Extracting Locations of Disease Outbreaks in News Articles

Yi Zhang and Bing Liu
Department of Computer Science
University of Illinois at Chicago

Abstract

This paper studies the problem of extracting locations of disease outbreaks from news articles. A novel technique is proposed based on two types of supervised sequential rule mining, i.e., label sequential rules and class sequential rules. The two types of rules are also combined for extraction. In learning, instead of using sentences as the training sequences, paths from dependency trees are employed for the purpose. Our experimental results based on a large number of health news from Google News and reports from ProMED-mail show that the proposed technique works effectively. It outperforms the state-of-the-art information extraction technique conditional random field dramatically.

1 Introduction

Although information extraction from text documents has been studied by many researchers, it remains to be a challenging problem. In this paper, we study a particular problem, i.e., extracting locations of disease outbreaks from news articles.

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.

Technically, this task involves two main steps. 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).

Since it is possible to obtain a list of known disease names, our task is thus confined to extract only the location of the disease. Also, since we aim to monitor the news stream constantly, the outbreak time is almost always the time near to the press time. In most cases, there is not a precise date and time that an outbreak occurs. The time that the disease is reported in the news is sufficiently informative. We thus use the press time of the first report of the disease. The problem studied in this paper is defined as follows:

Problem definition: Given a news article that contains a report of disease outbreak, extract the location of the outbreak.

Note that it is assumed that the given news article contains a disease outbreak report. Clearly, the first step of identifying whether a news article reports a disease outbreak is also a challenging problem (which can be dealt with as a super-

vised learning problem). The problem has been studied in (Zhang and Liu, 2007).

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 paper 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 Section 4 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 embeds the essential relationships of concepts in a sentence, while the information contained in a raw sentence can be quite diverse, which makes it difficult to mine key patterns to be used for extraction. The details will be given in Section 3.

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

1. 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 Section 3. This sentence is usually the first or the second sentence in the news report, which is not surprising.
2. 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 "National Center for Preparedness, Detection, and Control of Infectious Diseases" (http://www.cdc.gov/ncpdcid/).
3. Extract relevant paths of each dependency tree to form the sequence data for mining and learning. The detail will be discussed in Section 3.
4. 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.
5. Test the classifier using unseen test data using cross-validation to assess its precision recall and F value.

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 in 17 days and historic reports from ProMED-mail. 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.

2 Related Work

Information extraction aims at extracting structured data from unstructured data such as text. Common tasks include named entity extraction and relation extraction. Entity extraction locates entities in natural language text and identifies their types (Okanohara et al., 2007; Ji and Grishman, 2005). Relation extraction recognizes the relations in entities (Pantel and Pennacchiotti, 2006; Girju et al., 2006; Jiang and Zhai, 2007). A comprehensive review on information extraction is given by Mooney and Bunescu (2005). Conditional random fields (Lafferty et al., 2001) so far gives the best performance for information extraction in general.

Existing relational extraction works focus on semantic relations between entities, for example, ACE (Automatic Content Extraction) Evaluation 2007 defines seven relations {Artifacts, GEN-Affiliation, Metonymy, Org-Affiliation, Part-While, Person-Social, Physical}. None of these relations can address our problem. However, there is a directly related work to ours, a system called Proteus BIO, which was developed by Grishman et al. (2002). This system uses finite-state machines to extract infectious disease outbreak information. Their extraction patterns are

regular expressions. The system gets F-score of 0.54 on the author's corpora. This result is much lower than our method's results which will be shown in Section 5. We also show our method is much better than conditional random fields on the same dataset.

Information extraction has been applied to other domains. For example, in the biomedical domain, biological knowledge such as protein interactions and protein names are extracted from literature text (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; Nitin and Liu 2006). Nitin and Liu also used sequential rules in their application. However, their methods of using these rules are quite different. For example, class sequential rules are used to classify whether a sentence is a comparative sentence or not. In our case, they are used purely for extraction.

3 Problem Definition

3.1 Named Entity

Before we give the definition of the problem, we first define some terminologies that will be used. We start by introducing several types of named entities.

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.

- **Disease Named Entity (Disease NE)** is a Named Entity of a disease. In this work, we are only interested in infectious diseases.
- **Location Named Entity (Location NE)** is a Named Entity of a location.

For our application, location named entities can be further divided into 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.

3.2 Emergent Disease Report Sentences

Since in this work, we only study news articles that report disease outbreaks, and our extraction task is only based on the first few sentences of each article, we introduce two types of sentences:

- 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."

3.3 Problem Statement

In this work, 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 Section 4). 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 Proposed Method

Now we present our proposed method to identify the EDR location NE and disease NE pair using sequential pattern mining on paths in dependency trees built from sentences.

4.1 Dependency Tree

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.

Figure 1 shows an example of a dependency tree, which is for the example sentence given preciously in Section 3.1. Some words such as auxiliary words and preposition words are placed on the path to save space. Note that each named entity such as "bird flu" and "South Korea" is represented by one node even though 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 1, has a very similar dependency tree.

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

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*).

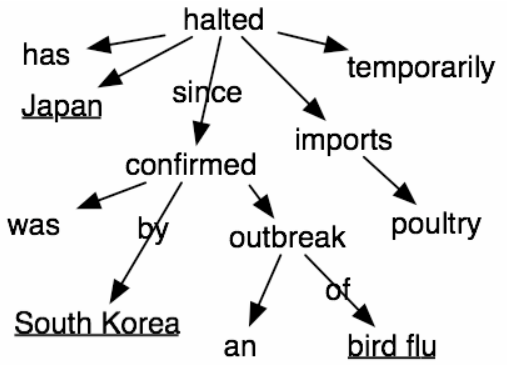

Figure 1. Dependency tree for "Japan has temporarily halted poultry imports since an outbreak of bird flu was confirmed by South Korea." Named Entities are underlined.

In a dependency tree, the **node path** $path(n_1, n_2)$ for node n_1 and node n_2 is defined as ($p(n)$ is the parent node of n):

- $\langle n_1\ p(n_1)\ p(p(n_1))\ldots\ n_2\rangle$ 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,
- otherwise $\langle n_1\ p(n_1)\ \ldots\ q\ \ldots\ p(n_2)\ n_2\rangle$ if 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.

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

4.2 Named Entity

After dependency tree is built, two types of named entities: disease NE and location NE will be detected using Minipar.

Disease NE: We are only interested in infectious diseases, and the number of all known infectious diseases is small and usually unchanged. So we gathered the names of infectious diseases and their alias from the "National Center for Preparedness, Detection, and Control of Infectious Diseases" and augmented Minipar's database with these names. Since disease NE was not in the original Minipar package, so a Named Entity type called Disease NE has been created in Minipar so that disease names can be recognized as disease NE.

Candidate Location NE

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 to cover almost all location NE. Clearly, this will introduce many false positives. We will use sequential rules 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 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 $I = \{i_1, i_2, \ldots, i_n\}$ 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 $\langle a_1 a_2 \ldots a_n \rangle$, 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 $s_1 = \langle a_1 a_2 \ldots a_r \rangle$ is a *subsequence* of another sequence $s_2 = \langle b_1 b_2 \ldots b_m \rangle$ or s_2 is a *supersequence* of s_1, if there exist integers $1 = j_1 < j_2 < \ldots < j_{r-1} \leq j_r$ 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.

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, $I \cap Y = \emptyset$. 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

$X \rightarrow y$, where X 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 1: Table 1 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:

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

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

	Data Sequence	Class
1	$\langle \{1\}\{3\}\{5\}\{7, 8, 9\} \rangle$	c_1
2	$\langle \{1\}\{3\}\{6\}\{7, 8\} \rangle$	c_1
3	$\langle \{1, 6\}\{9\} \rangle$	c_2
4	$\langle \{3\}\ \{5, 6\} \rangle$	c_2
5	$\langle \{1\}\ \{3\}\{4\}\ \{7, 8\} \rangle$	c_2

Table 1: An example of sequence database with classes

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.

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 2: Table 2 gives an example sequence database with 5 sequences and the minimum support of 30% and minimum confidence of 30%. We have the following,

$\langle \{1\}\{3\}\ \{7, *\} \rangle \rightarrow \langle \{1\}\{3\}\{7, 8\} \rangle$
[sup = 2/5, conf = 3/4]

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

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 non-EDR location depending on the rule and the confidence of the rule. Again, mining of this type of rules can be found in (Liu 2006).

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

Table 2: An example sequence database

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:

Training: Training consists of three steps,

1) From training sentences, build node path between each pair of disease NE and candidate location NE, and annotate the candidate location NE.
2) Construct two sequence databases from the node paths: SD_A and SD_B. Every node path corresponds to one sequence in SD_A and one sequence with class in SD_B.
3) Mine LSR and CSR from SD_A and SD_B, respectively.

Testing: For each test sentence, we first build its dependency tree and then extract relevant node paths. For a candidate location NE n_c in a node path, we determine if n_c is an EDR location NE by matching the mined LSR and CSR rules.

Below we give more details for each step.

Data Annotation: For each pair of disease NE and candidate location NE, its node path is built and n_l is annotated manually (for training) with a class c, $c \in C$ = {EDR-LOC, nonEDR-LOC, non-LOC}, where EDR-LOC represents EDR location, nonEDR-LOC represents non-EDR location, and non-LOC represents none location.

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) = ⟨n_1 n_2 … n_m⟩, then two sequences S_A and S_B are generated:

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

where $conv(n_i)$ =

- {D} if n_i is a disease NE;
- {L} if n_i is a candidate location NE;
- {n_i.word, n_i.POS} otherwise.

S_B= ⟨{n_l.*word*}⟩ class = c

S_A is added to SD_A, and S_B with class c is added to SD_B.

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

- ⟨{D}s{*}⟩ → ⟨{D}s{L}⟩
- ⟨{*}s{D}⟩ → ⟨{L}s{D}⟩

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 *.

CSR Mining: Class sequential rules are mined from SD_B with the user-specified $minsup_B$ and $minconf_B$. Likewise the rules are sorted first by confidence and then by support to be used as a classifier.

Apply LSR and CSR: Given a test sentence, we first build its dependency tree. For each pair of disease NE (n_d) and candidate location NE (n_l), we find its node path from the dependency tree. Two sequences S_A and S_B are generated as described above. We then find the highest confidence LSR rule *lr* that matches S_A and return the word matching the location token L as the class of n_l. If no matching LSR is found in the above step, we find the highest confidence CSR *cr*, whose left-hand-side matches S_B and return *cr*'s associated class as the result of n_l. CSR rules basically determine whether a word or phrase is a location or not based on the training annotation of location words. It does not use any path in the dependency tree. If the class is nonEDR-LOC (EDR-LOC), it means that the location NE is not an EDR location or is an EDR location. .

If still no match is found, we return the default class $c_{default}$ as the class of the candidate location NE, which is the majority class in C, i.e., EDR location in our case.

5 Experiments

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 (2008) and 596 are from 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 3 shows the distribution of the classes in the tagged data.

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

Table 3: Class distribution in tagged data.

Conditional Random Field (CRF). 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 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 $conv2(n_i)$ =

- {D} if n_i is a disease NE;
- {n_i.word} if n_i is a candidate Location NE;
- {n_i.word+'/'+n_i.POS} otherwise.

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 $conv3(n_i)$ =

- {Candidate-LOC-NE} 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 $t(n_i)$ =

- {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 (2004) in our experiments.

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.

Experimental Results. The average precision, recall, and F-value results are reported in Table 4 (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.

	$minsup_A$	$c_{default}$	P	R	F
CSR before LSR	0.014	EDR-LOC	0.79	0.98	0.87
		non-LOC	0.84	0.92	0.87
	0.02	EDR-LOC	0.78	0.98	0.87
		non-LOC	0.83	0.91	0.87
LSR before CSR	0.014	EDR-LOC	0.80	0.97	0.88
		non-LOC	0.83	0.93	0.88
	0.02	EDR-LOC	0.79	0.98	0.87
		non-LOC	0.82	0.93	0.87
CRF_1	N/A		0.76	0.88	0.82
CRF_2			0.68	0.96	0.80

Table 4: Evaluation results. CRF_1 and CRF_2 use *conv2* and *conv3* in the CRF input construction, respectively. P stands for Precision, R for Recall, and F for F-score.

Some limitations

For combined locations consisting of locations atdifferent levels, e.g., "Los Angels, CA", they should ideally be recognized as a single bcation,

i.e., Los Angels in CA, instead of two locations. We are recognizing them as two locations, but this problem can be solved by utilizing geographical databases such as GeoNames (http://www.geonames.org), so that we can combine two locations if one belongs to the other. In many cases, the formats that the locations are written already tell us their relationships such as the example above. Also, our current technique does not deal with pronoun resolution in multiple sentences, which will be investigated in the future.

6 Summary

In this paper, we studied the problem of extracting disease outbreak locations from sentences in new articles. A novel technique is proposed to use dependency tree and two types of sequential rules to solve the problem. Experimental results have shown that the proposed method is effective and performs significantly better than the state-of-the-art method conditional random fields.

References

Agrawal, Rakesh and Ramakrishnan Srikant. 1994. Mining Sequential Patterns. *The 11th International Conference on Data Engineering*.

Ayres, Jay, Johannes Gehrke, Tomi Yiu, and Jason Flannick. 2002. Sequential PAttern Mining Using Bitmaps. *The 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

Bunescu, Razvan, Ruifang Ge, Rohit J. Kate, Edward M. Marcotte, Raymond J. Mooney, Arun K. Ramani and Yuk Wah Wong. 2005. Comparative experiments on learning information extractors for proteins and their interactions. *Artificial Intelligence in Medicine*, volume 33, issue 2.

Carenini, Giuseppe, Raymond T. Ng, and Ed Zwart. 2005. Extracting knowledge from evaluative text. *The 3rd International Conference on Knowledge Capture*.

Girju, Roxana, Adriana Badulescu and Dan Moldovan. 2006. Automatic Discovery of Part-Whole Relations. *Computational Linguistics*, 32(1): 83-135.

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

Grishman, Ralph, Silja Huttunen and Roman Yangarber. 2002. Information extraction for enhanced access to disease outbreak reports. *Journal of Biomedical Informatics*, volume 35, Issue 4.

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

Jiang, Jing and ChengXiang Zhai. 2007. A Systematic Exploration of the Feature Space for Relation Extraction.

Jindal, Nitin and Liu, Bing. 2006. "Mining Comparative Sentences and Relations." *Proceedings of 21st National Conference on Artificial Intelligence*, July 16.20, 2006, Boston, Massachusetts, USA.

Lafferty, John, Andrew McCallum, and Fernando Pereira. 2001. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. *The 18th International Conference on Machine Learning*.

Lin, Dekang and Patrick Pantel. 2001. Discovery of Inference Rules for Question Answering. *Natutral Language Engineering*, volume7-4.

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

Mooney, Raymond J. and Razvan Bunescu. 2005. Mining Knowledge from Text Using Information Extraction. *SIGKDD Explorations*, volume 7, issue 1.

Okanohara, Daisuke, Yusuke Miyao, Yoshimasa Tsuruoka and Jun'ichi Tsujii. 2006. 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*.

Pantel, Patrick and Marco Pennacchiotti. Leveraging Generic Patterns for Automatically Harvesting Semantic Relations. 2006. *The 21st International Committee for Computational Linguistic and the 44th annual meeting of the Association for Computational Linguistics*.

Popescu, Ana-Maria and Oren Etzioni. 2005. Extracting Product Features and Opinions from Reviews. *Conference on Human Language Technology and Empirical Methods in Natural Language*.

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

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

Zhang, Yi and Bing Liu. 2007. Semantic Text Classification of Emergent Disease Reports. *The 11th European Conference on Principles and Practice of Knowledge Discovery*.