Extracting Semantic Frames from Thai Medical-Symptom Unstructured Text with Unknown Target-Phrase Boundaries

Peerasak INTARAPAIBOON^{†a)}, Student Member, Ekawit NANTAJEEWARAWAT^{†b)},
and Thanaruk THEERAMUNKONG^{†c)}, Members

SUMMARY Due to the limitations of language-processing tools for the Thai language, pattern-based information extraction from Thai documents requires supplementary techniques. Based on sliding-window rule application and extraction filtering, we present a framework for extracting semantic information from medical-symptom phrases with unknown boundaries in Thai unstructured-text information entries. A supervised rule learning algorithm is employed for automatic construction of information extraction rules from hand-tagged training symptom phrases. Two filtering components are introduced: one uses a classification model to predict rule application across a symptom-phrase boundary based on instantiation features of rule internal wildcards, the other uses weighted classification confidence to resolve conflicts arising from overlapping extractions. In our experimental study, we focus our attention on two basic types of symptom phrasal descriptions: one is concerned with abnormal characteristics of some observable entities and the other with human-body locations at which primitive symptoms appear. The experimental results show that the filtering components improve precision while preserving recall satisfactorily.

key words: information extraction; medical informatics; rule learning

1. Introduction

Standard knowledge representation languages for the Semantic Web, such as RDF [1] and OWL [2], have been recently developed and are now in place. A crucial question still remains: how will we feed machines with the relevant knowledge in an application domain? In this article, we present an information extraction (IE) framework towards bridging the gap between medical-symptom descriptions in Thai text and the world of concepts, which represent abstractions of human thoughts. IE techniques usually involve linguistic patterns and domain-specific lexicons, coupled with a conceptual description of an application domain, i.e., a domain ontology. While an ideal domain ontology is possibly language-independent, linguistic patterns and lexicons rely heavily on the language in use. Due to language-structure differences, when an IE framework that works well in one language is applied to a different language, supplementary components are often necessary.

As part of a broader-scope project on constructing a large-scale medical-related knowledge base in Thailand from information sources available on the Internet [3], we

aim to develop a system for extracting semantic information from unstructured-text (free-text) symptom descriptions in Thai and representing the extracted results in a form of machine-processable frames. IE based on linguistic analysis of Thai text is not currently feasible due to the lack of basic supporting language-processing tools. Not to mention a full parser for Thai sentences, neither a shallow-parsing (chunking) tool nor a fairly accurate part-of-speech analyzer is currently available, much owing to the high ambiguity of the structure of written Thai.

However, by incorporation of ontology-based semantic annotations and appropriate extraction filtering techniques, it is expected that IE based on patterns of triggering class tags and triggering plain words can be realized for Thai documents without part-of-speech tagging and text chunking. We focus on two types of symptom descriptions: one is concerned with abnormal characteristics of some observable entities and the other with human-body locations at which primitive symptoms appear. A well-known supervised rule learning algorithm, called WHISK [4], is used as the core algorithm for constructing IE rules automatically from a set of hand-tagged training symptom phrases. The technical challenges we address in this paper are twofold:

1. *IE from unstructured text with unknown target-phrase boundary:* From symptom phrases tagged with desired extractions in a training corpus, WHISK generates a set of IE rules, each of which yields an extracted frame when its pattern matches a target symptom phrase. The information sources of our target IE task are, however, collections of unstructured-text paragraphs, called *information entries*, describing diseases, rather than collections of text portions identified beforehand as potential symptom phrases. Each information entry typically contains several symptom phrases along with other text portions. Locating potential symptom phrases in an information entry requires a chunk parser and is thus not currently achievable for Thai text. A method, called *rule application using sliding windows (RAW)*, is introduced for applying IE rules to unstructured-text information entries without predetermining symptom-phrase boundaries.
2. *Extraction filtering techniques:* Using sliding windows, IE rules are applied to text portions regardless of symptom-phrase boundaries and, therefore, tend to make many false-positive extractions. Two extraction

Manuscript received June 2, 2010.

Manuscript revised October 1, 2010.

[†]The authors are with School of Information and Computer Technology, Sirindhorn International Institute of Technology, Thammasat University, Thailand.

a) E-mail: ipeerasak@siit.tu.ac.th

b) E-mail: ekawit@siit.tu.ac.th

c) E-mail: thanaruk@siit.tu.ac.th

DOI: 10.1587/transinf.E94.D.465

filtering modules, called *wildcard-instantiation filtering (WIF)* and *overlapping-frame filtering (OFF)*, are proposed for removing incorrect extractions. The first module uses a binary classifier for prediction of rule application across a symptom-phrase boundary; the second one uses weighted classification confidence to resolve conflicts arising from overlapping extractions. It is desirable that these filtering modules improve extraction precision while minimally sacrificing recall.

Basic ideas behind WIF and OFF were initially presented in [5]. In this work, in addition to technical improvements on feature selection for classifier learning and on extraction conflict resolution, we extend [5] by comparing our framework to extraction with known target-phrase boundaries, and also by evaluating the framework in four other application domains, i.e., soccer match reports, soccer player transfers, stock prices, and company dividends.

To begin with, Sect. 2 briefly describes Thai language-processing barriers and then gives an overview of our framework. After introducing inductive rule learning in WHISK, Sect. 3 explains the RAW method. Section 4 describes WIF and OFF. Section 5 presents experimental results in our target application. Section 6 shows experimental results when applying the framework in other domains. Section 7 discusses related works.

2. Framework Overview

Like many other South Asian and South East Asian languages, Thai is an analytic language—its syntax and meaning are shaped by the use of particles and word order rather than by word inflection. Working with Thai text introduces many problems, primarily founded on the ambiguity of the language structure. In the Thai writing system, words are consecutively written without delimiters, and, unlike the Chinese and Japanese languages, there is no common rule for using punctuation—the end point of a sentence is usually not specified explicitly and punctuation marks occur only

occasionally [6]—resulting in severe difficulty in dividing a piece of writing into sentences and phrases [7]. Moreover, the main subject, verb, or object can sometimes be omitted from a sentence and the sentence is still considered valid. The high language-structure ambiguity seriously hinders the development of basic language-processing tools. Among very few such tools available, only a word segmentation program (a word boundary detector) is commonly used.

2.1 Preprocessing

Considering the language-processing limitations, the IE framework outlined in Fig. 1 is proposed. Paragraph-like Thai unstructured-text descriptions, referred to as *information entries*, are taken as input documents for our target IE task. Word segmentation is applied to all information entries as part of a preprocessing step. A domain-specific ontology, along with a lexicon for it, is then employed to partially annotate word-segmented phrases with tags denoting the semantic classes of occurring words with respect to the lexicon. Figure 2 illustrates a portion of an obtained word-segmented and partially annotated information entry, where ‘|’ indicates a word boundary, ‘~’ signifies a space, and the tags “sec,” “ch,” “ptime,” “sym,” “org,” and “col” denote the semantic classes “Secretion,” “Characteristic,” “Time period,” “Symptom,” “Organ,” and “Color,” respectively, in the domain ontology. The portion contains five symptom phrases of our target types, which are underlined in the figure. Figure 3 provides a literal English translation of this text portion; translations of the five target phrases are also underlined.

2.2 Extracted Frames and IE Rules

Table 1 shows the frames required to be extracted from the five target symptom phrases in Fig. 2. The first frame, for example, contains three slots, i.e., OBS, ATTR, and PER, and the second frame contains two slots, i.e., SYM and LOC, where OBS, ATTR, PER, SYM, and LOC stand for “observed

Fig. 1 An overview of the proposed framework.

เป็นโรคที่พบบ่อยหลังจากเป็นไข้หวัด~ผู้ป่วยมักจะมี[sec น้ำมูก]ลักษณะ[ch ใส]~เป็นระยะเวลา~[ptime 4-10 วัน]~และมีอาการอื่นด้วย~
 เช่น~[sym ไอ]จน[org กล้ามเนื้อท้อง]มี[sym อาการเจ็บ]~มี[sym อาการเจ็บ]ที่บริเวณ[org หน้าอก]เป็นเวลานาน~[ptime 6-12 วัน]~อาจตรวจพบ
 [sec เสมหะ]ใน[org คอ]เป็น[col สีเขียว]~และมี[sym อาการแน่น]ใน[org หน้าอก]เป็นเวลานาน~ภายใน~[ptime 2-3 สัปดาห์]ผู้ป่วยจะมีอาการดีขึ้น
 ~โดยความรุนแรงของโรคจะขึ้นอยู่กับชนิดของเชื้อโรคที่ผู้ป่วยได้รับ~ผู้ป่วยอาจมีสุขภาพทั่วไปแข็งแรง

Fig. 2 A portion of a partially annotated information entry describing acute bronchitis.

It is a disease that often begins after flu. A patient may have [ch clear] [sec nasal mucus], lasting [ptime 4-10 days], and may have other symptoms,
 for example, a [sym cough] leading to a [sym pain] at the [org abdominal muscle], a [sym pain] in the [org chest], which lasts [ptime 6-12 days]. [col green]
 [sec mucus] may be found in the [org throat] and the patient may have a [sym congestion] in the [org chest] for a long period. Within [ptime 2-3 weeks] the
 patient will be better. The severity of the disease depends on the type of germs by which the patient is infected. The patient may be generally healthy.

Fig. 3 A literal English translation of the partially annotated Thai text in Fig. 2.

Table 1 Extracted frames obtained from the target phrases in Fig. 2.

Phrase	Extracted frame	English translation
1st	{OBS [sec น้ำมูก]}{ATTR [ch ใส]}{PER [ptime 4-10 วัน]}	{OBS [sec nasal mucus]}{ATTR [ch clear]}{PER [ptime 4-10 days]}
2nd	{SYM [sym อาการเจ็บ]}{LOC [org กล้ามเนื้อท้อง]}	{SYM [sym pain]}{LOC [org abdominal muscle]}
3rd	{SYM [sym อาการเจ็บ]}{LOC [org หน้าอก]}{PER [ptime 6-12 วัน]}	{SYM [sym pain]}{LOC [org chest]}{PER [ptime 6-12 days]}
4th	{OBS [sec เสมหะ]}{ATTR [col สีเขียว]}	{OBS [sec mucus]}{ATTR [col green]}
5th	{SYM [sym อาการแน่น]}{LOC [org หน้าอก]}	{SYM [sym congestion]}{LOC [org chest]}

Fig. 4 Semantic representation of the first extracted frame in Table 1.

Fig. 5 Semantic representation of the third extracted frame in Table 1.

entity,” “attribute,” “period,” “symptom,” and “location,” respectively. Figure 4 and Fig. 5 depict the semantic representations of the first and the third frames, respectively, in Table 1. Figure 6 gives a typical example of an IE rule. Its

Pattern: *(sym)*(org)*นาน*(ptime)
 Output template: {SYM \$1}{LOC \$2}{PER \$3}

Fig. 6 An IE rule example.

pattern part contains three triggering class tags, one triggering plain word, and four instantiation wildcards. The three triggering class tags also serve as *slot markers*—the terms into which they are instantiated are taken as fillers of their respective slots in the resulting extracted frame. When instantiated into the third target symptom phrase in Fig. 2, this rule yields the third frame in Table 1.

2.3 Rule Construction and Classifier Construction

From manually identified target phrases in a training corpus, the WHISK algorithm is used for automatically constructing a set of IE rules. During rule learning, symptom-phrase lengths are observed when a rule makes correct extractions on the training instances and they are used for determining an appropriate window size for RAW. In order to prepare training data for construction of binary classifiers for predicting rule instantiation across target-phrase boundaries in the WIF module, the IE rules obtained from WHISK are applied using RAW to the partially annotated information entries in the training corpus. Based on their classification confidence, the obtained classifiers are also used in the OFF

module for resolving conflicts caused by overlapping extractions.

3. Rule Learning and Rule Application

3.1 Inductive Rule Learning Using WHISK

WHISK [4] uses a covering algorithm to construct a set of multi-slot extraction rules. It takes training instances that are hand-tagged with desired extraction outputs to guide rule creation. The algorithm induces rules top-down, starting from the most general rule that covers all training instances, and then specializing the initial rule by adding triggering terms one at a time in order to prevent rule application with incorrect extractions. Figure 7 shows the top level of the WHISK algorithm. In each iteration, a new training instance is retrieved from a reservoir of partially annotated text (Line 5), information to be extracted from the instance is manually specified by a user (Line 6), and the instance is then added to the training set (Line 7). Each rule in the current rule repository is then applied to the new training instance, and if it yields incorrect extraction with respect to the instance, the rule is discarded (Line 8). The obtained rule set is then tested whether it covers all instances in the current training set. If some instance is not covered, the GROWRULE procedure is activated for generating a new rule.

Reasons for selecting WHISK include not only its previous success in a variety of English-text IE applications, but also its capability to generate multi-slot extraction rules, which enable extracted slots to be semantically connected, e.g., an observed entity and its abnormal characteristic. Other rule learning algorithms with performance compara-

ble to WHISK, e.g., RAPIER [8] and SRV [9], can generate only single-slot (individual-field) extraction rules [10], [11] and do not suit our application requirements.

3.2 Rule Application Using Sliding Windows (RAW)

A WHISK rule does not have the ability to automatically segment an unstructured-text information entry so that the rule can be applied to the relevant portion of text [10], [11]. When working with unstructured text, a WHISK-based IE system normally uses a part-of-speech tagger and a shallow parser to group together words into larger syntactic chunks, from which the boundaries of relevant text portions are determined. In the absence of such supporting tools, a sliding window technique is introduced to locate text portions for rule application, and filtering methods are used to remove potentially incorrect extractions afterwards.

Using a k -word sliding window, a rule r is applied to each k -word portion of an information entry one-by-one sequentially. More precisely, assume that an information entry E consisting of n words is given and for any l, m such that $1 \leq l \leq m \leq n$, the $[l, m]$ -portion of E is the portion beginning at the l th word position and ending at the m th word position of E . Then r is applied to the $[i, i + k - 1]$ -portion of E for each i such that $1 \leq i \leq n - k + 1$. An application that results in a duplicated frame is discarded.

Example 1: When a WHISK rule is applied, triggering class tags and triggering plain words in its pattern part match those occurring in a target text portion one-by-one from left to right in a greedy manner, i.e., once the first successful match for the pattern part is found, no search for alternative matching is made. As shown in Fig. 8, when the rule in Fig. 6 is applied to the information entry in Fig. 2 using a 12-word sliding window, it makes extractions from the [37, 48]-portion, the [38, 49]-portion, and the [59, 70]-portion of the entry. Table 2 shows the resulting extracted frames. Only the extraction made from the [38, 49]-portion is correct. When the rule is applied to the [37, 48]-portion, the slot filler taken through the first slot marker of the rule, i.e., “sym,” does not belong to the symptom phrase containing the filler taken through the second slot marker of it, i.e., “org,” whence an incorrect extraction occurs. The same situation arises when the rule is applied to the [59, 70]-portion. As will be described in the next section, WIF is designed to filter out incorrect extractions of this kind. ■

```

1: WHISK(Reservoir)
2:   RuleSet = NULL
3:   Training = NULL
4:   Repeat at user's request
5:     Select a batch of NewInst from Reservoir
6:     (User tags the NewInst)
7:     Add NewInst to Training
8:     Discard rules with errors on NewInst
9:     For each Inst in Training
10:      For each Tag of Inst
11:        If Tag is not covered by RuleSet
12:          Rule = GROWRULE(Inst, Tag, Training)

```

Fig. 7 The top level of the WHISK algorithm.

Fig. 8 Text portions from which extractions are made when the rule in Fig. 6 is applied to the information entry in Fig. 2 using a 12-word sliding window.

Table 2 Frames extracted from the text portions in Fig. 8 by the rule in Fig. 6.

Portion	Extracted frame	Correctness
[37, 48]	{SYM [sym อาการเจ็บ]}{LOC [org หน้าอก]}{PER [ptime 6-12 วัน]}	Incorrect
[38, 49]	{SYM [sym อาการเจ็บ]}{LOC [org หน้าอก]}{PER [ptime 6-12 วัน]}	Correct
[59, 70]	{SYM [sym อาการแน่น]}{LOC [org หน้าอก]}{PER [ptime 2-3 สัปดาห์]}	Incorrect

Table 3 Instantiation features for the first internal wildcard of the rule in Fig. 6 observed at the three text portions in Fig. 8.

Portion	Instantiation of the 1st internal wildcard	English translation	No. of spaces	No. of plain words	Presence of “ที่”	Presence of “บริเวณ”	Instantiation feature vector
[37, 48]	~ มี [sym อาการเจ็บ] ที่ บริเวณ	“a [sym pain] in the”	1	3	1	1	[1, 3, 1, 1]
[38, 49]	ที่ บริเวณ	“in the”	0	2	1	1	[0, 2, 1, 1]
[59, 70]	ใน	“in”	0	1	0	0	[0, 1, 0, 0]

The possibility that a symptom phrase fits into a window increases as the window size increases; but the risk of rule application across a symptom-phrase boundary is also higher. Appropriate window size for different rules may be different. In the evaluation presented in Sect. 5, the base window size for an individual rule is set to the length of the longest training symptom phrase observed when the rule makes correct extractions in the rule generation process.

4. Extraction Filtering

Two modules are proposed for filtering out incorrect extractions, i.e., false positives, resulting from RAW. The first module, called *wildcard-instantiation filtering (WIF)*, employs a classification model to predict incorrect extractions based on instantiation features of rule internal wildcards. The second one, called *overlapping-frame filtering (OFF)*, uses weighted confidence values of classification decisions to resolve conflicts caused by overlapping frames.

4.1 Wildcard-Instantiation Filtering (WIF)

WHISK learns an extraction pattern in terms of triggering terms for making an extraction from a “single” symptom phrase. Using RAW, however, an obtained IE rule may be instantiated across a symptom-phrase boundary (cf. Example 1), yielding an extracted frame containing unrelated slot fillers, which is definitely a false positive. Instantiations of wildcards occurring between the first and the last slot markers of a rule, called *internal wildcards*, provide a clue to detect such an undesirable extraction—if an internal wildcard of a rule is instantiated across a symptom-phrase boundary, then unrelated slot fillers are obtained. A wildcard that is not internal does not give the same clue since it is never instantiated into a word string enclosed by two slot fillers of the same extracted frame.

Predicting whether an internal wildcard is instantiated across a target-phrase boundary can be regarded as a binary classification problem. A classifier is constructed for each rule based on observations obtained by applying the rule to information entries in the training corpus using RAW.

Under the assumption that a rule obtained from WHISK is error-free with respect to training target phrases,[†] an incorrect extraction in the training corpus implies the existence of an internal-wildcard instantiation across a target-phrase boundary. Given a text portion T and a rule r containing m internal wildcards $w_1, \dots, w_m$, the *instantiation feature vector* for w_i ($1 \leq i \leq m$) observed at T is a vector $\vec{f}_i = [f_{i1}, \dots, f_{in}]$ consisting of n features selected for representing the instantiation of w_i when r is applied to T , and the *wildcard-instantiation feature vector* for r observed at T is then defined as a vector $\vec{x} = \vec{f}_1 \parallel \vec{f}_2 \parallel \dots \parallel \vec{f}_m$, where $\vec{f}_j$ ($1 \leq j \leq m$) is the instantiation feature vector for w_j observed at T and “ $\parallel$ ” denotes vector concatenation.

Two kinds of features are used for representing wildcard instantiations: first, the number of spaces, the number of plain words, and the number of annotated words occurring in a text portion into which a wildcard is instantiated; and secondly, the presence or absence of certain specific terms in such a text portion. For feature selection, the *information gain* calculated from training data is used as a measure of the relevance of each feature in our experiments (Sect. 5). The information gain of a feature f , denoted by $IG(f)$, is given by

$$IG(f) = \left(\sum_{i=1}^m -Pr(c_i) \log_2 Pr(c_i) \right) - \sum_{j=1}^n Pr(a_j) \left(\sum_{i=1}^m -Pr(c_i|a_j) \log_2 Pr(c_i|a_j) \right),$$

where $c_1, \dots, c_m$ are classes being considered, $a_1, \dots, a_n$ are possible values of f , $Pr(c_i)$ is the probability of an instance to be labeled with c_i , $Pr(a_j)$ is the probability of f to take a value a_j , and $Pr(c_i|a_j)$ is the probability of an instance to be labeled with c_i given that $f = a_j$.

Example 2: The rule in Fig. 6 has three internal wildcards, i.e., those occurring between “sym” and “ptime” in its pattern. Table 3 illustrates the instantiation feature vectors for

[†]WHISK uses an error tolerance threshold to accept rules with incorrect extractions. However, this threshold is normally small.

the first internal wildcard of this rule observed at the three text portions in Fig. 8 (cf. Example 1) when the number of spaces, the number of plain words, and the presence of the specific plain words indicated in the titles of the fifth and the sixth columns are selected as features. At the [37, 48]-portion, for example, the first internal wildcard is instantiated into a string consisting of one occurrence of space and three plain words, two of which are the indicated specific words, yielding [1, 3, 1, 1] as the resulting instantiation feature vector. Likewise, the instantiation feature vectors for this wildcard observed at the [38, 49]-portion and the [59, 70]-portion are [0, 2, 1, 1] and [0, 1, 0, 0]. ■

4.2 Overlapping-Frame Filtering (OFF)

One symptom phrase is independent of another symptom phrase. Accordingly, when two distinct extracted frames overlap, i.e., when they share a slot filler taken from the same text position, one of them is necessarily a false positive. After removing extractions by WIF, remaining overlapping frames are resolved based on the confidence of class predictions made in the removal process.

The *classification confidence* of a classifier C for an extraction E , denoted by $\text{conf}(C, E)$, indicates how likely C makes a correct prediction for E . Different classification models calculate classification confidence in different ways. In a Naïve Bayes (NB) model, a confidence value is the predicted-class conditional probability for the feature vector of an instance being classified [12]. For a Decision Tree (DT) model, prediction confidence is normally calculated from class distribution of leaf nodes into which an instance is classified [13]. For k -Nearest Neighbor (k NN) and Support Vector Machine (SVM) models, prediction confidence is obtained from calibrating a classifier score, i.e., transforming a classifier score into a class membership probability. A classifier score in a k NN model is the proportion of votes for the winning class among the k training instances closest to a test instance [14], and a classifier score in an SVM model is estimated based on the distance from the classified instance to its closest separating hyperplane [15]. Most machine learning tool sets provide a classification confidence value for each prediction. Table 4 shows the formulas used by the Weka machine learning suite, which is used in our experiment (Sect. 5), for determining the confidence

of classifying an extraction E , represented by a feature vector $\vec{f} = [f_1, \dots, f_n]$, into a class c , where

- $Pr(c)$ is the prior probability of c , $Pr(\vec{f})$ is the probability of $\vec{f}$ in a training vector space, and $Pr(f_j|c)$ is the probability of f_j given c ;
- M is the set of all leaf nodes in a tree, and w_m and $Pr_m(c|\vec{f})$ are the weight and the conditional probability of c given $\vec{f}$, respectively, at a leaf node m ;
- Sim is a function for measuring the similarity between feature vectors and $\mathfrak{N}_k(\vec{f}, c)$ is the set of k nearest neighbors of $\vec{f}$ restricted to c ; and
- ν is a target function for determining a classification score from a feature vector, a and b are tuning parameters for fitting a sigmoid curve to data points determined by classification scores and classes of feature vectors in a training vector space, and β is a constant used for avoiding a scaling problem.

Overlapping extractions often result from application of more than one rule. Since classifiers for different rules are constructed from different sets of wildcard-instantiation feature vectors, their classification confidence values should be weighted in order to make fair comparisons between them. Given a rule r , let C_r denote the classifier for r used in WIF and N_r the number of wildcard-instantiation feature vectors used for constructing C_r . Suppose that R is the set of rules obtained from WHISK and $N_R = \sum_{r \in R} N_r$. Then, for any rule $r \in R$, the *weighted classification confidence* of C_r for an extraction E , denoted by $wc(C_r, E)$, is given by

$$wc(C_r, E) = \frac{N_r}{N_R} \times acc(C_r) \times \text{conf}(C_r, E),$$

where $acc(C_r)$ is the accuracy of C_r with respect to its training set, i.e., the proportion of correct predictions to all predictions when evaluating C_r on training data.

Based on weighted classification confidence, OFF resolves overlapping extractions as follows: An extraction E is identified with a pair $\langle T, r \rangle$, where T is the text portion from which E is obtained and r is the rule that extracts E from T . Assume that $\mathbf{E}_{\text{TEST}}$ is an initially given set of extractions made on a test set. Overlapping extractions in $\mathbf{E}_{\text{TEST}}$ are removed by repeatedly performing the following steps until no overlapping extraction remains:

1. Determine the set $\mathbf{O}$ of all overlapping extractions currently belonging to $\mathbf{E}_{\text{TEST}}$.
2. Determine the set $\text{Min}_{wc}(\mathbf{O})$ consisting of all extractions $E = \langle T, r \rangle \in \mathbf{O}$ such that for every extraction $E' = \langle T', r' \rangle \in \mathbf{O}$,

$$wc(C_r, E) \leq wc(C_{r'}, E').$$
3. Select one arbitrary extraction in $\text{Min}_{wc}(\mathbf{O})$ and remove it from $\mathbf{E}_{\text{TEST}}$.

Table 4 Classification confidence formulas.

Model	Classification confidence
NB	$\frac{Pr(c)}{Pr(\vec{f})} \prod_{j=1}^n Pr(f_j c)$
DT	$\sum_{m \in M} w_m Pr_m(c \vec{f})$
k NN	$\frac{1}{k} \sum_{\vec{f}_p \in \mathfrak{N}_k(\vec{f}, c)} \left(1 - \sqrt{\frac{(Sim(\vec{f}_p, \vec{f}))^2}{n}} \right)$
SVM	$\frac{1}{\exp(-a - b\nu(\vec{f})) + \beta}$

5. Experimental Results and Discussion

5.1 Data Sets and Output Templates

5.1.1 Information Entries and Preprocessing

This work is part of a project supported by the Thailand Research Fund (TRF) and the National Electronics and Computer Technology Center (NECTEC), aiming at the development of a framework for constructing a large-scale medical-related knowledge base in Thailand from various information sources available on the Internet [3]. The overall project framework includes data acquisition, keyword extraction, link construction, and ontology-based knowledge representation. A set of supporting tools for gathering medical text data from Thai web pages were developed and a number of medicinal and pharmaceutical web sites (2759 URLs) were selected as seeds. The obtained data covers 474 diseases and 770 medicinal chemical substances, with approximately 6600 and 3350 information entries, respectively. Disease information entries were organized into disease characteristics, symptoms, cause, treatment, etc.

From textual data gathered in this project, unstructured-text symptom information entries are collected and divided into 3 data sets, i.e., D1, D2, and D3, based on their disease (or illness) groups. D1 comprises information entries obtained from 5 disease groups, i.e., the circulatory system, the urology system, the reproductive system, the eye system, and the ear system; D2 from 6 groups, i.e., the skin/dermal system, the skeletal system, the endocrine system, the nervous system, parasitic diseases, and venereal diseases; D3 from 4 groups, i.e., the respiratory system, the gastrointestinal tract system, infectious diseases, and accidental illness (e.g., illness caused by poisonous spider bites). The collected information entries are preprocessed using a word segmentation program, called CTTEX,[†] and are then partially annotated with semantic class tags with reference to a predefined ontology lexicon using an implemented annotation tool.

Class tags used for semantic annotation, e.g., “Symptom,” “Organ,” and “Hormone,” are taken from entity types collected as part of our medical-related knowledge base construction project [3] and are added to an upper ontology given in [16]. An ontology lexicon is developed by collecting 31,365 Thai words associated with those class tags from the Royal Institute Dictionary of Medicine. The annotation process is semi-automatic: when a word referring to multiple concepts in the lexicon (i.e., a homonym) is found, an appropriate one is selected manually. Among 2,426 resulting annotated words in the three data sets, only 137 words require such manual selection.^{††}

Table 5 characterizes each data set in terms of the number of information entries and their lengths, the number of class tags used in the annotation, and the ontology coverage, i.e., the proportion of annotated words to all words in

Table 5 Data set characteristics.

Data set	No. of info. entries	No. of words per info. entry			No. of class tags	Ratio of annotated words to all words
		Max.	Avg.	Min.		
D1	59	135	60.0	10	22	24.49%
D2	56	155	57.5	8	23	22.23%
D3	58	141	58.1	6	24	25.04%

Table 6 Output templates and their meanings.

Type	Output Template	Meaning
MD1	{OBS <i>O</i> }{ATTR <i>A</i> }{PER <i>T</i> }	An abnormal characteristic <i>A</i> is found at an observed entity <i>O</i> for a time period <i>T</i> .
MD2	{SYM <i>S</i> }{LOC <i>P</i> }{PER <i>T</i> }	A primitive named symptom <i>S</i> appears at a human-body part <i>P</i> for a time period <i>T</i> .

Table 7 Target phrase information.

Type	Data set	No. of distinct target phrases	Target-phrase length			No. of target phrases per info. entry		
			Max.	Avg.	Min.	Max.	Avg.	Min.
MD1	D1	90	11	3.5	2	7	3.6	1
MD1	D2	136	11	3.4	2	11	2.9	1
MD1	D3	160	14	3.3	2	11	3.6	0
MD2	D1	80	15	4.1	2	3	1.4	0
MD2	D2	66	13	4.3	2	5	1.2	0
MD2	D3	65	13	3.8	2	5	1.3	0

the data set.

5.1.2 Symptom Phrases and Output Templates

A collected information entry typically contains several symptom phrases, which provide several kinds of symptom-related information. Two basic types of symptom phrases, referred to as MD1 and MD2, are considered in our experiments. Table 6 gives the output-template forms for the two types along with their intended meanings. The slot PER in the MD1 template is optional. One of the slots Loc and PER, but not both, may be omitted in the MD2 template. The first and the fourth frames in Table 1, for example, are of type MD1, while all other frames in the same table are of type MD2. Table 7 provides some key characteristics of each template type in the data sets, e.g., the number of distinct symptom phrases, target-phrase lengths (in words), and target-phrase density.

5.2 Training Process

5.2.1 Rule Learning

D1 is used as the training corpus. All MD1 and MD2 symptom phrases occurring in D1 are manually tagged with desired output frames for rule learning. Using our implementation of WHISK, 27 rules for the MD1 template and 11

[†]Available at <http://www.mm.co.th/pub/firefox-thai>

^{††}In technical and scientific domains such as medicine, homonyms rarely occur because of the restricted use of language in narrow semantic domains [17], [18].

Table 8 Top five rules for each template type obtained from WHISK.

Type	Pattern	Output template
MD1	*(org)*(col)	{OBS \$1}{ATTR \$2}
MD1	*(org)*(szq)	{OBS \$1}{ATTR \$2}
MD1	*(sec)*(col)	{OBS \$1}{ATTR \$2}
MD1	*(ch)* [†] *(org)	{OBS \$2}{ATTR \$1}
MD1	*(ch)* [†] *(org)	{OBS \$2}{ATTR \$1}
MD2	*(sym)*(org)	{SYM \$1}{LOC \$2}
MD2	*(sym)* [†] *(org)	{SYM \$1}{LOC \$2}
MD2	*(sym)* [†] *(org)	{SYM \$1}{LOC \$2}
MD2	*(sym)*(org)* [†] *(ptime)	{SYM \$1}{LOC \$2}{PER \$3}
MD2	*(sym)*(org)* [†] *(ptime)	{SYM \$1}{LOC \$2}{PER \$3}

rules for the MD2 template are generated. Table 8 shows the top five obtained rules with the lowest Laplacian errors for each template. The length of the longest target phrase observed when a rule yields correct extractions on its training set is taken as the base window size for the rule. Since semantic class tags used for the annotation process represent domain concepts that are rather general (e.g., “Symptom,” “Organ,” “Secretion”) and are common to all the three data sets, it is expected that rules learned from D1 can be used for extracting frames from D2 and D3, which are used as test sets.

5.2.2 Classifier Learning

By applying the obtained rules to the information entries in their respective training data sets using RAW, wildcard instantiations are observed and used as training data for constructing classifiers for the WIF module. The types of features described in Sect. 4.1 are used. Top 10% of features ranked according to their information gain values are selected.[†] The number of selected features accordingly obtained for different rules varies from 2 to 8, with an average of 4.7 selected features per rule. The number of space occurrences is the most commonly selected feature. The average number of specific words used as selected features for a rule is 2.3.

5.3 Experimental Results

The Weka machine learning suite^{††} is employed for classifier learning and evaluation, using its default parameters. Four standard models are used, i.e., Support Vector Machine (SVM) based on the RBF kernel, *k*-Nearest Neighbor (*k*NN), Naïve Bayes (NB), and Decision Tree (DT) using C4.5. As observed during the learning process, 3NN performs slightly better than 1NN, 5NN, and 7NN on our training data sets, and is chosen as a representative of *k*NN.

5.3.1 Evaluation

The proposed framework is evaluated using D2 and D3 as test sets. Recall and precision are used as performance measures, where the former is the proportion of correct extractions to relevant symptom phrases and the latter is that of

Table 9 Results obtained by using SVM classifiers in WIF and OFF.

Type	Test set	RAW		RAW+WIF		RAW+WIF+OFF	
		Recall	Precision	Recall	Precision	Recall	Precision
MD1	D2	74.38	85.00	74.38	92.97	73.75	95.93
MD1	D3	71.63	87.13	70.67	93.04	70.67	97.35
MD2	D2	100.00	40.35	97.10	98.53	97.10	100.00
MD2	D3	95.89	34.15	95.89	87.50	94.52	95.83

correct extractions to all obtained extractions. Table 9 shows the evaluation results obtained from using RAW without any extraction filtering, RAW with WIF (RAW+WIF), and RAW with both WIF and OFF (RAW+WIF+OFF) when SVM classifiers are used (recall and precision are given in percentage). Compared to the results obtained using RAW alone, each of RAW+WIF and RAW+WIF+OFF improves precision while satisfactorily preserving recall of RAW in every row of this table. In particular, for MD2, where RAW has low precision (40.35 for D2 and 34.15 for D3), both WIF and WIF with OFF yield significant precision gains.

Table 10 shows the results obtained by using *k*NN, NB, and DT in WIF and OFF. Compared to Table 9, SVM and these three models give similar filtering performance.

5.3.2 Recall Improvement

As seen in Table 9, recall for MD1 appears to be relatively lower than that for MD2. A detailed examination of MD1 symptom phrases in D2 and D3 reveals that there are 101 target symptom phrases from which RAW fails to make any extraction, i.e., 101 false negatives. These false negatives are divisible into two groups: symptom phrases that match the pattern part of some existing rule, and those that do not. 36 of the 101 false negatives belong to the first group, and the rest belong to the second group. By increasing the size of a window in use, extractions can be made from false negatives in the first group but not from those in the second one.

Further analysis of the obtained MD1 rules shows that most of them have low coverage, i.e., they tend to be overfitting. Referring to the partial domain ontology in Fig. 9, each of the rules contains a tag denoting a subclass of “Gradable quality” in the domain ontology, e.g., the tags “col” (Color), “szq” (Size), and “ch” (Characteristic) in the five MD1 rules in Table 8. Moreover, some rule differs from another only at tags denoting subclasses of “Gradable quality” and can be merged together by generalizing such tags into “gq” (Gradable quality). For example, consider the first two MD1 rules in Table 8. The first rule differs from the second one only at their second tags, i.e., “col” and “szq.” Merging them by generalizing “col” and “szq” upwards into “gq” yields a new rule with higher coverage.

Two approaches are taken so as to reduce the number of false negatives: first, double the window size previously

[†]When chi-square is used instead of information gain, the same selected features are obtained in our experiments. Moreover, when top 20%, instead of top 10%, of all observed features are selected, the overall performance of WIF remains the same.

^{††}Available at <http://www.cs.waikato.ac.nz/ml/weka>

Table 10 Results obtained by using k NN, NB, and DT classifiers.

Type	Test set	RAW		Classifier	RAW+WIF		RAW+WIF+OFF	
		Recall	Precision		Recall	Precision	Recall	Precision
MD1	D2	74.38	85.00	k NN	74.38	92.97	73.75	95.93
MD1	D2	74.38	85.00	NB	74.38	92.97	73.75	95.93
MD1	D2	74.38	85.00	DT	74.38	92.97	73.75	95.93
MD1	D3	71.63	87.13	k NN	70.67	93.04	70.67	97.35
MD1	D3	71.63	87.13	NB	70.19	92.99	70.19	97.33
MD1	D3	71.63	87.13	DT	71.15	91.93	70.67	97.35
MD2	D2	100.00	40.35	k NN	98.55	100.00	97.10	100.00
MD2	D2	100.00	40.35	NB	97.10	98.53	97.10	100.00
MD2	D2	100.00	40.35	DT	94.20	97.10	94.20	98.48
MD2	D3	95.89	34.15	k NN	95.89	83.33	94.52	93.24
MD2	D3	95.89	34.15	NB	95.89	86.42	94.52	94.52
MD2	D3	95.89	34.15	DT	95.89	86.42	94.52	95.83

Fig. 9 Gradable quality subconcepts.

used for each rule, and, secondly, merge rules that can be made identical by merely generalizing tags denoting subclasses of “Gradable quality” upwards into “gq.” Referring to the first approach as 2W and the second one as RG, Table 11 presents the obtained evaluation results when SVM is employed, where 2W+RG denotes the combination of the two approaches. The table shows that RG and 2W+RG yield higher recall gain than 2W in both D2 and D3. Like the results in Table 9, each of RAW+WIF and RAW+WIF+OFF improves precision while satisfactorily preserving recall in every row of Table 11.

Table 12 gives the results obtained by using k NN, NB, and DT classifiers in WIF and OFF when the RG and 2W+RG improvements are made. Compared to Table 11, no significant difference in classification performance is seen.

5.3.3 Comparison with Known-Boundary Extraction

To investigate the performance of our framework in comparison with rule application when target-phrase boundaries are known, we manually locate all target phrases in the test data sets and apply the rules obtained from WHISK directly to these manually identified text portions. Table 13 compares such direct rule application with RAW+WIF+OFF when SVM classifiers and the 2W+RG improvement are employed. It shows that the performance of our framework

is close to that of known-boundary extraction in terms of both recall and precision. Observe that recall obtained from RAW itself for MD1 is already the same as that obtained from known-boundary extraction in both D2 and D3, meaning that no improvement of recall for MD1 can be gained by further extension of window size.

5.3.4 Comparison with Some Baseline Methods

The proposed framework is also compared with three baseline methods. Adopting the idea presented in [19],[20], the first baseline, referred to as *Baseline-1*, divides multi-slot extraction into two phases, i.e., binary-relation extraction and frame construction. Binary classifiers are constructed for predicting whether two consecutive annotated terms (possibly with plain words in between) are slot fillers of the same frame, i.e., whether they are semantically related. Obtained pairs of slot fillers are merged into one multi-slot frame if some of their fillers are extracted from the same text location. The second baseline, referred to as *Baseline-2*, employs a classifier learned from each combination of (two or more) class tags lying within a text window of a fixed size to directly predict whether a combination of annotated terms constitutes a multi-slot frame. The window size used for this baseline is twice as large as the length of the longest target phrase observed on training data. In the third baseline, referred to as *Baseline-3*, a trained Conditional Random Field (CRF) model is used for labeling annotated terms with slot names and also labeling spaces appearing outside target-phrase boundaries. Such outside-boundary space labeling determines whether two consecutive extracted slot fillers belong to the same frame and is necessary for our extraction task since an information entry often contains more than one target phrase. The Weka machine learning suite is again used for the first two baselines, with features and the feature selection process being the same as those employed in Sects. 5.3.1–5.3.3. The CRF++ toolkit[†] is used for implementing the third baseline, with features being words and class tags.

Table 14 compares RAW+WIF+OFF with the first two

[†] Available at <http://crfpp.sourceforge.net>

Table 11 Recall improvement by doubling the window size and rule generalization.

Type	Test set	Improved by	RAW		RAW+WIF		RAW+WIF+OFF	
			Recall	Precision	Recall	Precision	Recall	Precision
MD1	D2	2W	81.25	73.45	80.00	90.14	78.75	96.18
MD1	D2	RG	86.25	78.41	86.25	93.24	86.25	97.18
MD1	D2	2W+RG	89.38	60.59	89.38	91.08	87.50	95.89
MD1	D3	2W	83.17	71.78	82.21	90.48	79.81	97.65
MD1	D3	RG	86.54	76.92	85.58	93.68	85.58	96.22
MD1	D3	2W+RG	89.42	56.19	88.46	91.54	88.46	96.84

Table 12 Results obtained by using *k*NN, NB, and DT classifiers, with recall improvement.

Type	Test set	Improved by	RAW		Classifier	RAW+WIF		RAW+WIF+OFF	
			Recall	Precision		Recall	Precision	Recall	Precision
MD1	D2	RG	86.25	78.41	<i>k</i> NN	85.00	93.15	85.00	95.77
MD1	D2	RG	86.25	78.41	NB	85.00	93.15	85.00	95.77
MD1	D2	RG	86.25	78.41	DT	86.25	93.88	86.25	97.18
MD1	D2	2W+RG	89.38	60.59	<i>k</i> NN	87.50	90.32	87.50	95.89
MD1	D2	2W+RG	89.38	60.59	NB	86.25	91.39	86.25	96.50
MD1	D2	2W+RG	89.38	60.59	DT	89.38	89.94	87.50	95.89
MD1	D3	RG	86.54	76.92	<i>k</i> NN	85.58	94.18	85.10	95.68
MD1	D3	RG	86.54	76.92	NB	85.58	93.68	85.58	96.22
MD1	D3	RG	86.54	76.92	DT	86.06	91.79	86.06	96.76
MD1	D3	2W+RG	89.42	56.19	<i>k</i> NN	89.42	91.18	87.50	95.79
MD1	D3	2W+RG	89.42	56.19	NB	87.02	91.41	87.02	96.79
MD1	D3	2W+RG	89.42	56.19	DT	88.46	88.46	87.98	96.83

Table 13 Comparison with known-boundary extraction.

Type	Test set	Known-boundary		RAW		RAW+WIF		RAW+WIF+OFF	
		Recall	Precision	Recall	Precision	Recall	Precision	Recall	Precision
MD1	D2	89.38	96.96	89.38	60.59	89.38	91.08	87.50	95.89
MD1	D3	89.42	97.38	89.42	56.19	88.46	91.54	88.46	96.84
MD2	D2	100.00	100.00	100.00	40.35	97.10	98.53	97.10	100.00
MD2	D3	98.63	96.00	95.89	34.15	95.89	87.50	94.52	95.83

Table 14 Comparison with Baseline-1 and Baseline-2.

Type	Test set	Classifier	RAW+WIF+OFF			Baseline-1			Baseline-2		
			Recall	Precision	F-measure	Recall	Precision	F-measure	Recall	Precision	F-measure
MD1	D2	SVM	87.50	95.89	91.50	67.50	83.72	74.74	80.63	89.58	84.87
MD1	D2	<i>k</i> NN	87.50	95.89	91.50	66.30	82.20	73.40	80.63	93.48	86.58
MD1	D2	NB	86.25	96.50	91.09	79.38	88.81	83.83	76.88	86.62	81.46
MD1	D2	DT	87.50	95.89	91.50	62.50	83.33	71.43	81.88	93.57	87.33
MD1	D3	SVM	88.46	96.84	92.46	78.37	90.56	84.03	76.92	87.43	81.84
MD1	D3	<i>k</i> NN	87.50	95.79	91.46	50.48	72.41	59.49	80.29	88.83	84.34
MD1	D3	NB	87.02	96.76	91.63	81.25	93.37	86.89	77.40	88.46	82.56
MD1	D3	DT	87.89	96.83	92.14	63.94	85.81	73.28	80.29	89.30	84.56
MD2	D2	SVM	97.10	100.00	98.53	97.10	97.10	97.10	60.87	71.19	65.63
MD2	D2	<i>k</i> NN	97.10	100.00	98.53	88.41	76.25	81.88	91.30	91.30	91.30
MD2	D2	NB	97.10	100.00	98.53	95.65	72.53	82.50	84.06	78.38	81.12
MD2	D2	DT	94.20	98.48	96.29	95.65	81.48	88.00	92.75	95.52	94.12
MD2	D3	SVM	94.52	95.83	95.17	86.30	86.30	86.30	82.19	80.00	81.08
MD2	D3	<i>k</i> NN	94.52	93.24	93.88	84.93	76.54	80.52	90.41	90.41	90.41
MD2	D3	NB	94.52	94.52	94.52	93.15	72.34	81.44	87.67	75.29	81.01
MD2	D3	DT	94.52	95.83	95.17	78.08	72.15	75.00	86.30	87.50	86.90

baselines when SVM, *k*NN, NB, and DT classifiers are used and the 2W+RG improvement is made. The traditional F-measure, i.e., the harmonic mean of recall and precision, defined by

$$\text{F-measure} = \frac{2 \times \text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}},$$

is shown in the table. The results reveal that our framework performs better than these two baselines. Only in one case, i.e., the 9th row (MD2 in D2 using SVM), the first baseline gives comparable performance to RAW+WIF+OFF; and also only in one case, i.e., the 12th row (MD2 in D2 using DT), the second baseline offers comparable perfor-

Table 15 Comparison with Baseline-1, Baseline-2, and Baseline-3 (average performance values over SVM, k NN, NB, and DT classifiers are used for Baseline-1 and Baseline-2).

Type	Test set	RAW+WIF+OFF			Baseline-1			Baseline-2			Baseline-3		
		Recall	Precision	F-measure	Recall	Precision	F-measure	Recall	Precision	F-measure	Recall	Precision	F-measure
MD1	D2	87.19	96.04	91.40	68.92	84.52	75.92	80.00	90.81	85.06	51.25	67.21	58.16
MD1	D3	87.72	96.56	91.92	68.51	85.54	76.08	78.73	88.51	83.33	54.81	69.51	61.29
MD2	D2	96.38	99.62	97.97	94.20	81.84	87.59	82.25	84.10	83.16	43.48	78.95	56.07
MD2	D3	94.52	94.86	94.69	85.62	76.83	80.99	86.64	83.30	84.94	56.16	85.42	67.77

mance to it. The table also shows that the choice of classification models does not obviously affect the performance of RAW+WIF+OFF, while the performance of the two baselines significantly varies with classification models.

Table 15 shows the average performance values[†] over the four classification models presented in Table 14, along with the performance of third baseline (CRF-based extraction). It indicates that RAW+WIF+OFF clearly outperforms all the three baseline frameworks. The obtained performance gain of RAW+WIF+OFF over the third baseline are in line with the observation reported in [21] that an extraction framework based on regular expressions (e.g., a WHISK-based framework) tends to perform better than a CRF-based method when a training set is rather small.

6. Experimental Application in Other Domains

The framework is also evaluated in four other domains, i.e., *soccer match reports* (SM), *soccer player transfers* (PT), *stock prices* (SP), and *company dividends* (CD). Thai unstructured-text information entries in these four domains are collected from on-line newspapers. Table 16 describes the obtained information entries after they are preprocessed (cf. Table 5). A lexicon containing 803 soccer player names and 533 soccer team names, collected as part of an ongoing project on developing an alias extraction system [22], is used for semantic annotation in the domains SM and PT. From the SET company list^{††} provided by the Stock Exchange of Thailand, 565 company names along with their acronyms are collected for constructing a lexicon for the domains SP and CD. Table 17 shows the output template used in each domain and its intended meaning. The slot DIF in the SP template and the slot Tot in the CD template are optional. For each domain, 50% of the information entries are randomly selected for rule learning and the rest are used as test data. Table 18 characterizes target phrases in the obtained data sets (cf. Table 7). Using WHISK, 7, 20, 18, and 5 rules, respectively, are generated for the SM, PT, SP, and CD templates. The feature types and the feature selection process used in Sect. 5 are also employed for classifier learning in these four domains.

Table 19 gives the evaluation results in these four domains when SVM classifiers are used in WIF and OFF. Like the results in the medical-symptom domain, both RAW+WIF and RAW+WIF+OFF improve precision while preserving high recall of RAW in every row of the table.

Table 16 Data set characteristics.

Domain	No. of info. entries	No. of words per info. entry			No. of class tags	Ratio of annotated words to all words
		Max.	Avg.	Min.		
SM	294	242	61.2	20	14	16.77%
PT	448	161	54.2	16	13	13.23%
SP	132	330	94.7	20	15	14.70%
CD	191	312	72.6	20	13	10.67%

Table 17 Output templates and their meanings.

Domain	Output Template	Meaning
SM	{PLY P }{ACT A }{MIN N }	A player P takes a game action A in the N th minute.
PT	{CL1 $C1$ }{CL2 $C2$ }{FEE F }{PLY P }	A club $C1$ pays a club $C2$ a transfer fee F for a player P .
SP	{COM C }{PRC S }{DIF D }	The share price of a company C closes at an amount S , with an increase or decrease of an amount D .
CD	{COM C }{DIV D }{TOT T }	A company C announces a dividend of an amount D per share, with a total payout of an amount T .

Table 18 Target phrase information.

Domain	Data set	No. of distinct target phrases	Target-phrase length			No. of target phrases per info. entry		
			Max.	Avg.	Min.	Max.	Avg.	Min.
SM	Training	104	32	15.5	5	3	0.6	0
SM	Test	112	38	15.9	4	8	0.9	0
PT	Training	90	64	26.5	11	2	0.5	0
PT	Test	108	57	29.1	12	2	0.4	0
SP	Training	109	74	18.0	5	6	1.4	0
SP	Test	122	49	13.0	8	8	2.3	0
CD	Training	106	73	19.7	5	7	1.2	0
CD	Test	117	66	18.1	5	7	1.2	0

Table 20 compares our framework, when the base window size is doubled, to rule application with known target-phrase boundaries. Again, the performance of our framework is close to that of known-boundary extraction.

Refer to the three baseline methods described in Sect. 5.3.4. Table 21 compares our framework, again with the base window size being doubled, to the first two baselines when SVM, k NN, NB, and DT classifiers are used. Table 22 shows average results obtained from the two baselines over the four classifiers, along with the results obtained from the third baseline. Again, our framework outperforms the

[†]F-measure is recalculated using average recall and precision.

^{††}Available at <http://www.set.or.th/th/company/companylist.html>

three baselines. Note that the performance of each method in the domain PT is lower than that in the other three domains, the reason being that in PT a semantic class tag does not uniquely determines a slot, i.e., a term annotated as a soccer club may be a filler of the slot CL1 or the slot CL2, making the extraction task more difficult.

Table 19 Results obtained by using SVM classifiers in WIF and OFF.

Domain	Window size	RAW		RAW+WIF		RAW+WIF+OFF	
		Recall	Precision	Recall	Precision	Recall	Precision
SM	1W	86.61	54.19	83.93	85.45	83.03	95.87
SM	2W	94.64	39.85	91.07	85.00	90.18	93.52
PT	1W	76.85	45.60	75.93	80.39	75.93	85.42
PT	2W	94.44	27.57	91.67	67.81	90.74	82.35
SP	1W	86.07	51.72	86.07	78.36	86.07	89.74
SP	2W	93.44	38.00	91.80	69.14	91.80	93.33
CD	1W	83.76	68.53	82.91	87.39	82.91	89.81
CD	2W	96.58	56.50	94.87	84.09	94.87	88.10

Table 20 Comparison with known-boundary extraction.

Domain	Known-boundary		RAW	RAW+WIF+OFF	
	Recall	Precision		Recall	Precision
SM	94.64	94.64	94.64	90.18	93.52
PT	94.44	85.71	94.44	90.74	82.35
SP	95.90	92.86	93.44	91.80	93.33
CD	97.44	91.20	96.58	94.87	88.10

Table 21 Comparison with Baseline-1 and Baseline-2.

Domain	Classifier	RAW+WIF+OFF			Baseline-1			Baseline-2		
		Recall	Precision	F-measure	Recall	Precision	F-measure	Recall	Precision	F-measure
SM	SVM	90.18	93.52	91.82	57.14	55.17	56.14	82.14	89.32	85.58
SM	kNN	87.50	93.33	90.32	29.46	37.93	33.17	80.36	89.11	84.51
SM	NB	83.93	95.92	89.53	72.73	70.55	71.62	77.68	89.69	83.25
SM	DT	88.39	91.67	90.00	35.71	51.95	42.33	80.36	88.24	84.11
PT	SVM	90.74	82.35	86.34	45.37	24.50	31.82	36.11	54.17	43.33
PT	kNN	85.19	77.31	81.06	29.63	16.84	21.47	47.22	62.20	53.68
PT	NB	89.81	83.62	86.60	81.48	33.72	47.70	50.93	59.14	54.73
PT	DT	85.19	77.31	81.06	43.52	77.31	55.69	38.89	61.76	47.73
SP	SVM	90.98	92.50	91.73	71.31	75.00	73.11	79.51	83.62	81.51
SP	kNN	90.98	88.10	89.52	40.16	36.84	38.43	77.87	82.61	80.17
SP	NB	92.62	90.40	91.50	89.34	77.86	83.21	77.87	83.33	80.51
SP	DT	91.80	88.19	89.96	72.95	65.93	69.26	78.69	86.49	82.40
CD	SVM	94.87	88.10	91.36	70.94	79.81	75.11	81.20	75.40	78.19
CD	kNN	93.16	87.90	90.45	85.47	83.33	84.39	80.34	79.66	80.00
CD	NB	95.73	88.89	92.18	54.70	63.37	58.72	81.20	78.51	79.83
CD	DT	92.00	85.71	88.74	84.62	80.49	82.50	80.34	80.34	80.34

Table 22 Comparison with Baseline-1, Baseline-2, and Baseline-3 (average performance values over SVM, kNN, NB, and DT classifiers are used for Baseline-1 and Baseline-2).

Domain	RAW+WIF+OFF			Baseline-1			Baseline-2			Baseline-3		
	Recall	Precision	F-measure	Recall	Precision	F-measure	Recall	Precision	F-measure	Recall	Precision	F-measure
SM	87.50	93.61	90.45	48.76	53.90	51.20	80.13	89.09	84.37	17.86	62.50	27.78
PT	87.73	80.15	83.77	50.00	38.09	43.24	43.29	59.32	50.05	6.48	87.50	12.07
SP	91.60	89.80	90.69	68.44	63.91	66.10	78.48	84.01	81.15	43.44	55.79	48.85
CD	93.94	87.65	90.69	73.93	76.75	75.31	80.77	78.48	79.61	52.99	83.78	64.92

7. Related Works

Application of WHISK to IE tasks in medical-related domains was reported in [23] and [24]—the former is concerned with information on biomedical events and the latter with drug treatment information. Both of them take English documents as information sources and use linguistic tools, such as part-of-speech taggers and chunk parsers, along with ontology-based semantic tagging for text pre-processing. While a preliminary investigation was given in [24] without introducing any supplementary technique, an extraction verification module was proposed in [23] for removing incorrectly extracted biomedical events based on a maximum entropy (ME) classification method. The verification module in [23] employs a learned ME classifier to predict a class of the an extracted slot filler, and removes an extracted frame whose components contradict the class assigned by the classifier. By contrast, the WIF module in our framework uses a classifier to predict rule application across a symptom-phrase boundary, compensating for the unavailability of a phrase boundary analyzer.

Very few works on IE from Thai text were reported in the literature. Sukhahuta and Smith [25] proposed strategies for Thai-text IE using corpus-based syntactic surface analysis based on predefined context-free grammar rules. The extraction precision of their developed system is still rela-

tively low; as pointed out in [25] itself, one main cause of errors comes from the ambiguity of the sentence structure, due to which a parser is unable to determine sentence boundaries, resulting in parse-tree construction failure, in particular, when constituents such as subject or verb do not appear in a sentence as expected in the grammar rules. Only hand-crafted triggering-term patterns were considered in [25]; extraction-pattern learning was not discussed. Narupiyakul et al. [26] introduced a method for automated Thai-text IE in a housing advertisement corpus by using rule-based syllable segmentation for text preprocessing and applying Hidden Markov Models, with the Viterbi algorithm, to extract individual target fields independently. Target fields along with their prefixes and suffixes are tagged in the level of syllables, which are far less meaningful than words and semantic classes. Moreover, individual-field extraction, such as that in [26], has a serious limitation for a significant number of applications, in particular, when an information entry contains fillers of more than one frame, e.g., it cannot relate a particular person with his address when an information entry contains several person names and addresses.

8. Conclusions

From a set of manually collected symptom phrases, IE rules are created using our implementation of WHISK. To apply the obtained rules to unstructured-text information entries without predetermining symptom-phrase boundaries, rule application using sliding windows is introduced. Filtering techniques are proposed for the removal of false positives resulting from rule application across symptom-phrase boundaries and those resulting from overlapping extractions. The experimental results show that these techniques improve extraction precision while satisfactorily preserving high recall of sliding-window rule application. When the window size is sufficiently large, the performance of the proposed framework appears to be fairly close to that of rule application with manually located target phrases. Further works include extension of the types of target phrases and in-depth analysis of how the ontology-based semantic frames extracted from symptom phrases facilitate logic-based medical diagnosis reasoning.

Acknowledgement

This work was supported by the Thailand Office of Higher Education Commission and the Thailand Research Fund (TRF), under Grant No. PHD/0056/2550 (Royal Golden Jubilee Ph.D. Program) and National Research University Project.

References

- [1] G. Klyne and J.J. Carroll, "Resource description framework (RDF): Concepts and abstract syntax," W3C Recommendation, Feb. 2004, www.w3.org/TR/rdf-concepts
- [2] P.F. Patel-Schneider, P. Hayes, and I. Horrocks, "OWL Web ontology language: Semantics and abstract syntax," W3C Recommendation, Feb. 2004, www.w3.org/TR/owl-semantics
- [3] T. Theeramunkong, P. Iamtana-anan, C. Nattee, A. Suriyawongkul, E. Nantajeewarawat, and P. Aimmanee, "A framework for constructing a Thai medical knowledge base," Proc. 2nd International Conference on Knowledge, Information and Creativity Support Systems, pp.45–50, Ishikawa, Japan, 2007.
- [4] S. Soderland, "Learning information extraction rules for semi-structured and free text," Machine Learning, vol.34, no.1-3, pp.233–272, 1999.
- [5] P. Intarapaiboon, E. Nantajeewarawat, and T. Theeramunkong, "Extracting semantic frames from Thai medical-symptom phrases with unknown boundaries," Proc. 3rd Asian Semantic Web Conference, Bangkok, Thailand, Lect. Notes Comput. Sci., vol.5367, pp.390–404, 2008.
- [6] N. Danvivathana, The Thai Writing System, Helmut Buske Verlag, Hamburg, 1987.
- [7] W. Aroonmanakun, "Thoughts on word and sentence segmentation in Thai," Proc. 7th International Symposium on Natural Language Processing, pp.85–90, Pattaya, Thailand, 2007.
- [8] M.E. Califf and R.J. Mooney, "Bottom-up relational learning of pattern matching rules for information extraction," J. Mach. Learn. Res., vol.4, pp.177–210, 2003.
- [9] D. Freitag, "Machine learning for information extraction in informal domains," Mach. Learn., vol.39, no.2-3, pp.169–202, 2000.
- [10] C.-H. Chang, M. Kaye, M.R. Girgis, and K.F. Shaalan, "A survey of Web information extraction systems," IEEE Trans. Knowl. Data Eng., vol.18, no.10, pp.1411–1428, 2006.
- [11] I. Muslea, "Extraction patterns for information extraction tasks: A survey," Proc. AAAI-99 Workshop on Machine Learning for Information Extraction, pp.1–6, Orlando, Florida, 1999.
- [12] P. Domingos and M. Pazzani, "Beyond independence: Conditions for the optimality of the simple Bayesian classifier," Proc. 13th International Conference on Machine Learning, pp.105–112, Bari, Italy, 1996.
- [13] N.V. Chawla and D.A. Cieslak, "Evaluating probability estimates from decision trees," Proc. AAAI Workshop on Evaluation Methods for Machine Learning, pp.18–23, Boston, MA, 2006.
- [14] A.F. Atiya, "Estimating the posterior probabilities using the k-nearest neighbor rule," Neural Comput., vol.17, no.3, pp.731–740, 2005.
- [15] J. Abfal, H.-P. Kriegel, A. Pryakhin, and M. Schubert, "Multi-represented classification based on confidence estimation," Proc. 11th Pacific-Asia Conference on Knowledge Discovery and Data Mining, Nanjing, China, Lecture Notes in Artificial Intelligence, vol.4426, pp.23–34, 2007.
- [16] H. Helbig, Knowledge Representation and the Semantics of Natural Language, Springer, Berlin Heidelberg, 2006.
- [17] S.B. Johnson, "A semantic lexicon for medical language processing," J. American Medical Informatics Association, vol.6, no.3, pp.205–218, 1999.
- [18] R. Grishman and R. Kittredge, Analyzing Language in Restricted Domains: Sublanguage Description and Processing, Erlbaum Associates, Hillsdale, 1986.
- [19] R. McDonald, F. Pereira, S. Kulick, S. Winters, Y. Jin, and P. White, "Simple algorithms for complex relation extraction with applications to biomedical IE," Proc. 43rd Annual Meeting of Association for Computational Linguistics, Ann Arbor, Michigan, pp.491–498, 2005.
- [20] Y. Liu, Z. Shi, and A. Sarkar, "Exploiting rich syntactic information for relation extraction from biomedical articles," Proc. North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Companion Volume, pp.97–100, Rochester, New York, 2007.
- [21] Y. Li, R. Krishnamurthy, S. Raghavan, and S. Vaithyanathan, "Regular expression learning for information extraction," Proc. 2008 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, pp.21–30, Honolulu,

- Hawaii, 2008.
- [22] T. Suwanapong, T. Theeramunkong, and E. Nantajeewarawat, "The vector space models for finding co-occurrence names as aliases in Thai sports news," *Proc. 2nd Asian Conference on Intelligent Information and Database Systems*, pp.122–130, Hue City, Vietnam, 2010.
 - [23] E. Kim, Y. Song, C. Lee, K. Kim, G. Lee, and B.-K. Yi, "Two-phase learning for biological event extraction and verification," *ACM Trans. Asian Language Information Processing*, vol.5, no.1, pp.61–73, 2006.
 - [24] C.-H. Lee, J.-C. Na, and C. Khoo, "Towards ontology enrichment with treatment relations extracted from medical abstracts," *Proc. 9th International Conference on Asian Digital Libraries*, Kyoto, Japan, *Lect. Notes Comput. Sci.*, vol.4312, pp.419–428, 2006.
 - [25] R. Sukhahuta and D. Smith, "Information extraction strategies for Thai documents," *International Journal of Computer Processing of Oriental Languages*, vol.14, no.2, pp.153–172, 2001.
 - [26] L. Narupiyakul, C. Thomas, N. Cercone, and B. Sirinaovakul, "Thai syllable-based information extraction using hidden markov Models," *Proc. 5th International Conference on Intelligent Text Processing*, *Lect. Notes Comput. Sci.*, vol.2945, pp.537–546, Seoul, Korea, 2004.

Peerasak Intarapaiboon is currently a PhD student at the School of Information, Communication, and Computer Technology, Sirindhorn International Institute of Technology, Thammasat University. His research interests include knowledge representation, information extraction, automated reasoning, and machine learning.

Ekawit Nantajeewarawat received his DEng in Computer Science from the Asian Institute of Technology. His research interests include knowledge representation, information extraction, automated reasoning, and formal ontology languages.

Thanaruk Theeramunkong received his doctoral degree in Computer Science from Tokyo Institute of Technology. His current research interests include data mining, machine learning, natural language processing, and information retrieval.