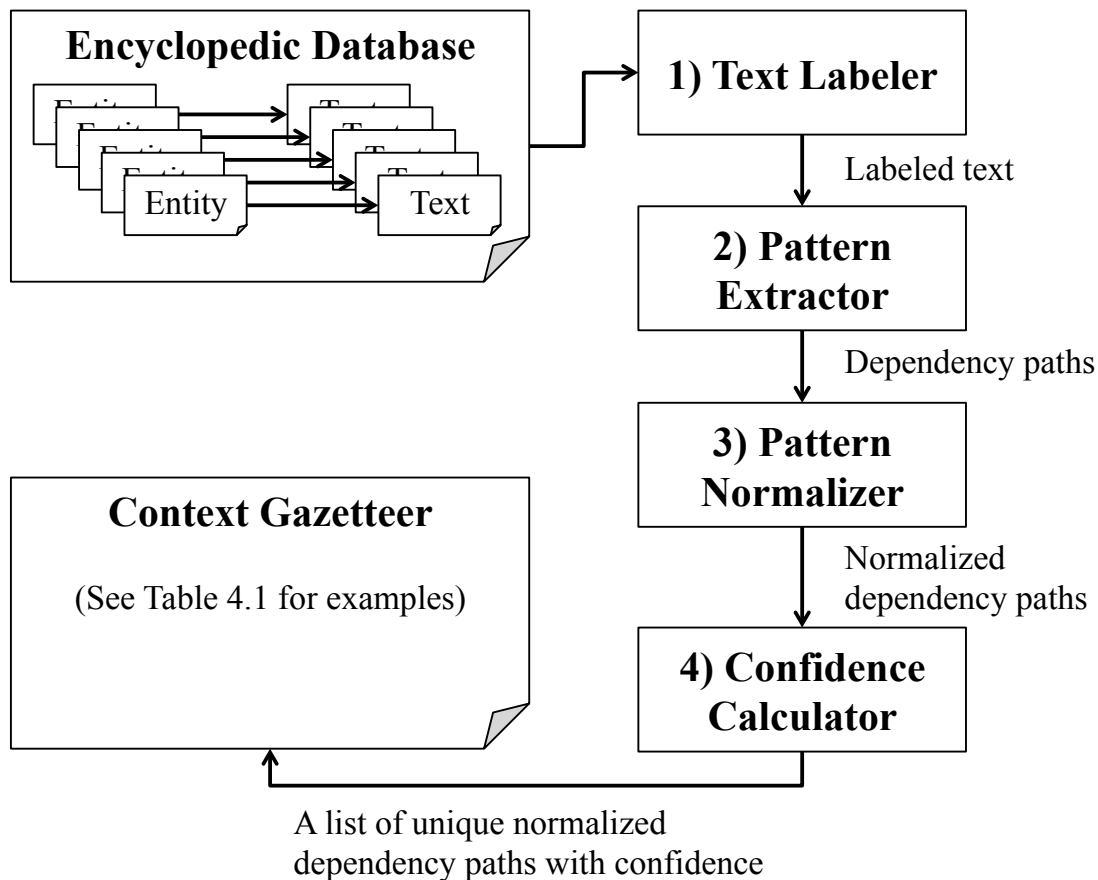


Figure 4.3: Building a context gazetteer from an encyclopedic database.

Hewlett-Packard and Ford Motor Company), diseases (e.g. Alzheimer disease), places (e.g., Washington, D.C and St. Paul, Minnesota), and so on.

Step 2.

The labeled text are then parsed. The dependency paths (contexts) involving entity words are extracted. Because of the excessive number of possible contexts, we applied two constraints to context generation. First, the contexts that have no content words (nouns, verbs and adjectives) except an entity word are removed because these contexts are often too general to be effective contexts. Second, we limit the maximum length of contexts depending on the data size (we used the

maximum length 5 for the experiment in this study).

Step 3.

For each context, an entity word is replaced with a wildcard character that matches any word. We think that additional normalization at this stage can further increase the coverage of a context gazetteer. For example, we can use stems (or lemmas) rather than words. After normalization, we remove duplicated contexts and keep them unique.

Step 4.

Contexts are often ambiguous even if they frequently appear with target entity mentions. We solve this problem by assigning confidence to each context. Presuming that data D is annotated automatically with the mentions of T different entity types², then, the confidence (conditional probability) of an entity type t given a context c is defined as in

$$\text{confidence}(t|c) = p(t|c) = \frac{C(t, c)}{C(c)} = \frac{\sum_{e_t \in D} C(e_t, c)}{C(c)}, \quad (4.1)$$

where e_t is an entity word of the semantic type $t \in T$ in the data D . The estimated confidence is pessimistic, meaning that they are usually lower than they should be because automatically annotated data have high precision but low recall.

4.4 Evaluation

In this section, we create a context gazetteer of gene names from the EntrezGene database [79], and demonstrate its usefulness by applying it to the BioCreative 2 gene name recognition task [115]. We performed two experiments: one to assess

²The set T includes non-entity type O too.

the effect of different featurization methods and the other one to evaluate the relationship between a context gazetteer and a entity gazetteer.

4.4.1 Data

Context Gazetteer.

For gazetteer generation, we use the gene names (including synonyms) and the human curated reference information in the EntrezGene. At the first step in Figure 4.3, 358,049 abstracts including titles are extracted from the MEDLINE database³ by using the reference information of the EntrezGene. In each abstract, gene names that have reference links from the EntrezGene to the abstract are labeled based on the exact string matching. The labeled gene names are highly precise because of the references information between the gene names and the abstracts in the EntrezGene.

Second, the labeled text are parsed by using the Stanford POS tagger [127] and dependency parser [32] included in the CoreNLP tool⁴. Then, we extracted the dependency paths (contexts) that involve entity words. Contexts that have no content words aside from entity words are filtered out since they are very general patterns and are not much informative. The maximum length is set to 5 experimentally.

Third, the entity words of the extracted contexts are anonymized. In the biomedical domain, many entity mentions include symbols and numbers. For domain-specific normalization, continuous numbers and symbols of the words in the contexts are converted into a representative number (0) and symbol (underbar), respectively. Lastly, confidence values are assigned to each context using Equation 4.1. Contexts appearing less than 10 times are removed in this process because the their estimated confidence values can be unreliable.

³MEDLINE is the U.S. National Library of Medicine’s (NLM) premier bibliographic database.

⁴<http://nlp.stanford.edu/software/corenlp.shtml>

Several extracted contexts having high confidence are presented in Figure 4.4. At the beginning of this study, we expected to obtain contexts similar to predicate-argument structure (PAS) and domain specific relations. For example, the second context in this table indicates that X is likely to be a gene if it appears in a relation with a gene name *C-jun* as in “... interaction between X and C-Jun.” The fourth and fifth context patterns are in the form of PAS using the nominal and verbal predicates respectively. However, we also found unexpected but interesting context patterns too. First, many contexts capture factual knowledge. The first context is very simple but highly confident pattern meaning that X is likely to be a gene if it is a *globin*. Second, some contexts represent procedural information. The third context, for instance, indicates that there is a screening process for analyzing mutations of a gene. Lastly, the sixth context, seemingly uninformative at first glance, means that discovering the function of a gene is a common task as in “Although the exact function of RPE65 is not yet known, a role in vitamin A metabolism has been proposed, and ...”

Entity Gazetteer.

We use four entity gazetteers compiled from the EntrezGene, Universal Protein Resource (UniProt) [26], Unified Medical Language System (UMLS) [9] and the Open Biological and Biomedical Ontologies (OBO)⁵. For improving the coverage of these gazetteers, continuous numbers and symbols of the entity mentions are normalized into a representative number and symbol (0 for numbers and under-bar for symbols), and all alphabet characters are lower-cased. This process also applies to the input text.

For the entity gazetteers compiled from the EntrezGene and the UniProt, we use the single semantic categories: gene and protein. However, the UMLS and the OBO gazetteers have multiple categories, some of which are related to gene names such as peptides and amino acids, but many of which are different biomedical entity

⁵<http://www.obofoundry.org/>

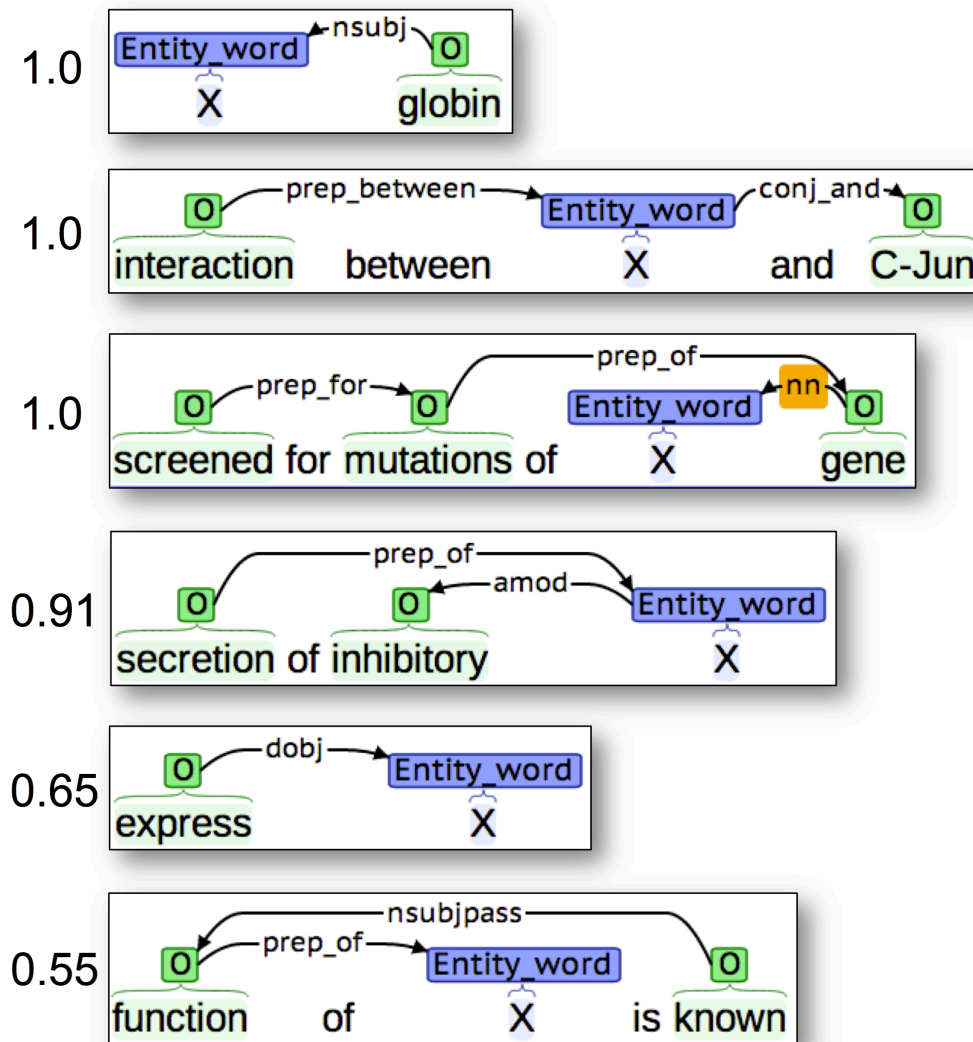


Figure 4.4: Examples of high confidence extracted context patterns. Scores at the left side show the confidence of context patterns. (X is a place-holder, *nsubj* is nominal subject, *conj_and* is conjunction and, *nn* is noun compound modifier, *amod* is adjectival modifier, *dobj* is direct object, and *nsubjpass* is passive nominal subject.)

categories. During NER system development, we found that not only gene-related categories but also other categories are beneficial for increasing performance.

GENETAG corpus.

The BioCreative 2 gene mention recognition task uses the GENETAG corpus [124] comprising 20,000 sentences, of which 15,000 sentences were used for training and 5,000 sentences were used for testing.

We processed raw text to obtain additional syntactic information for use in feature generation. Raw text consisting of sentences are split into tokens by using a fine-grained tokenization scheme that uses whitespace and non-alphanumeric characters as token boundary markers. When a string is tokenized at non-alphanumeric character, this character also becomes a single character token (e.g., “p53-activated” to “p53,” “-,” and “activated”). Next, the tokenized text is fed to the GENIA tagger [128] for lemmatization, POS-tagging, and chunking. For each entity gazetteer, the sequences of tokens that appear in the gazetteer are tagged by using the BIO labels (e.g., “B-EntrezGene,” “I-EntrezGene,” “B-UniProt”). Lastly, for the EntrezGene context gazetteer, the tokens surrounded by the contexts of the gazetteer are tagged with context gazetteer class label. Six types of featurization methods for a context gazetteer will be explained in the next section.

4.4.2 Machine Learning and Featurization

For machine learning, we use the CRFsuite [95], which implements the first-order linear-chain Conditional Random Fields [70]. The regularization parameter (C) is optimized by using the first 90% of the original training data as the training data and the rest, 10%, as the development data. Eleven C values (0.25, 0.5, 0.75, 1, 2, 3, 4, 5, 6, 8, and 10) are tested and the best performing one is chosen.

A set of features used in the experiment is described in Table 4.1, and the symbols are explained in Table 4.2. For the featurization of the EntrezGene context

Class	Description
Token	$\{w_{t-2}, \dots, w_{t+2}\} \wedge y_t, \{w_{t-2,t-1}, \dots, w_{t+1,t+2}\} \wedge y_t,$ $\{\bar{w}_{t-2}, \dots, \bar{w}_{t+2}\} \wedge y_t, \{\bar{w}_{t-2,t-1}, \dots, \bar{w}_{t+1,t+2}\} \wedge y_t$
Lemma	$\{l_{t-2}, \dots, l_{t+2}\} \wedge y_t, \{l_{t-2,t-1}, \dots, l_{t+1,t+2}\} \wedge y_t,$ $\{\bar{l}_{t-2}, \dots, \bar{l}_{t+2}\} \wedge y_t, \{\bar{l}_{t-2,t-1}, \dots, \bar{l}_{t+1,t+2}\} \wedge y_t$
POS	$\{p_{t-2}, \dots, p_{t+2}\} \wedge y_t, \{p_{t-2,t-1}, \dots, p_{t+1,t+2}\} \wedge y_t$
Lemma & POS	$\{l_{t-2}p_{t-2}, \dots, l_{t+2}p_{t+2}\} \wedge y_t,$ $\{l_{t-2,t-1}p_{t-2,t-1}, \dots, l_{t+1,t+2}p_{t+1,t+2}\} \wedge y_t$
Chunk	$\{c_t, w_{t_last}, \bar{w}_{t_last}, the_{lhs}\} \wedge y_t$
Char.	Character 2,3,4-grams of w_t
Ortho.	All capitalized, all numbers, contain Greek letters, ... (Refer to [73] for the detailed explanation)
E. gaz.	$\{ge_{t-2}, \dots, ge_{t+2}\} \wedge y_t, \{ge_{t-2,t-1}, \dots, ge_{t+1,t+2}\} \wedge y_t,$ $\{ge_{t-2}l_{t-2}, \dots, ge_{t+2}l_{t+2}\} \wedge y_t, \{ge_{t-2,t-1}l_{t-2,t-1}, \dots, ge_{t+1,t+2}l_{t+1,t+2}\} \wedge y_t$
C. gaz.	1) $gc_t^n \wedge y_t$ 2) $C(gc_t^n) \wedge y_t$ 3) $C_{[k,k+0.1]}(gc_t^n) \wedge y_t$ where $0.0 \leq k < C(gc_t^n)$ and $k \in \{0.0, 0.1, \dots, 0.9\}$ 4) $\{gc_t^n, gc_t^n l_t\} \wedge y_t$ 5) $\{C(gc_t^n), C(gc_t^n) l_t\} \wedge y_t$ 6) $\{C_{[k,k+0.1]}(gc_t^n), C_{[k,k+0.1]}(gc_t^n) l_t\} \wedge y_t$ where $0.0 \leq k < C(gc_t^n)$ and $k \in \{0.0, 0.1, \dots, 0.9\}$

Table 4.1: Features used for experiments. Char. stands for character features, Ortho. for orthographical features, E. gaz. for entity gazetteer, and C. gaz. for context gazetteer.

gazetteer, we tested six methods as shown at the last row of Table 4.1. The first one is a simple binary feature that is true if a context around a word appears in the context gazetteer. This feature can be triggered more than once for the same word if different contexts around the word appear in the context gazetteer. The second one is a real-valued feature that uses estimated confidence as explained in

Symbol	Description
w_t	A t -th word. (e.g., p53, expresses)
\bar{w}_t	A normalized t -th word where successive numbers and symbols are converted into a single zero and under-bar. (e.g., p0)
l_t	A t -th lemma. (e.g., express)
\bar{l}_t	A normalized t -th lemma. (e.g., express)
p_t	A t -th POS-tag. (e.g., NN (noun, singular or mass), NNS (noun, plural), JJ (adjective), VB (verb, base form))
c_t	The chunk type of w_t . (e.g., noun phrase, verb phrase, prepositional phrase)
w_{t_last}	The last word of a current chunk.
\bar{w}_{t_last}	The normalized last word of a current chunk.
the_{lhs}	True if 'the' exists from the beginning of a current chunk to w_{t-1} .
ge_t^n	A label of the entity gazetteer n for a t -th word. (e.g., $ge_3^{EntrezGene} = Gene$, $ge_3^{UniProt} = Protein$, $ge_8^{UMLS} = Disease$, $ge_{12}^{OBO} = Chemical_substance$)
gc_t^n	A label of the context gazetteer n for a t -th word. (e.g., $gc_3^{EntrezGene} = Gene$)
$C(gc_t^n)$	The confidence of a context label gc_t^n .
$C_{[k,k+0.1)}(gc_t^n)$	A quantized confidence symbol of the context label gc_t^n based on its confidence $C(gc_t^n)$. (<i>This is a symbol, not real value.</i>)

Table 4.2: Symbols used for features (see Table 4.1).

Equation 4.1. We can see the effect of confidence in the use of context gazetteer features by comparing the results of these two featurization methods. The third one quantizes real-value confidence into ten binary features by increasing the confidence 0.1 by 0.1 from 0.1 to 1.0, and uses these features that have the confidence range lower than the confidence of a context pattern. For example, if a context pattern around the second word appears in the context gazetteer and has the confidence 0.38, it will trigger three binary features, namely $C_{[0.0,0.1)}(gc_2^{EntrezGene})$,

$C_{[0.1,0.2]}(gc_2^{EntrezGene})$, and $C_{[0.2,0.3]}(gc_2^{EntrezGene})$. We designed this type of features since the contribution of a context feature to entity word recognition may not be linearly proportional to its confidence. Suppose that context features become very informative when it crosses a certain threshold. Then, a machine learning algorithm can give sigmoid-like weights to quantized features. In such a situation, this type of featurization will be more suitable than using a real-valued confidence feature. The other three featurization methods use lexicalized features of the previous three methods in addition to the original unlexicalized features. A lexicalized feature is the combination of an unlexicalized feature and a normalized current lemma, \bar{l}_t , as shown in Table 4.2. Lexicalized features are useful, especially when high confidence context patterns co-occur with non-entity words. Pronouns such as *it*, *this*, and *that* are most obvious examples.

4.4.3 Experiment Results

The first experiment evaluates the effect of six featurization methods described in the previous section. Table 4.3 shows the performance of the NER models using these featurization methods. We use two evaluation measures: one is based on the strict-match and the other one is based on the relaxed-match (official evaluation scheme). The relaxed-match evaluation result is shown within a pair of parentheses. The baseline model uses all features explained in Table 4.1 except the context features. The GC1, GC2, and GC3 models use a recognized context pattern as a simple binary feature, a real value feature, and a group of quantized binary features respectively. The GC1-LEX, GC2-LEX, and GC3-LEX models use lexicalized features in addition to the original unlexicalized features.

The GC1 model, which uses a recognized context pattern in the simplest form as a binary feature, increases recall by 0.35 percent while slightly losing its precision by 0.08 percent compared to the baseline model. The improvement in recall comes from the new non-local contexts that are usually more general information than local contexts, which are frequently part of entity mentions. A slight decrease in

Model	Precision	Recall	F1-score
Baseline	79.00 (89.06)	71.99 (82.78)	75.33 (85.81)
GC1	78.92 (88.91) ↓	72.34 (83.18) ↑	75.49 (85.94) ↑
GC2	79.26 (89.23) ↑	72.78 (83.65) ↑	75.88 (86.35) ↑
GC3	78.12 (88.36) ↓	72.42 (83.54) ↑	75.16 (85.88) ↓
GC1-LEX	79.87 (90.02) ↑	70.54 (81.33) ↓	74.92 (85.45) ↓
GC2-LEX	79.35 (89.37) ↑	72.53 (83.43) ↑	75.79 (86.30) ↑
GC3-LEX	78.87 (89.10) ↓	72.44 (83.54) ↑	75.51 (86.23) ↑

Table 4.3: Performance evaluation using six types of context pattern featurization methods. The upward and downward arrows indicate the change of performance compared to the baseline model.

precision may result from ambiguous context patterns. We can resolve this problem by distinguishing unambiguous context patterns from ambiguous ones. The GC2 model, which adopts this idea, uses a recognized context pattern as a real-valued feature based on its estimated confidence. It further improves precision and recall by 0.26 and 0.79 percent respectively and achieves an F1-score of 75.88. The benefit of using estimated confidence can be also verified by comparing the learnt weights of these features in the GC1 and GC2 models. The binary feature of the GC1 model has the weight around 0.1, whereas the real-value feature of the GC2 model has the weight about 0.9. It indicates that estimated confidence correctly reflect the quality of context patterns so that a machine learning algorithm can reliably depend on this feature. Contrary to our expectation, however, using a context pattern as a group of quantized binary features results in poor performance due to the decrease of precision as shown in the GC3 model. We found that a machine learning algorithm experiences difficulty in estimating the proper weights of the quantized features of high confidence ranges (e.g., $C_{[0.7,0.8)}gc_t$, $C_{[0.8,0.9)}gc_t$, $C_{[0.9,1.0]}gc_t$) since they do not appear frequently in the training data. Figure 4.5 shows the weights of quantized context gazetteer features for each label. We can

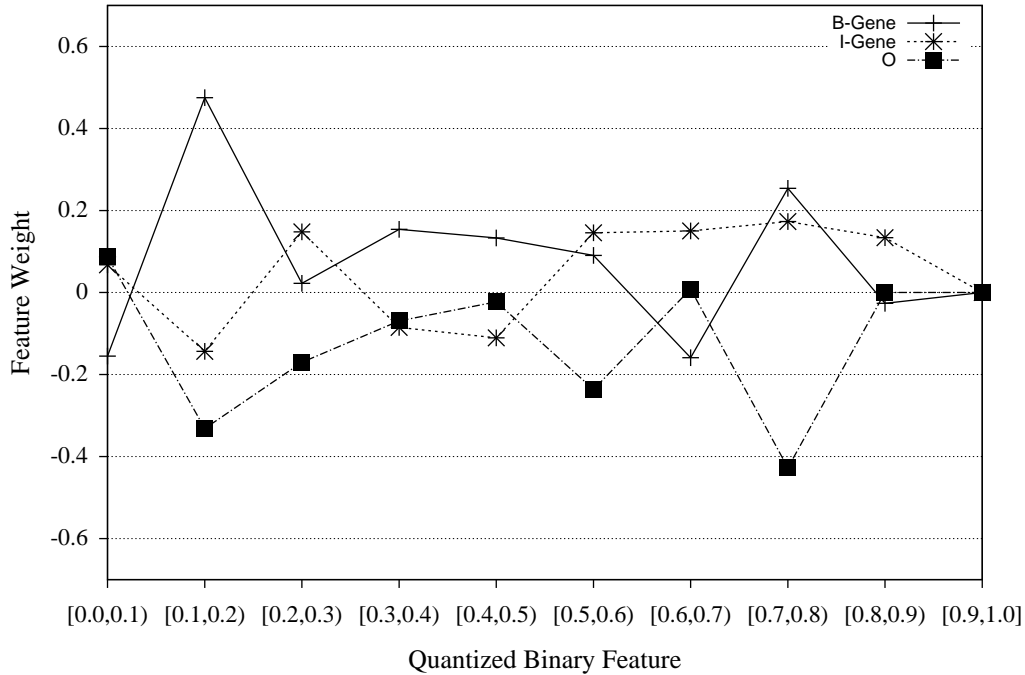


Figure 4.5: The feature weights of quantized binary features.

see that feature weights related two entity related labels (*B-Gene* and *I-Gene*) are fluctuating and totally different from the shape of the sigmoid function.

In addition to these unlexicalized features, the GC1-LEX, GC2-LEX, and GC3-LEX models use lexicalized context features that are the combination of an unlexicalized feature and the normalized lemma of a current word (\bar{l}_t). Surprisingly, the performance of these lexicalized models is inferior to that of unlexicalized models. In the experiment results, lexicalized features improves a small amount of precision, whereas it decreases similar amount of recall. Since recall is relatively lower than precision in general⁶, the overall performance of lexicalized models becomes lower than that of unlexicalized models. One of the reasons that we consider is the sparsity of lexicalized features. Since many context patterns are already complex, combining them with lemma makes lexicalized features very sparse and prone to over-fitting.

⁶In NER, recall is relatively lower than precision because of the skewed label distribution in training data where one class label, *O*, dominates all other classes [54].

Model	E. Gaz.	C. Gaz.	Precision	Recall	F1-score
Base line	None	None	77.43 (87.99)	70.13 (81.71)	73.60 (84.73)
Ctx-Gaz	None	EG	77.98 (88.25)	70.35 (81.60)	73.97 (84.79)
Ent-Gaz	ALL	None	79.00 (89.06)	71.99 (82.78)	75.33 (85.81)
All-Gaz	ALL	EG	79.26 (89.23)	72.78 (83.65)	75.88 (86.35)

Table 4.4: Performance evaluation using entity and context gazetteers. In the second column, “E. Gaz” means entity gazetteer where “EG” stands for EntrezGene and “ALL” for the gazetteers compiled from four databases, the EntrezGene, UniProt, UMLS, and OBO. In the third column, “C. Gaz.” means context gazetteer.

Lastly, we conducted the statistical significant test between the baseline model and the best-performing model (GC2) to verify whether the improvements is meaningful. We performed the statistical significance test using the bootstrap re-sampling method [115]. More specifically, from the set of 5,000 sentences in the test data, new 5,000 sentences are randomly sampled with replacement for 10,000 times. Then, the performance of two models is evaluated on the 10,000 sets of sampled test data. The p -value of the GC2 model is 0.0040, which means that it achieves better performance than the baseline model for 9,960 times among 10,000 times.

The second experiment is designed to investigate the relation between entity and context gazetteers. If the effect of these two types of gazetteers is independent, as we assume, recognition performance will be enhanced by the sum of performance improvement by them. We tested various combinations of the four entity gazetteers and one context gazetteer. A context gazetteer pattern is used as a real-valued feature. Table 4.4 shows the experiment results. The baseline model does not use any gazetteers. The Ctx-Gaz model exploits the context gazetteer built from the EntrezGene and the Ent-Gaz model utilizes four entity gazetteers compiled from the EntrezGene, UniProt, UMLS, and OBO databases. Lastly, the All-Gaz model,

which is equivalent to the GC2 model, uses all of these gazetteers.

In terms of precision, the use of the context and entity gazetteers in the Ctx-Gaz and Ent-Gaz models improves precision by 0.55 and 1.57 percent respectively. The All-Gaz model achieves a precision of 79.26 (+1.83 percent) that is slightly lower than the sum of the improvement of the previous two models (+2.12 percent). On the other hand, the recall in the Ctx-Gaz and Ent-Gaz models increases 0.22 and 1.86 percent. Contrary to the previous case, the improvement in the recall of the All-Gaz model (+2.65 percent) is more than the sum of the improvement in the Ctx-Gaz and Ent-Gaz models (+2.08 percent). Considering the trade-off between precision and recall, we can conclude that the effect of entity and context gazetteers is almost independent.

4.4.4 Result Analysis

We manually compared about 20% of the output of two models, Ent-Gaz and All-Gaz, to see how the context gazetteer features affect the tagging results.

There are 32 gene names correctly recognized by the All-Gaz model but not by the Ent-Gaz model. In all of these cases, one or more context gazetteer features are triggered. Figure 4.6 shows three examples in which the Ent-Gaz model identified 8 gene names marked with red color and the All-Gaz model recognized 11 gene names marked with dark green color.

Two context gazetteer features are triggered for the gene name “MEQ,” “`dobj(encode, X)`” and “`appos(X, protein)`.” The second feature is a strong evidence of X being a gene name because a word X is in apposition with the word protein. In the second example, “I-92” has a feature “`prep_of(association, X) \wedge prep_with(X, p0)`” meaning that X is likely to be a part of gene name if it is associated with the gene name “p0” where 0 is a normalized number. Contexts of these kinds are the fragments of domain specific knowledge and usually have high confidence (0.5 for this context). In the last example, the gene name “IP-30” has a context gazetteer feature “`prep_of(function, X)`” and a more specific one “`nsubjpass(known, func-`

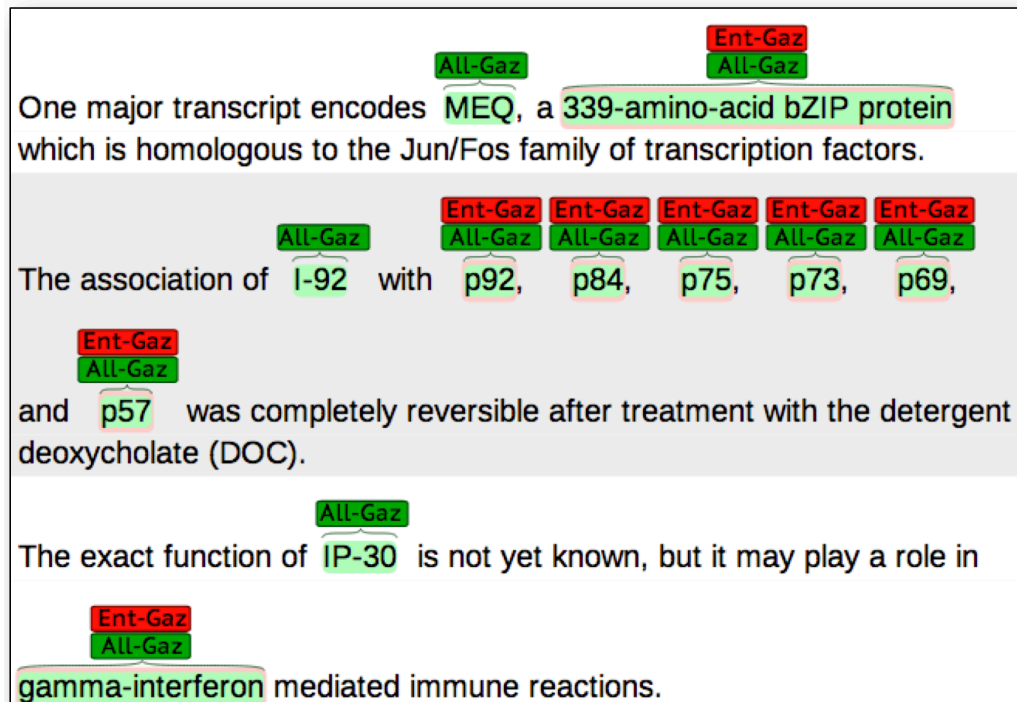


Figure 4.6: Three sentences excerpted from the test data. Three gene names are newly recognized by using the context gazetteer.

tion) \wedge prep_of(function, X)” with confidence 0.44 and 0.54. These contexts can be interpreted as domain-specific expressions where figuring out the function of a gene is a much more important task than others (54% vs. the rest).

However, 15 gene names are recognized by the Ent-Gaz model, but not by the All-Gaz model. Among them, three gene names did not have any context gazetteer features. Since we use the words (not stems or lemmas) in the contexts, the coverage might be not sufficiently high. For the other 12 cases, context gazetteer features are fired, but these gene names are not recognized.

4.5 Summary

In this study, we presented the concept of a new resource, a context gazetteer, and a method to automatically create it from a certain type of unlabeled data, encyclopedic database. By taking advantage of both feature engineering and semi-supervised approaches, we could overcome the difficulties of each approach and bring improvement in recognition performance. Compared to the feature aggregation methods [15, 66, 100], the proposed method can be easily applied to streaming data such as tweets and pre-processed data with sentence selection where recognizing document (or discourse) boundary is difficult. In addition, the proposed method is based on a single-pass algorithm; therefore, it is not necessary to worry about the semantic drift problem.

However, we also uncovered difficulties. First, for this research, we used words and their dependencies as contexts. However, these contexts sometimes include uninformative words in the middle of contexts. If it is possible to generalize the contexts by replacing these unimportant words with POS-tags or wildcards, then the coverage of the context gazetteer can be enhanced. Second, gene names (or parts of them) often appear as a part of contexts. Although these contexts often have very high confidence, they may not be general patterns. They can be more useful if they were replaced by some general gene name wildcards.

Chapter 5

Conclusion

5.1 Contribution of this Thesis

Named entity recognition (NER) has been considered as an important research issue in natural language processing and information extraction areas and also utilized as a fundamental application for various information processing systems. While most recent NER systems achieve impressive performance by employing supervised learning techniques, they often suffer from the data-sparseness problem due to the limited size of training data and the use of complex features.

In this thesis, we proposed two methods to address this problem in the viewpoint of feature generalization. Chapter 3 deals with the segmentation problem of NER tasks. NER, which is formalized as a sequence labeling problem, consists of two sub-tasks, the segmentation and classification tasks. Most previous studies tried to solve these two problems at the same time by incorporating the segmentation task into the classification task. They used a set of entity labels that are augmented with a set of segment labels. We pointed out that this approach makes a feature space very sparse and proposed a new feature generation method to overcome this problem. The proposed method incorporates multiple segment label sets into a single model as feature functions. By utilizing both complex and general segment label sets within a single model, it can exploit not only sophisti-

cated features capturing the characteristics of entity words appearing at specific positions but also robust features that are not much sensitive to the positions of entity words. In the experiment, we demonstrated that the proposed method consistently improved the performance of baseline NER systems.

Chapter 4 deals with the problem of combinatorial features, especially syntactic contexts. While syntactic contexts seem to be very informative at first glance, these contexts obtained from a small manually annotated data are too sparse to be effective in most cases. On the other hand, generalizing these contexts result in highly ambiguous contexts that barely improve recognition performance. To overcome these problems, we take advantage of both feature engineering and semi-supervised approaches. We present the concept of a new resource, a context gazetteer, which comprises a large number of context patterns; therefore, we can use it in the form of a gazetteer without overly simplifying them. We also propose a method that automatically generates a context gazetteer from a certain type of unlabeled data, encyclopedic database, similar to semi-supervised approach. However, the proposed method is based on a single pass algorithm, which performs annotation and extraction processes only once, to avoid the problem of the semantic drift and annotation noise. The experiment results show that an NER model utilizing a context gazetteer improves both precision and recall compared to state-of-the-art NER models.

In conclusion, this thesis presented two novel methods for dealing with the data-sparseness problem. Considering that the size of training data is always limited, the proposed methods will be valuable techniques that can better utilize these data.

5.2 Future Work

There were several problems that we faced in the course of research but could not solved yet. The first proposed method, which utilizes multiple segment label

sets, inevitably increases the number of features and slows down training speed. Although we use a simple feature selection method that eliminates the features that occur only once in training data, the number of features is still very large and the training of the most complex model takes about eight times longer than the best conventional model. Therefore, it is necessary to devise an effective feature selection method to make the proposed method more practical.

In the second study, a context gazetteer has been proved as a useful resource in this thesis. However, we found that its coverage is relatively lower than our expectation. Currently, we use context patterns consists three elements: words, dependency labels between words, and POS-tags of words. While examining context patterns, we found that some part of context patterns need to be generalized. For example, nouns and verbs can be normalized into their singular forms without much problem. A little more complicated context patterns often involve a part of a target entity mention as in "... interaction between X and C-Jun..." Therefore, the next step of this study is to investigate how to generalize context patterns and develop appropriate methods.

In addition to solving remaining issues, there are different NLP tasks that can take advantage of using multiple SRs. For example, previous studies in the word segmentation and shallow parsing tasks mostly use the best SR that is empirically chosen for a given corpus or integrates the outputs of multiple models using different SRs in a pipe-line system. The proposed method can provide greater benefit to these tasks than these previous approaches.

While conducting the second study, we found that NER and relation extraction, which are mostly tackled by a pipe-line architecture, are tightly related. In the current pipe-line architecture, failing to recognize an entity mention X in the NER stage cannot be recovered in the RE stage. However, many relation-like context patterns such as "... interaction between X and C-Jun..." and "... association of X with p92 ..." suggest that identifying an entity mention with informative contexts around it can be used to solve this problem. Therefore, a unified framework for

these two tasks will benefit each other.

References

- [1] Beatrice Alex, Claire Grover, Barry Haddow, Mijail Kabadjor, Ewan Klein, Michael Matthews, Stuart Roebuck, Richard Tobin, and Xinglong Wang. Assisted curation: Does text mining really help?. In *Pacific Symposium on Biocomputing*, volume 13, pages 556–567. Citeseer, 2008.
- [2] Enrique Alfonseca and Suresh Manandhar. An unsupervised method for general named entity recognition and automated concept discovery. In *Proceedings of the 1st International Conference on General WordNet, Mysore, India*, pages 34–43, 2002.
- [3] Stephen F Altschul, Warren Gish, Webb Miller, Eugene W Myers, and David J Lipman. Basic local alignment search tool. *Journal of molecular biology*, 215(3):403–410, 1990.
- [4] Douglas E Appelt, Jerry R Hobbs, John Bear, David Israel, Megumi Kameyama, David Martin, Karen Myers, and Mabry Tyson. Sri international fastus system: Muc-6 test results and analysis. In *Proceedings of the 6th conference on Message understanding*, pages 237–248. Association for Computational Linguistics, 1995.
- [5] Masayuki Asahara and Yuji 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-Volume 1*, pages 8–15. Association for Computational Linguistics, 2003.
- [6] Michele Banko and Eric Brill. Mitigating the paucity-of-data problem: ex-

- ploring the effect of training corpus size on classifier performance for natural language processing. In *Proceedings of the first international conference on Human language technology research*, pages 1–5. Association for Computational Linguistics, 2001.
- [7] Eckhard Bick. A named entity recognizer for danish. In *Language Resources and Evaluation Conference*, 2004.
- [8] Daniel M Bikel, Scott Miller, Richard Schwartz, and Ralph Weischedel. Nymble: a high-performance learning name-finder. In *Proceedings of the fifth conference on Applied natural language processing*, pages 194–201. Association for Computational Linguistics, 1997.
- [9] Olivier Bodenreider. The unified medical language system (umls): integrating biomedical terminology. *Journal of Nucleic Acids Research*, 32(suppl 1): D267–D270, 2004. doi: 10.1093/nar/gkh061.
- [10] Andrew Borthwick, John Sterling, Eugene Agichtein, and Ralph Grishman. Nyu: Description of the mene named entity system as used in muc-7. In *Proceedings of the Seventh Message Understanding Conference (MUC-7)*, 1998.
- [11] Eric Brill. A simple rule-based part of speech tagger. In *Proceedings of the third conference on Applied natural language processing*, ANLC '92, pages 152–155, Stroudsburg, PA, USA, 1992. Association for Computational Linguistics. doi: 10.3115/974499.974526. URL <http://dx.doi.org/10.3115/974499.974526>.
- [12] Peter F. Brown, Peter V. deSouza, Robert L. Mercer, Vincent J. Della Pietra, and Jenifer C. Lai. Class-based n-gram models of natural language. *Journal of Computational Linguistics*, 18(4):467–479, December 1992. ISSN 0891-2017.
- [13] Razvan Bunescu, Ruifang Ge, Rohit J Kate, Edward M Marcotte, Raymond J Mooney, Arun K Ramani, and Yuk Wah Wong. Comparative exper-

- iments on learning information extractors for proteins and their interactions. *Artificial intelligence in medicine*, 33(2):139–155, 2005.
- [14] Michael Chau, Jennifer J Xu, and Hsinchun Chen. Extracting meaningful entities from police narrative reports. In *Proceedings of the 2002 annual national conference on Digital government research*, pages 1–5. Digital Government Society of North America, 2002.
- [15] Hai Leong Chieu and Hwee Tou Ng. Named entity recognition with a maximum entropy approach. In *Proceedings of the Seventh Conference on Natural Language Learning (CoNLL-2003)*, pages 160–163, 2003. doi: 10.3115/1119176.1119199.
- [16] N. A. Chinchor. Overview of MUC-7/MET-2. In *Proceedings of the Seventh Message Understanding Conference (MUC7)*, April 1998.
- [17] Nancy Chinchor and Patricia Robinson. Muc-7 named entity task definition. In *Proceedings of the Seventh Conference on Message Understanding*, 1997.
- [18] Han-Cheol Cho, Naoaki Okazaki, and Kentaro Inui. Inducing context gazetteers from encyclopedic databases for named entity recognition. In *Advances in Knowledge Discovery and Data Mining (PAKDD2013)*, pages 378–389. Springer, 2013. doi: 10.1007/978-3-642-37453-1_31. URL http://link.springer.com/chapter/10.1007%2F978-3-642-37453-1_31#page-1.
- [19] Han-Cheol Cho, Naoaki Okazaki, Makoto Miwa, and Jun’ichi Tsujii. Named entity recognition with multiple segment representations. *Information Processing & Management*, 49(4):954–965, 2013. doi: 10.1016/j.ipm.2013.03.002. URL <http://www.sciencedirect.com/science/article/pii/S0306457313000368>.
- [20] Sam Coates-Stephens. The analysis and acquisition of proper names for the understanding of free text. *Computers and the Humanities*, 26(5-6):441–456, 1992.

- [21] Aaron M Cohen. Unsupervised gene/protein named entity normalization using automatically extracted dictionaries. In *Proceedings of the ACL-ISMB workshop on linking biological literature, ontologies and databases: Mining biological semantics*, pages 17–24. Association for Computational Linguistics, 2005.
- [22] Aaron M Cohen and William R Hersh. A survey of current work in biomedical text mining. *Briefings in bioinformatics*, 6(1):57–71, 2005.
- [23] William W Cohen, Pradeep D Ravikumar, Stephen E Fienberg, et al. A comparison of string distance metrics for name-matching tasks. In *IJWeb*, volume 2003, pages 73–78, 2003.
- [24] Michael 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. Association for Computational Linguistics, 2002.
- [25] Michael Collins and Yoram Singer. Unsupervised models for named entity classification. In *Proceedings of the joint SIGDAT conference on empirical methods in natural language processing and very large corpora*, pages 189–196, 1999.
- [26] The UniProt Consortium. Reorganizing the protein space at the universal protein resource (uniprot). *Journal of Nucleic Acids Research*, 40(D1):D71–D75, 2012. doi: 10.1093/nar/gkr981.
- [27] Alessandro Cucchiarelli and Paola Velardi. Unsupervised named entity recognition using syntactic and semantic contextual evidence. *Computational Linguistics*, 27(1):123–131, 2001.
- [28] Silviu Cucerzan and David Yarowsky. Language independent named entity recognition combining morphological and contextual evidence. In *Proceedings of the 1999 Joint SIGDAT Conference on EMNLP and VLC*, pages 90–99, 1999.

- [29] James R Curran, Tara Murphy, and Bernhard Scholz. Minimising semantic drift with mutual exclusion bootstrapping. In *Proceedings of the 10th Conference of the Pacific Association for Computational Linguistics*, pages 172–180, 2007.
- [30] Joaquim Ferreira Da Silva, Zornitsa Kozareva, and José Gabriel Pereira Lopes. Cluster analysis and classification of named entities. In *Proceedings of Conference on Language Resources and Evaluation*, 2004.
- [31] Hoa Trang Dang, Diane Kelly, and Jimmy J Lin. Overview of the trec 2007 question answering track. In *TREC*, volume 7, page 63. Citeseer, 2007.
- [32] Marie-Catherine De Marneffe, Bill MacCartney, and Christopher D Manning. Generating typed dependency parses from phrase structure parses. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC)*, volume 6, pages 449–454, 2006.
- [33] George R Doddington, Alexis Mitchell, Mark A Przybocki, Lance A Ramshaw, Stephanie Strassel, and Ralph M Weischedel. The automatic content extraction (ace) program-tasks, data, and evaluation. In *Proceedings of Conference on Language Resources and Evaluation*, 2004.
- [34] Rezarta Islamaj Doğan and Zhiyong Lu. An improved corpus of disease mentions in pubmed citations. In *Proceedings of the 2012 Workshop on Biomedical Natural Language Processing*, pages 91–99. Association for Computational Linguistics, 2012.
- [35] Oren Etzioni, Michael Cafarella, Doug Downey, Ana-Maria Popescu, Tal Shaked, Stephen Soderland, Daniel S Weld, and Alexander Yates. Unsupervised named-entity extraction from the web: An experimental study. *Artificial Intelligence*, 165(1):91–134, 2005.
- [36] Richard Evans and Stafford Street. A framework for named entity recognition in the open domain. *AMSTERDAM STUDIES IN THE THEORY AND HISTORY OF LINGUISTIC SCIENCE*, page 267, 2004.

- [37] Dimitra Farmakiotou, Vangelis Karkaletsis, John Koutsias, George Sigletos, Constantine D Spyropoulos, and Panagiotis Stamatopoulos. Rule-based named entity recognition for greek financial texts. In *Proc. of the Workshop on Computational lexicography and Multimedia Dictionaries (COMLEX 2000)*, pages 75–78. Citeseer, 2000.
- [38] Jenny Finkel, Shipra Dingare, Huy Nguyen, Malvina Nissim, Christopher Manning, and Gail Sinclair. Exploiting context for biomedical entity recognition: from syntax to the web. In *Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and its Applications (JNLPBA)*, pages 88–91, 2004.
- [39] Jenny Rose Finkel, Trond Grenager, and Christopher Manning. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics (ACL)*, pages 363–370, 2005. doi: 10.3115/1219840.1219885.
- [40] David Fisher, Stephen Soderland, Fangfang Feng, and Wendy Lehnert. Description of the umass system as used for muc-6. In *Proceedings of the 6th conference on Message understanding*, pages 127–140. Association for Computational Linguistics, 1995.
- [41] Radu Florian, Abe Ittycheriah, Hongyan Jing, and Tong Zhang. Named entity recognition through classifier combination. In *Proceedings of the Seventh Conference on Natural Language Learning (CoNLL) at HLT-NAACL 2003 - Volume 4*, pages 168–171, 2003. doi: 10.3115/1119176.1119201.
- [42] Jordi Fortuny and Bernat Corominas-Murtra. On the origin of ambiguity in efficient communication. *Journal of Logic, Language and Information*, 22(3):249–267, 2013. ISSN 0925-8531. doi: 10.1007/s10849-013-9179-3. URL <http://dx.doi.org/10.1007/s10849-013-9179-3>.
- [43] Ken-ichiro Fukuda, Tatsuhiko Tsunoda, Ayuchi Tamura, Toshihisa Takagi, et al. Toward information extraction: identifying protein names from biological papers. In *Pac Symp Biocomput*, pages 707–718, 1998.

- [44] Robert Gaizauskas, Kevin Humphreys, Hamish Cunningham, and Yorick Wilks. University of sheffield: description of the lasie system as used for muc-6. In *Proceedings of the 6th conference on Message understanding*, pages 207–220. Association for Computational Linguistics, 1995.
- [45] Martin Gerner, Goran Nenadic, and Casey M Bergman. Linnaeus: a species name identification system for biomedical literature. *BMC bioinformatics*, 11(1):85, 2010.
- [46] Ralph Grishman and Beth Sundheim. Message understanding conference-6: A brief history. In *Proceedings of International Conference on Computational Linguistics*, volume 96, pages 466–471, 1996.
- [47] Bertram M. Gross. *The Managing of Organizations: The Administrative Struggle*. The Free Press New York 1964, 1964.
- [48] Marti A Hearst. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the 14th conference on Computational linguistics-Volume 2*, pages 539–545. Association for Computational Linguistics, 1992.
- [49] Lynette Hirschman and Robert Gaizauskas. Natural language question answering: The view from here. *Natural Language Engineering*, 7(4):275–300, 2001.
- [50] Lynette Hirschman, Alexander Yeh, Christian Blaschke, and Alfonso Valencia. Overview of biocreative: critical assessment of information extraction for biology. *BMC bioinformatics*, 6(Suppl 1):S1, 2005.
- [51] C.N. Hsu, Y.M. Chang, C.J. Kuo, Y.S. Lin, H.S. Huang, and I.F. Chung. Integrating high dimensional bi-directional parsing models for gene mention tagging. *Journal of Bioinformatics*, 24(13):i286–i294, 2008.
- [52] Hideki Isozaki and Hideto Kazawa. Efficient support vector classifiers for named entity recognition. In *Proceedings of the 19th international conference on Computational linguistics - Volume 1*, COLING '02, pages 1–7, Stroudsburg, PA, USA, 2002. Association for Computational Linguistics.

tics. doi: 10.3115/1072228.1072282. URL <http://dx.doi.org/10.3115/1072228.1072282>.

- [53] Mark Przybocki Jonathan, Jonathan G. Fiscus, John S. Garofolo, and David S. Pallett. Hub-4 information extraction evaluation. In *Proceedings of the DARPA Broadcast News Workshop*, pages 13–18. Morgan Kaufmann, 1999.
- [54] Nanda Kambhatla. Minority vote: at-least-n voting improves recall for extracting relations. In *Proceedings of the COLING/ACL on the main conference poster sessions*, pages 460–466, 2006.
- [55] Jun’ichi Kazama and Kentaro Torisawa. Exploiting wikipedia as external knowledge for named entity recognition. In *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing (EMNLP) and Computational Natural Language Learning (CoNLL)*, pages 698–707, 2007.
- [56] Jun’ichi Kazama and Kentaro Torisawa. Inducing gazetteers for named entity recognition by large-scale clustering of dependency relations. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics (ACL): Human Language Technologies (HLT)*, pages 407–415, 2008.
- [57] Jun’ichi Kazama, Takaki Makino, Yoshihiro Ohta, and Jun’ichi Tsujii. Tuning support vector machines for biomedical named entity recognition. In *Proceedings of the ACL-02 workshop on Natural language processing in the biomedical domain - Volume 3*, BioMed ’02, pages 1–8, Stroudsburg, PA, USA, 2002. Association for Computational Linguistics. doi: 10.3115/1118149.1118150. URL <http://dx.doi.org/10.3115/1118149.1118150>.
- [58] J.-D. Kim, T. Ohta, Y. Tateisi, and J. Tsujii. Genia corpus—a semantically annotated corpus for bio-textmining. *Bioinformatics*, 19(suppl 1):i180–i182, 2003. doi: 10.1093/bioinformatics/btg1023. URL http://bioinformatics.oxfordjournals.org/content/19/suppl_1/i180.abstract.

- [59] Jin-Dong Kim, Tomoko Ohta, Yoshimasa Tsuruoka, Yuka Tateisi, and Nigel Collier. 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, JNLPBA '04*, pages 70–75, Stroudsburg, PA, USA, 2004. Association for Computational Linguistics. URL <http://dl.acm.org/citation.cfm?id=1567594.1567610>.
- [60] Jin-Dong Kim, Sampo Pyysalo, Tomoko Ohta, Robert Bossy, Ngan Nguyen, and Jun'ichi Tsujii. Overview of bionlp shared task 2011. In *Proceedings of the BioNLP Shared Task 2011 Workshop*, pages 1–6, 2011. ISBN 9781937284091.
- [61] Seonho Kim and Juntae Yoon. Experimental study on a two phase method for biomedical named entity recognition. *IEICE - Trans. Inf. Syst.*, E90-D(7):1103–1110, July 2007. ISSN 0916-8532. doi: 10.1093/ietisy/e90-d.7.1103. URL <http://dx.doi.org/10.1093/ietisy/e90-d.7.1103>.
- [62] Ioannis P Klapaftis and Suresh Manandhar. Unsupervised named entity resolution. In *Proc of the 3rd IEEE International Conference on Multimedia Com-munications, Services and Security. Krakow, Poland*, pages 1–6, 2010.
- [63] Corinna Kolárik, Roman Klinger, Christoph M Friedrich, Martin Hofmann-Apitius, and Juliane Fluck. Chemical names: terminological resources and corpora annotation. In *Workshop on Building and evaluating resources for biomedical text mining (6th edition of the Language Resources and Evaluation Conference)*, volume 36, 2008.
- [64] Michael Krauthammer, Andrey Rzhetsky, Pavel Morozov, and Carol Friedman. Using blast for identifying gene and protein names in journal articles. *Gene*, 259(1):245–252, 2000.
- [65] Saul A Kripke. *Naming and necessity*. Springer, 1973.
- [66] Vijay Krishnan and Christopher D. Manning. An effective two-stage model for exploiting non-local dependencies in named entity recognition. In *Pro-*

- ceedings of the 21st International Conference on Coling and the 44th annual meeting of the ACL*, pages 1121–1128, 2006. doi: 10.3115/1220175.1220316.
- [67] George R Krupka and Kevin Hausman. Description of the netowlTM extractor system as used for muc-7. In *Proceedings of the Seventh Message Understanding Conference (MUC-7)*, 1998.
- [68] Taku Kudo and Yuji Matsumoto. Chunking with support vector machines. In *Proceedings of the second meeting of the North American Chapter of the Association for Computational Linguistics (NAACL) on Language technologies*, pages 1–8, 2001.
- [69] Seth Kulick, Ann Bies, Mark Liberman, Mark Mandel, Ryan McDonald, Martha Palmer, Andrew Schein, Lyle Ungar, Scott Winters, and Pete White. Integrated annotation for biomedical information extraction. In *Proc. of the Human Language Technology Conference and the Annual Meeting of the North American Chapter of the Association for Computational Linguistics (HLT/NAACL)*, pages 61–68, 2004.
- [70] John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning (ICML)*, pages 282–289, 2001. ISBN 1-55860-778-1.
- [71] Robert Leaman and Graciela Gonzalez. Banner: an executable survey of advances in biomedical named entity recognition. *Journal of Pacific Symposium on Biocomputing*, pages 652–663, 2008.
- [72] Robert Leaman, C Miller, and G Gonzalez. Enabling recognition of diseases in biomedical text with machine learning: corpus and benchmark. In *Proceedings of the 2009 Symposium on Languages in Biology and Medicine*, 2009.
- [73] Ki-Joong Lee, Young-Sook Hwang, Seonho Kim, and Hae-Chang Rim. Biomedical named entity recognition using two-phase model based on svms. *Journal of Biomedical Informatics*, 37(6):436–447, 2004.

- [74] Lishuang Li, Rongpeng Zhou, and Degen Huang. Two-phase biomedical named entity recognition using {CRFs}. *Computational Biology and Chemistry*, 33(4):334 – 338, 2009. ISSN 1476-9271. doi: 10.1016/j.compbiolchem.2009.07.004. URL <http://www.sciencedirect.com/science/article/pii/S1476927109000590>.
- [75] Y. Li, H. Lin, and Z. Yang. Incorporating rich background knowledge for gene named entity classification and recognition. *Journal of BMC bioinformatics*, 10(1):223, 2009.
- [76] Dekang Lin. Automatic retrieval and clustering of similar words. In *Proceedings of the 17th international conference on Computational linguistics-Volume 2*, pages 768–774. Association for Computational Linguistics, 1998.
- [77] Zhiyong Lu, Hung-Yu Kao, Chih-Hsuan Wei, Minlie Huang, Jingchen Liu, Cheng-Ju Kuo, Chun-Nan Hsu, Richard Tsai, Hong-Jie Dai, Naoaki Okazaki, et al. The gene normalization task in biocreative iii. *BMC bioinformatics*, 12(Suppl 8):S2, 2011.
- [78] Peter Lyman and Hal Varian. How much information 2003? UC Berkeley School of Information Management and Systems, 2004. URL <http://www2.sims.berkeley.edu/research/projects/how-much-info-2003/index.htm>.
- [79] Donna Maglott, Jim Ostell, Kim D. Pruitt, and Tatiana Tatusova. Entrez gene: Gene-centered information at ncbi. *Journal of Nucleic Acids Research*, 33(suppl 1):D54–D58, 2005. doi: 10.1093/nar/gki031.
- [80] Xinnian Mao, Wei Xu, Yuan Dong, Saike He, and Haila Wang. Using non-local features to improve named entity recognition recall. In *Proceedings of the 21th Pacific Asia Conference on Language, Information and Computation*, pages 303–310, 2007.
- [81] James Mayfield, Paul McNamee, and Christine Piatko. Named entity recognition using hundreds of thousands of features. In *Proceedings of the seventh*

conference on Natural language learning at HLT-NAACL 2003-Volume 4, pages 184–187. Association for Computational Linguistics, 2003.

- [82] Andrew McCallum and Wei Li. Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003 - Volume 4*, CONLL '03, pages 188–191, Stroudsburg, PA, USA, 2003. Association for Computational Linguistics. doi: 10.3115/1119176.1119206. URL <http://dx.doi.org/10.3115/1119176.1119206>.
- [83] Andrei Mikheev. A knowledge-free method for capitalized word disambiguation. In *Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics*, pages 159–166. Association for Computational Linguistics, 1999.
- [84] Scott Miller, Jethran Guinness, and Alex Zamanian. Name tagging with word clusters and discriminative training. In Daniel Marcu Susan Dumais and Salim Roukos, editors, *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL*, pages 337–342, Boston, Massachusetts, USA, May 2 - May 7 2004. Association for Computational Linguistics.
- [85] Tomohiro Mitsumori, Sevrani Fation, Masaki Murata, Kouichi Doi, and Hirohumi Doi. Gene/protein name recognition based on support vector machine using dictionary as features. *BMC bioinformatics*, 6(Suppl 1):S8, 2005.
- [86] Alvaro E Monge, Charles Elkan, et al. The field matching problem: Algorithms and applications. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, pages 267–270, 1996.
- [87] David Nadeau. *Semi-Supervised Named Entity Recognition*. PhD thesis, University of Ottawa, 2007.
- [88] David Nadeau and Satoshi Sekine. A survey of named entity recognition and classification. *Journal of Linguisticae Investigationes*, 30:3–26, 2007.

- [89] Gonzalo Navarro, Ricardo Baeza-Yates, and João Marcelo Azevedo Arcoverde. Matchsimile: a flexible approximate matching tool for searching proper names. *Journal of the American society for Information Science and Technology*, 54(1):3–15, 2003.
- [90] Mariana Neves, Alexander Damaschun, Andreas Kurtz, and Ulf Leser. Annotating and evaluating text for stem cell research. In *Third Workshop on Building and Evaluation Resources for Biomedical Text Mining (BioTxtM 2012)*.(to appear), 2012.
- [91] Tomoko Ohta, Sampo Pyysalo, Makoto Miwa, Jin-Dong Kim, and Jun’ichi Tsujii. Event extraction for post-translational modifications. In *Proceedings of the 2010 Workshop on Biomedical Natural Language Processing*, pages 19–27. Association for Computational Linguistics, 2010.
- [92] Tomoko Ohta, Sampo Pyysalo, Makoto Miwa, and Jun’ichi Tsujii. Event extraction for dna methylation. In *Semantic Mining in Biomedicine*, 2010.
- [93] Tomoko Ohta, Sampo Pyysalo, and Jun’ichi Tsujii. From pathways to biomolecular events: opportunities and challenges. In *Proceedings of BioNLP 2011 Workshop*, pages 105–113. Association for Computational Linguistics, 2011.
- [94] Tomoko Ohta, Sampo Pyysalo, Jun’ichi Tsujii, and Sophia Ananiadou. Open-domain anatomical entity mention detection. In *Proceedings of the Workshop on Detecting Structure in Scholarly Discourse*, pages 27–36. Association for Computational Linguistics, 2012.
- [95] Naoaki Okazaki. Crfsuite: a fast implementation of conditional random fields (crfs), 2007. URL <http://www.chokkan.org/software/crfsuite/>.
- [96] Marius Pasca, Dekang Lin, Jeffrey Bigham, Andrei Lifchits, and Alpa Jain. Organizing and searching the world wide web of facts-step one: the one-million fact extraction challenge. In *AAAI*, volume 6, pages 1400–1405, 2006.

- [97] Jon Patrick, Casey Whitelaw, and Robert Munro. Slinerc: The sydney language-independent named entity recogniser and classifier. In *proceedings of the 6th conference on Natural language learning-Volume 20*, pages 1–4. Association for Computational Linguistics, 2002.
- [98] Sampo Pyysalo, Filip Ginter, Juho Heimonen, Jari Björne, Jorma Boberg, Jouni Järvinen, and Tapio Salakoski. Bioinfer: a corpus for information extraction in the biomedical domain. *BMC bioinformatics*, 8(1):50, 2007.
- [99] Sampo Pyysalo, Tomoko Ohta, Han-Cheol Cho, Dan Sullivan, Chunhong Mao, Bruno Sobral, Jun’ichi Tsujii, and Sophia Ananiadou. Towards event extraction from full texts on infectious diseases. In *Proceedings of the 2010 Workshop on Biomedical Natural Language Processing*, pages 132–140. Association for Computational Linguistics, 2010.
- [100] Lev Ratinov and Dan Roth. Design challenges and misconceptions in named entity recognition. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL)*, pages 147–155, 2009. ISBN 978-1-932432-29-9.
- [101] Lisa F Rau. Extracting company names from text. In *Proceedings of Conference on Artificial Intelligence Applications of IEEE*, volume 1, pages 29–32. IEEE, 1991.
- [102] Yael Ravin and Nina Wacholder. *Extracting names from natural-language text*. Citeseer, 1997.
- [103] Dietrich Rebholz-Schuhmann, Antonio Yepes, Chen Li, Senay Kafkas, Ian Lewin, Ning Kang, Peter Corbett, David Milward, Ekaterina Buyko, Elena Beisswanger, Kerstin Hornbostel, Alexandre Kouznetsov, René Witte, JonasB Laurila, ChristopherJO Baker, Cheng-Ju Kuo, Simone Clematide, Fabio Rinaldi, Richárd Farkas, György Móra, Kazuo Hara, LauraI Furlong, Michael Rautschka, Mariana Neves, Alberto Pascual-Montano, Qi Wei, Nigel Collier, Md Chowdhury, Alberto Lavelli, Rafael Berlanga, Roser Morante, Vincent Van Asch, Walter Daelemans, José Marina, Erik van

- Mulligen, Jan Kors, and Udo Hahn. Assessment of ner solutions against the first and second calbc silver standard corpus. *Journal of Biomedical Semantics*, 2(5):1–12, 2011. doi: 10.1186/2041-1480-2-S5-S11. URL <http://dx.doi.org/10.1186/2041-1480-2-S5-S11>.
- [104] Kashif Riaz. Rule-based named entity recognition in urdu. In *Proceedings of the 2010 Named Entities Workshop*, pages 126–135. Association for Computational Linguistics, 2010.
- [105] Ellen Riloff and Rosie Jones. Learning dictionaries for information extraction by multi-level bootstrapping. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications Proceedings of the National Conference on Artificial Intelligence and Conference on Innovative Applications of Artificial Intelligence (AAAI/IAAI)*, pages 474–479, 1999.
- [106] Ellen Riloff and Jessica Shepherd. A corpus-based approach for building semantic lexicons. In *Proceedings of the Second Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 117–124, 1997.
- [107] Erik F. Tjong Kim Sang and Jorn Veenstra. Representing text chunks. In *Proceedings of the Ninth Conference on European Chapter of the Association for Computational Linguistics (EACL)*, pages 173–179, 1999.
- [108] Diana Santos, Nuno Seco, Nuno Cardoso, and Rui Vilela. Harem: An advanced ner evaluation contest for portuguese. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC)*, pages 22–28, 2006.
- [109] Sunita Sarawagi and William W. Cohen. Semi-markov conditional random fields for information extraction. In *In Advances in Neural Information Processing Systems 17*, pages 1185–1192, 2004.
- [110] Satoshi Sekine and Hitoshi Isahara. Irex: Ir and ie evaluation project in japanese. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC)*, pages 1977–1980, 2000.

- [111] Satoshi Sekine et al. Nyu: Description of the japanese ne system used for met-2. In *Proc. of the Seventh Message Understanding Conference (MUC-7)*, volume 17, 1998.
- [112] Burr Settles. Biomedical named entity recognition using conditional random fields and rich feature sets. In *Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and its Applications, JNLPBA '04*, pages 104–107, Stroudsburg, PA, USA, 2004. Association for Computational Linguistics. URL <http://dl.acm.org/citation.cfm?id=1567594.1567618>.
- [113] Burr Settles and Mark Craven. An analysis of active learning strategies for sequence labeling tasks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1070–1079. Association for Computational Linguistics, 2008.
- [114] Matthew S Simpson and Dina Demner-Fushman. Biomedical text mining: A survey of recent progress. In *Mining Text Data*, pages 465–517. Springer, 2012.
- [115] Larry Smith, Lorraine Tanabe, Rie Ando, Cheng-Ju Kuo, I-Fang Chung, Chun-Nan Hsu, Yu-Shi Lin, Roman Klinger, Christoph Friedrich, Kuzman Ganchev, Manabu Torii, Hongfang Liu, Barry Haddow, Craig Struble, Richard Povinelli, Andreas Vlachos, William Baumgartner, Lawrence Hunter, Bob Carpenter, Richard Tsai, Hong-Jie Dai, Feng Liu, Yifei Chen, Chengjie Sun, Sophia Katrenko, Pieter Adriaans, Christian Blaschke, Rafael Torres, Mariana Neves, Preslav Nakov, Anna Divoli, Manuel Mana-Lopez, Jacinto Mata, and W John Wilbur. Overview of biocreative ii gene mention recognition. *Journal of Genome Biology*, 9(Suppl 2):S2, 2008. ISSN 1465-6906. doi: 10.1186/gb-2008-9-s2-s2.
- [116] Larry H. Smith and W. John Wilbur. The value of parsing as feature generation for gene mention recognition. *Journal of Biomedical Informatics*, 42(5):895–904, October 2009. ISSN 1532-0464. doi: 10.1016/j.jbi.2009.03.011.

- [117] Wee Meng Soon, Hwee Tou Ng, and Daniel Chung Yong Lim. A machine learning approach to coreference resolution of noun phrases. *Computational linguistics*, 27(4):521–544, 2001.
- [118] Pontus Stenetorp, Sampo Pyysalo, Goran Topić, Tomoko Ohta, Sophia Ananiadou, and Jun’ichi Tsujii. Brat: a web-based tool for nlp-assisted text annotation. In *Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 102–107. Association for Computational Linguistics, 2012.
- [119] Beth M Sundheim. Overview of results of the muc-6 evaluation. In *Proceedings of a workshop on held at Vienna, Virginia: May 6-8, 1996*, pages 423–442. Association for Computational Linguistics, 1996.
- [120] Charles Sutton and Andrew McCallum. An introduction to conditional random fields for relational learning. In Lise Getoor and Ben Taskar, editors, *Introduction to Statistical Relational Learning*. MIT Press, 2007.
- [121] Koichi Takeuchi and Nigel Collier. Bio-medical entity extraction using support vector machines. *Artificial Intelligence in Medicine*, 33(2):125–137, 2005.
- [122] Lorraine Tanabe and W. John Wilbur. Tagging gene and protein names in biomedical text. *Bioinformatics*, 18(8):1124–1132, 2002. doi: 10.1093/bioinformatics/18.8.1124. URL <http://bioinformatics.oxfordjournals.org/content/18/8/1124.abstract>.
- [123] Lorraine Tanabe and W John Wilbur. Tagging gene and protein names in full text articles. In *Proceedings of the ACL-02 workshop on Natural language processing in the biomedical domain-Volume 3*, pages 9–13. Association for Computational Linguistics, 2002.
- [124] Lorraine Tanabe, Natalie Xie, Lynne Thom, Wayne Matten, and W John Wilbur. Genetag: a tagged corpus for gene/protein named entity recognition. *Journal of BMC Bioinformatics*, 6(Suppl 1):S3, 2005. ISSN 1471-2105. doi: 10.1186/1471-2105-6-S1-S3.

- [125] Erik F. Tjong Kim Sang. Introduction to the conll-2002 shared task: language-independent named entity recognition. In *Proceedings of the 6th Conference on Natural Language Learning - Volume 20*, COLING-02, pages 1–4, Stroudsburg, PA, USA, 2002. Association for Computational Linguistics. doi: 10.3115/1118853.1118877. URL <http://dx.doi.org/10.3115/1118853.1118877>.
- [126] Erik F. Tjong Kim Sang and Fien De Meulder. Introduction to the conll-2003 shared task: language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning (CoNLL) at HLT-NAACL 2003 - Volume 4*, pages 142–147, 2003.
- [127] Kristina Toutanova, Dan Klein, Christopher D. Manning, and Yoram Singer. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL) on Human Language Technology (HLT) - Volume 1*, pages 173–180, 2003. doi: 10.3115/1073445.1073478.
- [128] Yoshimasa Tsuruoka and Jun’ichi Tsujii. Bidirectional inference with the easiest-first strategy for tagging sequence data. In *Proceedings of the Conference on Human Language Technology (HLT) and Empirical Methods in Natural Language Processing (EMNLP)*, pages 467–474, 2005. doi: 10.3115/1220575.1220634.
- [129] Yu Usami, Han-Cheol Cho, Naoaki Okazaki, and Jun’ichi Tsujii. Automatic acquisition of huge training data for bio-medical named entity recognition. In *Proceedings of BioNLP 2011 Workshop*, pages 65–73, 2011. ISBN 978-1-932432-91-6.
- [130] A Vlachos. Tackling the biocreative2 gene mention task with conditional random fields and syntactic parsing. *Proceedings of the Second BioCreative Challenge Evaluation Workshop; 23 to 25 April 2007; Madrid, Spain*, pages 85–87, 2007.

- [131] Francis Wolinski, Frantz Vichot, and Bruno Dillet. Automatic processing of proper names in texts. In *Proceedings of the seventh conference on European chapter of the Association for Computational Linguistics*, pages 23–30. Morgan Kaufmann Publishers Inc., 1995.
- [132] Nianwen Xue. Chinese word segmentation as character tagging. *Journal of International Journal of Computational Linguistics and Chinese*, 2003.
- [133] Kaoru Yamamoto, Taku Kudo, Akihiko Konagaya, and Yuji Matsumoto. Protein name tagging for biomedical annotation in text. In *Proceedings of the ACL 2003 workshop on Natural language processing in biomedicine-Volume 13*, pages 65–72. Association for Computational Linguistics, 2003.
- [134] Zhihao Yang, Hongfei Lin, and Yanpeng Li. Exploiting the contextual cues for bio-entity name recognition in biomedical literature. *Journal of biomedical informatics*, 41(4):580–587, 2008.
- [135] Roman Yangarber, Winston Lin, and Ralph Grishman. Unsupervised learning of generalized names. In *Proceedings of the 19th international conference on Computational linguistics-Volume 1*, pages 1–7. Association for Computational Linguistics, 2002.
- [136] Hai Zhao, Chang-Ning Huang, Mu Li, and Bao-Liang Lu. Effective tag set selection in chinese word segmentation via conditional random field modeling. In *Proceedings of the 20th Asian Pacific Conference on Language, Information and Computation*, pages 87–94, 2006.