DESRM: a disease extraction system for real-time monitoring

Minh-Tien Nguyen*
Hung Yen University of Technology and Education (UTEHY), Administrative Building, Nhan Hoa Commune, My Hao District, Hung Yen Province, Vietnam
Email: tiennm@utehy.edu.vn
*Corresponding author

Tri-Thanh Nguyen
Vietnam National University (VNU),
University of Engineering and Technology (UET),
144, Xuan Thuy Street, Cau Giay District, Hanoi, Vietnam
Email: ntthanh@vnu.edu.vn

Abstract: In this paper, we proposed a method that combines semantic rules and machine learning to extract infectious disease events in Vietnamese electronic news for a real-time monitoring system of spreading status. Our method includes two important steps: detecting disease events from unstructured text and extracting information of the disease event. The detection phrase uses semantic rules and machine learning to detect a disease event; in the later step, named entity recognition (NER), rules, and dictionaries are utilised to capture the event’s information. The performance of the two steps has F-score of 77.33% (2.36% better than the baseline’s) and 91.89% (4.31% better than the baseline’s) correspondingly. The promising results from the comparisons showed that our method is suitable for extracting disease events in Vietnamese text.

Keywords: data mining; information extraction; event extraction; disease event extraction; disease monitoring system.


Biographical notes: Minh-Tien Nguyen obtained his Bachelor of Technology degree from Hung Yen University of Technology and Education (UTEHY), Hung Yen, Vietnam in 2009. He completed his Master of Science degree at University of Engineering and Technology (UET), Vietnam National University, Hanoi (VNU) in July 2014. Since October 2009, he has worked as an Assistant Lecturer for Faculty of Information Technology, UTEHY.

Tri-Thanh Nguyen received his Bachelor in Information Technology at University of Natural Science, VNU in 1999; Master of Science degree at Asian Institute of Technology (AIT), Thailand in 2002; and PhD in Information Science at Japan Advanced Institute of Science and Technology (JAIST) in 2008. Since February 2003, he has worked as a Lecturer for Faculty of
Information Technology, College of Technology [later changed to University of Engineering and Technology (UET)], VNU. He was granted the Associate Professor title in Information Technology in January 2015.

This paper is a revised and expanded version of a paper entitled ‘Extraction of disease events for a real-time monitoring system’ presented at the SoICT, Da Nang, 5–6 December 2013.

1 Introduction

Information from electronic newspapers provides valuable inputs for public health surveillance, early outbreak detection, and disease monitoring systems. When the presence of a disease is announced by the government and published on a web page, it is typically called ‘a disease event or an infectious disease (ID) outbreak’. Unfortunately, the electronic resources of IDs are multidimensional, chaotic, and not well organised, so extracting useful patterns from these sources is really challenging. ‘How to detect an ID event (or disease event for short)?’ and ‘how to extract information of the event?’ are two important questions which we strongly focus on this paper.

Disease detection and disease spreading/outbreak monitoring are extremely meaningful issues for society, especially when the diseases are dangerous and have high ability of infection. Since an ID normally gets an outbreak in a short time and very quickly over a large area, it can bring to emergency circumstances not only for the citizens, but also for the government and economy. Therefore, monitoring ID outbreak is survival in prevention, handing diseases and helping the authorities to make suitable decisions.

Previous works, e.g., Huttunen et al. (2002), Grishman et al. (2002a), Volkova et al. (2010), Sani et al. (2012), and Hogenboom et al. (2013), have solved event extraction by using knowledge-driven approach. This approach achieves promising results in extracting because rules were built in a narrow domain. However, it also has problems: there is an imbalance between high precision (thanks to the handcrafted rules of experts) and low recall; it took time to build the rules after a careful analysis over the data. This approach cannot be applied to Vietnamese, since the fundamental linguistic resources (e.g., Wordnet or parsing utilities) are either unavailable or not good enough performance. This makes a challenge for solving the extraction by rules. In addition, there are several systems which extracted and monitored disease events, e.g., Ralph et al. (2002b), Doan et al. (2008), Freifeld et al. (2008), Rortais et al. (2010), Zamite et al. (2010), and Pivovarova et al. (2013), but none of them solves the extraction in Vietnamese. This motivated us to propose a model to automatically detect and extract information of human ID events (called events for short) from Vietnamese web pages. Our contributions are:
We adapted a hybrid approach for solving disease event extraction problem. i.e., combines semantic rules and machine learning in two key components: event detector and extractor. In the first component, semantic rules were used as a data filter which finds out relevant data (articles mentioning about diseases). After finding relevant data, machine learning (classification) was applied to identify whether an article contains an event or not. In the second component, semantic rules were used to extract time when the event occurred; a disease dictionary was used as the clue to extract disease name; and machine learning (named entity recognition – NER) was utilised to capture locations of the event.

We successfully applied our model to extract disease events in Vietnamese free texts. Our results can be combined with other sources (e.g., TV, social networks, news services, etc.) to support the authorities in dealing disease situations in Vietnam.

We set our problem as the extraction of disease events from online news articles. To overcome this issue we have to solve two problems. The first is the detection of an event and the second is the extraction when an event was already detected. In the first step, the detection is equivalent to a classification problem. Moreover, since articles from the internet are very diverse; therefore, in order to improve the performance of classification, removing irrelevant data by a data filter is a suitable approach. In the second step, the model has to extract three elements: time, disease name, and locations. While disease name can be extracted by a dictionary, the two remaining elements are challenging. To extract the time we used semantic rules. This is because the time was mentioned in either relative or absolute form; hence, using rules to capture this information is one of good approaches. The final element was captured by machine learning (NER). This is because extracting location is a main task in NER and many natural language processing (NLP) tools solve this task very well. From the investigation, we decided to combine semantic rules and machine learning for extracting disease events in Vietnamese texts.

Our paper is organised as follows: we mention related work in Section 2; we show characteristics, problem, solution, and model for detecting and extracting disease events in Section 3; we then give results and discussions about the detection and extraction in Section 4; finally, we conclude the achievements and future directions in Section 5.

2 Related work

Event extraction was first introduced as an important topic in 1987 in Message Understanding Conference (MUC) (Grishman and Sundheim, 1996) and continued to mention in automatic content extraction (ACE) programme (Doddington et al., 2004). In ACE, events were divided into eight types. In the work of Allan et al. (1998), an event was defined to include four attributes: modality, polarity (positive or negative), genericity (specific or generic), and tense (past, present, future, or unspecified). Grishman et al. (2002a) gave a definition of a disease event as a template: disease name, date, location, victim number, victim description, victim status, victim type, and parent event. Pyysalo et al. (2011) has overviewed the extraction of IDs in the BioNLP share task. All teams in this task had to extract ID events from recent full-text PMC open access documents selected by ID domain experts (Virginia Tech team). The highest-performing system achieved 56% of F-score.
Event extraction from free texts can be applied in many fields, especially in disease data domain. Grishman et al. (2002a) used linguistic event patterns (120 patterns) for analysing sentences to capture information of a disease event. These linguistic patterns were built on word classes and relation among them. For example, pattern ‘np (DISEASE) vp (KILL) np (VICTIM)’, where np and vp are a noun phrase and a verb phrase, would match a clause like ‘Cholera killed 23 inhabitants’. An event was recognised based on the trigger of two noun phrases: ‘outbreak of…’ and ‘people died from…’. These patterns were applied to extract disease events and achieved the F1 of 53.98%. Normally, applying linguistic patterns can achieve high results if the pattern set has a high coverage over the data domain. Nonetheless, preparing these patterns is always time-consuming and requires domain experts. Moreover, the patterns must be changed when the data change. Finally, because the patterns were built on word classes, so the authors must identify these classes (e.g., noun phrase, verb phrase, etc.), but in some other languages (e.g., Vietnamese), this approach is challenging due to the lack of required resource (i.e., the ontology of words).

Volkova et al. (2010) mixed entity recognition and sentence classification to extract animal disease events. The event recognition consisted of three main steps: the first step was entity recognition from unstructured texts; secondly, sentences were classified based on these entities; finally, the entities within an event sentence were combined into structured tuples. In the event recognition, correct events should contain a disease name and a disease-related verb. The authors got the precision of 75% and 65% in event tuple recognition and the sentence classification, correspondingly, with the features from WordNet and Google-Set corpus. However, using a list of verbs to confirm an event can badly affect the event extraction in Vietnamese language due to the lack of resources for NLP (e.g., Vietnamese WordNet or Google-Set like corpus) or the low performance of parsing utility.

Doan et al. (2008) built a Global Health Monitor system which showed the disease spreading situation around the world. The system included three main steps: topic classification, NER, and disease/location detection. Naïve Bayes was used in topic classification with the precision of 88.10%, F1 was 76.97% in entity recognition step with support vector machine (SVM), and the final step achieved the precision of 93.40% with BioCaster ontology. However, there are some limitations in this system. The first limitation is the location ambiguity, because some locations are not clearly mentioned in input data (they are only provinces/cities, lack of country name), then the system cannot recognise the location exactly. An alternative limitation is that BioCaster system cannot detect new diseases or locations that are not in the ontology.

Our approach uses both advantages of semantic rule-based method and machine learning in two mains components: event detector and extractor. In the event detector, while the semantic rules have the role as a data filter, the classification distinguishes whether a news article contains an event or not. Since our rules act as the filter, they are simpler than those of Grishman et al. (2002a). The rules in our study are short phrases which are a mixture of noun and verb phrases instead of complete sentences. Moreover, we do not use a list of verbs to trigger the existence of events as Volkova et al. (2010), because, typically, this method depends on the coverage of verbs and building these verbs always takes time. In the event extractor, our approach is similar to that of Doan et al. (2008). We use rules to extract time; a disease dictionary to extract disease names; and a NER tool in combination with a location dictionary to extract locations.
3 ID event extraction

3.1 ID event characteristics

An investigation on our data indicated that a formal ID event usually includes a disease name, occurred time, location(s) where the disease was discovered, and the number of infected victim(s). In some cases, it may have additional information, such as the method or environment of infection. Though Grishman et al. (2002a) used the name of the disease, the time and location of the outbreak, the number of affected victims, and type of victims as the information of a disease event, we only focus on three basic information: the time and the locations of the outbreak and the name of an ID. We ignore the methods or environment information because the model collects data from web pages instead of medical reports, so this information is not mentioned clearly, in most cases. An event in MUC, Grishman and Sundheim (1996) must include an actor, in our study, the actor is equivalent to a disease.

In addition, a closer examination on disease news articles showed that a disease name is sometimes as similar as a symptom, so this similarity can be the confusion in the event extraction. For example, pneumonia is the symptom of bird flu (A/H5N1), but it is detected as a disease, in some cases.

3.2 Problem definition

As mentioned previously, our problem is the extraction of disease events from online news articles. The extraction problem can be formally defined as follows:

- **Input**: a news article.
- **Output**: the news article contains an ID event or not? If yes, extract the information of the event.

In our research, an event \( E \) is defined as a tuple that has three attributes:

\[
E = (\text{name}, \text{time}, \text{place})
\]

where \( \text{name} \) is the disease name which is mentioned in a news article; \( \text{time} \) is the time when the disease was discovered; and \( \text{place} \) is a set of locations where the disease spread.

3.3 Solution

Hogenboom et al. (2011) provided a general guideline on how to select a suitable method for event extraction. The guideline indicated that the event extraction approaches can be listed as data-driven, knowledge-driven, and hybrid. The authors compared advantages and drawbacks among these methods. Finally, the authors pointed out the hybrid approach prevail. Using the hybrid approach has two advantages: firstly, this method combines the advantages of both data-driven and knowledge-driven methods, as well as it can help to avoid the limitation of rule-based approach (i.e., the need of domain experts); Secondly, using the hybrid method can extract complicated elements of an event. For example, in our research, the time can be easily extracted by rules while the place is better when machine learning (NER) is applied. Our hybrid model is proposed as illustrated in Figure 1.
DESRM: a disease extraction system for real-time monitoring

Figure 1  The model of disease event extraction (see online version for colours)

The extraction model includes four components: the crawler retrieves data from the internet by using RSS; the event detector decides whether a news article contains a disease event or not; the event extractor captures the information of the event in a given news article; finally, the visualisation plots the disease events on an online geographic information system (GIS) map.

We have built a real-time monitoring system of other events (e.g., fire, accident, crime), thus the crawler and visualisation are already available. In this paper, we only focus on two key components: event detector and extractor which are described in Sections 3.4 and 3.5.

3.4 Event detector

The goal of event detector is to detect events from free text which can be solved by binary classification (i.e., decide whether an article contains an event or not). In reality, the articles from the internet are very diverse; hence classifying a large number of articles takes time and faces noisy data. A natural idea to avoid this problem is to identify relevant articles before classifying them. From this idea, we divided this component into two modules. The first is a data filter which removes irrelevant data and the second is the classification. The process of event detector is illustrated in Figure 2(a).

Figure 2  The detail steps of detection and extraction, (a) disease event detection process (b) disease event extraction process (see online version for colours)
In Figure 2(a), the filter module receives data from the crawler and pre-processes to remove HTML tags; figures out relevant data (disease related news) by using rules; finally, the relevant data is transferred into the classifier to distinguish whether a news article contains an event or not.

3.4.1 Data filtering

The goal of semantic rules is to retrieve relevant data for the classification. We do not use rules as single words/phrases (i.e., verbs or nouns) as in Grishman et al. (2002a) and Volkova et al. (2010). To build the rules we propose to identify a set of frequent words/phrases (of noun or verb) that appear in documents containing a disease event by carrying out a statistic on a large set of news articles from ‘Sức khỏe’ (health) (SU CKHOE, http://www.baomoi.com/Home/SucKhoе.epi) category of ‘Báo mới’ (BAOMOI news website) (BAOMOI, http://www.baomoi.com/). The number of the frequent words is 34 and top 10 words are given in Table 1, where the third column shows the number of articles containing the corresponding words in the second column.

We also examined the data domain carefully and identified that most of news’ titles express their main content. It means that the title of a news article has enough evidence to trigger the existence of a disease event. Therefore, the filter can be seen as title filtering instead of content.

Based on the frequent-word set (Table 1), we proposed Pattern 1 as follows:

$$\text{Pattern 1} = \text{noun} \# \text{verb}$$

(2)

where noun phrase and verb phrase belong to the frequent-word set.

Sample examples of the Pattern 1 are:

- bệnh nhân tử vong # nhiễm (died patient # infected)
- dịch tả # bùng phát (cholera # got an outbreak)
In order to extract events containing the diseases that have not appeared in the news, however they can appear in the future, we propose Pattern 2 as follows:

\[
\text{Pattern 2} = \text{disease name} \# \text{verb phrase}
\]  

where

- \textit{disease name} is retrieved from the BioCaster Ontology (Collier et al., 2010) and The Ministry of Health of Vietnam (2011) (http://www.moh.gov.vn/), dated June 24th, 2011
- \textit{verb phrase} is in the frequent-word set.

Some examples of Pattern 2 are:

- tiêu chảy cấp # nhiễm (acute diarrhea # infected)
- tiêu chảy cấp # phát hiện (acute diarrhea # discovered)
- tiêu chảy cấp # lây lan (acute diarrhea # spread)
- tiêu chảy cấp # bùng phát (acute diarrhea # got an outbreak)
- tiêu chảy cấp # chết (tử vong) (acute diarrhea # died)
- tiêu chảy cấp # dương tính (acute diarrhea # was positive).

3.4.2 Event detection

This module has to identify whether an article contains an event or not. It is possible to solve this task by using rules as Grishman et al. (2002a), however, due to the keyword-based rules, the precision of the detection is very high while recall is low. An alternative method is to transform the problem into a binary classification (based on machine learning) which can balance between the precision and recall, and enhance the performance as well as the adaptability on another data domain.

We follow the method using binary classification, i.e., decide an article contains an event (EVENT class) or not (NOT_EVENT class). To apply classification, we have to prepare the training as well as the testing data for evaluation. From the investigation over the data, we found that the title and abstract of articles has enough information as a representative of the content, therefore, they are utilised to create the features for classifier instead of the whole article content in order to reduce noise, and improve the classification speed.

After classification, articles labelled as EVENT will be used as the input for the next component: event extractor.

3.5 Event extractor

Event extractor is responsible for capturing the information of an event in a text document, and its model is illustrated in Figure 2(b). The information of an event includes three attributes: time, disease name, and place. We divided event extraction into three sub-modules responsible for extracting each, namely, the first module uses rules to extract the time; the second utilises a disease dictionary in extracting the disease; and the third combines NER and a location dictionary to capture the place.
3.5.1 Time extraction

Extracting time in an article is a traditional task in information extraction and can be seen as time resolution (Mani and Wilson, 2000; UzZaman and Allen, 2010). This task can be solved by machine learning as some studies in MUC and ACE tasks, or by rule-based approach (Verhagen and Pustejovsky, 2008), Chang and Manning, (2012), which deemed to be one of the best approaches. Due to the fact that the time is usually mentioned either absolutely (e.g., January 10th, 2013) or relatively (e.g., yesterday morning), or both forms (e.g., early morning of January 10th, 2013), in this problem, we need to have absolute time. Hence, machine learning is not suitable for our situation. In addition, our model can not apply directly the rules of Mani and Wilson (2000), and UzZaman and Allen (2010) due to the difference of language (i.e., Vietnamese in our study).

We created a set of Vietnamese rules in form of regular expressions. An investigation on the data showed that the absolute time has the form of DD/MM/YYYY (e.g., 14/02/2015), where DD, MM and YYYY are numbers representing day, month and year, correspondingly. The regular expression in Java programming language for extracting absolute time is '{nd}+/{nd}+/{nd}'. For the relative form of time, there is a time-related phrase, such as ‘hôm qua’ (yesterday), ‘tối qua’ (yesterday evening), and in some situations, in combination with an absolute time, for example, ‘Sáng ngày 15/01/2012’ (early morning, January 15, 2012). It is possible to list all the Vietnamese time-related phrases, such as ‘vào’ (on), ‘ngày’ (the prefix of a date), ‘sáng’ (morning), ‘hôm nay’ (today), ‘sáng hôm nay’ (this morning), ‘chiều’ (afternoon), ‘hôm qua’ (yesterday), ‘tối qua’ (yesterday evening), ‘rạng sáng’ (early morning), ‘tháng’ (month), thus, it is easily to capture them in the text. A special case of time-related phrase, such as ‘hai ngày trước’ (two days ago), ‘ba tháng trước’ (three months ago), we prepared a rule to extract time in the past:

\[<RELATIVE\ TIME> \geq <\ number> <\ ngày (days)\ [tuần(weeks)]\ [tháng(month)]\ [năm(years)] > "trước"
\]

and another rule to extract future time:

\[<RELATIVE\ TIME> \geq <\ number> <\ ngày (days)\ [tuần(weeks)]\ [tháng(month)]\ [năm(years)] > "tới(next) >"\]

In our data domain, the future time is not available. The time-related phrase is converted into RELATIVE TIME by using a mapping from the phrase into a number as depicted in Table 2. In case there is no absolute time followed by the time-related phrase, we use the published date of the article as the absolute time. We follow the time rule of Tran et al. (2012) to capture the exact time as showed in equation (4).

\[EXACT\ TIME =<RELATIVE\ TIME> + <ABSOLUTE\ TIME>\]

We set relative time extraction rules higher priority than absolute ones. Below are examples of time extraction:

‘Ngày 12/03/2012, Bộ Y tế công bố dịch cúm A/H5N1 đã tái phát tại Quảng Ngãi’ (On March 12th, 2013, the Ministry of Health announced A/H5N1 flu has recurred at Quang Ngai). The time-related phrase is ‘On’, absolute time is ‘March 12th, 2013’, RELATIVE TIME(‘On’)=0, thus, exact time is ‘March 12th, 2013’.
DESRM: a disease extraction system for real-time monitoring

‘Sáng ngày 15/01/2012, Sở Y tế Hà Nội thông báo bệnh nhân đầu tiên nhiễm cúm A/H5N1 đã tử vong’ (In the morning of January 15th, 2012, Hanoi Department Health announced the first patient who was infected A/H5N1 flu has died). Time-related phrase is ‘morning’, absolute time is ‘January 15th, 2012’, RELATIVE TIME(‘morning’) = 0, thus, exact time is ‘January 15th, 2012’.

Table 2 Example of relative time conversion

<table>
<thead>
<tr>
<th>Phrase</th>
<th>Offset</th>
<th>Phrase</th>
<th>Offset</th>
</tr>
</thead>
<tbody>
<tr>
<td>hôm qua (yesterday)</td>
<td>−1</td>
<td>ngày mai (tomorrow)</td>
<td>+1</td>
</tr>
<tr>
<td>sáng hôm qua (yesterday morning)</td>
<td>−1</td>
<td>tối qua (yesterday evening)</td>
<td>+1</td>
</tr>
<tr>
<td>đêm qua (last night)</td>
<td>−1</td>
<td>tuần trước (last week)</td>
<td>−7</td>
</tr>
<tr>
<td>hai ngày trước (two days ago)</td>
<td>−2</td>
<td>bốn tháng trước (four months ago)</td>
<td>−120</td>
</tr>
<tr>
<td>tuần tới (next week)</td>
<td>+7</td>
<td>tháng trước (last month)</td>
<td>−30</td>
</tr>
</tbody>
</table>

3.5.2 Disease extraction

This task can be easily solved by using a disease ontology. Unfortunately, the Vietnamese version of the resource is not available, thus, we instead use a disease dictionary (i.e., a list of disease names) which can be built by retrieving from BioCaster ontology (Collier et al., 2010) and The Ministry of Health of Vietnam (2011) (http://www.moh.gov.vn/) (dated June 24th, 2011).

Extracting disease name can be done in three steps:
1. pre-processing (i.e., word segmentation)
2. enumerating candidates in the disease dictionary by searching each word in the article
3. checking the candidates with the original article to find out the correct ones.

An example of the disease extraction process is illustrated as follows:

‘Điểm cúm A/H5N1 bùng phát tại Bến Tre’ (A/H5N1 flu spread at Ben Tre province). After word-segmentation, the words cúm (flu) and A/H5N1 have candidates in the dictionary, namely, cúm (flu), cúm A/H5N1 (A/H5N1 flu), and cúm gia cầm (bird flu). Each of the candidate is matched against the original article to find out the longest item. In this example, it is cúm A/H5N1 (A/H5N1 flu).

Figure 3 An intuition of location taxonomy in which the finding marks the red node as a candidate and the red line is the complete location information (see online version for colours)
3.5.3 Location extraction

This task is more challenging than the two previous ones due to ambiguity among locations. In fact, a proper name can be named for several places (e.g., ‘Dong Hai’ town is a location in both ‘Tra Vinh’ and ‘Quang Ninh’ provinces). Hence, if an article does not mention the absolute locations clearly, the place information is confused.

Location extraction can be seen as the toponym resolution task (Kitamoto and Sagara, 2012; Leidner, 2007). This task can be well solved by NER because location extraction is one of main tasks of NER and many NLP tools provide this function. However, using NER can not solve this task perfectly due to the cases where only partial information of a location is mentioned, i.e., in the context of a certain province, the article only mentions about a district, thus, NER can not resolve to an absolute location. To deal with this issue, we combine the NER and a location dictionary to improve the performance of the location extraction. The location extraction can be showed in two steps:

1. (partial) location retrieval via a NER tool
2. normalisation to get the absolute locations.

We organised the dictionary in the form of a taxonomy as showed in Figure 3 to facilitate the normalisation, where:
- \( T \) is the abbreviation of the town
- \( C \) is the abbreviation of the commune.

In this taxonomy, the highest level is the root node; level 1 represents 63 provinces; 692 districts are showed in level 2; and 11,101 towns and communes are represented by nodes in the level 3. Firstly, a NER tool JvnTextPro (http://jvntextpro.sourceforge.net/) is applied to detect locations (which are marked by <LOC> and </LOC> tags) in a news article. Then the list of locations is collected by extracting the text between <LOC> and </LOC> tags. In the second step, if a location is relative (e.g., missing the some information), we search the taxonomy from a matched leaf (a commune) to the root to restore the full information. Below is an example:

‘Ngày 12/04/2013, Sở y tế Quảng Ngãi thông báo dịch cúm A/H5N1 đã bùng phát tại thị trấn Sông Vệ (On April 12th, 2013, Quảng Ngãi Department Health announced A/H5N1 flu has spread in Song Vê town).

This example mentions only the ‘Song Vê’ town where A/H5N1 flu spread, while the district and province are absent. As the extracting, ‘Song Vê’ was recognised as a location, while ‘Quảng Ngãi’ was recognised as an organisation. Obviously, ‘Song Vê’ is a relative location which needs to be normalised. The model looked up ‘Song Vê’ in the location taxonomy, when a matched leaf was found, the model traversed from the current node to the root node in the taxonomy to extract the absolute location (i.e., ‘Song Vê’ town, ‘Tu Nghia’ district, and ‘Quảng Ngai’ province) that can be shown in a GIS map. This example also shows that our method helps to tolerate the error of the NER tool (e.g., incorrect recognition of ‘Quảng Ngai’ as an organisation).

Finally, the extracted time, disease name, and location(s) from the article are combined to create an event in which the place is the set of location(s). The event is stored in an event database which is used for the visualisation component in a real-time monitoring system.
4 Experiments and results

4.1 Data preparation

Our data was retrieved from ‘Báo mới’ (BAOMOI news website) (BAOMOI, http://www.baomoi.com/), since this site automatically crawls a lot of pages from most of famous Vietnamese electronic news. The dataset was 3,842,137 news articles. We used title, abstract, published time, URL, and content of an article for the later steps.

After crawling data, the model used Pattern 1 (2) and Pattern 2 (3) to get relevant pages. The rules of Pattern 1 were built by mixing 52 noun phrases and ten verb phrases to create 520 rules, however, only 43 rules matched the real data. Similarly, we combined 186 disease names and six verb phrases to create 1,116 rules following Pattern 2.

After removing irrelevant data by the above rules, the number of relevant articles was 1,668 and we called it relevant dataset which is the input for the later steps.

4.2 Event detection evaluation

4.2.1 Data filtering evaluation

Since we did have enough time to evaluate on the big dataset (of 3,842,137 articles), we evaluated on the relevant dataset by sampling. To evaluate the performance of this module we used error rate, i.e., the percentage of irrelevant data in the dataset. This measure is calculated by equation (5).

\[
\text{ErrorRate} = \frac{\#\text{irrelevant}}{\#\text{total}}
\]  

where:

- \#irrelevant: is the number of articles which are not related to disease.
- \#total: is the total number of articles.

We randomly selected 100 articles from the relevant dataset and divided into ten folds. For each part, we checked the error rate manually. The result was showed in Table 3, where the fifth column was the error rate, that was calculated by corresponding results in the fourth and third column.

<table>
<thead>
<tr>
<th>Fold</th>
<th>Total document</th>
<th>Correct</th>
<th>Incorrect</th>
<th>Error rate (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>100</td>
<td>64</td>
<td>36</td>
<td>36</td>
</tr>
</tbody>
</table>

We accept a high error rate (36%) to increase the recall of this module. The performance of the detection will be improved in the classification step. Moreover, from an investigation of the error rate, we found that there are confusions affecting the data filtering module. Such confusions will be explained in Section 4.4.
4.2.2 Classification evaluation

Classification can be seen as a binary classifier which assigns a news article into either EVENT or NOT_EVENT. To train the model, the relevant dataset was manually annotated, then the documents were represented by vectors of 2-grams, 3-grams, and 4-grams features. After selecting the model retrieved 4,552 features which were used to train the model. Some features are showed in Table 4.

We utilised the maximum entropy (ME) model (MAXENT, (http://www.cs.princeton.edu/maxent) as the classifier due to its good performance of classification issue (Nigam et al., 1999) in sparse data. After training, the model was used to detect disease events. The news articles which are labelled EVENT will become the input for the event extractor. To evaluate performance of the classification, we designed an additional programme (called baseline) which only used machine learning. We used precision, recall, and F1 based on ten-fold cross-validation to compare the performance of the two experiments.

In our approach, we used semantic rules and machine learning to detect a disease event. The training dataset of our approach was 1,668 articles which passed the data filter module. On the contrary, in the baseline, we only utilised machine learning. We did not use the semantic rules by adding 50 articles into the dataset. These articles were dissembled by the semantic rules and randomly selected from the remaining data of 1,668 news articles. Therefore, the training dataset of the baseline was 1,718 articles.

<table>
<thead>
<tr>
<th>No.</th>
<th>Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Dịch tả chân miệng (Hand, foot, and mouth disease -HFMD)</td>
</tr>
<tr>
<td>2</td>
<td>Tiêu chảy (diarrhea)</td>
</tr>
<tr>
<td>3</td>
<td>Trẻ tử vong (dead children)</td>
</tr>
<tr>
<td>4</td>
<td>Ổ dịch (disease source)</td>
</tr>
<tr>
<td>5</td>
<td>Dương tính (positive)</td>
</tr>
<tr>
<td>6</td>
<td>Dịch cúm gia cầm (bird flu)</td>
</tr>
<tr>
<td>7</td>
<td>Ca tử vong (dead cases)</td>
</tr>
<tr>
<td>8</td>
<td>Bùng phát dịch (spread)</td>
</tr>
<tr>
<td>9</td>
<td>Dịch cúm (flu)</td>
</tr>
<tr>
<td>10</td>
<td>Bệnh nhân tử vong (dead patient)</td>
</tr>
</tbody>
</table>

After preparing the training dataset, we compared the performance of the two experiments. The comparison of two classifiers was showed in Table 5 where the results of the baseline are in three columns on the right, while the results of our approach are showed in three columns on the left. The average of F1 in the two experiments indicates that our method outperforms the baseline 2.36%. The difference between two classifiers is not big because we only add 50 articles into the baseline dataset. The performance will be much better if we add more raw articles into the baseline dataset.

After detecting event the model retrieved 152 news articles which were labelled EVENT. These articles are input for the extraction.
### 4.3 Event extraction evaluation

Since an event $E$ was defined as a tuple that includes time, disease name, and place (location(s)) in equation (1), so a correct event should completely contain all elements. In some cases, we can accept an event which does not have the time information. However, in these cases, the time is the time when a news article was published. In other cases, if an event does not include either a disease or locations, then it is considered to be an incorrect event.

To evaluate the precision of the event extraction, we carried out two experiments. The first as our approach that combined rules and NER and the second was baseline that only utilised rules. Both of the two experiments use 152 news articles which are passed the detection as the input data.

For the baseline, we used rules as in Tran et al. (2012) to extract location:

\[
\begin{align*}
&< \text{LOC PREP} \geq \{\text{ở (in), tại (at), trên (in), gần (nearby), trong (in)}\} \\
&< \text{LOC PREFIX} >= \{\text{thành phố (city), tỉnh (province), quận (district),} \\
&\text{phố (street), đường (street), huyện (town), thị xã (town), thị trấn (town),} \\
&\text{thôn (village)}\} \\
&< \text{LOCATION} >= \{\text{loc | loc} \in \text{location dictionary}\}
\end{align*}
\]

A location is extracted if it matched the rule:

\[
< \text{LOC PREP} <= \text{LOC PREFIX} <= \text{LOCATION} >
\]

We used precision ($P$), recall ($R$), and $F_1$ to compare the performance of the two experiments. These measures are denoted by equations (6), (7), and (8) as the following:

\[
P = \frac{\#\text{correct}}{\#\text{correct} + \#\text{incorrect}}
\]
where

- \( \#_{\text{correct}} \): is the number of correct events
- \( \#_{\text{incorrect}} \): is the number of incorrect events

\[
R = \frac{\#_{\text{correct}}}{\#_{\text{correct}} + \#_{\text{not\_found}}} \tag{7}
\]

where

- \( \#_{\text{correct}} \): is the number of correct events
- \( \#_{\text{not\_found}} \): is the number of events which the model does not found

\[
F1 = \frac{2 \times P \times R}{P + R} \tag{8}
\]

Based on the equations (6), (7), and (8), we compared the performance of our approach and the baseline. The comparison was showed in Table 6, where the second row is the result of our approach whereas the third row is the result of the baseline.

In our approach, the F1 is 91.89%, in comparison with 87.58% of the baseline. The causes of the performance difference between two experiments will be explained in the next section.

Table 6  The comparison of our approach and the baseline

<table>
<thead>
<tr>
<th>Experiment</th>
<th>Total</th>
<th>Correct</th>
<th>Incorrect</th>
<th>Precision</th>
<th>Recall</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Our approach</td>
<td>152</td>
<td>136</td>
<td>16</td>
<td>89.47</td>
<td>94.44</td>
<td>91.89</td>
</tr>
<tr>
<td>Baseline</td>
<td>152</td>
<td>127</td>
<td>25</td>
<td>83.55</td>
<td>92.02</td>
<td>87.58</td>
</tr>
</tbody>
</table>

4.4 Error analysis and discussion

In the first component (event detector), the results in Table 3 suggest that there is a confusion in the data filter module. To find out the cause we checked manually 100 titles which were selected in Section 4.2.1. An investigation on the results indicated that in error cases, some rules of Pattern 1 (2) and Pattern 2 (3) were not efficient to filter out articles, since several topics (i.e., not related to diseases) can be expressed by some verbs in the frequent-word set. For instance, the verb ‘tử vong’ (die) may belong to the topics of both disease and vaccination as showed in following example, where the article was captured by the rule ‘bệnh nhân # tử vong’ (patient # died):

\[\text{Uống thuốc hạ sốt sau 30 phút bệnh nhân tử vong} \text{ (30 minutes after drinking fever pills, the patient died).}\]

Another kind of confusion came from articles concerning about diseases, but not a disease spread event as given in following example:

\[\text{‘Phát hiện chủng virus mới gây bệnh tay chân miệng’} \text{ (A new virus strains causing the hand, foot, and mouth disease was detected).}\]

A rule of Pattern 2 (3) ‘bệnh tay chân miệng # phát hiện’ (hand, foot, and mouth disease# detected) matched this sentence, so it is marked as relevant, though it mentions about a new virus rather than a detection of the hand, foot, and mouth disease.
Final disease name error came from the fact that the name is not included in the dictionary as in the example no. 8 in Table 8 where nhiễm khuẩn liên cầu lởm (Streptococcal infection) is not in the dictionary.

For the next component (event extractor), the results in Table 6 indicate that the precision of our approach outperforms the baseline of 5.92%.

To find out the source of errors we manually checked the incorrect articles in the two experiments (mentioned in Section 4.3). The checking results were showed in Table 7 and Table 8, respectively, where the fourth column shows the extracted information, and the third column is the correct one.

The statistic from Tables 7 and 8 indicates that the cause of errors in both the experiments originates from the location and diseases extraction. The main cause of location extraction came from the low performance of NER tool. In a few cases, NER did not detect locations exactly due to the abbreviation of places in articles (similarly to the rule-based method). For example, the location ‘Thành phố Hồ Chí Minh’ (Ho Chi Minh City) can be written by TPHCM. In other cases, NER mis-recognised a location as an organisation as showed in the following example:

‘Ngày 12/03/2012, dịch tiêu chảy cấp đã bùng phát tại Hà Nội, Hải Phòng, Quảng Ninh, Bến Tre, Cần Thơ’. (On December 03rd, 2012, the cholera got an outbreak in Hanoi, Hai Phong, Quang Ninh, Ben Tre, Can Tho).

### Table 7 Ten examples of 16 errors in our approach

<table>
<thead>
<tr>
<th>No.</th>
<th>Doc ID</th>
<th>Error detail</th>
<th>Correct</th>
<th>Extracted</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>16</td>
<td>Thanh Long village, Phuoc My commune, Quy Nhon City</td>
<td>Binh Dinh</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>17</td>
<td>Giao Thuy district, Nam Dinh, A (H5N1) flu</td>
<td>Nam Dinh, Flu</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>21</td>
<td>Me So, Van Giang, Hung Yen</td>
<td>Hung Yen</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>23</td>
<td>Ba Ria-Vung Tau</td>
<td>NULL</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>25</td>
<td>4 village, Hoa An commune, Krong Pac district, Dak Lak</td>
<td>Hoa An Commune, Chiem Hoa district, Tuyen Quang</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>26</td>
<td>8 Commune, 5 District, Ho Chi Minh City (P.8, Q.5, TP. HCM)</td>
<td>NULL</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>32</td>
<td>7 Commune, 8 District, HCM City (P.7, Q.8, TP. HCM)</td>
<td>NULL</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>39</td>
<td>Mo Cay Nam, Mo Cay Bac, Giong Trom, Thanh Phu, Chau Thanh Ba Tri, Cho Lach</td>
<td>Ben Tre</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>40</td>
<td>6 Commune, 8 District (P.6, Q.8)</td>
<td>TP. HCM</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>45</td>
<td>Hung Yen, Yen Dinh, Thanh Hoa, Vinh Phuc, Ba Dinh, Hanoi</td>
<td>Hanoi, Vinh Phuc</td>
<td></td>
</tr>
</tbody>
</table>

The locations ‘Hanoi’, ‘Hai Phong’, ‘Quang Ninh’, ‘Ben Tre’, and ‘Can Tho’ were mis-recognised as organisations (<ORG>), thus, they were ignored during extraction process.

The NER tool also caused another error due to its incomplete location recognition, e.g., only some parts of a location was recognised as shown in row no. 3 in Table 7, and row no. 5 of Table 8 (where only Tam Quan commune was recognised). In case, the
article mentions additional locations where the disease did not appear, however, they were extracted and considered as the location of the event as in row no. 5 in Table 7, where the correct location was missed by the NER.

Rule-based approach (in the baseline) also failed to extract location element, in case it was abbreviated as illustrated in the following example:

‘Phát hiện một trường hợp bệnh nhân nhiễm cúm A/H5N1 tại P.7, Q.8, TP. HCM’. (A patient infected A/H5N1 flu was discovered at 7 commune, 8 district, HCM city).

In this example, 7 commune, 8 district, and HCM city are abbreviated as P.7, Q.8, TP. HCM, correspondingly.

Table 8  Ten examples errors in the baseline

<table>
<thead>
<tr>
<th>No.</th>
<th>Doc ID</th>
<th>Error description</th>
<th>Correct information</th>
<th>Extraction</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>89</td>
<td>A/H1N1 flu</td>
<td>Pneumonia (Symptom)</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>92</td>
<td>A/H1N1 flu</td>
<td>Tuberculosis</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>96</td>
<td>Ea T’ling towns and communes: Nam Dong, Tam Thang, D’Dak Rong</td>
<td>NULL</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>105</td>
<td>Cholera</td>
<td>Acute diarrhea</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>108</td>
<td>Tam Quan commune, Tam Dao province, Quan Noi, Quan Ngoai, Lang Chanh village, Lang Mau village, and Nhan Ly</td>
<td>Tam Quan commune</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>111</td>
<td>51/6 the Khanh Gia zoon, Vinh Nguyen Ward, Nha Trang City A/H1N1 flu</td>
<td>Nha Trang, Khanh Hoa Pneumonia (Symptom)</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>115</td>
<td>Co., Ltd.-TM-DV XD Huang De-partment, at the address 199 Ma Lo Quarter 6, Binh Tri Dong A Ward, Binh Tan District, Ho Chi Minh City</td>
<td>Ho Chi Minh City, Phuong Mai Ward, Hanoi</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>118</td>
<td>nhiễm khuẩn liên cầu lơ (Strep-tococcal infection)</td>
<td>NULL</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>120</td>
<td>Kon Tom</td>
<td>NULL</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>123</td>
<td>Tan Binh, Tan Phu, Tan Binh, Go Vap</td>
<td>HCM City, Tan Binh province</td>
<td></td>
</tr>
</tbody>
</table>

In both our approach and the baseline, some extracted disease names were incorrectly recognised due to a fact that the disease dictionary contains some names which are equivalent to the symptom of a disease. For instance, in Table 8, a disease name of A/H1N flu in the document 89th was detected as pneumonia, while pneumonia is a symptom of the A/H1N flu.

In addition, there are some factors which badly affected to the event extraction. Firstly, typing error of the location in articles reduced the performance of the location extraction. For instance, ‘Đắk Lắk’ was written as ‘Đắc Lắc’, but ‘Đắc Lắc’ did not appear in the location dictionary. Therefore, the location information was missing. Secondly, some places were not described clearly such as ‘các huyện phía Tây của tỉnh Bến Tre’ (the western districts of ‘Ben Tre’ province), so the NER can not recognise these locations. Finally, one of the important causes was the geo-ambiguity that reduced
the precision of event extraction. In fact, one proper name can be named for several places, so in case of articles did not mention the places clearly, the location information can be confused as showed in the following example:

‘Ngày 05/10/2012, Sở Y tế Quảng Ninh thông báo đã phát hiện vi khuẩn tả tại thị trấn Đồng Hai’ (On May 10th, 2012, Quảng Ninh Health Department announced that cholera bacterium was detected at Dong Hai town).

In this example, ‘Dong Hai’ town is a location in both ‘Tra Vinh’ and ‘Quang Ninh’ province, but the article only mentions Dong Hai town, so the model did not have enough information to decide where the disease was detected, i.e., ‘Quang Ninh’ or ‘Tra Vinh’?

5 Conclusions

In this paper, we introduced a method that combines both semantic rules and machine learning to extract disease outbreak events in Vietnamese. In the event detection, while rules helped to reduce irrelevant data, an event was detected by the classification. After that, information of an event was extracted in the event extraction. The model combined three approaches: using location dictionary, rules, and the NER to extract the elements of an event. Subsequently, the model combined these elements to create a disease event and store it into an event database. Furthermore, we have described briefly our system process, especially we emphasise two key components: event detector and event extractor. Our results were integrated into Vn-Loc (http://www.vnloc.com) system where user can follow some event types: fire, crime, and transport accident.

However, our method needs to have some improvements to enhance the quality in the future. Firstly, the coverage of semantic rules and the performance of the classifier must be enhanced the by adding useful information. Secondly, the precision of event extraction can be increased by improving the performance of the NER tool. Beside, the geo-ambiguity and confusion between diseases and symptoms should be improved. Finally, the relations among disease events should be consider to enhance the quality of the monitoring system.

References


