AUTOMATIC EXTRACTION OF OUTBREAK INFORMATION FROM NEWS

BY

YI ZHANG
B.E., University of Science and Technology of China, 2002

THESIS

Submitted as partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Computer Science
in the Graduate College of the
University of Illinois at Chicago, 2008
Chicago, Illinois

This thesis is dedicated to my wife, Wei, without whom it would never have been accomplished.

ACKNOWLEDGMENTS

I would like to thank all people who have helped and inspired me during my doctoral study.

My deep gratitude to my advisors Prof. Bing Liu and Prof. Peter Nelson for their support and advising. I've learned not just data mining from them, but also the professional attitude and methodology in conducting research. I would also like to thank my other thesis committee, Prof. Natalia Maltsev, Prof. Clement Yu, and Prof. Philip Yu, for their assistance.

I want to thank everyone at bioinformatics group at Mathematical Computer Science Division in Argonne National Laboratory, Prof. Natalia Maltsev, Dr. Elizabeth M. Glass, Dr. Mark D'Souza, Syed H. Mustafa, Alexis Rodriguez, and Dinanath Sulake. I really enjoy the time working with all of you.

The staff and college students at National Center for Data Mining supported me in my research work, and I am obliged to David Turkington, Michal Sabala, Jia Chen, Xiaowen Ding, Feng Tian, and all other people in the lab.

YZ

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

TC	Text Categorization
EDR	Emergent Disease Report
NE	Named Entity
SVM	Support Vector Machine
NB	Naïve Bayes
CRF	Conditional Random Fields
KNN	K-Nearest Neighbors
TF-IDF	Term Frequency–Inverse Document Frequency
SPM	Sequential Pattern Mining
CSR	Class Sequential Rule
LSR	Label Sequential Rule
NLP	Natural Language Processing
POS	Part Of Speech
IE	Information Extraction
URL	Uniform Resource Locator
HTML	Hyper Text Markup Language
DOM	Document Object Model
HMM	Hidden Markov Model

SUMMARY

With the explosion of unstructured data on the Web, especially in the form of text, there has been a lot of interest to mine knowledge from these data for variety of purposes. In this thesis, we study a particular problem: how to extract disease outbreak information from news.

By defining Emergent Disease Report, we focus on extracting disease name and outbreak location from the news report emergent disease outbreaks. First, we study the problem how to classify those sentences reporting disease outbreak, and propose to a new method by integrating semantic features with the bag-of-words scheme. Experimental result shows the integrated approach is better than each individual approach alone. Second, a novel method based on sequential rules is introduced to extract the outbreak locations from the outbreak reporting sentences, and the new method outperforms conditional random fields in our experimental data. Finally, we discuss how to do classification and extraction together using label sequential rules and how to geocode the extracted location named entities into geographical locations accurately. Evaluations on classification-extraction including geocoding are conducted, and the proposed method is shown to improve the overall performance.

1 INTRODUCTION

1.1 Background

With the availability of massive news in the electronic format, the interest of automatic dealing with the news rises. For example, automatic categorization of news documents into different topics has been studied extensively in the last two decades (Sebastiani, 2002). This thesis studies a particular problem: automatically extracting emergent disease outbreaks from the news, i.e., we want to use computer to find emergent disease outbreak's time, location, and the disease name.

Technically, this task involves three main sub tasks. First, one needs to monitor the news streams constantly to identify articles that report disease outbreaks. This can be regarded as a classification problem, i.e., to classify each article as reporting a disease outbreak or not reporting a disease outbreak (two classes). Second, from each disease outbreak news article, one then extracts the name of the disease, the location and the time of the outbreak. For example, in the following sentence in a news report, "Four people were reported dead this morning from a cholera outbreak in Country X", we want to extract the disease name "cholera", the outbreak location "Country X", and the time "this morning" (which can be translated to an absolute date and time based on the press time).

Finally, the outbreak location words are mapped to geographical locations so that we can location the outbreak accurately.

All of the three sub tasks are very challenging problems. As we will discuss later, we find the first sub task very different from the traditional text classification (i.e., text categorization) problem. Traditional text classification studied in the information retrieval and machine learning literature is mainly based on topics, while our first sub task requires more refined classification on some semantic perspectives. We propose a novel technique to tackle this problem. The second sub task is an information extraction problem, and we introduce a new method based on sequential rules. For the third sub task, a greedy algorithm for geocoding and solutions for geocoding ambiguity are proposed. Finally, we give a mixed method that attempts to solve both sub tasks together. For each method, experiments based on large number of health news are conducted, and comparisons with existing other methods are reported.

1.2 Statement of the Problem

Suppose we are given a set of news documents D, we want to find information about any emergent disease outbreak from each piece of news d ($d \in D$).

An emergent disease outbreak, as it is named, is any disease outbreak happened in a recent short time period, so a flu outbreak happened threes years ago is clearly not emergent, and thus doe not belong to this category. The information we want to get about

an emergent disease outbreak includes: time and location of the outbreak, and name of the disease.

Time and disease name are self-explained. For the location, there are two levels of information. One level is the location word literally appearing in the news, such as “Cook County”, and the other level is the geographical location referred by the location words, such as “Cook County in Illinois” which is referred by the word “Cook County”. Notice that one location word could refer to multiple geographical locations. For example, for “Cook County”, besides the one in Illinois, it could also mean the one in Georgia, or the one in Minnesota, or even the one in New Zealand. We will discuss how to disambiguate location words to a single geographical location in Chapter 5. Location words are location named entities, and we will introduce several types of named entities that will be used throughout this thesis.

1.2.1 Named entity

A **Named Entity (NE)** is a word or a phrase that has a designated meaning, such as a location, the name of a person or the name of an organization. In this work, two types of named entities are of particular interest: Disease Named Entity and Location Named Entity.

1.2.1.1 Disease named entity

A **Disease Named Entity (Disease NE)** is a Named Entity of a disease. In this work, we are only interested in infectious diseases, so in this thesis, by any disease, we mean infectious disease. We gathered the names of infectious diseases and their alias from the Web (Centers for Disease Control and Prevention, 2008).

1.2.1.2 Location Named Entity

Location Named Entity (Location NE) is a Named Entity of a location. One location NE may correspond one or more than one geographical locations. Our definition of Location NE is slightly different from the traditional definition (Grishman and Sundheim, 1996; Sekine et al., 2002). For example, "Indonesia Health Department" is defined as an organization named entity as a whole in traditional definition, but in our definition, this is still defined as a Location NE, since the word "Indonesia" does refer a location. This does not mean that every organization named entity is considered as Location NE in our definition: only those organizations refer to a physical location are. In another example, "County Health Administer" is not a Location NE, since it has no specific location. More over, out definition of Location NE includes adjective forms of location words such as "Chinese" and "Australian".

In our application, each location named entity falls into one of the following two subtypes:

- **Emergent Disease Report Location Named Entity (EDR Location NE)** is a named entity of a location where a disease outbreak happened.
- **Non-Emergent Disease Report Location Named Entity (nonEDR Location NE)** is a location named entity, where no disease outbreak happened.

In the following example:

> Japan has temporarily halted its poultry imports because of a recent bird flu outbreak in South Korea.

The disease "bird flu" (which is also called "avian influenza") is a disease NE, the location "Japan" is a nonEDR location NE, and the location "South Korea" is an EDR location NE.

1.2.2 Emergent disease report news

Any news that reports an emergent disease outbreak is an **Emergent Disease Report News (EDR News)**.

Any news that do not reports any emergent disease outbreak is a **Non-Emergent Disease Report News (nonEDR News)**. A nonEDR News may talk about many different topics, such as disease research, previous outbreak investigation, and outbreak preparedness.

Note that any EDR News *must* contain at least one EDR Sentence, and any nonEDR News *must not* contain any EDR Sentence.

1.2.3 Emergent disease report sentences

Similar to EDR News, we defined EDR Sentence, i.e., we categorize sentences into two classes:

An **Emergent Disease Report Sentence (EDR Sentence)** is a sentence that reports an emergent disease outbreak.

A **Non-Emergent Disease Report Sentence (nonEDR Sentence)** is a sentence that does not report any emergent disease outbreak.

Not all sentences containing disease names are EDR sentences, as showed in the following nonEDR sentence:

> Researchers at the University of Texas Southwestern Medical Center in Dallas may have found a way to stop the transmission of HIV in women.

In an EDR sentence, it's not necessary that all Location NE are EDR Location NE. E.g., in the previous example in 1.2.1, the sentence is an EDR sentence, but there is also a nonEDR Location NE "Japan" in the sentence.

Sentence, instead of news, is usually the basic unit to report emergent disease outbreak. The reason is, an EDR news consisting of multiple sentences, often has some sentences reporting the outbreak, and some sentences giving other information, e.g.,

previous outbreaks, how to prepare for the disease. Thus this thesis will mainly investigate on sentence level in Chapter 3 and Chapter 4, and a mixed method will be introduced in Chapter 5.

Among nonEDR sentences, those related to disease outbreak are the most difficult to differentiate from EDR sentences. 1) Some nonEDR sentences talk about old outbreaks, and they are commonly seen after EDR sentences, as people give historic outbreak of a disease after reporting its recent outbreak. 2) Some other nonEDR sentences just give a hypothesis, and the outbreaks were not actually true. For example, "*Thirty million people could die if a human-to-human strain of bird flu spreads over the nation*". 3) Another type of nonEDR sentences negates disease outbreaks. It may announce the end of an outbreak, e.g., "*Vietnam announces the success over last month's bird flu outbreak*"; or it may deny any report of an outbreak, e.g., "*Iranian health officials have asserted that the news of a bird flu outbreak in northern and western parts of the country is baseless*" and "*Samples from a farm in south-eastern Romania have revealed no case of bird flu*". These sentences use similar words as EDR sentences do, and the traditional text classification methods based on bag of words do not work well on these sentences.

1.2.4 Outbreak time and disease name

Since we are extracting from real-time news, the time when an outbreak happens is approximately the same as the report date, under the assumption that the outbreak we

extracted is an *emergent* disease outbreak. In other words, we took the reporting date as the time of the outbreak, and will not extract time from news explicitly.

For disease name, there are some cases that an EDR news do not use any specific disease names, however, we observe only 1.4% news in our EDR news corpus do not have any specific disease name. After we regard general words (such as "disease", "illness", "outbreak") as Disease NE, this percentage further reduced to 0.2%. General disease words and the statistics will be discussed in detail in 5.2.1.

1.3 Significance of the Study

Although topics based text classification has been studied extensively, semantic based text classification remains an open problem. Existing work on semantic classification still focus on classification of whole text documents (news level in our case), using the 20-newsgroups (20 Newsgroups Dataset, 2008) and Reuters-21578 (Reuters-21578 Dataset, 2008), which are typical topic-based classification data sets. However, our task focuses on sentence level classification, which requires more semantic information, i.e. more delicate features from sentences based on the dependency tree (e.g., center noun, negation word, determiner, tense, etc). We also extract dates and treat them as features, which to our knowledge have not been done before for text classification.

Information extraction has also been studied by numerous researchers, but it remains to be a challenging problem. The novel method described in Chapter 4, uses

sequential rules on paths generated from dependency trees, and we will show experimental results that this method outperforms the state-of-the-art information extraction technique conditional random field.

Finally, in Chapter 5, we introduce a novel method which attempts to solve the classification problem and extraction problem together. We also discuss how to geocode the extracted location words into geographical locations accurately. Experiments on classification-extraction including geocoding are conducted, and the proposed method is shown to improve the overall performance.

1.4 Significance of the Problem

Extracting disease outbreaks is important with many applications. For example, it helps authorities control the spread of infectious diseases by travelers, planes and ships. It also enables health organizations to take preventive actions to alert citizens traveling to infected areas. The most up to date disease outbreak reports are usually from news articles around the world. Although it is possible to collect such reports manually by reading all the health related news from all over the world, it is a daunting task, highly labor intensive and time consuming. It is thus useful to develop automated techniques to extract such reports automatically, i.e., to find the time and the location of each disease outbreak. In fact, this thesis work is supported by the Great Lakes Protection Fund, and the automatically extracted outbreak information will be used by environmental agencies

to monitor infectious disease outbreak, so that appropriate measures can be taken when a ship having visited an infected area is going to enter Great Lakes.

1.5 Contribution of This Thesis

A number of novel techniques have been proposed in this work.

In Chapter 3, we introduced a technique for classification of emergent disease report sentences, and the new technique uses semantic feature derived from a sentence's dependency tree.

A novel method for extraction of disease outbreak location named entity is proposed in Chapter 4. The class sequential rules and label sequential rules used in the method are mined from disease-location paths derived from dependency tree.

In Chapter 5, we propose to use label sequential rules for both classification and extraction tasks.

In the same chapter, we formulize the geocoding problem for mapping a location named entity into geographical entries, and introduce a greedy algorithm for disambiguation of geocoding. We also discuss how to utilize the background location in disambiguation.

This thesis consists of six chapters. Chapter 2 reviews the related work, which includes text classification studied in data mining, information extraction studied in information retrieval, dependency tree studied in natural language processing, and several existing systems for public health monitoring. Our proposed new methods are given in Chapter 3, Chapter 4, and Chapter 5. Chapter 3 introduces the semantic classification algorithm for classification of EDR Sentences. Chapter 4 discusses a novel algorithm based on Class Sequential Rule and Label Sequential Rule to extract EDR Location NE from EDR Sentences, and introduction of Class Sequential Rule and Label Sequential Rule are given in the same chapter. In Chapter 5, we discuss how to do EDR Sentence classification and EDR Location NE extraction together, and we also investigate how to map the extracted EDR Location NE to geographical locations. Empirical results will be shown for each new method. Chapter 6 concludes the thesis and outlines directions for future work.

2 RELATED WORK

In this chapter, we review related work in text classification, information extraction, dependency tree that has been studied in data mining, natural language processing, and information retrieval, respectively. Several existing systems that monitor disease outbreaks are given and their cons and pros are discussed. The existing work on sequential rule, including label sequential rule and class sequential rule are given in Chapter 4.

2.1 Text Classification

Given a set of documents D, a set of classes C, and a test document d_{test}, such that each document d_i ($d_i \in D$) is labeled with a class c_j ($c_j \in C$), the goal of text classification is to predict the label of d_{test}. This is called supervised text classification since the label of every document in set D is known. There is also unsupervised text classification, where no labeled document is available. This thesis focus on supervised text classification only.

The traditional topic-based text classification, also called text categorization, has been studied extensively previously (Sebastiani, 2002; Yang and Liu, 1999). In these studies, the class set C contains coarse-grained topics such as politics, sports, and science.

2.1.1 Data representation

The supervised learning algorithm cannot take the text document directly as input. Usually, each text document d is represented by a vector $< u_{1,d}, u_{2,d}, \ldots\ldots, u_{m,d} >$, where m is the number of terms and $u_{i,d}$ is the weight of the ith term t_i.

2.1.1.1 Term weight function

The most common forms for the weights of the term are binary and TF-IDF.

In binary form,

$$u_{i,d} = \begin{cases} 0 & c(d,t_i) = 0 \\ 1 & c(d,t_i) > 0 \end{cases}$$

where $c(d,t_i)$ is the number of occurrences of term t_i in document d.

Term Frequency–Inverse Document Frequency (TF-IDF) is more popularly used because it accounts for how important a term is.

$$u_{i,d} = \frac{\sum_{d \in D} c(d,t_i)}{\sum_{d \in D,i} c(d,t_i)} \cdot \log \frac{|D|}{|\{d | d \in D, c(d,t_i) > 0\}|}$$

where on the right hand side of the equation, the first item is the term frequency and the second one is the inverse document frequency.

2.1.1.2 Term

A term is typically a word and hence this representation is called “bag-of-words”. The bag-of-words is often sufficient in text categorization because a topic can usually be characterized by a set of topic-specific keywords.

N-gram model takes the subsequence of n words and use them as a term. It is shown that while 2-gram and 3-gram improves the classification, longer gram reduces the performance (Fürnkranz, 1998).

Recently, There are several works on using linguistic information for text classification. Most of them are based on the idea of carefully choosing additional keywords or phrases. Lewis uses word phrases as terms and found this representation gives inferior classification result than word-based representation does (Lewis, 1992). Noun phrases have been used in text classification and higher precision but lower recall were reported (Furnkranz et al., 1998). Aizawa incorporated terms features extracted based on Part-Of-Speech tagging (Aizawa, 2001). The text classification performance was improved by using bag-of-concepts (Sahlgren and Coster, 2004). Complex nominals have been used as features (Moschitti and Basili, 2004). Classification results based on several keyword extraction methods, i.e., 1-grams, 2-grams, and 3-grams, noun phrase chunks, and frequently occurring POS patterns, are reported to give better result when these features are combined (Hulth and Megyesi, 2006).

Text summarization has been used to help classification. Assigning term weights based on the importance of each sentence, which is determined by a text summarization system, is observed to improve classification performance (Ko et al., 2004). Mihalcea and Hassan also used automatic extractive summarization in text classification, while their approach is to integrate a graph-based method for automatic summarization with a text classifier (Mihalcea and Hassan, 2005). Existing summaries of texts has also been used in text classification (Li et al., 2003).

2.1.2 Classifier

In supervised learning, a classifier builds the model from the training data. In text categorization, many classifiers have been used: i.e., Support Vector Machine (SVM), k-Nearest Neighbors (KNN), Neural Network, Naïve Bayes (NB), and etc. SVM has reported giving the best result (Yang and Liu, 1999). We will not go in detail for classifier since our work is focused on features.

2.1.3 Sentence level classification

Sentence Level classification has similar setting as the document level classification, except that each document is a sentence. Sentence level classification is commonly applied in sentiment analysis or opinion mining, where the system determines whether a sentence expresses a positive or a negative opinion (Dave et al., 2003; Hatzivassiloglou and Wiebe, 2002; Hu and Liu, 2004; Ng et al., 2006; Pang et al., 2002;

Riloff and Wiebe, 2003; Turney, 2002). They mainly use opinion words (e.g., “great”, “wonderful”, “bad”, and “poor”) or phrases in the process.

2.2 Information Extraction

Information extraction aims at extracting structured data from unstructured data such as text, and it commonly has three basic tasks: named entity extraction, and relation extraction, and event extraction.

2.2.1 Named entity extraction

Named entity extraction, also called named entity recognition, locates entities in natural language text and identifies their types (Okanohara et al., 2006; Ji and Grishman, 2002a). It provides the foundation for other extraction tasks. Some work focus on domain specific entities such as proteins (Mooney and Bunescu, 2005), while some work use general purposed entities, for example, Automatic Content Extraction (Automatic Content Extraction, 2007) defines seven types of entities: Person, Organization, Location, Facility, Weapon, Vehicle, and Geo-Political Entity.

Various methods have been used for named entity extraction. Gazetteer and regular expressions, which are used in early work of named entity extraction (Maynard et al., 2001), are still commonly used. To accommodate new domains, systems using gazetteer and regular expressions need to be manually tuned, but this could be difficult and tedious. As a result, methods use automatically learned patterns are more adaptive

and reporting better performance. The learned patterns can be statistics based models such as Hidden Markov Model (HMM) (Zhou and Su, 2001), and Conditional Random Fields (CRF) (Lafferty et al., 2001; Li and McCallum, 2003; Mcdonald and Pereira, 2005). CRF gives the best result so far (Sarawagi and Cohen, 2005).

After named entities have been recognized, it's also useful to resolve the coreference, i.e., identify all noun phrases that refer to the same object. Both linguistics based approach and machine learning based approach have been used on coreference resolution (Soon and Ng, 2001; Ng and Cardie, 2001).

2.2.2 Relation extraction

Relation extraction detects and characterizes the semantic relation among extracted named entities. E.g., Automatic Content Extraction (Automatic Content Extraction, 2007) defines seven relations: Artifacts, GEN-Affiliation, Metonymy, Org-Affiliation, Part-Whole, Person-Social, and Physical. Notice that none of these relations can address the relationship between a disease outbreak and the location of the outbreak.

Semi-automatically derived semantic constraints have been used for extraction of part-whole relations (Girju et al., 2006). While most other work for relation extraction automatically collect and/or construct features from the sentence which containing the entities, and then use statistical machine learning approach to identify the entities' relation. The features can be simply the text between the two entities, or linguistics

features such as Part-Of-Speech (POS) tagging and dependency relationships (Zhao and Grishman, 2005; Cultotta and Sorensen, 2004). Besides kernel methods (Zelenko et al., 2003; Cultotta and Sorensen, 2004; Jiang and Zhai, 2007), HMM (Skounakis et al., 2003) and CRF (Sutton and McCallum, 2007) have also been used in relation extraction as well. Web info has been used to filter incorrect instances (Pantel and Pennacchiotti, 2006).

2.2.3 Event extraction

Event extraction detects the events that the entities participated in and identifies their types. Similarly, the specifications for events vary by domains. For example, Automatic Content Extraction defines five types of events: Interaction, Movement, Transfer, Creation, and Destruction. Event extraction is a difficult problem because an even is usually indirectly express in a sentence. Linguistics knowledge has been showed useful in event extraction (Fillmore et al., 2006).

2.2.4 Application of information extraction

Information extraction has been applied to many different domains. In bioinformatics, information extraction helps biologists to retrieve biological knowledge such as protein interactions and protein names from literature (Bunescu et al., 2005; Mooney and Bunescu, 2005). In opinion mining, product review opinions and comparisons are extracted from reviews (Popescu and Etzioni, 2005; Carenini et al., 2005; Jindal and Liu 2006). Research paper search engine relies on extraction of meta-data from research paper (Peng and McCallum, 2004). With more and more news available on

Internet, it becomes a good source for information extraction. For example, corporate acquisitions are extracted from news wires (Califf and Mooney, 1999), and disease outbreak information is extracted from news reports.

2.3 Natural Language Processing

The recent advance of natural language processing has brought some interesting outcomes such as dependency tree. We will use dependency tree throughout this thesis. A dependency tree describes the semantic relationships between pair of words in a sentence. Dependency tree has been used in classification before, it is showed that sub-trees of dependency trees are helpful in classification, but using n-gram produces comparable results (Kudo and Matsumoto, 2004). Dependency tree has also been applied in information extraction. It has been used to extract dependency word micro contexts from Czech sentences (Holub and Böhmová, 2000). Another work uses tree-based patterns for Japanese information extraction, and each pattern is acquired from the paths of the dependency tree of a sentence. Tree-based patterns are found superior to the patterns derived from plain text (Sudo et al., 2001). Some researchers investigated using dependency tree in relation extraction, and using dependency tree kernel, higher precision but lower recall is reported comparing to bag-of-words kernel (Cultotta and Sorensen, 2004).

Throughout this thesis, we use MINIPAR (Lin and Pantel, 2001) to construct dependency tree. MINIPAR is a broad-coverage English parser; besides extracting dependency relationships, it also produces Part-Of-Speech tagging and tags named entities. Figure 1 shows the raw output of MINIPAR for a sample sentence "Japan has temporarily halted poultry imports since an outbreak of bird flu was confirmed by South Korea" is shown in, and the corresponding tree is illustrated in Figure 2. In a dependency tree, arrows point from a parent node to a set of children nodes that the parent node governs. The dependency relationship between a parent node and a child varies. For example, nodes "Japan" and "halted" have a subject-and-verb relationship while "poultry" and "imports" have a noun-noun-modifier relationship. Note that each named entity such as "bird flu" and "South Korea" is represented by one node in spite of that they have multiple literal words.

The dependency tree of a sentence captures the overall structure of a sentence. Because of the flexibility of the natural language, there are many ways to express the same meaning, but their dependency trees can be quite similar or the same. For example, the following sentence, which is different from the one in Figure 2, has a very similar dependency tree.

> Poultry imports have been temporarily halted by Japan since South Korea confirmed a bird flu outbreak.

E1	(()	fin C	*		(atts (perf +) (tense present)))	
1	(Japan	~ N	4	s	(gov halt)	(atts (sem (+country +location))))
2	(has	have have	4	have	(gov halt)	(atts (perf +) (tense present)))
3	(temporarily		~ A	4	amod	(gov halt))
4	(halted	halt V	E1	i	(gov fin)	(atts (perf +) (tense present)))
E3	(()	Japan N	4	subj	(gov halt)	(antecedent 1) (atts (sem (+country +location))))
5	(poultry		~ N	6	nn	(gov import))
6	(imports		import N	4	obj	(gov halt))
7	(since	~ SentAdjunct	4	mod	(gov halt))	
E0	(()	fin C	7	comp1	(gov since)	(atts (perf -) (tense past)))
8	(an	~ Det	9	det	(gov outbreak))	
9	(outbreak		~ N	14	s	(gov confirm))
10	(of	~ Prep	9	mod	(gov outbreak))	
11	(bird	~ U	12	lex-mod	(gov bird flu))	
12	(flu	bird flu N	10	pcomp-n	(gov of)	(atts (sem (+disease))))
13	(was	be be	14	be	(gov confirm)	(atts (perf -) (tense past)))
14	(confirmed		confirm V	E0	i	(gov fin) (atts (perf -) (tense past)))
E4	(()	outbreak N	14	obj	(gov confirm)	(antecedent 9))
15	(by	~ Prep	14	by-subj	(gov confirm))	
16	(South	~ U	17	lex-mod	(gov South Korea))	
17	(Korea	South Korea N	15	pcomp-n	(gov by)	(atts (sem (+country +location))))
18	(.	~ U	*	punc)		

Figure 1. Raw MINIPAR output for "Japan has temporarily halted poultry imports since an outbreak of bird flu was confirmed by South Korea."

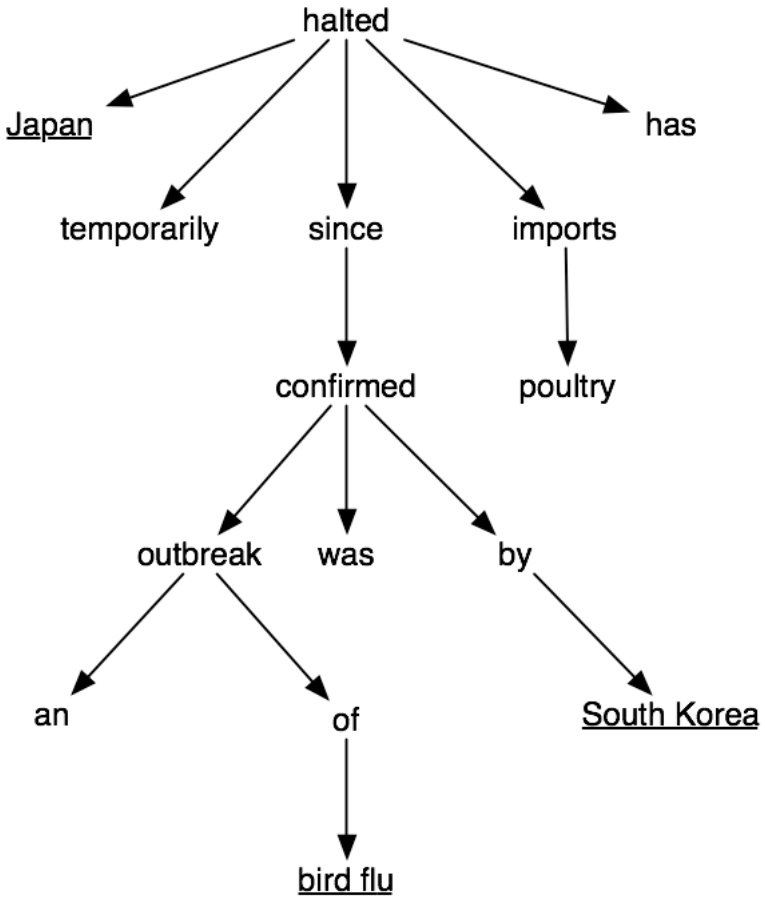

Figure 2. Dependency tree for "Japan has temporarily halted poultry imports since an outbreak of bird flu was confirmed by South Korea." Named entities are underlined.

2.4 Existing Emergent Disease Report System

There are some existing systems for detecting and monitoring emergent disease outbreak. ProMED-mail(ProMED-mail, 2007) is a mailing list about epidemic outbreak used by health workers worldwide, but it is totally manually operated.

Global Public Health Intelligence Network (GPHIN) (Global Public Health Intelligence Network, 2008) is a proprietary system developed by the Public Health Agency of Canada and it uses a filtering system (unpublished) and human experts to identify potential infectious disease outbreaks from news.

Proteus-BIO (Grishman et al., 2002) is an early system that automatically extracts infectious disease outbreak information. The extraction core is based on finite-state machine, so the extraction system can only work if a sentence match one of the patterns in its knowledge base. Considering the variety of natural language, it's no surprise that this method yields a 41% recall for extraction. There is no geocoding function in this system.

HealthMap (Freifeld et al., 2008; Brownstein and Freifeld, 2007) is a freely available web-based system that collects and visualizes disease outbreaks. By searching news for certain keywords like "outbreak", and using news sources that only report disease outbreak, HealthMap assumes all coming news are EDR, and thus it generally does not discriminate between EDR and nonEDR.

Global Health Monitor (Doan et al., 2008) is a similar system to HealthMap, in that both are web-based, and both detect and visualize outbreaks. One of the advantages of Global Health Monitor is its classification on news, though the classification is traditionally topic-based.

Our work is different from the existing work. First, our classification focuses on the semantic level difference between EDR and nonEDR. While only Global Health Monitor classifies the news in a traditionally topic-based way. Second, our extraction algorithm extracts the outbreak's location based on sequential rules that capture the semantic and/or syntactical patterns, while Global Health Monitor's location extraction only depends on the location's frequency, and HealthMap simply take all locations appearing in the news as the location of the outbreak. Finally, we use a comprehensive geographical database GeoNames, which consists of over eight million geographical entries, and it allows us to locate an outbreak accurately. For comparison, Health Map uses a dictionary of 2,300 locations, and Global Health Monitor's geographical database includes 4,268 locations. A side effect of using a comprehensive geographical database is a severe ambiguity problem, and we solve this problem by using a disambiguation algorithm and the geographical locations where the news sites are from.

2.5 Summary

In this chapter, we reviewed the related work in text classification, information extraction, and natural language processing. We also introduced several existing systems for emergent disease outbreak detection, and how our work is distinct from them.

3 SENTENCE LEVEL CLASSIFICATION OF EMERGENT DISEASE REPORT

In this chapter, we study how to classify Emergent Disease Report Sentences and nonEmergent Disease Report Sentences. Traditional text classification studied in the information retrieval and machine learning literature is mainly based on topics. That is, each class or category represents a particular topic, e.g., sports, politics or sciences. However, many real-world problems require more refined classification based on some *semantic perspectives*. For example, in a set of documents about a disease, some documents may report outbreaks of the disease, some may describe how to cure the disease, some may discuss how to prevent the disease, etc. To classify text at this semantic level, the traditional bag-of-words model is no longer sufficient. We study semantic text classification of disease reporting on sentence level, and show that both keywords and sentence semantic features are very useful for the classification. Our experimental results demonstrated that this integrated approach is highly effective.

The setting of semantic text classification is the same as traditional topic-based text classification. Given a set of documents D, a set of classes C, and a test document d_{test}, such that each document d_i ($d_i \in D$) is labeled with a class c_j ($c_j \in C$), the goal is to predict the label of d_{test}. The difference between semantic text classification and the traditional text classification is mostly on the classes C. In traditional text classification,

classes are coarse-grained topics, for example, politics and sports, however, in semantic text classification, the classes are more refined, for example, emergent disease outbreak and emergent disease preparation. This is a supervised classification problem where the training data's labels are known.

As a case study, we investigate the disease reporting domain. We want to classify sentences that report disease outbreaks, and sentences that do not. For example, the following sentence reports a possible disease outbreak "*the district hospital reported today that 10 people were diagnosed with cholera this morning*". However, the following sentence does not report an outbreak, "*the district hospital reported today that they have successfully tested a new cholera treatment procedure*". Both sentences are on the topic of cholera. However, they are entirely different semantically. The problem is how to separate sentences based on the required semantic categories, i.e., reporting a possible outbreak or not in this case. Note that sentence, rather than document is the focus of classification because one document may contain many different aspects of a topic, for example, besides outbreak reporting, a news report of a disease outbreak may contain the research on this disease, how to prepare for the disease, the history of previous disease outbreak, etc.

3.1 Proposed Method

We propose to integrate the bag-of-words scheme and semantic features extracted from texts for classification. Here, we define *semantic information* as any information extracted from the text that is not based on keywords or n-grams. Clearly, there are multiple levels of semantic information. At the highest level, it is the full understanding of the text, which is still not possible with the current technology. At lower levels, we have features with different amounts of semantic contents, which can be extracted from sentences based on the current NLP techniques. The exact features used in this work will be discussed in the next section.

An interesting question is whether the bag-of-words representation is still useful in semantic text classification. We believe that the answer is yes for two reasons:

To express a particular semantic meaning, certain specific keywords are still more likely to be used, although the same words can be used to express other information but with less frequency.

Semantic feature extraction is still not perfect. There may be many errors. Keywords can help offset some of these errors.

Figure 3 illustrates the difference between traditional text classification and semantic text classification as described in this work. Note that we do not make a difference of the types of classes or texts used in a classification task because

classification of any type of categories can be assisted by some level of semantic information.

3.2 Important Semantic Features

Our aim is to classify sentences that report possible disease outbreaks (EDR Sentences) and those that do not (nonEDR Sentences), which is a classification problem. We will use a supervised machine learning algorithm, e.g., Naïve Bayes (NB) or support vector machines (SVM). Thus, we only need to design and construct features. As we mentioned above, we use both keywords and semantic features. Keyword features are obtained in the same way as in traditional text classification. Here, we only focus on semantic features.

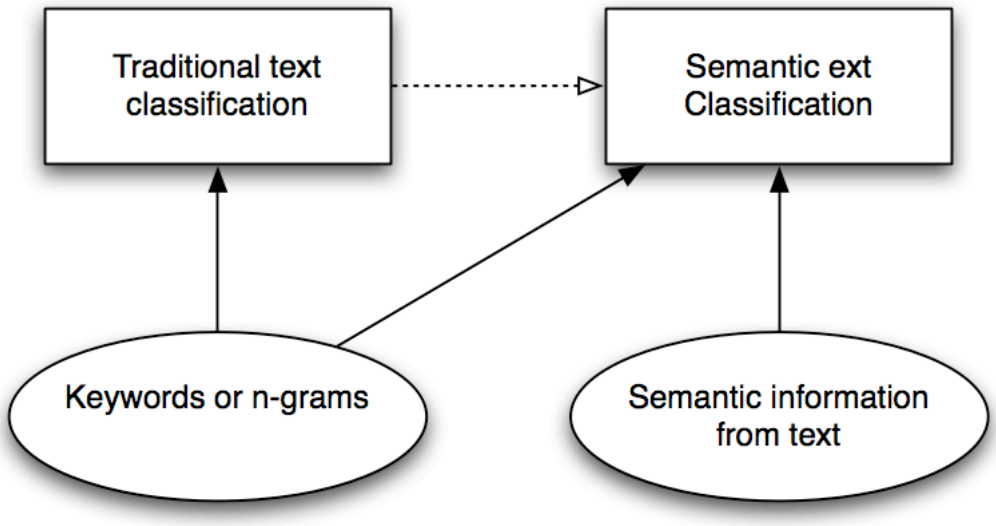

Figure 3. Traditional text classification and semantic text classification.

3.2.1 Noun phrase containing a disease word

Center word of a noun phrase: Noun phrase is the basic building brick of a sentence's structure. In the traditional bag-of-words approach, all words in a noun phrase are treated equally. However, the center word in a noun phrase has a more direct influence on the overall semantic meaning than any other word in the noun phrase. For example, in the sentence "*the district hospital reported today that their new cholera treatment procedure had been very successful*", the center word of the noun phrase "*their new cholera treatment procedure*" is "*procedure*". While in "*10 cases of cholera have been reported early today*", the center word of the noun phrase "*10 cases of cholera*" is "*cases*". Thus, although the disease name "cholera" appeared in both sentences, it has different center words, which lead to different semantic meanings. Note that we only use noun phrases that contain a disease word because such phrases are more likely to be relevant to our classification task, i.e., sentence containing a disease named entity is much more likely to be an EDR Sentence than sentence not without one. There are some cases that no disease name will show up in an EDR Sentence, but in Chapter 5 we will show that the percentage of such cases are very small, and we can further reduce this percentage by including general disease words as disease names.

Negation modifier and determiner word: Other important features in a noun phrase include negative modifiers such as "no", and determiner words such as "every" and "a". Their importance can be illustrated by the following examples: "*No case of cholera has been found yesterday*" indicates no disease found, and "*For every case of*

mad cow disease in Switzerland, 100 animals may carry the infection silently" gives a study result on the disease rather than a specific case.

3.2.2 Verb phrase

Verb and adjective: The verb serves as the main skeleton of a sentence and thus an important feature. Sometimes, a verb is too common to have a specific meaning. In that case, the adjective word after the verb becomes important. For example, the verbs "is" and "become" are not specific, but "is ill" and "become sick" are.

Tense: Another characteristic of a verb phrase is the verb's tense, which is also important in the semantic meaning because tense may show the time or the subjunctive mood. Past perfect tense usually means something happened in the long past. For example, "*West Nile Virus had plagued US*" refers to an old disease outbreak. Subjunctive mood expressed by the tense of past-future perfect is often used for conjectures or assumptions. For example, "*a bird flu outbreak could have killed millions of people*" is a conjecture of the disease's impact rather than a report.

Auxiliary word: Subjunctive mood can also be expressed by a verb's auxiliary words such as "can" and "may". Again, it shows a hypothetical case instead of a fact.

Verb phrase being an if-whether clause: Using word "if" or "whether" is yet another way to express the subjunctive mood. Similarly, it does not describe a fact.

Negation word of verb phrase: A verb with negation modifiers usually has the opposite meaning to the verb alone. Examples of such modifiers are “not”, “rarely”, “seldom”, “never”, and so on.

Verb phrase being an adjective clause: If a verb phrase appears in an adjective clause of a sentence, it often gives complementary information, not the main content of the sentence. For instance, in the sentence “*the team reports on their investigation of a Canadian farm where an outbreak of pneumonia in pigs began in October 1999*,” the main interest is not in reporting the outbreak, as the outbreak most likely happened in the past and has been noticed before.

Subject and object: The subject word and object word of a verb are also important. As in “*The real name for mad cow disease is Bovine Spongiform Encephalopathy*”, the subject “name” and the object “*Bovine Spongiform Encephalopathy*” suggest that the sentence is not about any disease outbreak.

3.2.3 Dates

For disease outbreak reporting, dates used in sentences are important. If the date appearing in a sentence refers to a long time ago, the sentence is unlikely to report a new outbreak. Although the verb tense can show whether the time is in the past, present, or future, it is ambiguous as it is unclear how far in the past or in the future.

Date information can be expressed in a large number of ways. We focus on the most common ones in this work. Thus, our description below is by no means complete, but is quite sufficient for our data. A piece of date information is usually expressed by a prep word (implied if it is missing or omitted) followed by a *date phrase*. We call the date expressed in the text as the *expressed date*, and the date of the context as the *context date* (e.g., the date when a news report was published)

Prep word: A prep word decides the relationship between an *expressed date* and the date phrase that follows. We summarize the prep words and the corresponding relationships in TABLE I. If a prep word is omitted, in most cases it's the same as the first relationship in the table. For example, "*The alert was given last Tuesday*".

Adjective and adverb: Some adjectives and adverbs may also be associated with dates, e.g., "ago" as in "three months ago", "last" as in "last month", etc. Grammar rules related to them will be given below. In general, a *date phrase* expresses either an *absolute date* or a *relative date*. We will not discuss time in this paper as the date information is sufficient for our application task. The ways to express a specific time are not as diverse as those for date and can be dealt with in a similar way.

Absolute date: As its name suggests, an absolute date expresses a specific date without ambiguity regardless when it is seen. There are two main types:

TABLE I PREPOSITION WORDS AND RELATIONSHIPS BETWEEN EXPRESSED DATES AND DATE PHRASES

Relationship	Prep word	Example
expressed date is the date phrase	in, at, on, during	on Monday
expressed date is before the date phrase	before	before winter
expressed date is after the date phrase	after	after May 1, 2006
expressed date ends within the date phrase	in, within	in two days
expressed date ends by the date phrase	by, as of, until, till, no later than	by today
expressed date spans the two date phrases	Between … and… from…to…	between January and February
expressed date starts from the date phrase	since	since last year

- *Historic period*: It is a time period in history. Its duration is usually very long, and it has a specific name, e.g., “Stone Age”.
- *Formal date*: It specifies an absolute time period in quantitative terms that can be:

  a century (e.g., “18th century”),
  a decade (e.g., “1980’s”),
  a year (e.g., 1998),
  a season (e.g., summer of 2007),
  a month (e.g., May 1998),
  a day (e.g., October 22, 2005),
  a time period of a specific day (e.g., morning of Mar 22, 2005),

Relative date: The absolute date of a *relative date* can only be determined based on the *context date*.

- *Recurrent named date*: Such a relative date occurs repetitively, e.g., annual festival, season, month of year, day of month, day of week, etc. For example, “May 22” refers to the date in the year determined by context. Other examples include, “last Christmas”, “next morning”, and “this Thursday”. A restrictive modifier is usually mandatory, although sometimes it is omitted based on convention. For example, in “*an outbreak was reported on Monday*”, “Monday” usually refers to “the *past* Monday”.

- *Other named dates*: Such dates include “today” and “tomorrow” or *special words* (e.g., “now” and “recently”) that are dedicated to some relative dates. No restrictive modifier such as “next” or “last” is needed before them.
- *Number-unit*: This is also popularly used in date phrases, e.g., “three months” in “three months ago” and “ten years” in “past ten years”. Similar to a recurrent date, a modifier is also required for this type, e.g., “ago” and “past”.

A date phrase may have a refiner, such as “*early* 2007” and “the *end* of last month”. Now we give a formal definition of date phrases in Backus–Naur form. Due to space limitations, some rules use suspension points in place of similar entries.

<DatePhrase> ::= [<Refiner>]<FormalDate>
| [<Refiner>]<HistoricPeriod> | [<Refiner>]<FormalPeriod>
| [<Refiner>]<Modifier><DateName> | <SpecialWord>
| [<Refiner>]<Modifier>[<Number>]<Date Unit>
| [<Number>]<Date Unit>[<PostModifier>] | [<Refiner>]<SpecialDay>
<Refiner> ::= fiscal | late | early | end of | beginning of
| middle of | mid | ……
<Modifier> ::= last | previous | next | coming | past|……
<PostModifier> ::= ago | later | early | ……
<DateUnit> ::= century | decade | year | season | quarter
| month | week | day | hour | minute | second
<SpecialDay> ::= today | tomorrow | the day before yesterday | ……

<SpecialWord> ::= now | recently | …

<FormalDate> ::= [<Month>] <Year> | [<Season>] <Year>

| [<Festival>] <Year> | [<DayofWeek>]<Month>[/]<Day>[/][<Year>]

<HistoricPeriod> ::= Stone Age | ……

<FormalPeriod> ::= <OrdinalNumber> century | <year>[']s | ……

<DateName> ::= <Festival> | <DayOfWeek> [<TimeOfDay>]

| <Season> | <MonthOfYear>

<Month> ::= <Digit><Digit> | <MonthOfYear>

<Day> ::= 1 | 1st | …… | 31 | 31st

<Year> ::= <Digit><Digit><Digit><Digit>

<Season> ::= spring | summer | fall | autumn | winter

<MonthOfYear> ::= jan | january | feb | feburary | ……

<DayOfWeek> ::= mon | monday | tue | tuesday | ……

<TimeOfDay> ::= morning | noon | afternoon | evening | ……

<Festival> ::= Christmas [eve] | [post] Christmas |……

<OrdinalNumber> ::= 1st | first | 2nd | second | ……

<Number> ::= <Digit>+ | one | two | ……

<Digit> ::= 0 | 1 | …… | 9

Most date phrases can be used after any prep word, with some exceptions, e.g., "three days ago" is usually not used with "during".

3.3 Feature Extraction

Last section introduced several features that are important for semantic classification of disease sentences. Now we describe how to extract these features from a sentence.

3.3.1 Named entity recognition

Named entities representing disease names and dates are essential for feature extraction because most features described in last section can only be found based on correct recognition of the corresponding named entities, i.e., disease names or dates. Thus, recognizing named entities is a necessary step. The named entity recognition system that we use is MINIPAR (Lin and Pantel, 2001). Note that a typical named entity recognizer also recognizes locations, person names, organization names, etc, but they are not needed in this work. Besides named entity, MINIPAR also produces the dependency tree of a sentence.

3.3.2 Feature extraction and construction

After a sentence's named entities have been recognized and its dependency tree has been built, we extract features in the following way:

We start from a named entity of any infectious disease, and find the noun phrase that contains the disease. The center word of a noun phrase is the highest node in the noun phrase's dependency tree. Negation modifier and determiner words can be found among children nodes of the center word.

The verb can be determined easily as it is always the nearest ancestral verb node of the noun phrase. If there is an adjective node between the verb and the noun phrase, it is taken as the adjective word feature. The tense of a verb phrase does not depend solely

on the form of the center verb. The forms of auxiliary words such as "*do*", "*have*" and "*be*" count as well. For example, in "*an outbreak of cholera has been reported*", although the center verb "*reported*" is of the preterit form, it is not past tense because of the auxiliary word "*has*". All auxiliary words are children nodes of the center word. So are negation modifiers, and subject/object words.

To get other features of a verb phrase, we need to check sibling or parent nodes of the verb node in the dependency tree. If a verb phrase is an adjective clause, the verb node normally has a sibling node of "wh-" word and a relationship of complementary to its parent node. Here is an example, "*the team reports on their investigation of a Canadian farm where an outbreak of pneumonia in pigs began in October 1999"*. If a verb phrase is an if-whether clause, there will be a sibling node of "if" or "whether". By scanning the verb's sibling nodes, these features can all be found easily.

For the date feature, it can be recognized using the definitions given in Section 3.2. However, it is not trivial to normalize dates so that they will be comparable to each other. One solution is to translate all dates into absolute ones, and construct features that include a date's year, month, and day. But for nonspecific dates, such as "*during the last decade*", accurate translation is impossible. In this application, we are only interested in recent disease outbreaks, so we generally treat a relative date in the scope of current year as "recent", i.e., things happened in sometime last year and before would be "old". Any date

after the report date is considered a “future” date. Thus, the date feature has three possible values, “recent”, “old” and “future”.

In some sentences, there are multiple disease names in one sentence, and then additional features are created as long as they correspond to different noun phrases.

3.3.3 Implementation

We use the English parser MINIPAR for dependency tree generation and for named entity recognition. In order to recognize infectious diseases, we supplemented the standard MINIPAR database with infectious disease names extracted from Centers for Disease Control and Prevention (Centers for Disease Control and Prevention, 2008). Another modification is to recognize some date phrases such as “last week”, which MINPAR fails to recognize. The feature construction algorithm, which is implemented in Perl, then reads the generated dependency trees and outputs features.

3.4 Experiments

This section evaluates the proposed technique. We discuss the experimental data, evaluation settings and the results in turn.

3.4.1 Experimental data

The corpus consists of sentences related to infectious diseases. Some of the sentences are emergent disease reports (EDR Sentences), and others are not (nonEDR

Sentences) but still contain the disease names. The sentences are extracted from disease report documents from ProMED-mail (ProMED-mail, 2007). We labeled the sentences into two classes: EDR Sentences and nonEDR Sentences. The data set has 1660 nonEDR Sentences and 682 EDR Sentences.

3.4.2 Experimental settings

Two popular supervised learning algorithms are used to build models, Support Vector Machines (SVM) and Naïve Bayes (NB). Both algorithms are provided in the latest version of the Rainbow package (McCallum, 1996), which is used in our experiments. Different types of features are employed and compared:

a. *sentence*: only bag-of-words representation with *i*-grams: 1-gram, 2-gram, 3-gram and 4-gram.

b. *s-features*: semantic features (including the date feature).

c. *s-features*+*sentence*: semantic features and bag-of-words features in a sentence are combined.

To ensure reliable results, we run each technique 10 times. In each run, 80% of the data (randomly selected) is used for training and 20% of the data is used for testing. The results are then averaged and reported below. The evaluation measure is F-score on EDR Sentences. F-score is the harmonic mean of precision (p) and recall (r), i.e., $F = 2pr/(p+r)$, which is commonly used in text classification.

3.4.3 Experimental results

Figure 4 shows the average F-scores of all methods. We observe that semantic features (*s-features*) are very helpful. Both SVM and NB produce much better results when sentences and semantic features are both used. SVM (*s-features+sentence*) with 3-gram for sentences gives the best F-score and it also performs the best for 2-gram and 4-gram, except for 1-gram, in which NB (*s-features+sentence*) is better.

We also single out the date feature to see how it effects classification as intuitively the disease reporting dates are important for EDR sentences.

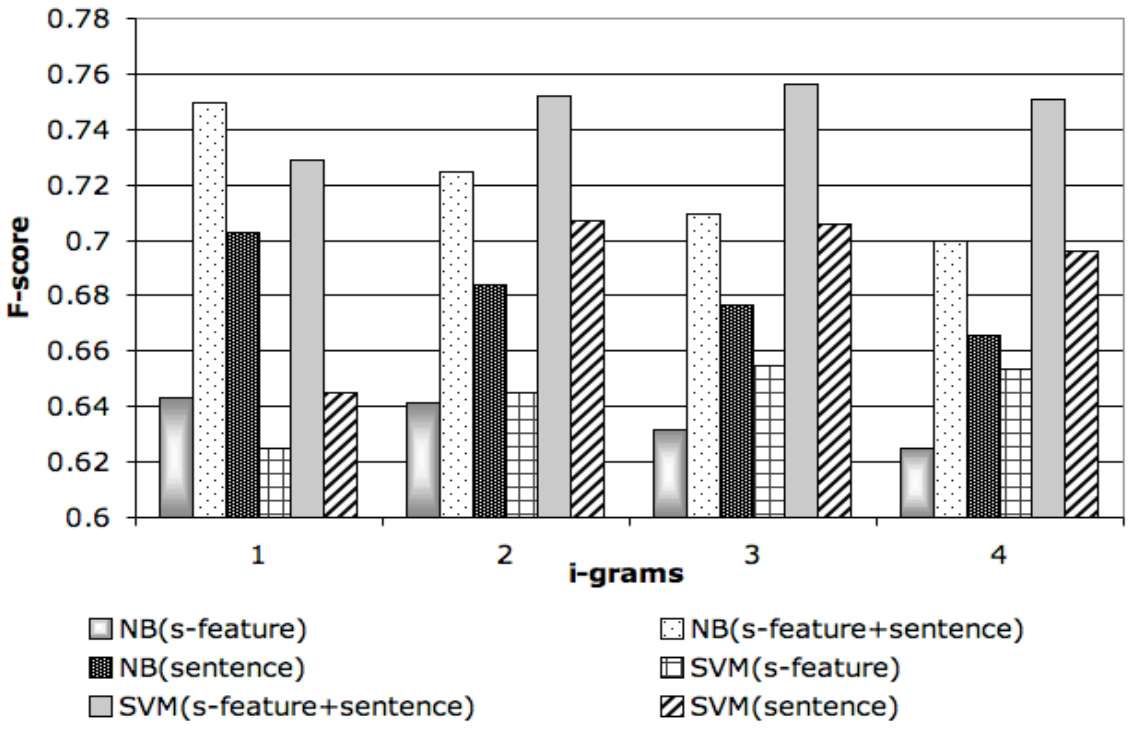

Figure 4. Experimental results on Emergent Disease Report Sentence Classification.

The date feature is indeed helpful, as shown in Figure 5. For NB, the F-scores with date features are always better than without date features. For SVM, the results are also better for 1-gram and 2-gram. All the results here use both s-features and sentences. Note that there are 271 nonEDR and 240 EDR Sentences with "recent" for the date feature. Thus, the classification cannot be done trivially using dates alone.

Since the date feature has shown its importance, it will probably help more if the weight of the date feature is increased. We thus increase its weight. Multiplying each date feature by 3 gives the best results. Figure 5 shows that "with triple date feature" (the other settings remain the same) gives better F-scores for both NB and SVM in almost all cases. NB with 1-gram produces the best result. Due to this success, we also tried to increase the weights of all s-features ("with all features doubled") but without improvements.

In summary, we can conclude that combining bag-of-words and semantic features indeed improves classification. The date feature is also very helpful.

3.5 Summary

In this chapter, we study the problem of classifying disease reporting at the semantic level. It is shown that both keywords and semantic features are valuable for the task. We also investigate the representation of dates, which will be useful to other

applications. Experimental results demonstrated that the proposed integrated approach significantly outperforms each individual approach alone.

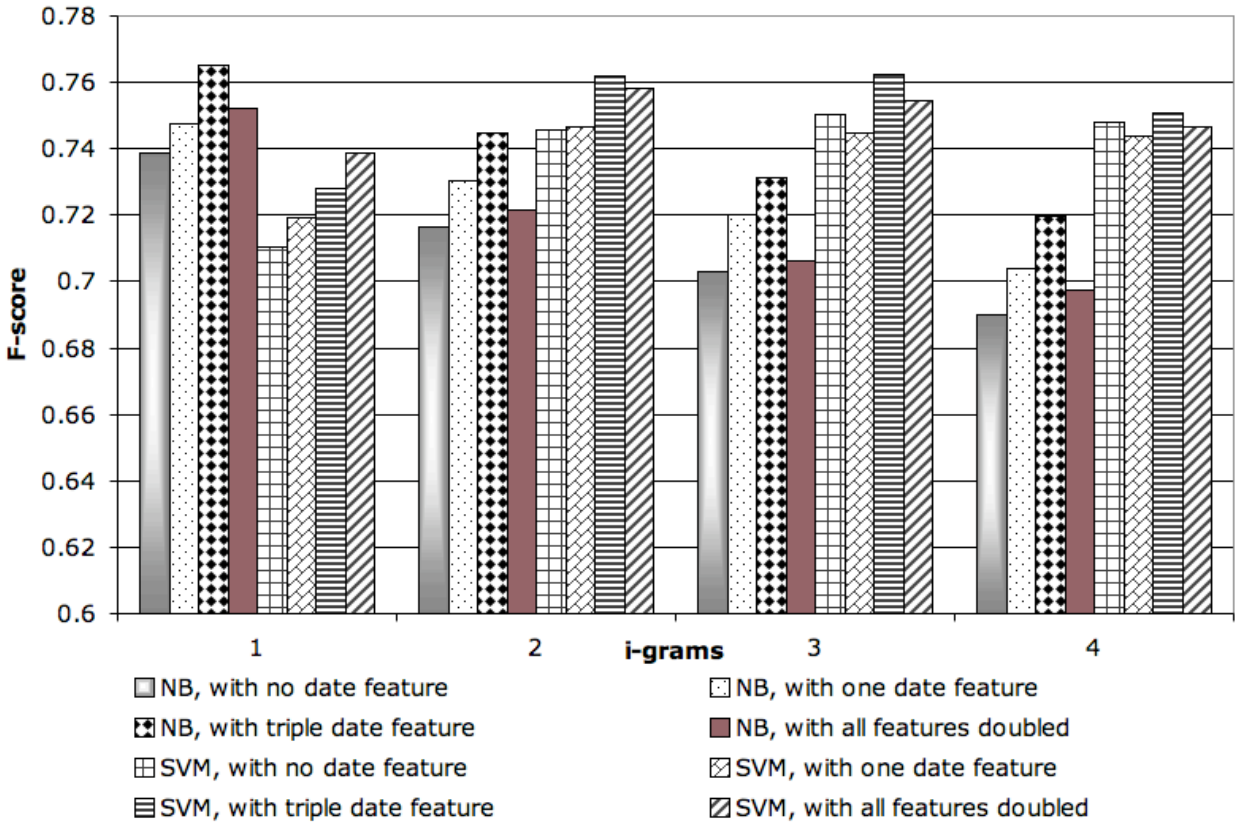

Figure 5. Experimental results on Emergent Disease Report Sentence Classification: with different weights of date feature.

4 EXTRACTION OF OUTBREAK LOCATION FROM EMERGENT DISEASE REPORT

Last chapter discusses how to classify EDR Sentences from non EDR Sentences. Once EDR Sentences are identified, the next step is to extract information about an outbreak from the identified EDR Sentences. There are three kinds of information we are interested: outbreak location, outbreak time and name of the disease. For outbreak time, since we are dealing with real-time news, any EDR News and EDR Sentences must report an outbreak in a recent sort time period, so the report date of the EDR News can be regarded as the outbreak time approximately. For name of the disease, it is possible to obtain a list of known disease names, and we will discuss how to deal with EDR Sentence and EDR News without any disease name. Thus, in this chapter our task is confined to extract only the location of the outbreak, more specifically, we study the problem of extracting EDR Location Named Entities from EDR Sentences.

4.1 Introduction

The problem studied in this chapter is defined as follows:

Problem definition: Given an EDR Sentence, extract the EDR Location Named Entities of the outbreak.

Note that it is assumed that the given sentences are EDR Sentences. The problem of classification of EDR Sentences has been studied in the last chapter.

Clearly, our problem is an information extraction problem. The general information extraction problem has been studied by numerous researchers and many existing techniques have also been reported in the literature. Conditional random field (Lafferty et al., 2001) is perhaps the most effective general approach to solving the problem. However, we will show that it does not perform well based on our data. This chapter proposed a novel technique based on sequential pattern mining to generate extraction rules. These rules are then used to match and to extract disease locations.

Specifically, we will use label sequential rules and class sequential rules for the purpose. These rules are described in 4.3.3 together with their mining techniques as traditional sequential pattern mining in data mining does not generate any rules, but only produces frequent sequences that meet a user-specified minimum support. Thus, each type of rule mining consists of two steps, frequent sequential pattern mining (unsupervised) and rule generation (supervised).

Another important novelty of the proposed technique is that the data used for mining or learning are sequences obtained in dependency trees. That is, only some important paths in dependency trees are used rather than the original sentences. This is because the structure in a dependency tree embeds the essential relationships of concepts in a sentence, while the information contained in a raw sentence can be quite diverse,

making it difficult to mine key patterns to be used for extraction. The details will be given later in this chapter.

The whole process of the proposed technique consists of the following steps:

a. Obtain the first sentence that contains a disease and a candidate location from each news article. We will define what we mean by candidate location in 4.3.2.2. This sentence is usually the first or the second sentence in the news report, which is not surprising.

b. Build a dependency tree of each sentence using a parser. In our work, we use MINIPAR (Lin and Pantel, 2001) for the purpose. The list of diseases is also input into MINIPAR so that it can also recognize disease names that we are interested in. Our disease names are obtained from Centers for Disease Control and Prevention (Centers for Disease Control and Prevention, 2008).

c. Extract relevant paths of each dependency tree to form the sequence data for mining and learning. The detail will be discussed in 4.4.4.

d. Mine sequential patterns from the path sequence data, and generate label and class sequential rules based on the manually tagged data of disease locations. Label sequential rules and class sequential rules are combined to form a classifier to be used to recognize whether a candidate location is a disease outbreak location.

e. Test the classifier using unseen test data using cross-validation to assess its precision, recall, and F-score.

To the best of our knowledge, the proposed technique has not been used in existing approaches. To evaluate the proposed technique, we use a large number of health news articles crawled from Google News (Google News, 2008) in 17 days and historic reports from ProMED-mail (ProMED-mail, 2007). The extraction result demonstrated the effectiveness of the technique. The proposed technique is also compared with the current state-of-the-art extraction algorithm conditional random field. Our method outperforms conditional random fields significantly.

4.2 Problem Statement

In this chapter, we focus on identifying outbreak information at the sentence level, so the problem is stated for sentences.

For each news article, we extract the first sentence that contains at least one disease name and one candidate location (to be defined later in 4.3.2.2). We then identify the actual location of the disease outbreak.

Although this is a restricted problem, it is still a challenging problem because many sentences mention multiple locations, and some locations are not disease occurring locations. For example, in the sentence “Japan has temporarily halted its poultry imports

because of a recent bird flu outbreak in South Korea," "Japan" is not the EDR location, but "South Korea" is.

This is clearly an information extraction problem that extracts structured information, pairs of the form (disease_name, disease_location), from natural language sentences. The simplest method is to use a named entity tagger to detect the disease NE and location NE from a sentence and assume that the disease occurs in the location. However, this method has two problems. First, the named entity taggers make many mistakes, i.e., it can tag person names and organization names are location names, and vice versa. Second, the extracted diseases and locations may not have the required relations that a disease occurs in a location. This problem thus should be solved as a relation extraction problem. However, as we discussed in the related work, the existing results for solve this specific problem based on regular expressions is quite poor.

4.3 Some Definitions

Before we present our proposed method, we first introduce the definition of node path, class sequential rule, and label sequential rule.

4.3.1 Dependency tree and node path

The dependency tree of a sentence is a tree where,

- Each word or phrase is a node in the tree;

- Each node in the tree points to a parent node that it depends on.

We use MINIPAR (Lin and Pantel, 2001) to construct dependency tree. In the dependency tree output of MINIPAR, each node *n* contains the information such as the literal word/phrase (*n.word*), stemmed form of the word (*n.root*), Part-of-Speech tagging (*n.POS*), and Named Entity type recognized by MINIPAR (*n.NE*).

In a dependency tree, the *node path, path*(n_1, n_2) for node n_1 and node n_2 is defined as:

- $\langle \{n_1\}\ \{p(n_1)\}\ \{p(p(n_1))\}\ \ldots\ \{n_2\} \rangle$ if n_2 is an ancestor of n_1,
- $\langle \{n_2\}\ \{p(n_2)\}\ \{p(p(n_2))\}\ \ldots\ \{n_1\} \rangle$ if n_2 is an ancestor of n_1,
- $\langle \{n_1\}\ \{p(n_1)\}\ \ldots\ \{q\}\ \ldots\ \{p(n_2)\}\ \{n_2\} \rangle$ otherwise, where q is the nearest common ancestor of n_1 and n_2. Note that if n_1 appears after n_2 in the sentence, the order of this node path is inverted. When n_1 and n_2 belongs to two clauses, q will be the visual root node of the dependency tree, and no node path is defined.

where $p(n)$ is the parent node of n.

Such paths will be used in sequential pattern mining to be discussed later.

4.3.2 Named entity

After dependency tree is built, two types of named entities: Disease NE and Location NE will be detected using MINIPAR.

4.3.2.1 Disease named entity

Disease NE follows the same definition in 1.2.1. We gathered the names of infectious diseases and their alias from the Centers for Disease Control and Prevention (Centers for Disease Control and Prevention, 2008), and augmented MINIPAR's database with these names. Since Disease NE was not in the original MINIPAR package, a Named Entity type called Disease NE has been created in MINIPAR so that disease names can be recognized as Disease NE.

4.3.2.2 Candidate location named entity

For identifying location NE, MINIPAR's existing named entity recognition system is not accurate enough. It often takes a Location NE as a person's name and vice versa. To solve this problem, we introduce candidate Location NE, which covers almost all location NE. Clearly, this will introduce many false positives. We will use sequential rules later to find the correct ones. A node *n* is a candidate Location NE if:

- (*n*.NE != Date Named Entity) and
- (*n*.NE != Number Named Entity) and
- (*n*.NE != Disease Named Entity) and

- (n.POS = N) and
- The first letter of n.word is capitalized.

4.3.3 Class and label sequential rules

Our proposed technique uses two types of sequential rules. Such rules are mined based on sequential patterns (Agrawal and Srikant, 1994). Given a set of input sequences, sequential pattern mining (SPM) finds all subsequences (called *sequential patterns*) that satisfy a user-specified minimum support threshold. Below, we first explain some notations, and then define two types of rules, *class sequential rules* (CSR) used in classification of location names, and *label sequential rules* (LSR) used in EDR location extraction. For more details about these types of rules and their mining algorithms, please refer to (Liu, 2006).

Let $I = \{i_1, i_2, ..., i_n\}$ be a set of items. A *sequence* is an ordered list of *itemsets*. An *itemset X* is a non-empty set of items. We denote a sequence s by $\langle a_1 a_2 ... a_n \rangle$, where a_i is an itemset, also called an *element* of s. We denote an element of a sequence by $\{x_1, x_2, \ldots, x_m\}$, where x_j is an item. An item can occur only once in an element of a sequence, but can occur multiple times in different elements. A sequence $s_1 = \langle a_1 a_2 ... a_r \rangle$ is a *subsequence* of another sequence $s_2 = \langle b_1 b_2 ... b_m \rangle$ or s_2 is a *supersequence* of s_1, if there exist integers $1 \leq j_1 < j_2 < ... < j_{r-1} \leq j_r \leq m$ such that $a_1 \subseteq b_{j_1}$, $a_2 \subseteq b_{j_2}$, ..., $a_r \subseteq b_{j_r}$. We also say that s_2 *contains* s_1.

4.3.3.1 Class sequential rules

Let S be a set of data sequences. Each sequence is labeled with a class y. Let Y be the set of all classes, $I \cap Y = \varnothing$. Thus, the input data D for mining is represented with $D = \{(s_1, y_1), (s_2, y_2), \ldots, (s_n, y_n)\}$, where s_i is a sequence and $y_i \in Y$ is its class. A *class sequential rule* (CSR) is an implication of the form

$$X \rightarrow y, \text{ where } X \text{ is a subsequence, and } y \in Y.$$

A data instance (s_i, y_i) is said to *cover* the CSR if X is a subsequence of s_i. A data instance (s_i, y_i) is said to *satisfy* a CSR, if X is a subsequence of s_i and $y_i = y$. The *support* (sup) of the rule is the fraction of total instances in D that satisfies the rule. The *confidence* (conf) is the proportion of instances in D that covers the rule also satisfies the rule.

Example: TABLE II gives an example sequence database with five sequences and two classes, c_1 and c_2. Using the minimum support of 20% and the minimum confidence of 40%, one of the discovered CSRs is:

$$\langle \{1\}\{3\}\{7, 8\} \rangle \rightarrow c_1 \quad [\text{sup} = 2/5 \text{ and conf} = 2/3]$$

Data instances 1 and 2 satisfy the rule, and data instances 1, 2 and 5 cover the rule.

Given a labeled sequence data set *D*, a minimum support (*minsup*) and a minimum confidence (*minconf*) threshold, CSR mining finds all class sequential rules in D.

TABLE II AN EXAMPLE OF SEQUENCE DATABASE WITH CLASSES

	Data Sequence	Class
1	⟨{1}{3}{5}{7, 8, 9}⟩	c_1
2	⟨{1}{3}{6}{7, 8}⟩	c_1
3	⟨{1, 6}{9}⟩	c_2
4	⟨{3} {5, 6} ⟩	c_2
5	⟨{1} {3}{4} {7, 8}⟩	c_2

4.3.3.2 Label Sequential Rules

A *label sequential rule* (LSR) is of the form,

$$X \rightarrow Y,$$

where *Y* is a sequence and *X* is a sequence produced from *Y* by replacing some of its items with wildcards. A wildcard, denoted by a '*', matches any item. The definitions of support and confidence are similar to those above.

Example: TABLE III gives an example sequence database with 5 sequences. From this sequence database and minimum support of 30% and minimum confidence of 30%. We can get the following label sequential rule,

$$\langle\{1\}\{3\}\ \{7, *\}\rangle \rightarrow \langle\{1\}\{3\}\{7, 8\}\rangle$$

[sup = 3/5, conf = 3/4]

Data sequences 1, 2, 4, and 5 contain $\langle\{1\}\{3\}\{7, *\}\rangle$, and data sequences 1, 2, and 4 contain $\langle\{1\}\{3\}\{7, 8\}\rangle$.

TABLE III AN EXAMPLE

SEQUENCE DATABASE

	Data Sequence
1	$\langle\{1\}\{3\}\{5\}\{7, 8, 9\}\rangle$
2	$\langle\{1\}\{3\}\{6\}\{7, 8\}\rangle$
3	$\langle\{1, 6\}\{9\}\rangle$
4	$\langle\{1\}\ \{3, 6\}\ \{7, 8\}\rangle$
5	$\langle\{1\}\ \{2, 3\}\{7, 9\}\rangle$

Such rules are useful because we want to predict some items in an input sequence, e.g., item 8 above. The confidence of the rule tells us the probability that the '*' is 8 if an input sequence matches ⟨{1}{3} {7, *}⟩. In our application, the "*" can match an EDR location or a nonEDR location depending on the rule and the confidence of the rule. Again, mining of this type of rules can be found in (Liu, 2006).

4.4 Proposed Method

Now we introduce our method to identify the EDR location NE and disease NE pair. Because disease NE can be identified very accurately, we just need to find the location NE paired with the disease NE. The overall flow of the algorithm is as follows:

4.4.1 Training

Training consists of three steps:

a. From training EDR Sentences, build node path between each pair of disease NE and candidate Location NE, if there is any, and annotate the candidate Location NE.

b. Construct two sequence databases SD_A and SD_B from the node paths. Every node path corresponds to one sequence (without class) in SD_A and one sequence with class in SD_B.

c. Mine Label Sequential Rules from SD_A and mine Class Sequential Rules from SD_B.

4.4.2 Testing

Testing also consists of three steps:

a. For each test sentence, we first build its dependency tree, and then identify all candidate Location NE and Disease NE from the tree. For each pair of candidate Location NE n_l and Disease NE n_d in the same tree, build their node path $path(n_l, n_d)$, if there is any.

b. Each node path $path(n_l, n_d)$ is converted to sequence S_A and sequence S_B as described in 4.4.4.

c. Apply the mined Label Sequential Rules and Class Sequential Rules on S_A and S_B, to assign a class to n_l, i.e., to determine if it is an EDR Location NE. Details are given in 4.4.6.

4.4.3 Data annotation

Each candidate Location NE is annotated manually with a class c, $c \in C$ = {EDR-LOC, nonEDR-LOC, non-LOC}, where EDR-LOC represents EDR Location NE, nonEDR-LOC represents nonEDR Location NE and non-LOC represents anything that is not Location NE.

4.4.4 Sequence data generation

For a candidate Location NE n_l and a disease NE n_d, if n_l is annotated with a class c, and the node path is $path(n_1, n_d) = \langle \{n_1\}\ \{n_2\}\ \ldots\ \{n_m\} \rangle$, then two sequences S_A and S_B are generated.

Each itemset in S_A corresponds to a node in $path(n_1, n_d)$ following the same order. The candidate Location NE and Disease NE are converted to special itemset dedicated to its NE type, and other node is converted to an itemset consisting of the node's literal word and Part-Of-Speech tagging. Formally,

$$S_A = \langle conv1(n_1)\ \ conv1(n_2)\ \ \ldots\ \ \ conv1(n_m) \rangle$$

where $conv(n_i)$ =

- $\{TOKEN_d\}$ if n_i is a disease NE;
- $\{TOKEN_c\}$ if n_i is a candidate Location NE and $i \neq 1$ and $i \neq m$;
- $\{n_i$'s class$\}$ if n_i is a candidate Location NE and (i=1 or i=m);
- $\{n_i.word, n_i.POS\}$ otherwise.

$TOKEN_d$ and $TOKEN_c$ are two special itemsets dedicated to Disease NE and candidate Location NE, respectively.

S_B has exactly one itemset, consisting of the candidate Location NE's word.

$$S_B = \langle \{n_l.word\} \rangle \text{ with class} = c$$

For node path used in training, being converted to sequences, S_A is added to SD_A, and S_B with class c is added to SD_B.

4.4.5 Rule mining and rule matching

4.4.5.1 Label sequential rule mining

Label sequential rules of the following forms are mined from SD_A with the user-specified $minsup_A$ and $minconf_A$.

$$\langle \{TOKEN_d\} s \{*\} \rangle \rightarrow \langle \{TOKEN_d\} s \{c\} \rangle$$

$$\langle \{*\} s \{TOKEN_d\} \rangle \rightarrow \langle \{c\} s \{TOKEN_d\} \rangle$$

where $c \in C$ and s is a subsequence (or empty sequence). We call c the associated label of the LSR.

All mined LSRs are sorted first by confidence and then by support in a decreasing order. The set of rules can be used for extraction, etc, to extract the node in the test sequence that matches *.

4.4.5.2 Class sequential rule mining

Class sequential rules are mined from SD_B with the $minsup_B$ and $minconf_B$. Likewise the rules are sorted first by confidence and then by support in decreasing order. We use the same settings here: $minsup_B = minsup_A$ and $minconf_B = minconf_B$.

4.4.5.3 Rule matching

Recall the definition of subsequence and supersequence, we define a class sequential rule *csr matches* a sequence *s* if *s* contains the left hand side of *csr*, which is the same as the conventional notion. However, for the mined label sequential rule, we have a slightly different definition of matching.

For those label sequential rules mined from SD_A, we define a label sequential rule *lsr matches* with a sequence *s* if *s* contains the left hand side of *lsr*, and their first itemset are the same, and their last itemset are the same. Formally,

A label sequential rule *lsr* of the form $\langle a_1 a_2 \ldots a_r \rangle \rightarrow Z$ *matches* sequence *s* ($s = \langle b_1 b_2 \ldots b_m \rangle$) if:

- exist integers $1 \leq j_1 < j_2 < \ldots < j_{r-1} \leq j_r \leq m$ such that $a_1 \subseteq b_{j_1}$, $a_2 \subseteq b_{j_2}$, …, $a_r \subseteq b_{j_r}$, and
- $a_1 = b_1$, and
- $a_r = b_m$.

4.4.6 Apply rules

Given a test sentence, we build its dependency tree and identify Disease NE and candidate Location NE from the tree. For each pair of disease NE n_d and candidate Location NE n_l, its node path is built from the dependency tree, if there is any. Sequences

S_A and S_B are generated as described above. Now with S_A, S_B, the mined Label Sequential Rules, and Class Sequential Rules, we apply the rules in the following way:

a. First try to find the LSR with the highest confidence that matches S_A, and return the associate label of the LSR as the class of n_l.

b. For example, we have mined the following LSR r:

$\langle \{*\}\ \{\text{report, VERB}\}\ \{TOKEN_d\} \rangle \rightarrow \langle \{\text{EDR-LOC}\}\ \{\text{report, VERB}\}\ \{TOKEN_d\} \rangle$

c. then sequence $S_1 = \langle \{TOKEN_c\}\ \{\text{report, VERB}\}\ \{TOKEN_d\} \rangle$ is matched by r, and EDR-LOC will be returned as the class of the first node in S_1.

d. If no LSR matching S_A is found in the above step, we try to find the CSR cr with the highest confidence that matches S_B and return cr's associated class as the result of n_l. The mined CSR rules basically determine whether a word or phrase is an EDR Location NE or not based on the class distribution of the annotated training data. It does not use any path in the dependency tree.

e. If still no CSR matching S_A is found, we return the default class $c_{default}$ as the class of the candidate Location NE. $c_{default}$ is a pre-set value, and $c_{default} \in C$. In our experiment, we will compare the performance of the technique by using different value of $c_{default}$.

If the returned class is EDR-LOC, the candidate location is an EDR Location; if the return class is nonEDR-LOC or non-LOC, the candidate location is not an EDR Location NE, and it could be a nonEDR Location NE or it may not be a Location NE at all.

4.5 Experiments

4.5.1 Data collection and tagging

We manually collected 1158 sentences from EDR news (which report disease outbreaks) based on the condition described in Section 4.2, in which 562 are from Google News (Google News, 2008) and 596 are from ProMED-mail (ProMED-mail, 2007). Each sentence always contains at least one disease NE and one candidate Location NE. For each sentence, node paths between all pairs of disease NEs and candidate Location NEs are constructed, and the candidate Location NE in each node path is manually annotated with a class c, $c \in C$ and C = {EDR-LOC, nonEDR-LOC, non-LOC}. TABLE IV shows the distribution of the classes in the tagged data.

4.5.2 Conditional random fields

We compare our method with CRF, which has been reported as one of the best methods for information extraction (Mooney and Razvan, 2005). As we have mentioned in related work section, there is also an existing system for infectious disease outbreak extraction (Grishman et al., 2002). However, the system is giving poor performance. We

TABLE IV CLASS DISTRIBUTION OF TAGGED DATA

EDR-LOC	nonEDR-LOC	non-LOC	Total
1168	25	511	1705

will show that our method is giving much better results, but a direct comparison with the existing system is not possible, since their system and dataset are not available.

CRF requires two inputs, a raw sequence and a tagged sequence. Suppose $path(n_1, n_d) = \langle n_1\ n_2\ \ldots\ n_m \rangle$, then we create a raw sequence S_C in the following way:

$$S_C = \langle conv2(n_1)\ conv2(n_2)\ \ldots\ conv2(n_m) \rangle$$

where $conv2(n_i)$ =

- $\{TOKEN_d\}$ if n_i is a disease NE;
- $\{n_i.\text{word}\}$ if n_i is a candidate Location NE;
- $\{n_i.\text{word}+'/'+n_i.\text{POS}\}$ otherwise.

Here '+' is a string concatenation operator.

Another way to create S_C is by replacing the candidate Location NE's word with a unique token, i.e., let

$$S_C = \langle conv3(n_1)\ conv3(n_2)\ \ldots\ conv3(n_m) \rangle$$

where $conv3(n_i)$ =

- $\{TOKEN_c\}$ if n_i is a candidate Location NE;
- $conv2(n_i)$ otherwise.

S_C's tagged sequence is:

$$T_C = \langle t(n_1)\ t(n_2)\ \ldots\ t(n_m)\rangle$$

where $t(n_i)$ =

- $\{TOKEN_d\}$ if n_i is a disease NE;
- $\{n_i.class\}$ if n_i is a candidate Location NE;
- {non-LOC} otherwise.

We use the CRF package developed by Sarawagi (Sarawagi, 2004) in our experiments.

4.5.3 Experimental settings

For CRF, we experimented with both ways of sequence construction (*conv2* and *conv3*). For our method, we experimented all combinations of $minsup_A$=*0.014* and *0.02*, $minconf_A$= 0.8, and $c_{default}$ = EDR-LOC and non-LOC. Note that $minsup_A$= 0.014 gives the best results, but any support below 2% (0.02) produce similar results. We have also experimented with switching the order of CSR and LSR, i.e., apply LSR before CSR. For both methods, five-fold cross validations are used.

4.5.4 Experimental results

The average precision, recall, and F-value results are reported in TABLE V (based on five-fold cross validation). We observed that our method is more effective. All the results of our method achieve 5% to 8% higher F-score comparing with CRF. Of the two CRF sequence construction methods, *conv2* gives higher precision, lower recall, and a higher F-score overall. Among all the parameters settings, our method got the best result when LSR is applied before CSR and $minsup_A$ is 0.014. Changing the default class $c_{default}$ from EDR-LOC to non-LOC increases the recall and decreases the precision, but does not influence the F-score.

4.6 Summary

We introduce a novel technique to extract EDR Location NE from EDR Sentences. This technique uses label sequential rule and class sequential rule, and the sequences are derived from dependency tree of a sentence. Experimental results show that this technique outperforms the Conditional Random Fields on our data.

TABLE V EVALUATION ON EXTRACTION OF EDR LOCATION NE

Method	$minsup_A$	$c_{default}$	P[a]	R[b]	F[c]
CSR before LSR	0.014	EDR-LOC	0.787	0.977	0.871
		non-LOC	0.835	0.921	0.875
	0.02	EDR-LOC	0.779	0.980	0.867
		non-LOC	0.828	0.915	0.868
LSR before CSR	0.014	EDR-LOC	0.798	0.971	0.876
		non-LOC	0.833	0.931	0.879
	0.02	EDR-LOC	0.789	0.979	0.873
		non-LOC	0.824	0.931	0.874
CRF_1[d]	N/A		0.763	0.878	0.815
CRF_2[e]			0.683	0.964	0.797

[a] P stands for Precision.

[b] R stands for Recall.

[c] F stands for F-score.

[d] CRF_1 uses *conv*2 in the CRF input construction.

[e] CRF_2 uses *conv*3 in the CRF input construction.

5 COMBINE CLASSIFICATION AND EXTRACTION TOGETHER

In the previous two chapters, we have discussed how to classify EDR Sentences from nonEDR Sentences and how to extract EDR Location NE from EDR Sentences. A very interesting question follows these two chapters: can we do EDR Sentence classification and EDR Location NE extraction together?

We think the answer is yes. In this chapter, we propose a novel technique based on label sequential rule to classify EDR Sentence and extract EDR Location NE together. We also discuss how to geocode the extracted EDR Location NEs, and how to disambiguate the geocoding. At last, we have a system using the new technique for classification and extraction, together with the geocoding and disambiguation algorithm, and we evaluate the performance of the system on manually annotated data.

5.1 Doing Classification and Extraction Together?

Recall in the last chapter, we build node paths between Disease NEs and candidate Location NEs from EDR Sentences' dependency trees, mine label sequential rules from the training data's node paths, and then apply the mined rules on testing data's node paths. The idea is that the label sequential rules can capture the expressions patterns that report disease outbreak in a certain location. If we follow the same idea, include

nonEDR Sentences in training data, we can then use mined label sequential rules to extract EDR Location NE directly from the mixed sentences consisting of both EDR Sentences and nonEDR Sentences, essentially, classification and extraction are done at the same time.

5.1.1 Training

The training process is similar as 4.4.1, except that training sentences include both EDR and nonEDR Sentences, and only one sequence database is built.

a. From training sentences (including EDR Sentences and nonEDR Sentences), annotate the candidate Location NE, and build node path between each pair of Disease NE and candidate Location NE, if there is any.

b. Construct sequence database SD_A from the node paths. Every node path corresponds to one sequence in SD_A.

c. Mine Label Sequential Rules from SD_A.

5.1.2 Testing

The testing process is similar as 4.4.2, except that testing sentences include both EDR and nonEDR Sentences, and only sequence S_A is created.

a. For each test sentence, we first build its dependency tree, and then identify all candidate Location NE and Disease NE from the tree. For

each pair of candidate Location NE n_l and Disease NE n_d in the same tree, build their node path $path(n_l, n_d)$, if there is any.

b. Each pathnode $path(n_l, n_d)$ is converted to sequence S_A the same way as described in 4.4.4.

c. Apply the mined Label Sequential Rules on S_A, to assign a class to n_l, i.e., to determine if it is an EDR Location NE.

5.1.3 Data annotation

Each candidate Location NE is annotated manually with a class c, $c \in C$ = {EDR-LOC, nonEDR-LOC, non-EDRLOC}, where EDR-LOC represents EDR Location NE, nonEDR-LOC represents nonEDR Location NE and non-EDRLOC represents anything that is not EDR Location NE. Clearly, any NE of the class nonEDR-LOC is also of the class non-EDRLOC. A candidate Location NE in EDR Sentence can be annotated with any of these three classes, while every candidate Location NE in nonEDR Sentences is be annotated with non-EDRLOC.

Besides annotating candidate Location NE, each sentence is annotated as EDR Sentence or nonEDR Sentence, each news is annotated as EDR News or nonEDR News, and for EDR News, the outbreak's locations are annotated.

5.1.4 Sequence data generation

Sequence S_A is built exactly the same way as 4.4.4. If the sequence is used in training, it's added to SD_A.

5.1.5 Rule mining

Label sequential rules are mined from SD_A, the same way as described in 4.4.5.1.

5.1.6 Negation node path

One kind of nonEDR Sentences are EDR Negation Sentences, as we mentioned in 1.2.3. To negate disease outbreak, these sentences usually contain explicit negation words, such as "no" and "not". However, the node path we constructed cannot represent these negation words, because the node path between a Disease NE and a candidate Location NE does not include modifier nodes, where negation words frequently appear.

We define a negation node path as a node path that one of its nodes is a negation word itself or has a negation word as modifier, and it can be identified by scanning each node in the path and its sub tree. In a negation node path, the candidate Location NE in the node path is not likely to be an EDR Location NE, and this can help us in extraction. Note that negation in node path rather than negation in sentence is used, because a sentence can be very long, and a negation word appearing in a sentence, does not necessarily mean the whole sentence is of negative meaning.

5.2 Revisit Named Entity

In the previous research, we have found some weakness of the way we identity and use the named entities. In this section we revisit this problem and propose some improvements.

5.2.1 Disease named entity

In a large annotated dataset which is introduced in 5.4, we observe some EDR Sentences do not contain any specific Disease NE, and here are two examples:

> An undiagnosed disease that has affected 200 and left at least four dead in Cacuaco, about 20km north of the Angolan capital, Luanda, has health organizations scrambling to identify the illness.
>
> 150 people were sickened and sent to hospital.

In the first sentence, though no specific disease name is present, it can still be identified as “an undiagnosed disease”. In the second sentence, however, no disease is mentioned at all, and we need to look at other sentences in the same news in order to identify the disease name.

Among 1942 EDR Sentences in our tagged dataset, there are 287 EDR Sentences (14.8%) without any Disease NE, and among 981 EDR News, there are 14 EDR News (1.4%) without any Disease NE. In order to address this problem, we enlarge the disease names we collected from the Centers for Disease Control and Prevention (Centers for

Disease Control and Prevention, 2008) to include several general words to express diseases such as “illness”, “disease”, and “outbreak”, and treat these words as Disease NE. With the enlarged disease name lists, the number of EDR Sentences without any Disease NE is reduced to 189 (9.7%) and the number of EDR News without any Disease NE is reduced to 2 (0.2%).

5.2.2 Candidate location named entity

The candidate Location NE found following 4.3.2.2 is observed to cover most Location NEs, but also include many false positives, i.e., words taken as candidate Location NEs but are not Location NEs. Here we explore several ways to reduce the number of false positives.

5.2.2.1 Location named entity that can be mapped to a geographical location

Since we are interested in the geographical locations of outbreaks, those candidate Location NEs that cannot be mapped to any geographical location are of no use and can be safely ignored. Thus, we check the candidate Location NEs in our geographical database and if no entry were found, we remove its candidacy. The checking is done as follows:

a. Check if the word or the first word of the phrase is an adjective form of location. If so, return it as a candidate Location NE, otherwise go to the next step. We compiled a list of 234 adjective form locations manually.

b. Search GeoNames Database for any entry matching the word or phrase, which will be covered later in 5.3.2.

5.2.2.2 Use other named entity taggers to reduce false positives

The GeoNames Database is very comprehensive, and a side effect of it is that we may find some geographical locations with names that are frequently used words but meaning something other than locations. For example, there is a populated place in France called “This”, and another populated place in India is named “Her”. These two are very frequently used words and when they are capitalized as the first word of a sentence, we will take them as candidate Location NE according to our previous definition. To the best of our knowledge, MINIPAR’s Named Entity recognition is based on dictionary and grammar, rather than the statistics learning.

To overcome this problem, we use a Named Entity tagger NER Package (NER Package, 2008) to remove those candidate Location NEs not being recognized as named entities by NER Package, with the following exceptions:

- The candidate Location NE does not have a suffix of location unit word such as “state” and “district”.
- The candidate Location NE does not have a parent node in the dependency tree with preposition word such as “in” and “at”.

5.2.2.3 Other post-process for candidate location named entity

A real Location NE should be used with its first letter capitalized at all time. Based on this observation, if a candidate Location NE's word or phrase appearing in the same news and without its first letter capitalized, it thus should not be a candidate Location NE.

5.3 Geocoding of Location Named Entity.

After we extracted EDR Location NEs, the next step is to map these Location NEs to one or multiple geographical locations. This task may seem easy, as all need to do is to search a geographical database and find the geographical locations for the given Location NEs. This is only true if the mapping relationship between Location NE and geographical location is one-to-one, but unfortunately, the relationship is many-to-one in many cases. Especially, when in a comprehensive geographical database such as GeoNames Database, this ambiguity problem can be very severe. For example, searching "Norfolk" for exact match returns 23 entries from 7 different countries, and if we relax the search to match "Norfolk" as the first word, the number of matched entries soars to 136.

A compromised solution is to use a rather simplified geographical database, which contains no ambiguity. This is the method that most existing health monitoring systems are using. For instance, Health Map uses a dictionary consisting of only 2,300

locations (Freifeld et al., 2008), and Global Health Monitor's geographical database includes only 4,268 locations (Doan et al., 2008).

5.3.1 Problem statement

Before give the problem statement, we have the following definitions.

A geographical location is represented by a geographical entry *g*. Given geographical entries g_1 and g_2, g_1 *belongs to* g_2 if g_1's geographical location is inside g_2's, e.g., "Chicago" is inside "Illinois" and "Chicago" is inside "United States".

If a Location NE *n* refers to a geographical location that is represented by an entry *g*, we say *g* covers *n*. There may be more than one geographical entry covering the same Location NE.

Now given a set of Location NEs *N* and a set of geographical entries *H*, the goal of geocoding is to find a set of geographical entries *G* ($G \subseteq H$) satisfying both of the following two conditions.

- For each $n \in N$, either there exists *g* ($g \in G$) such that *g* covers *n*, or there exists *g* ($g \in G$) and *g'* ($g' \notin G$), such that *g'* covers *n* and *g* belongs to *g'*.
- For each $g \in G$, there exists *n* ($n \in N$) such that *g* covers *n*.

Most of the time, the sets satisfy the criteria are not unique, and this ambiguity needs to be resolved, which will be studied in 5.3.3.

5.3.2 GeoNames database

The GeoNames geographical database (GeoNames geographical database, 2008) integrates geographical data such as names of places in various languages, elevation, and population. It currently contains over eight million geographical names, which cover 645 different features, such as political entity, populated area, hospital and school. The GeoNames Database is not complete yet, and its data distribution is very unbalanced, for example, 1.8 millions of the entries are from United States. Each entry in GeoNames contains information such as location name, country name, administration level, feature type, longitude and latitude.

To query the GeoNames Database with a single Location NE is relatively easy. First query for entry with the name equals to the location NE. Second, if the Location contains an organization or a location unit as suffix, remove the suffix and query for the remaining name. The reason for the second step is that for Location NE of the form "location + organization", if no entry with the full name can be found, there is a chance that the "location" has a matched entry. For example, "Caura Hospital" is not in GeoNames Database, but by removing "Hospital", we can find "Caura" as an entry. Similarly, location units can be used as suffix in some Location NEs such as "X City" and "Y District". We use heuristic rules to identify the organization or location units.

Since the suffix may tell us the location's feature type or the administration level, they are also used to filter search results. For example, after we remove the location unit from "Washington State", there are multiple entries with the name "Washington", and because the location unit is "State", only the entry whose administration level is state will be returned.

5.3.3 Disambiguation of geocoding.

In geocoding, one way to reduce the ambiguity is to require the geographical entry set to be of minimum size, to which we give a simple greedy algorithm in Figure 6.

INPUT: Location NE set N and geographical entry set H.
OUTPUT: Geocoding of N.

```
1   Let Q = N, G = {}
2   While Q is not empty{
3       Find n (n ∈ N) and g (g ∈ G) with maximum |A(Q, g)|
4       Remove n from Q, add g to G
5   }
6   return G.
```

Figure 6. A greedy algorithm for disambiguation of geocoding.

$A(Q, g)$ is the subset of Q, and each Location NE in the subset is either covered by g or covered by an entry that g belongs to. Formally,

$A(Q, g) = \{n \mid n \in Q$ and (g covers n or exits $g'(g' \in Q)$ g belongs to g' and g' covers $n)\}$

In line 3 of the algorithm, there may exist multiple pairs of n and g with the same maximum $|A(Q, g)|$, and our selection method is based on the following two observations.

One observation is that a news article normally refers to a location unambiguously from its readers' point view, so the ambiguity can usually be resolved if we know the location of the article's major audiences, which we call the *background location* of the news. For example, if "Norfolk" appears in a British national newspaper, it's likely to refer to the Norfolk in Britain, other than Norfolk in other countries. Another observation is that when referring a place in a different country, that country's name is usually appearing in the news, as well.

If a background location is known, for multiple pairs of n and g, our selection is thus in favor of the g that is in the same city (or county, or state, or country) as the background location, and not in favor of the g that is in a different county as the background location if the county name is not in N.

If multiple pairs still remain, we select the g of the highest administration level and the biggest population.

To get the background location of news, we first search the URL of the news with a database of news sites, if no hit is found, we check the country code in the URL. We

compiled a database of 3519 world newspapers, and a database of 213 country/area's country code.

5.4 Experiments

5.4.1 Dataset collection

Our dataset consists of 2052 Health news from Google News between Dec. 1 2007 and Dec. 31 2007. For each news, we extracted the first paragraph of the content and the title from the HTML page, and split the content into sentences.

5.4.1.1 Crawling

We crawled rss feed for Google News' Health section every one hour and other rss feeds with search for infectious diseases in Google News every 24 hours. Crawling time and news URL are stored with the web pages. For web pages with the same URL, only the first crawled one is kept. This ensures the crawling time approximately the same as the time when the news is published.

5.4.1.2 News content extraction from web page

From the crawled web pages, the title and content of the news are extracted using a heuristic algorithm. The main idea is that the content of a news web page usually has much less hyper links and much more text comparing to non-content things such as sidebars and navigation bars. The news title must come before the content, and it may be available on the HTML's title node that can be identified directly. The heuristic algorithm

is able to find the start of the content most of the time, but it can't find the end of the content very well. So from the start of the content, we extract text from the first DOM node, which usually contains the first paragraph of the content. For news with different URLs but exactly the same extracted text, again, only the first one is kept. After the content is extracted from web page, we split it into multiple sentences using the Sentence Segmentation Tool (Sentence Segmentation Tool, 2008).

If a news title is found, it will be used the same way as a sentence, unless its words are all capitalized. Because both the dependency tree builder and the named entity tagger have very poor performance on sentences whose words are all capitalized.

5.4.2 Dataset annotation

We annotate the dataset on both classification and extraction.

Every news in the dataset is annotated as either EDR or nonEDR, and the same annotation is done for sentences. For EDR News, the outbreak locations and their corresponding geographical entries are annotated. In EDR Sentence, each location is annotated as either EDR-LOC or nonEDR-LOC. We did not manually annotate any named entity in nonEDR Sentences, so all candidate Location NEs in a nonEDR Sentence are automatically considered as non-EDRLOC.

Disease NE is recognized using the revised method described in 5.2.1. Candidate Location NE are those words or phrases satisfying the previous conditions outlined in

4.3.2.2 and can be mapped to a geographical location as described in 5.2.2.1. The post process introduced in 5.2.2.2 and 5.2.2.3 to reduce false positive candidate Location NE is going to be applied later in the experiments in order to evaluate its effectiveness.

5.4.3 Dataset statistics

The statistics of the dataset and the annotation is shown in TABLE VI. The EDR class and nonEDR class are equally distributed on news level approximately, but on sentence level, nonEDR Sentences are almost twice as many as EDR Sentences.

TABLE VI DATASET STATISTICS

Category	Class	Number
News	EDR	981
	nonEDR	1071
Sentence	EDR	1942
	nonEDR	3477
Location NE in EDR Sentence	EDR-LOC	2199
	nonEDR-LOC	130

5.4.4 Evaluation

5.4.4.1 Training and testing data splitting

The dataset is divided into five folds by the news' publishing dates with a continuous six or seven days in one fold. All evaluations are done on five cross-validations, and average scores are reported.

5.4.4.2 Classification

A surprising observation is that the traditional text classification methods work very well in classification of the first paragraph, as shown in Figure 7. We experimented the combinations of two grams and three different classifiers. NB with trigram gives the overall best result with 97.0% precision and 93.5% recall. This could be explained as the first paragraph of EDR News usually uses common vocabulary to report disease outbreak. However, using these methods on classification of EDR Sentences still give poor performance, which is illustrated in Figure 8. Though recall is as high as 93.4% by using SVM with trigram, the precision is always lower than 60%. We use Rainbow package (McCallum, 1996) for this classification.

5.4.4.3 Evaluation on extraction

We evaluate the extraction of EDR Location NE using the following methods:

- *TC*: this method first classify the first paragraph, using NB with trigram, which performs the best in 5.4.4.2, and then it simply tags every candidate Location NE in an EDR paragraph as EDR Location NE.

- *TC II*: this method is the same as the previous *TC* method except that we apply the post process on candidate Location NE (introduced in 5.2.2.2 and 5.2.2.3).
- *Rule*: use Label Sequential Rules directly to extract EDR Location NE, as described in 5.1.
- *Rule-TC II*: this combined method only tags a candidate Location NE as EDR Location NE if both the method *TC II* and the method *Rule* tag it as EDR Location NE. Hence, post process on candidate Location NE is also applied within *TC II*.
- *Rule II-TC II*: this method is the same as *Rule-TC II*, except that the node paths are replaced with the nodes between the candidate Location NE and Disease NE in their order of in the flat sentence. This method is used to test if using dependency tree has any advantage against using simply the flat sentences.
- *Rule-TC II-NegationPathRemoval*: from EDR Location NE tagged by the method *Rule-TC II*, this method further removes any EDR Location NE if it is in a negation node path as defined in 5.1.6.

The experimental results are shown in Figure 9. The method *TC* gives the best recall of almost 80%, but a low precision of about 70%. The method *TC II* gives a higher precision with 75.4% with a littler lower recall. This shows that the post process on

candidate Location NE is working. *Rule* alone does not perform well; especially the precision is of the lowest among all methods. Combining *TC II* and *Rule* gives higher precision than that of using *TC II* and *Rule* alone, and with negation node path removal, the best precision of 78.9% is achieved. But the recalls of the combined methods also drop to around 68%. *Rule II-TC II* gives the lowest recall and approximately the same precision as *Rule-TC II* does, which suggests that the sequence derived from dependency tree is more effective than the sequence in flat sentence.

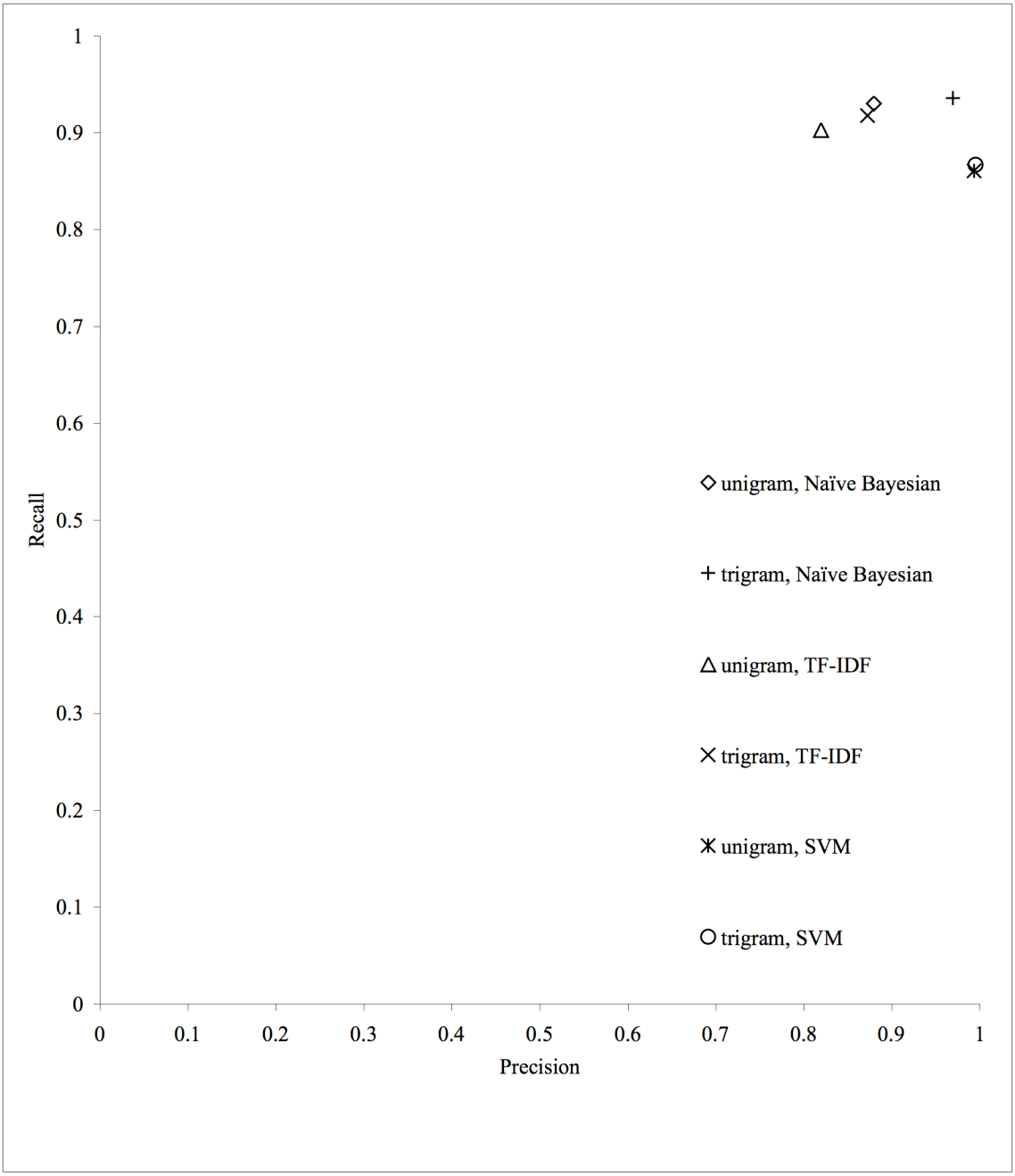

Figure 7. Classification of first paragraph with traditional text classification methods.

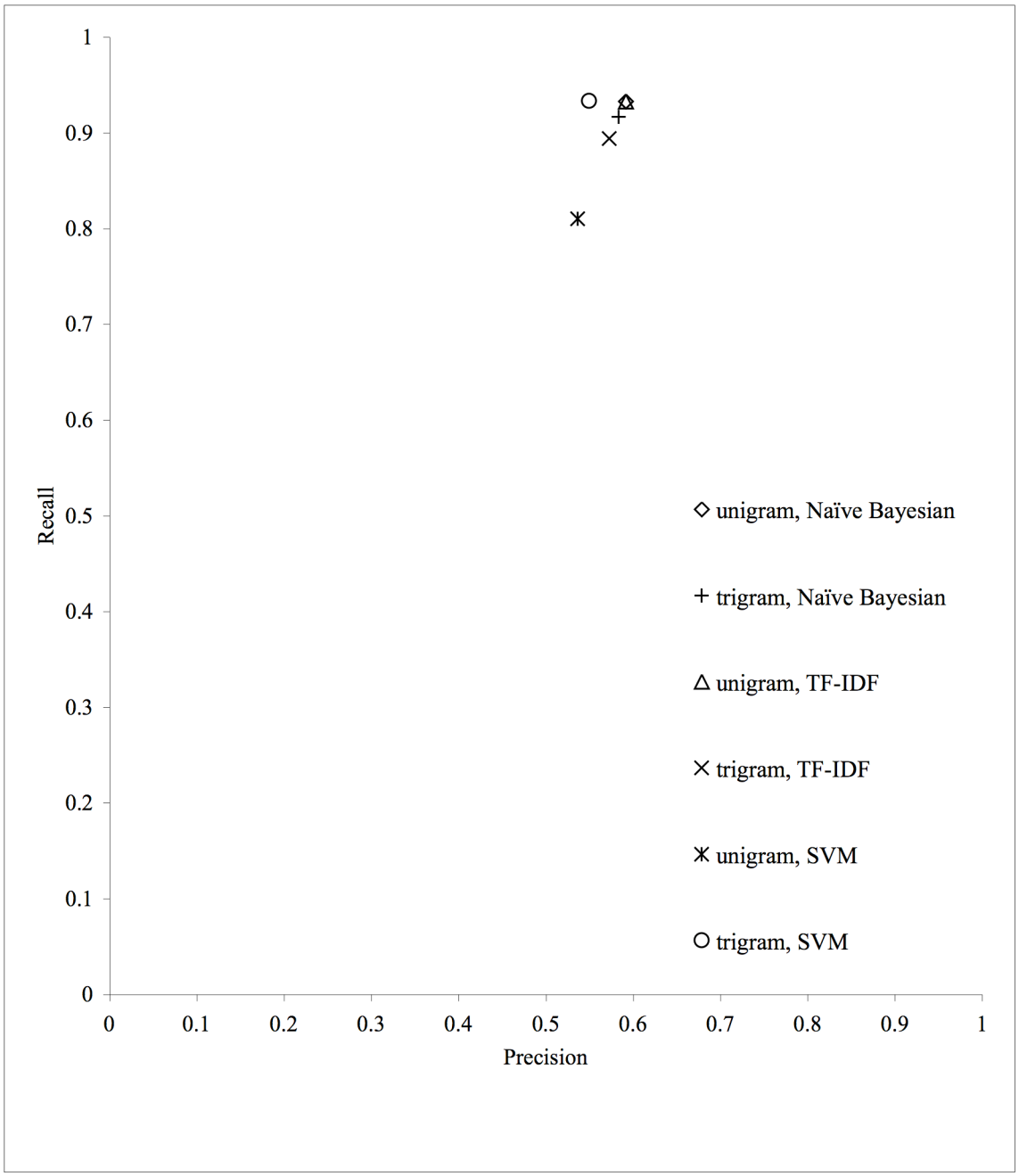

Figure 8. Classification of Emergent Disease Report Sentences with traditional text classification methods.

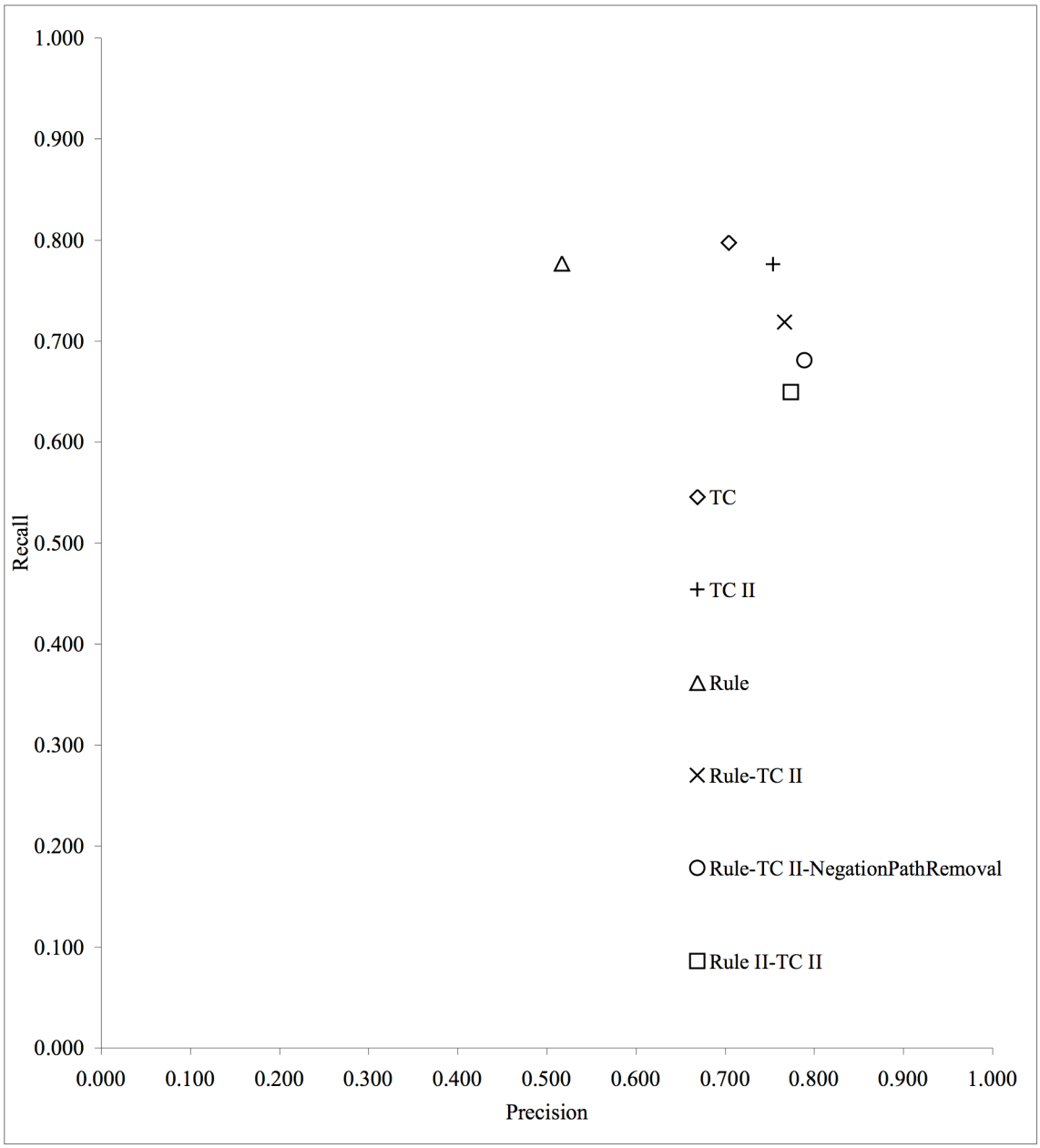

Figure 9. Evaluation on Emergent Disease Report Location Named Entity extraction.

5.4.4.4 Add geocoding into evaluation

Now we evaluate the extraction of EDR Location NE with geocoding, i.e., we evaluate the geographical entries that are identified as EDR Location NEs by geocoding. Evaluation on geographical entries extraction is more complicated than evaluation on EDR Location NE extraction, because the geographical entries form a tree-like structure and two entries' relationship is more than equality and inequality. For example, Chicago City and Naperville are not the same location, but they are both in State of Illinois, so they are equal on the state level. We will evaluate the geographical entry extracted on four levels, namely country level, state level, county level, and name level. Two entries are equal on county level if they have the same county name; they are equal on state level if they have the same state name and the same country name; they are equal on county level if they have the same county name, the same state name, and the same country name; finally, two entries are equal on name level if their names are the same. Thus, for two geographical entries in different countries but with the same name, they are equal on the name level, but not on any other three levels.

All the EDR geographical entries are computed directly from the EDR Location NEs. So the training process is the same as in extraction EDR Location NE, and no training news' EDR geographical entries are used. Since the test set consists of all news in a six days range, besides evaluation of EDR geographical entries by news, we are also interested to know the evaluation by the whole test set. Formally, the testing process is given in Figure 10.

INPUT: TS and M. TS is the testing set and M is a method for extracting EDR Location NE. (TS=$\{TS_1, TS_2, \ldots, TS_k\}$ k is the fold number)

OUPUT: the evaluations on geographical entry extraction by news and by test set.

1 $G_0 = \varnothing$, $G_0' = \varnothing$
2 For i = 1 to k do{
3 For j =1 to $|TS_i|$ do{
4 Get the set of EDR Location NE N from $t_{i,j}$ using M
5 Geocoding N to the set of geographical entries G
6 Disambiguate G
7 G' is the annotated geographical entries of G'
8 If ($G' \neq \varnothing$ and $G \neq \varnothing$){
9 Evaluate G with G', and calculate precision $p_{i,j}$, recall $r_{i,j}$, and f-score $f_{i,j}$.
10 $G_0 = G_0 \cup G$; $G_0' = G_0' \cup\cup G$;
11 }
12 }
13 Evaluate G_0 with G_0', and calculate precision $p_{i,0}$, recall $r_{i,0}$ and f-score $f_{i,0}$.
14 }
15 p = AVERAGE$\{p_{i,j} \mid 1 \le i \le k, 1 \le j \le |TS_i|\}$
16 r = AVERAGE$\{r_{i,j} \mid 1 \le i \le k, 1 \le j \le |TS_i|\}$
17 f = AVERAGE$\{f_{i,j} \mid 1 \le i \le k, 1 \le j \le |TS_i|\}$
18 p_0 = AVERAGE$\{ p_{i,0} \mid 1 \le i \le k\}$
19 r_0 = AVERAGE$\{ r_{i,0} \mid 1 \le i \le k\}$
20 f_0 = AVERAGE$\{ f_{i,0} \mid 1 \le i \le k\}$
21 return (p, r, f) as the evaluation by news and (p_0, r_0, f_0) as the evaluation by test set.

Figure 10. Testing process for evaluation on extraction of geographical entries.

This testing process can be applied on evaluation on any levels.

We select the methods that give good results in EDR Location NE for this evaluation, i.e., methods *TC II*, *Rule-TC II*, and *Rule-TC II-NegationPathRemoval*. *Rule II-TC II* is also tested.

TABLE VII shows the experimental results. When evaluated by news, *TC II* gives the best F-score on all levels, and on country level it achieves the highest F-score of 0.800. *Rule-TC*, the rules based method, continues to give higher precisions on all levels by about 0.02, while the recalls drop by 0.09 on state level and 0.05 on other levels. *Rule-TC-NegationPathRemoval* also performs similarly as in EDR Location extraction, mostly with better precisions than *Rule-TC*, but its recalls are the lowest of the three methods.

Usually, when a disease outbreak happens, it is reported by many different news. So if a method misses a location's outbreak in one news, it may capture it in another news, hence our methods' low recalls on news might be compensated in the evaluation on the whole test set. This is confirmed in the second part of TABLE VII: both *Rule-TC II* and *Rule-TC II-NegationPathRemoval* give closer recalls to those of *TC II* when evaluated by test set. *Rule-TC II-NegationPathRemoval*'s precisions were higher than *TC II*'s by 0.04 to 0.05, with better F-score on country level and very close F-score on other levels.

Rule II-TC II under performance *Rule-TC II* almost for every level, continues to suggest the effectiveness of the sequence derived from dependency tree over that derived from flat sentences.

5.5 Summary

In this chapter, we discuss how to use label sequential rules to classify EDR Sentence and extract EDR Location NE together and how to geocode the extracted EDR Location NE. Improvements on identification of Location NE and Disease NE, and techniques to disambiguate geocoding are also introduced. The experimental results show that the novel technique improves the overall performance on extracting the geographical locations of emergent disease outbreaks.

TABLE VII EVALUATION ON EDR LOCAITON NE EXTRACTION WITH GEOCODING.

Evaluation Type	Method	Country Level			State Level			County Level			Name Level		
		P [a]	R [b]	F [c]	P	R	F	P	R	F	P	R	F
Evaluation by News	*TC II*	0.803	0.798	0.800	0.687	0.636	0.660	0.513	0.420	0.462	0.624	0.633	0.628
	Rule-TC II	0.819	0.734	0.774	0.706	0.543	0.614	0.530	0.376	0.439	0.659	0.594	0.624
	Rule II -TC II	0.809	0.688	0.743	0.701	0.481	0.571	0.526	0.320	0.398	0.638	0.544	0.587
	Rule-TC II-Negation PathRemoval	0.828	0.703	0.760	0.719	0.521	0.604	0.522	0.343	0.414	0.668	0.569	0.614
Evaluation by Test Set	*TC II*	0.778	0.854	0.814	0.729	0.760	0.744	0.489	0.490	0.489	0.556	0.590	0.572
	Rule-TC II	0.812	0.868	0.839	0.762	0.709	0.734	0.525	0.445	0.482	0.593	0.552	0.572
	Rule II-TC II	0.813	0.847	0.830	0.773	0.692	0.730	0.516	0.404	0.453	0.591	0.504	0.544
	Rule-TC II-Negation PathRemoval	0.815	0.861	0.837	0.780	0.713	0.745	0.532	0.439	0.481	0.601	0.538	0.568

[a] P stands for Precision. [b] R stands for Precision. [c] F stands for Precision.

6 CONCLUSIONS AND FUTURE WORK

We investigate the problem of extracting information of emergent disease outbreak from news. Specifically, this work focuses on extracting disease name and outbreak location from Emergent Disease Report. First, we study the problem of classification of the sentences reporting emergent disease outbreak, and propose a new method by integrating semantic features with the bag-of-words scheme. Experimental result shows the integrated approach is better than each individual approach alone. Second, a novel method based on sequential rules is introduced to extract the outbreak locations from the outbreak reporting sentences, and this new method outperforms Conditional Random Fields in our experimental data. Finally, we discuss how to do classification and extraction together using label sequential rules and how to geocode the extracted location named entities into geographical locations accurately. Evaluations on classification-extraction including geocoding are conducted, and the proposed method is shown to improve the overall performance.

We observe some limitations of our work. 1) The sequential rules are mined from paths between disease names and candidate location words, but such path cannot take into account of the remaining words in the same sentence. 2) To build a node path, a sentence must have both disease name and location name, but some sentences don't have

any disease names, e.g., “100 people were sent to hospital in Chicago”, and some sentences only refer to a location, e.g. “Cholera is spotted in this country’s capital”.

For future research, we propose the following directions. 1) Study how to deal with pronoun resolution in multiple sentences. 2) Combine locations consisting of locations at different levels into a single location named entity. For example, “Chicago, IL” should be recognized as a single named entity ideally. 3) Investigate how to use previous news’ outbreak information to improve the classification and extraction accuracy on coming news. Similarly, previous news’ non-outbreak information could also be used to find out nonEDR news.

CITED LITERATURE

20 Newsgroups Dataset. http://kdd.ics.uci.edu/databases/20newsgroups/. 2008.

Agrawal, R., Srikant, S.: Mining Sequential Patterns. The 11th International Conference on Data Engineering. 1994.

Aizawa, A.: Linguistic techniques to improve the performance of automatic text categorization. Proceedings of the Sixth Natural Language Processing Pacific Rim Symposium. 2001.

Automatic Content Extraction. http://www.nist.gov/speech/tests/ace/2007/index.html. 2007.

Ayres, J., Gehrke, J., Yiu, T., Flannick, J.: Sequential PAttern Mining Using Bitmaps. The 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2002.

Brownstein, J. S., Freifeld, C. C.: HealthMap: the development of automated real-time Internet Surveillance for epidemic intelligence. Eurosurveillance. 12(48). 2007.

Bunescu, R., Ge, R., Kate, R., Marcotte, E., Mooney, R., Ramani, A., Wong, Y. W.: Comparative experiments on learning information extractors for proteins and their interactions. Artificial Intelligence in Medicine. volume 33, issue 2. 2005.

Califf, M. E., Mooney, R. J.: Relational learning of pattern-match rules for information extraction. Proceedings of the 16th National Conference on Artificial Intelligence. 1999.

Carenini, G., Ng, R. T., Zwart, E.: Extracting knowledge from evaluative text. The 3rd International Conference on Knowledge Capture. 2005.

Centers for Disease Control and Prevention. http://www.cdc.gov. 2008.

Culotta, A., Sorensen, J.: Dependency tree kernels for relation extraction. Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics. 2004.

Dave, K., Lawrence, S., Pennock, D. M.: Mining the peanut gallery: Opinion extraction and semantic classification of product reviews. The 12th International World Wide Web Conference. 2003.

Doan, S., Hung-Ngo, Q., Kawazoe, A., Collier, N.: Global Health Monitor - A Web-based System for Detecting and Mapping Infectious Diseases. Proceedings of the International Joint Conference on Natural Language Processing. Companion Volume, pp. 951-956. 2008.

Etzioni, O., Cafarella, M., Downey, D., Popescu, A.-M., Shaked, T., Soderland, S., Weld, D. S., Yates., A.: Unsupervised named-entity extraction from the Web: An experimental study. Artificial Intelligence. 165(1). 2005.

Fillmore, C. J., Narayanan, S., Baker, C. F.: What can linguistics contribute to event extraction? 21st Conference on Artificial Intelligence. Workshop on Event Extraction and Synthesis. 2006.

Freifeld, C. C., Mandl, K. D., Reis, B. Y., Brownstein, J. S.: HealthMap: Global Infectious Disease Monitoring through Automated Classification and Visualization of Internet Media Reports. Journal of the American Medical Informatics Association. 15(2):150-157. 2008.

Furnkranz, J., Mitchell, T., Riloff, E.: A case study using linguistic phrases for text categorization on the WWW. AAAI-98 Workshop on Learning for Text Categorization. 1998.

Fürnkranz, J.: A Study Using n-gram Features for Text Categorization. Technical Report OEFAI-TR-98-30. Austrian Research Institute for Artificial Intelligence. 1998.

GeoNames Geographical Database. http://www.geonames.org. Accessed 2008.

Girju, R., Badulescu, A., Moldovan, D.: Automatic Discovery of Part-Whole Relations. Computational Linguistics. 32(1): 83-135. 2006.

Google News. http://news.google.com. Accessed 2008.

Grishman, R., Huttunen, S., Yangarber, R.: Real-Time Event Extraction for Infectious Disease Outbreaks. Proceedings of the second international conference on Human Language Technology Research. 2002a.

Grishman, R., Huttunen, S., Yangarber. R.: Information extraction for enhanced access to disease outbreak reports. Journal of Biomedical Informatics. Volume 35, Issue 4. 2002b.

Grishman, R., Sundheim, B.: Message Understanding Conference-6: a brief history. International Conference On Computational Linguistics, Proceedings of the 16th conference on Computational linguistics. Volume 1, 1996.

Hatzivassiloglou, V., Wiebe. J.: Effects of Adjective Orientation and Gradability on Sentence Subjectivity. International Conference On Computational Linguistics, Proceedings of the 18th conference on Computational linguistics. 2000.

Holub, M., Böhmová, A.: Use of dependency tree structures for the microcontext extraction. Proceedings of the ACL-2000 workshop on Recent advances in natural language processing and information retrieval: held in conjunction with the 38th Annual Meeting of the Association for Computational Linguistics. Volume 11, Page 23-33. 2000.

Hu, M., Liu, B.: Mining and summarizing customer reviews. Proceedings of the 10th ACM SIGKDD international conference on Knowledge discovery and data mining. 2004.

Hulth, A., Megyesi, B.: A Study on Automatically Extracted Keywords in Text Categorization. The Annual Meeting of the Association for Computational Linguistics. 2006.

Ji, H., Grishman, R.: Improving name tagging by reference resolution and relation detection. The 21st International Committee for Computational Linguistic and the 43rd Association for Computational Linguistics. 2005.

Jiang, J., Zhai. C. X.: A Systematic Exploration of the Feature Space for Relation Extraction. Proceedings of the Human Language Technologies and the Annual Conference of the North American Chapter of the Association for Computational Linguistics. 2007.

Jindal, N., Liu, B.: Mining Comparative Sentences and Relations. Proceedings of 21st National Conference on Artificial Intelligence. 2006.

Ko, Y., Park, J., Seo. J.: Improving text categorization using the importance of sentences. Information Processing and Management. 40(1):65–79, 2004.

Kudo, T., Matsumoto, Y.: A boosting algorithm for classification of semi-structured text. Conference on Empirical Methods in Natural Language Processing. 2004.

Lafferty, J., McCallum, A., Pereira, F.: Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. The 18th International Conference on Machine Learning. 2001.

Lewis, D. D.: An evaluation of phrasal and clustered representations on a text categorization task. SIGIR-92, Proceedings of the15th ACM International Conference on Research and Development in Information Retrieval. 1992.

Li, C., Wen, J.-R., Li, H.: Text classification using stochastic keyword generation. Proceedings of the 20th International Conference on Machine Learning. 2003.

Li, W., McCallum, A.: Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. Proceedings of the 7th conference on natural language learning. Volume 4. 2003.

Lin, D., Pantel, P.: Discovery of Inference Rules for Question Answering. Natural Language Engineering. Volume 7-4. 2001.

Liu, B.: Web Data Mining: Exploring Hyperlinks, Contents and Usage Data. Springer, 2006.

Mawudeku, A., Blench, M.: Global Public Health Intelligence Network (GPHIN). 7th Conference of the Association for Machine Translation. 2007.

Maynard, D., Tablan, V., Ursu, C., Cunningham, H., Wilks, Y.: Named Entity Recognition from Diverse Text Types. Recent Advances in Natural Language Processing. 2001.

McCallum, A. K. Bow: A toolkit for statistical language modeling, text retrieval, classification and clustering. http://www.cs.cmu.edu/~mccallum/bow. 1996.

McDonald, R., Pereira, F.: Identifying gene and protein mentions in text using conditional random fields. BMC Bioinformatics. 6(Supplement 1): S6. 2005.

Mihalcea, R., Hassan. S.: Using the essence of texts to improve document classification. Proceedings of the Conference on Recent Advances in Natural Language Processing, 2005.

Mooney, R. J., Bunescu. R.: Mining Knowledge from Text Using Information Extraction. ACM SIGKDD Explorations Newsletter, volume 7, issue 1. 2005.

Moschitti, A., Basili. R.: Complex linguistic features for text classification: A comprehensive study. European Conference on Information Retrieval. 2004.

NER Package. http://l2r.cs.uiuc.edu/~cogcomp/asoftware.php?skey=NE. Accessed 2008.

Ng, V., Cardie C.: Improving machine learning approaches to coreference resolution. Proceedings of the 40th Annual Meeting on Association for Computational Linguistics. 2001.

Ng, V., Dasgupta, S., Niaz Arifin, S. M.: Examining the Role of Linguistic Knowledge Sources in the Automatic Identification and Classification of Reviews. Proceedings of the COLING/ACL on Main conference poster sessions. 2006.

Okanohara, D., Miyao, Y., Tsuruoka, Y., Tsujii, J.: Improving the Scalability of Semi-Markov Conditional Random Fields for Named Entity Recognition. The 21st International Committee for Computational Linguistic and the 44th annual meeting of the Association for Computational Linguistics. 2006.

Pang, B. Lee, L., Vaithyanathan, S.: Thumbs up? Sentiment classification using machine learning techniques. Conference on Empirical Methods in Natural Language. 2002.

Pantel, P., Pennacchiotti, M.: Espresso: leveraging generic patterns for automatically harvesting semantic relations. The 21st International Committee for Computational Linguistic and the 44th annual meeting of the Association for Computational Linguistics. 2006.

Peng, F., McCallum, A.: Accurate information extraction from research papers using conditional random fields. Proceeding of Human Language Technology Conference / North American Association for Computational Linguistics Annual Meeting. 2004.

Popescu, A.-M., Etzioni, O.: Extracting Product Features and Opinions from Reviews. Conference on Human Language Technology and Empirical Methods in Natural Language. 2005.

ProMED-mail. West Nile virus, humans - USA (Louisiana). 2002. 12 Jul: 20020712.4737. http://www.promedmail.org/. Accessed 2007.

Reuters-21578 Dataset. http://kdd.ics.uci.edu/databases/reuters21578/. 2008.

Riloff, E., Wiebe. J.: Learning extraction patterns for subjective expressions. Conference on Empirical Methods in Natural Language. 2003.

Sahlgren, M., Coster. R.: Using bag-of-concepts to improve the performance of support vector machines in text categorization. The 20th International Conference on Computational Linguistics. 2004.

Sarawagi, S., Cohen, W. W.: Semi-markov conditional random fields for information extraction. Advances in Neural Information Processing Systems. 2005.

Sarawagi, S.: The crf project: a java implementation. http://crf.sourceforge.net. 2004.

Sebastiani, F.: Machine learning in automated text categorization. ACM Computing Surveys. 34(1):1–47, 2002.

Sekine, S., Sudo, K., Nobata, C.: Extended Named Entity Hierarchy. Proceedings of Third International Conference on Language Resources and Evaluation. 2002.

Sentence Segmentation Tool. http://l2r.cs.uiuc.edu/~cogcomp/atool.php?tkey=SS. Accessed 2008.

Skounakis, M., Craven, M., Ray, S.: Hierarchical Hidden Markov Models for Information Extraction. Proceedings of the 18th International Joint Conference on Artificial Intelligence. 2003.

Soon, W. M., Ng, H. T.: A machine learning approach to coreference resolution of noun phrases. Computational Linguistics. 27(4). 2001.

Sudo, K., Sekine, S., Grishman, R.: Automatic Pattern Acquisition for Japanese Information Extraction. Proceedings of Human Language Technology. 2001.

Sutton, C., McCallum, A.: An introduction to conditional random fields for relational learning. Introduction to Statistical Relational Learning. MIT Press. 2007.

Turney, P.: Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews. Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics. 2002.

Yang, Y., Liu, X.: A re-examination of text categorization methods. Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval.1999.

Zelenko, D., Aone, C., Richardella, A.: Kernel methods for relation extraction. Journal of Machine Learning Research. 2003.

Zhang, Y., Liu, B.: Semantic Text Classification of Emergent Disease Reports. The 11th European Conference on Principles and Practice of Knowledge Discovery. 2007.

Zhao, S., Grishman, R.: Extracting relations with integrated information using kernel methods. Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics. 2005.

Zhou, G., Su, J.: Named entity recognition using an HMM-based chunk tagger. Proceedings of the 40th Annual Meeting on Association for Computational Linguistics. 2001.

VITA

NAME:	Yi Zhang
EDUCATION:	B.E., Computer Science and Technology, University of Science and Technology of China, Hefei, China, 2002 Ph.D., Computer Science, University of Illinois at Chicago, Chicago, 2008
TEACHING:	Department of Computer Science, University of Illinois at Chicago, Chicago, Illinois: Database Systems, 2003 Department of Computer Science, University of Illinois at Chicago, Chicago, Illinois: FORTRAN Programming for Engineers, 2004
PUBLICATIONS:	Zhang, Y., Liu, B.: Semantic Text Classification of Emergent Disease Reports. ECML/PKDD. 2007. Zhang, Y., Liu, B.: Semantic Text Classification of Disease Reporting. SIGIR-07 Poster. 2007. D'Souza, M., Glass, E., Syed, M. Zhang, Y., Rodriguez, A., Maltsev, N., Galperin, M. Y.: Sentra: a database of signal transduction proteins for comparative genome analysis. Nucleic Acids Research. (35) Database-Issue: 271-273, 2007. Maltsev, N., Glass, E., Sulakhe, D., Rodriguez, A., Syed, M., Bompada, T., Zhang, Y., D'Souza, M.: PUMA2 - grid-based high-

	throughput analysis of genomes and metabolic pathways. Nucleic Acids Research. (34) Database-Issue: 369-372, 2006. Berman, P., DasGupta, B., Mubayi, D., Sloan, R., Turan, G., Zhang, Y.: The inverse protein folding problem on 2D and 3D lattices. Discrete Applied Mathematics. 1556-7: 719-732, 2007 DasGupta, B., Enciso, G., Sontag, E., Zhang, Y.: Algorithmic and Complexity Results for Decompositions of Biological Networks into Monotone Subsystems. Fifth International Workshop on Experimental Algorithm. 253-264, 2006. Berman, P., DasGupta, B., Mubayi, D., Sloan, R., Turan, G., Zhang, Y.: The Protein Sequence Design Problem in Canonical Model on 2D and 3D Lattices. 15th Annual Combinatorial Pattern Matching Symposium. 2004.

AUTOMATIC EXTRACTION OF OUTBREAK INFORMATION FROM NEWS

Yi Zhang, Ph.D.
Department of Computer Science
University of Illinois at Chicago
Chicago, Illinois (2008)

Dissertation Chairperson: Dr. Bing Liu

With the explosion of unstructured data on the Web, especially in the form of text, there has been a lot of interest to mine knowledge from these data for variety of purposes. In this thesis, we study a particular problem: how to extract disease outbreak information from news.

By defining Emergent Disease Report, we focus on extracting disease name and outbreak location from the news report emergent disease outbreaks. First, we study the problem how to classify those sentences reporting disease outbreak, and propose to a new method by integrating semantic features with the bag-of-words scheme. Experimental result shows the integrated approach is better than each individual approach alone. Second, a novel method based on sequential rules is introduced to extract the outbreak locations from the outbreak reporting sentences, and the new method outperforms conditional random fields in our experimental data. Finally, we discuss how to do classification and extraction together using label sequential rules and how to geocode the extracted location named entities into geographical locations accurately. Evaluations on classification-extraction including geocoding are conducted, and the proposed method is shown to improve the overall performance.