PAPER

An Application of Intuitionistic Fuzzy Sets to Improve Information Extraction from Thai Unstructured Text

Peerasak INTARAPAIBOON[†a)] ***and*** **Thanaruk THEERAMUNKONG**[††b)], ***Members***

SUMMARY Multi-slot information extraction, also known as frame extraction, is a task that identify several related entities simultaneously. Most researches on this task are concerned with applying IE patterns (rules) to extract related entities from unstructured documents. An important obstacle for the success in this task is unknowing where text portions containing interested information are. This problem is more complicated when involving languages with sentence boundary ambiguity, e.g. the Thai language. Applying IE rules to all reasonable text portions can degrade the effect of this obstacle, but it raises another problem that is incorrect (unwanted) extractions. This paper aims to present a method for removing these incorrect extractions. In the method, extractions are represented as intuitionistic fuzzy sets, and a similarity measure for IFSs is used to calculate distance between IFS of an unclassified extraction and that of each already-classified extraction. The concept of k nearest neighbor is adopted to design whether the unclassified extraction is correct or not. From the experiment on various domains, the proposed technique improves extraction precision while satisfactorily preserving recall.

key words: *intuitionistic fuzzy set, similarity measure, multi-slot information extraction*

1. Introduction

Information extraction (IE) from unstructured text normally involves linguistic patterns, domain-specific lexicons, and conceptual descriptions of an application domain, i.e., domain ontologies. While an ideal domain ontology is arguably language-independent, linguistic patterns and lexicons rely heavily on the language in which the source textual information appears. Due to language-structure differences, some basic language processing tools available in one language may be unavailable in another language. When an IE framework is applied in a different language, the framework often needs modification and supplementary techniques are often necessary.

Multi-slot information extraction, also known as frame extraction, is a task that identify several related entities simultaneously. Most researches on this task are concerned with applying IE patterns (rules) to extract related entities from unstructured documents. A well-known supervised algorithm for constructing pattern-based IE rules is WHISK [1] It learns IE rules from a set of hand-tagged training phrases. Information sources of our target IE task are, however, collections of text paragraphs, called *information entries*, rather than collections of text portions identified beforehand as potential target phrases. Each information entry typically contains several target phrases along with other text portions. Locating potential target phrases in an information entry requires a chunk parser and is thus not currently achievable for Thai text. Applying IE rules to documents with unknown target-phrase locations tends to make false positives (incorrect extractions), since these rules probably match with text portions that do not convey information of interest. (Even potential target phrases are determined, incorrect extractions are probably made.) As such, several IE frameworks come up with components to alleviate the detriment suffered by the issue. The components can be grouped to two approaches. One approach is removing inefficient rules [2], [3]. Ideally, the approach is expected that only correct extractions should be produced, whenever the remaining rules match text portions. Each rule is usually applicable into a few text portion. Thus, the main problem for this direction is that many rules are required in order to extract all pieces of interesting information. An alternative approach uses the all IE rules with the assumption that every wanted information is extracted. Of course, more incorrect frames than the first approach are observed. Then, the IE frameworks based on this approach are embedded with incorrect-extraction filtering modules, e.g. [4]–[7]. From a machine-learning viewpoint, the task of detecting false extractions can be reduced to a binary classification problem.

Recently, intuitionistic fuzzy set (IFS) [8] has been much explored in both theory and application. Differing from representation of a fuzzy set (FS) [9], an IFS considers both the membership and non-membership of elements belonging or not belonging to such a set. IFS is therefore more flexible to handle the uncertainty than FS. Measuring similarity and distance between IFSs is one of most research areas to which many researchers have focused. After Dengfeng [10] gave the axiomatic definition of similarity measures between IFSs, various similarity measures have been proposed continuously [11]–[16]. One of most applications of IFS similarity measures is classification problems. Khatibiand and Montazerm [15] conducted experiments for bacterial classification using similarity measures for FSs and IFSs. The results indicated that each measure for IFSs outperformed that for FSs. In the Ye’s research [16], cosine and

Manuscript received December 28, 2017.
Manuscript revised April 13, 2018.
Manuscript publicized May 23, 2018.

[†]The author is with Department of Mathematics and Statistics, Faculty of Science and Technology, Thammasat University, Phathum Thani, Thailand.

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

a) E-mail: ipeerasak@siit.tu.ac.th (Corresponding author)
b) E-mail: thanaruk@siit.tu.ac.th

DOI: 10.1587/transinf.2017EDP7423

เป็น|โรค|ที่|พบ|บ่อย|หลัง|จาก|เป็น|ไข้หวัด|~|ผู้ป่วย|ส่วนใหญ่|มัก|จะ|มี|[sec เสมหะ]|เป็น|[col สีเขียว]|~|
มี|[sym อาการเจ็บ]|ที่|บริเวณ|[org หน้าอก]|อยู่|เป็น|เวลา|นาน|~|[ptime 6-12 วัน]|~|มี|[sym อาการไอ]|จน|
เกิด|[sym อาการเจ็บ]|ที่|[org ชายโครง]|อยู่|นาน|~|[ptime 3-4 วัน]|~|ผู้ป่วย|อาจ|มี|สุขภาพ|ทั่วไป|แข็งแรง|. . .

Fig. 1 A portion of a partially annotated word-segmented information entry

It is a disease that often begins after flu. A patient may have [col *green*] [sec *mucus*], *and may have a* [sym *pain*] *in his* [org *chest*], *which lasts* [ptime *6-12 days*], *and a* [sym *cough*] *that leads to a* [sym *pain*] *in his* [org *lower rib cage*] *lasting* [ptime *3-4 days*]. *A patient may have regular health . . .*

Fig. 2 A literal English translation of the partially annotated Thai text in Fig. 1

weighted similarity measures for IFSs were proposed and applied to a small medical diagnosis problem.

By the success of research in IFS, especially IFS-based techniques for classification problems, it is anticipated that IFS technologies will contribute to improve performance of an IE framework. This work presents an IFS-based method aimed to eliminate incorrect extractions. The main contribution of this work is twofold: (i) representation an extracted frame in terms of an IFS and (ii) applying a similarity measure between IFSs for removing incorrect extraction.

The remainder of the paper proceeds as follows: Section 2 provides a literature review about information extraction with incorrect extraction removal. Section 3 explains a pattern-based IE framework from Thai texts. Section 4 reviews IFS and similarity measures for IFSs. Section 5 presents our filtering method, then the experiments is detailed in Sect. 6. Finally, Sect. 7 gives conclusions and outlines future works.

2. Related Works

From a machine-learning viewpoint, the task of detecting false extractions can be reduced to a binary classification problem. A classification can be constructed to predict whether extractions are correct. In [4], biological events, each of which consists of three slots—one interaction type, one effect, and one reactant—were extracted from unstructured texts using a pattern-based strategy. In order to determine whether an extracted event is correct, a maximum entropy classifier is employed to assign one slot type to each slot filer in the event. When the slot type of a slot filler assigned by the classifier is inconsistent with that by the IE pattern the extracted event is discarded. Similarly, in [5], an pattern-based IE framework to extract multi-slot frames was proposed. To improve precision by removing false extraction, two extraction filtering modules were proposed. The first module uses a binary classifier, e.g. naïve bayes and support vector machine, for prediction of rule application across a target-phrase boundary; the second one uses weighted classification confidence to resolve conflicts arising from overlapping extractions. In [6], linguistic patterns were used for extracting medication information, including medical name, dosage, frequency, duration, and reason, from free-text medical records. Occasionally, medical records contain side effects which are out of scope and usually extracted as reasons. A hand-crafted semantic rule set was constructed and used to filter out such side-effect statements.

3. Information Extraction from Thai Texts

This section briefly explains the idea of domain-specific information extraction for Thai unstructured texts using extraction rules.

3.1 Preprocessing

By detecting paragraph breaks, a text document is decomposed into paragraphs, referred to as *information entries*, then word segmentation is applied to all information entries as part of a preprocessing step. A domain-specific ontology, along with a lexicon for concepts in the ontology, is then employed to partially annotate word-segmented phrases with tags denoting the semantic classes of occurring words with respect to the lexicon.

In the medical domain, as an example, suppose 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. Figure 1 illustrates a portion of word-segmented and partially annotated information entry describing acute bronchitis, obtained from the text-preprocessing phase, where '|' indicates a word boundary, '~' signifies a space, and the tags "sec," "col," "sym," "org," and "ptime" denote the semantic classes "Secretion," "Color," "Symptom," "Organ," and "Time period," respectively, in our medical-symptom domain ontology. The portion contains three target symptom phrases, which are underlined in the figure. Figure 2 provides a literal English translation of this text portion; the translations of the three target phrases are also underlined. Figure 3 shows the frame required to be extracted from the second underlined symptom phrase in Fig. 1. It contains three slots, i.e., Sym, Loc, and Per, which stand for "symptom," "location," and "period," respectively.

Target phrase: |มี|[sym อาการเจ็บ]|ที่|บริเวณ|[org หน้าอก]|อยู่|เป็น|เวลา|นาน|~|[ptime 6-12 วัน]|
English translation: *have a* [sym *pain*] *in his* [org *chest*], *which lasts* [ptime *6-12 days*]

Extracted frame: {Sym [sym อาการเจ็บ]}{Loc [org หน้าอก]}{Per [ptime 6-12 วัน]}
English translation: {Sym [sym *pain*]}{Loc [org *chest*] }{Per [ptime *6-12 days*]}

Fig. 3 A target phrase and an extracted frame

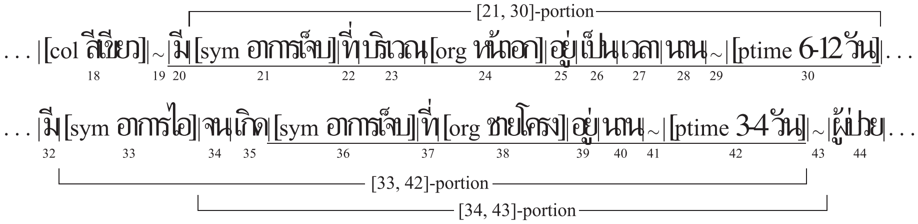
[21, 30]-portion
. . . |[col สีเขียว]|~|มี|[sym อาการเจ็บ]|ที่|บริเวณ|[org หน้าอก]|อยู่|เป็น|เวลา|นาน|~|[ptime 6-12 วัน]|. . .
18 19 20 21 22 23 24 25 26 27 28 29 30
. . . |มี|[sym อาการไอ]|จน|เกิด|[sym อาการเจ็บ]|ที่|[org ชายโครง]|อยู่|นาน|~|[ptime 3-4 วัน]|~|ผู้ป่วย|. . .
32 33 34 35 36 37 38 39 40 41 42 43 44
[33, 42]-portion
[34, 43]-portion

Fig. 5 Text portions from which extractions are made when the rule in Fig. 4 is applied to the information entry in Fig. 1 using a 10-word sliding window

Portion	Extracted frame	Correctness
[21, 30]	{Sym [sym อาการเจ็บ]}{Loc [org หน้าอก]}{Per [ptime 6-12 วัน]}	Correct
[33, 42]	{Sym [sym อาการไอ]}{Loc [org ชายโครง]}{Per [ptime 3-4 วัน]}	Incorrect
[34, 43]	{Sym [sym อาการเจ็บ]}{Loc [org ชายโครง]}{Per [ptime 3-4 วัน]}	Correct

Fig. 6 Frames extracted from the text portions in Fig. 5 by the rule in Fig. 4

Pattern: *(sym)*(org)*นาน*(ptime)
Output template: {Sym $1}{Loc $2}{Per $3}

Fig. 4 An IE rule example

3.2 IE Rules and Rule Application

A well-known supervised rule learning algorithm, called WHISK [1], is used as the core algorithm for constructing extraction rules. Figure 4 gives a typical example of an IE rule. Its pattern part contains (i) three triggering class tags, i.e., sym, org, and ptime, (ii) four internal wildcards, and (iii) one triggering word (between the last two 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 target phrase in Fig. 3, this rule yields the extracted frame shown in the same figure.

WHISK rules are usually applied to individual sentences. In the Thai writing system, however, the end point of a sentence is usually not specified. To apply IE rules to free text with unknown boundaries of sentences and potential target text portions, rule application using sliding windows (RAW) is employed. Roughly speaking, by RAW, a particular rule is applied to each l-word portion of an information entry one-by-one sequentially, where the window size, l, is predefined depending on the rule. As shown in Fig. 5, when the rule in Fig. 4 is applied to the information entry in Fig. 1 using a 10-word sliding window, it makes extractions from the $[21, 30]$-portion, the $[33, 42]$-portion, and the $[34, 43]$-portion of the entry. Figure 6 shows the resulting extracted frames. Only the extractions made from the first and third portions are correct. When the rule is applied to the second 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.

4. Intuitionistic Fuzzy Sets and Their Similarity Measures

In this section, some basic concepts for IFSs and their similarity measures are presented. For the convenience of explanation, the following notations are used hereinafter: $X = \{x_1, x_2, \ldots, x_h\}$ is a discrete universe of discourse and $IFS(X)$ is the class of all IFSs of X. Atanassov [8] defined an intuitionistic fuzzy set A in $IFS(X)$ as follows:

$$A = \{\langle x, \mu_A(x), \nu_A(x)\rangle | x \in X\} \tag{1}$$

which is characterized by a membership function $\mu_A(x)$ and a non-membership function $\nu_A(x)$. The two functions are defined as:

$$\mu_A : X \to [0, 1], \tag{2}$$
$$\nu_A : X \to [0, 1], \tag{3}$$

such that

$$0 \le \mu_A(x) + \nu_A(x) \le 1, \forall x \in X. \tag{4}$$

In the IFS theory, the hesitancy degree of x belonging to A is also defined by:

Table 1 Some similarity measures between IFSs.

Author	Expression
Dengfeng [10]	$SM1(A,B) = 1 - \frac{1}{\sqrt[p]{h}} \sqrt[p]{\sum_{i=1}^{h} \lvert\varphi_A(i) - \varphi_B(i)\rvert^p}$ where $\varphi_k(i) = (\mu_k(x_i) + 1 - \nu_k(x_i))/2, k = \{A, B\}$, and $p = 1, 2, 3, \ldots$
Mitchell [12]	$SM2(A,B) = \frac{1}{2}\left(\rho_\mu(A,B) + \rho_f(A,B)\right)$ where $\rho_\mu(A,B) = S_d^p(\mu_A(x_i), \mu_B(x_i))$ and $\rho_f(A,B) = S_d^p(1 - \nu_A(x_i), 1 - \nu_B(x_i))$
Ye [16]	$SM3(A,B) = \frac{1}{h}\sum_{i=1}^{h} \frac{\mu_A(x_i)\mu_B(x_i)+\nu_A(x_i)\nu_B(x_i)}{\sqrt{\mu_A^2(x_i)+\nu_A^2(x_i)}\sqrt{\mu_B^2(x_i)+\nu_B^2(x_i)}}$

$$\pi_A(x) = 1 - \mu_A(x) - \nu_A(x). \tag{5}$$

This degree expresses uncertainty whether x belongs to A or not.

A similarity measure S for $IFS(X)$ is a real function $S : IFS(X) \times IFS(X) \rightarrow [0, 1]$, which satisfies the following properties:

P1: $0 \leq S(A,B) \leq 1$,
P2: $S(A,B) = S(B,A), \forall A, B \in IFS(X)$,
P3: $S(A,B) = 1$ iff $A = B$,
P4: If $A \subseteq B \subseteq C$, then $S(A,C) \leq S(A,B)$ and $S(A,C) \leq S(B,C)$, for all A, B, and $C \in IFS(X)$.

Let $A = \{\langle x_i, \mu_A(x_i), \nu_A(x_i)\rangle | x_i \in X\}$ and $B = \{\langle x_i, \mu_B(x_i), \nu_B(x_i)\rangle | x_i \in X\}$ be in $IFS(X)$, Table 1 highlights some similarity measures between IFSs. SM1 and SM2 are distance-based measures, while SM3 is cosine-based measures.

5. IFS-Based Extraction Filtering

As we have seen, RAW probably produces false extractions. Hence, to improve the extraction accuracy, a method for removing unwanted extractions is necessary. This section describes our proposed method, to determine whether an extraction is correct or not. In the method, a classifier model for each IE rule r is constructed using the supervised learning approach. The rule r is applied to a training corpus, then we obtain the set of all extractions, denoted by E_r. An IFS characterizing each extraction, e_i in E_r is represented. If we have an extracted frame, e_t by r to be justified, an IFS corresponding to the frame is made. Like the concept of k nearest neighbor (k-NN), e_t is classified into the same group (either correct or incorrect) that is the most common among k nearest neighbors of its IFS representation.

5.1 Motivation for the Filtering Development

Using RAW, the rule r may be instantiated across a target-phrase boundary (e.g. the second frame in Fig. 6), which produces an incorrect extraction. Instantiations of the wildcards being between the first and the last slot makers of r, called the internal wildcards, provide a clue to detect such an undesirable extraction. Then, we have an assumption that the characteristics of the internal-wildcard instantiations producing the correct extractions from rule r should be more similar than those producing the incorrect ones.

Sentence similarity measures usually derive from symbolic, syntactic and structural information. Unlike European languages, there is limitation of linguistic tools for the Thai language. However, without facilitation of syntactic features, several works related with sentence similarity present acceptable results [5], [17]–[19].

In this work, we observe two main characteristics of the text portion into which an internal wildcard is instantiated: structural and symbolic information. The former type includes the length of tokens and the number of spaces. The later type includes words and class tags. The details of the two feature types will be explained more the next section. The precise steps of the proposed method are detailed as follows:

5.2 Preprocessing

5.2.1 Vector-Based Representation

(a1) The rule r is applied into all information entries in the training corpus, then semantic frames are obtained. The set of all extractions with respect to r is referred to as E_r.

(a2) When r matches with a text portion, we observe tokens[†], into which each internal wildcard is instantiated. (All wildcards except the first one are called an *internal wildcard*.)
After r is applied to the whole training corpus, two sets for each internal wildcard are constructed: one containing different words only when correct extractions are made; and the other containing those only when incorrect ones are made. For convenience, W_{cor}^s and W_{inc}^s are referred to the former set and the latter set, respectively, of the s-th internal wildcard.

(a3) Suppose the rule r contains n internal wildcards. A feature vector, namely $\vec{V}_i$, characterizing each extracted frame, e_i, in E_r is generated. The vector is defined as:

$$\vec{V}_i = \vec{v}_i^1 \parallel \vec{v}_i^2 \parallel \cdots \parallel \vec{v}_i^n,$$

where $\vec{v}_i^s$ is a 4-dimensional feature vector corresponding to the instantiation of the s-th internal wildcard in

[†] A token might be a word, a white space, or a semantic tag.

Table 3 An example of the proposed vector-based representation

Extraction	v_i^1				v_i^2				v_i^3				$\vec{V}_i$
	$f_{i,1}^1$	$f_{i,2}^1$	$f_{i,3}^1$	$f_{i,4}^1$	$f_{i,1}2$	$f_{i,2}^2$	$f_{i,3}^2$	$f_{i,4}^2$	$f_{i,1}^3$	$f_{i,2}^3$	$f_{i,3}^3$	$f_{i,4}^3$	
e_1	2	0	1	0	3	0	2	0	1	1	0	0	$[2, 0, 1, 0, 3, 0, 2, 0, 1, 1, 0, 0]$
e_2	4	0	0	3	1	0	0	0	1	1	0	0	$[4, 0, 0, 3, 1, 0, 0, 0, 1, 1, 0, 0]$
e_3	1	0	0	0	1	0	0	0	1	1	0	0	$[1, 0, 0, 0, 1, 0, 0, 0, 1, 1, 0, 0]$

Table 2 Instantiation of the internal wildcards of the rule in Fig. 4 into the information entry in Fig. 1.

Extraction	Text portion	Internal wildcard 1st	2nd	3rd
e_1	[21,30]	[22,23]	[25,27]	[29,29]
e_2	[33,42]	[34,37]	[39,39]	[40,40]
e_3	[34,43]	[37,37]	[39,39]	[40,40]

the rule pattern, and '||' refers to vector concatenation. The feature vector $\vec{v}_i^s$ is defined as:

$$\vec{v}_i^s = [f_{i,1}^s, f_{i,2}^s, f_{i,3}^s, f_{i,4}^s],$$

where $f_{i,1}^s$, $f_{i,2}^s$, $f_{i,3}^s$, and $f_{i,4}^s$ are the length of tokens, the number of spaces, the number of plain words or semantic tags in W_{cor}^k, and the number of plain words or semantic tags in W_{inc}^k observed from the text portion into which the internal wildcard is instantiated.

Example 1: This example illustrates the vector-based representation process. Suppose, in a training corpus, the rule shown in Fig. 4 can produce extractions only when it is applied to the information entry in Fig. 1. Then solely three extractions (cf. Figure 5 and 6) are made. Table 2 summarizes instantiation of the internal wildcards of the rule into the information entry in. To interpret, one can see, for example, that in the [33–42] portion, the first internal wildcard is instantiated into the [34–37] portion, including 3 plain words and 1 class tag ("Sym") and each on the second and third internal wildcards into an 1-token portion. To avoid the Thai writing in the text body, we use "w_i" referring to the i-th token in the information entry.

Observing the 1st internal wildcard instantiation from the three extraction, we know that

$$\begin{aligned} W_{cor}^1 &= \{w_{23}\}, \\ W_{inc}^1 &= \{w_{34}, w_{35}, \text{“}sym\text{”}\}, \\ W_{cor}^2 &= \{w_{26}, w_{27}\}, \\ W_{inc}^2 &= W_{cor}^3 = W_{inc}^3 = \emptyset. \end{aligned}$$

It is worthy to emphasis that, for tokens with semantic tags, we collect only their tags. For example, "sym" in W_{inc}^1 is the class tag of w_36. Following the (a3) step, we can construct a vector representation corresponding to each extraction as depicted in Table 3. ■

5.2.2 IFS-Based Document Representation

Recalling,

$$\vec{V}_i = \vec{v}_i^1 \parallel \vec{v}_i^2 \parallel \cdots \parallel \vec{v}_i^n,$$

a feature vector observed when the i-th frame is extracted. To convert $\vec{V}_i$ to an IFS, we propose one method which its conceptual idea is explained as follows.

Given the universe of discourse

$$X = \{x_1^1, x_2^1, x_3^1, x_4^1 \ldots, x_1^n, x_2^n, x_3^n, x_4^n\}.$$

It is noteworthy that the number of elements in X is equal to the dimension of $\vec{V}_i$, which is $4n$. We defined $A_i = \{\langle x_j^s, \mu_i(x_j^s), \nu_i(x_j^s)\rangle, \}$ is an IFS for the vector V_i, when j and s are indexes for feature types and internal wildcards, respectively. In this work, $\mu_i(x_j^s)$ presents a confidential level to say that $f_{i,j}^s$ in the feature vector of the i-th extraction is relatively high comparing to those values of the same feature type, j, and the same wildcard, s, in the other feature vectors. In contrast, $\nu_i(x_j^s)$ does a confidential level to say that $f_{i,j}^s$ in the i-th feature vector is not relatively high. The next example gives more details.

Example 2: Let consider the output the feature vectors from Example 1, i.e.

$$\begin{aligned} \vec{V}_1 &= [2, 0, 1, 0, 3, 0, 2, 0, 1, 1, 0, 0], \\ \vec{V}_2 &= [4, 0, 0, 3, 1, 0, 0, 0, 1, 1, 0, 0], \\ \vec{V}_3 &= [1, 0, 0, 0, 1, 0, 0, 0, 1, 1, 0, 0]. \end{aligned}$$

Since $f_{2,1}^1 > f_{1,1}^1 > f_{3,1}^1$, the confidential level to say that the first internal wildcard matches with a longer text portion for the second extraction than those for the rest extractions. Hence, $\mu_2(x_1^1) > \mu_1(x_1^1) > \mu_3(x_1^1)$ and $\nu_2(x_1^1) < \nu_1(x_1^1) < \nu_3(x_1^1)$. ■

Based on the idea discussed above, the process of transformation will be formally explained. Every value $f_{i,j}^s$ in the vector-based representation of the i-th extraction is then converted in terms of the three degrees of x_j^s as the following steps:

(b1) $f_{i,j}^s$ is normalized by:

$$z_{i,j}^s = \begin{cases} \frac{f_{i,j}^s - \overline{X}_j^k}{sd_j^s}, & sd_j^s \neq 0 \\ 0 & sd_j^s = 0 \end{cases}, \tag{6}$$

Table 4 An example of the proposed IFS-based representation from Example 3.

Information	Value
E	$\begin{bmatrix} 2 & 0 & 1 & 0 & 3 & 0 & 2 & 0 & 1 & 1 & 0 & 0 \\ 4 & 0 & 0 & 3 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \end{bmatrix}$
M	$\begin{bmatrix} 2.33 & 0.00 & 0.33 & 1.00 & 1.67 & 0.00 & 0.67 & 0.00 & 1.00 & 1.00 & 0.00 & 0.00 \end{bmatrix}$
SD	$\begin{bmatrix} 1.25 & 0.00 & 0.47 & 1.41 & 0.94 & 0.00 & 0.94 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \end{bmatrix}$
Z	$\begin{bmatrix} -0.27 & 0.00 & 1.41 & -0.71 & 1.41 & 0.00 & 1.41 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\ 1.34 & 0.00 & -0.71 & 1.41 & -0.71 & 0.00 & -0.71 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\ -1.07 & 0.00 & -0.71 & -0.71 & -0.71 & 0.00 & -0.71 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \end{bmatrix}$
D_μ	$\begin{bmatrix} 0.35 & 0.40 & 0.64 & 0.26 & 0.64 & 0.40 & 0.64 & 0.40 & 0.40 & 0.40 & 0.40 & 0.40 \\ 0.63 & 0.40 & 0.26 & 0.64 & 0.26 & 0.40 & 0.26 & 0.40 & 0.40 & 0.40 & 0.40 & 0.40 \\ 0.20 & 0.40 & 0.26 & 0.26 & 0.26 & 0.40 & 0.26 & 0.40 & 0.40 & 0.40 & 0.40 & 0.40 \end{bmatrix}$
D_ν	$\begin{bmatrix} 0.51 & 0.45 & 0.18 & 0.60 & 0.18 & 0.45 & 0.18 & 0.45 & 0.45 & 0.45 & 0.45 & 0.45 \\ 0.19 & 0.45 & 0.60 & 0.18 & 0.60 & 0.45 & 0.60 & 0.45 & 0.45 & 0.45 & 0.45 & 0.45 \\ 0.67 & 0.45 & 0.60 & 0.60 & 0.60 & 0.45 & 0.60 & 0.45 & 0.45 & 0.45 & 0.45 & 0.45 \end{bmatrix}$

where $\overline{X}_j^s$ and sd_j^s are the mean and the standard deviation, respectively, of the feature type j for the internal wildcard s over extractions. More precisely,

$$\overline{X}_j^s = \frac{\sum_{i=1}^{|E_r|} f_{i,j}^s}{|E_r|}, \tag{7}$$

and

$$sd_j^s = \left(\frac{\sum_{i=1}^{|E|} (f_{i,j}^s - \overline{X}_j^s)^2}{|E_r|} \right)^{1/2}. \tag{8}$$

(b2) Denoted by $\mu_i(x_j^s)$, a membership degree of x_j^s with respect to the extraction i and the wildcard s is determined by a weighted sigmoid function:

$$\mu_i(x_j^s) = r_j^s \frac{1}{1 + e^{-z_{i,j}^s}}, \tag{9}$$

where $0 < r_j^s \leq 1$ is a weight for x_j.

(b3) Denoted by $\nu_i(x_j^s)$, a non-membership degree of x_j^s with respect to the extraction i and the wildcard s is determined by a weighted sigmoid function:

$$\nu_i(x_j^s) = \bar{r}_j^s \frac{1}{1 + e^{z_{i,j}^s}}, \tag{10}$$

where $0 < \bar{r}_j^s \leq 1$ is a weight for x_j.

(b4) Denoted by $\pi_i(x_j^s)$, the hesitancy degree of the document i with respect to x_j^s is calculated by (5), i.e.,

$$\pi_i(x_j^s) = 1 - \mu_i(x_j^s) - \nu_i(x_j^s).$$

Example 3: This example illustrates how to convert a vector representation to an IFS representation using the steps (b1)-(b3). Consider three vectors, i.e., $\vec{V}_1, \vec{V}_2$, and $\vec{V}_3$ as shown in Example 2. For convenience, the vectors are represented in terms of the matrix E shown in Table 4. Next, we compute the mean and the standard deviation for each feature type of each internal wildcard, then the results are presented as the row matrices M and SD in the same table. More precisely, each entry of M and SD is obtained by columnwise computation of E, e.g. the first entry of M is the average of the first column of E. By the step (b1), we have the matrix Z. Suppose that the weights r_j^s and $\bar{r}_j^s$ are equal to 0.8 and 0.9, respectively. After applying (b2) and (b3), we have the membership and non-membership degrees which are represented as the two matrices D_μ and D_ν in the table. Finally, we can convert the feature vectors $\vec{V}_1$, $\vec{V}_2$, and $\vec{V}_3$ to IFSs by using D_μ, and D_ν. For instance, gathering the first row of the matrices, we can form an IFS, namely IFS_1 corresponding to $\vec{V}_1$:

$$\begin{aligned} IFS_1 = \{ & \langle x_1^1, 0.35, 0.51 \rangle \langle x_2^1, 0.40, 0.45 \rangle, \\ & \langle x_3^1, 0.64, 0.18 \rangle, \langle x_4^1, 0.26, 0.60 \rangle, \\ & \langle x_1^2, 0.64, 0.18 \rangle, \langle x_2^2, 0.40, 0.45 \rangle, \\ & \langle x_3^2, 0.64, 0.18 \rangle, \langle x_4^2, 0.40, 0.45 \rangle, \\ & \langle x_1^3, 0.40, 0.45 \rangle, \langle x_2^3, 0.40, 0.45 \rangle, \\ & \langle x_3^3, 0.40, 0.45 \rangle, \langle x_4^3, 0.40, 0.45 \rangle \} \end{aligned}$$

■

5.3 Extraction Classification

Recalling again that E_r is the set of all extractions—no matter whether each of them is correct or not—when apply the rule r into the training corpus, by the pre-process, we then have IFSs for those extractions. Let us refer them as IFS_1, $IFS_2, \ldots, IFS_m$, when m is the number of extractions in E_r.

To determine whether an extraction e_t made by the rule r is correct or not, it begins with representing e_t in terms of an IFS by the same values of parameters, i.e., means, standard deviations, and weights, used in the training process. The IFS representation of e_t here is referred to as IFS_t. Like the the concept of k-nearest neighbor classification, the extraction e_t is classified by assigning the label which is most frequent among the k IFSs corresponding to extractions in E_r nearest to IFS_t, where a distance is measured by an IFS

Table 5 Output templates and their meanings

Type	Output Template	Meaning
MD1	[Obs O][Attr A][Per T]	An abnormal characteristic A is found at an observed entity O for a time period of T.
MD2	[Sym S][Loc P][Per T]	A primitive named symptom S occurs at a human-body part P for a time period of T.
SR	[Ply P][Act A][Time N]	A player P takes a game action A in the Nth minute.
HA	[Area A][Bdr N][Rsr M]	A house of area size A has N bedrooms and M rest rooms.

similarity measure. Hereinafter, the parameter k is called the size of neighborhood.

6. Experiments and Discussion

6.1 Data Sets, Output Templates, and Training Process

6.1.1 Data Set Preparation

The proposed framework is evaluated in three different domains of Thai text: *medical-symptom descriptions (MD)*, *soccer match reports (SR)*, and *housing advertisements (HA)*. To prepare a data set for each domain, we begin with collecting information from web sites related to the domain. For the MD domain, the data set is obtained from pieces of disease information provided in the project 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 [20]. An information entry in the SR data set is a news-story-style unstructured text reporting a soccer match in details. An information entry in HA is a house-selling announcement collected from on-line classified advertisement sites. As results, 115, 86, and 189 information entries with the average length of 45.0, 68.6, and 64.3 words per entry in the MD, SR, and HA domains, respectively, are used in our exploratory evaluation.

Next, the collected information entries are preprocessed using a word segmentation program, called CTTEX developed by the National Electronics and Computer Technology Center, and are then partially annotated with semantic class tags using predefined ontology lexicons. Class tags for MD including, for example, “Symptom,” “Organ,” “Hormone,” are taken from entity types collected as part of the project [20]. A lexicon containing soccer player names and soccer team names, collected as part of a project on developing an alias extraction system [21], is used for semantic annotation in the domain SR, while a lexicon containing city names is used for HA. Moreover, regular expression based semi-automatic annotation is applied for tagging quantity information, e.g. “Period of Time,” “Minute,” and “Price.”

6.1.2 Output Templates

Four types of target phrases are considered in our experiments: two of them are from the MD domain, referred to as *Type-MD1* and *Type-MD2*; one from the SR domain, referred to as *Type-SR*; and the other one from the HA domain, referred to as *Type-HA*. Table 5 gives the output-template forms for the four types along with their intended meanings. The slot Per in the Type-MD1 template as well as the slot Time in the Type-SR template is optional. One of the slots Loc and Per, but not both, may be omitted in the Type-MD2 template. One arbitrary slot in the Type-HA template may be omitted. The first underlined phrase in Fig. 1 (also in Fig. 2) is an example of a text portion conforming to Type-MD1, while the second and third underlined phrases in the same figure are text portions conforming to Type-MD2.

6.1.3 Rule Learning

For each template type, extraction rules are created using WHISK by repeatedly performing the following steps until addition of the 10 most recently obtained training phrases causes no creation of any new rule:

1. Randomly select an information entry in its respective data domain.
2. Manually tag all target phrases of the template type in the selected information entry with desired output frames.
3. Add the obtained hand-tagged target phrases as new training instances in the rule learning process of WHISK.

All remaining information entries are then used as test data. Table 6 shows the number of all distinct target phrases in the training data set and the test data set accordingly obtained for each template type, and characterizes the data sets in terms of target-phrase length (in words). The table indicates, for example, that while target phrases of Type-HA are typically longer than those of other types, they occur less frequently than target phrases of Type-MD1 and Type-SR, and also that the proportion of words contained in target phrases of Type-HA to all words in their respective entire data sets is close to the same proportion of those in target phrases of Type-SR. Using our implementation of WHISK, IE rules are automatically generated. Table 7 summarizes information of the rule set for each template about the numbers of generated IE rules and the numbers of internal wildcards. For the MD1 template, as an example, there are 15 rules in which the maximum, average, minimum numbers

Table 6 Data set characteristics for each template type

Type	Data set	No. of distinct target phrases	Target-phrase length			No. of target phrases per entry		
			Max.	Avg.	Min.	Max.	Avg.	Min.
MD1	Training	90	11	3.5	2	7	3.6	1
MD1	Test	136	8	3.3	2	11	2.9	1
MD2	Training	80	15	4.1	2	3	1.4	0
MD2	Test	66	8	3.9	2	5	1.2	0
SR	Training	93	28	8.0	3	6	2.0	2
SR	Test	156	21	6.4	3	6	3.9	2
HA	Training	87	37	16.1	7	2	1.2	1
HA	Test	113	34	15.2	7	2	1.3	1

Table 8 Evaluation results using the base window size (1W)

Method	k	Template Type							
		MD1		MD2		HA		SR	
		R	P	R	P	R	P	R	P
RAW	-	76.33	83.60	100.00	39.12	81.50	55.29	82.93	40.57
RAW+SM1	1	75.85	97.52	98.93	92.50	79.77	92.62	81.46	85.20
	3	76.33	97.53	100.00	94.92	80.35	92.67	81.46	86.53
	5	76.33	95.76	99.47	93.47	80.92	92.11	80.98	86.01
RAW+SM2	1	75.85	97.52	98.93	92.50	79.77	92.62	80.49	84.18
	3	76.33	97.53	99.47	95.38	80.35	92.67	81.46	86.53
	5	76.33	95.76	99.47	93.47	79.77	91.39	80.98	85.57
RAW+SM3	1	75.85	98.13	98.93	93.43	79.77	92.62	81.95	85.71
	3	76.33	98.14	100.00	95.90	80.35	93.92	81.95	87.50
	5	76.33	96.34	100.00	94.44	80.92	93.33	81.95	86.60

Table 7 IE-rule characteristics for each template type

Type	No. of rules	No. of internal wildcards		
		Max.	Avg.	Min.
MD1	15	3	1.3	1
MD2	11	3	1.9	1
SR	8	3	2	1
HA	9	6	4.7	3

of internal wildcards are 3, 1.3, and 1, respectively.

6.2 Parameter Setting

The parameters in the proposed method including the weights r^s_j, $\overline{r}^s_j$, and the size of neighborhood k are determine as follows:

- The weights r^s_j and $\overline{r}^s_j$ are based on statistical characteristics of feature type by
$$r^s_j = \overline{r}^s_j = \frac{|1 - sd^s_j|}{|1 + sd^s_j|}.$$
- The neighborhood size k, in this experiment, is varied as 1, 3, and 5.

6.3 Experimental Results

The proposed framework is evaluated using the four test data sets for their respective template types (cf. Table 6). Recall and precision are used as performance measures, where the former is the proportion of correct extractions to relevant target phrases and the latter is the proportion of correct extractions to all obtained extractions. 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, denoted by 1W. From our experiment, the 1W values of rules are between 2 to 11 (3.5 on average) for MD1, between 2 to 15 (4.1 on average) for MD2, between 3 to 28 (8.0 on average) for SR, and between 7 to 37 (16.1 on average) for HA. During experiments, we also evaluated with the extension of the size for each rule by doubling (2W), tripling (3W) so on; but, we noticed that the recall of 3W is equal to that of 2W. Then, only the results from 1W and 2W are reported. Tables 8 and 9 shows the evaluation results obtained from 1W and 2W, respectively, where 'R' and 'P' stand for recall and precision, which are given in percentage.

Compared to the results obtained using RAW alone, filtering by each of the three similar measures improves precision while satisfactorily preserving the recall value of RAW in every experiment. In particular, for Type-MD2, Type-SR, and Type-HA, where RAW has low precision, filtering by each of the three similar measures yields significant precision gains. For example, considering the evaluation results of Type-MD2 in Table 8, RAW produced 100% of recall but only 39.12% of precision; however, when the filtering technique with SM1 and $k = 3$ was applied, the precision was increased up to 94.92% and the recall was preserved.

To analyze the performance of the three similarity measures, Figs. 7 and 8 show the average recall and precision, respectively, over the neighborhood sizes with 1W. Figures 9

Table 9 Evaluation results using the double base window size (2W)

Method	k	Template Type							
		MD1		MD2		HA		SR	
		R	P	R	P	R	P	R	P
RAW	-	87.44	61.36	100.00	34.00	95.95	59.71	92.68	35.32
RAW+SM1	1	84.54	97.22	96.26	93.75	92.49	91.43	90.24	88.10
	3	86.47	94.71	98.40	94.85	94.22	94.77	92.20	85.52
	5	86.96	93.75	99.47	90.73	93.64	94.19	92.20	84.38
RAW+SM2	1	84.54	97.22	96.26	93.75	92.49	91.43	90.24	88.52
	3	86.47	94.71	98.40	94.85	94.22	94.77	92.20	85.52
	5	86.96	93.75	98.93	90.69	93.64	94.19	92.20	85.14
RAW+SM3	1	85.51	95.68	98.93	94.87	94.22	97.02	91.22	88.63
	3	86.96	95.74	100.00	95.41	94.80	96.47	91.71	89.10
	5	87.44	93.78	100.00	94.92	94.80	95.35	92.20	86.70

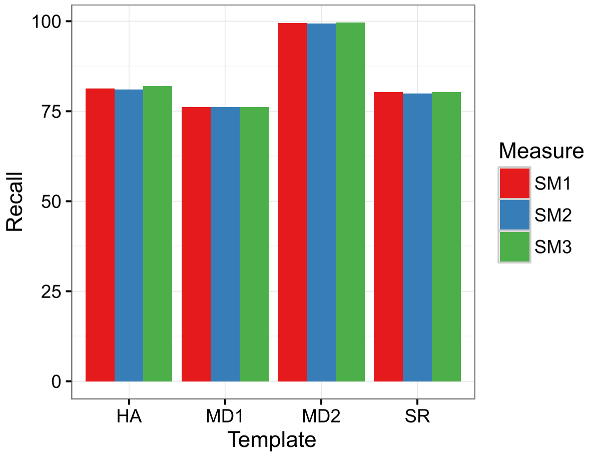

Fig. 7 Recall comparison of similarity measures using 1W.

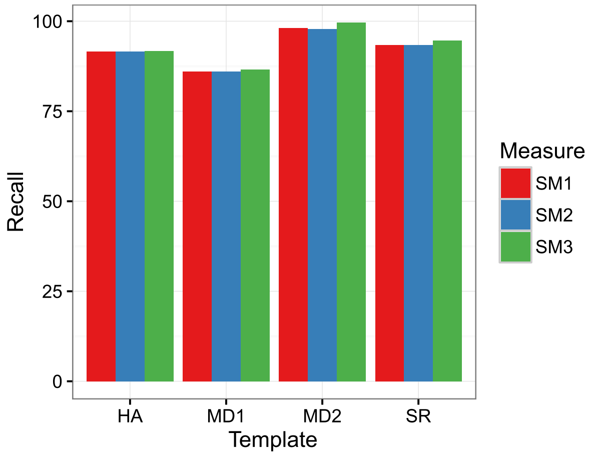

Fig. 9 Recall comparison of similarity measures using 2W.

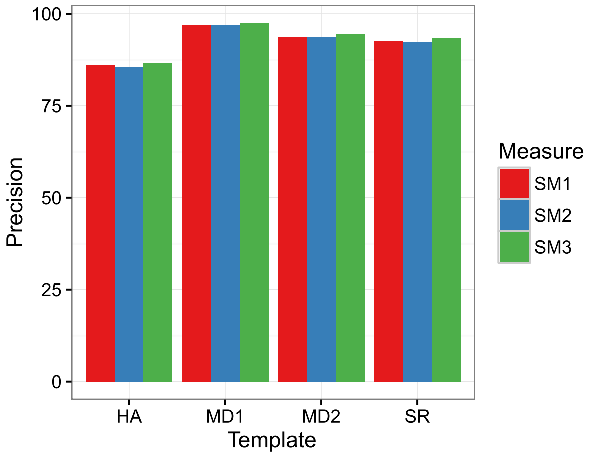

Fig. 8 Precision comparison of similarity measures using 1W.

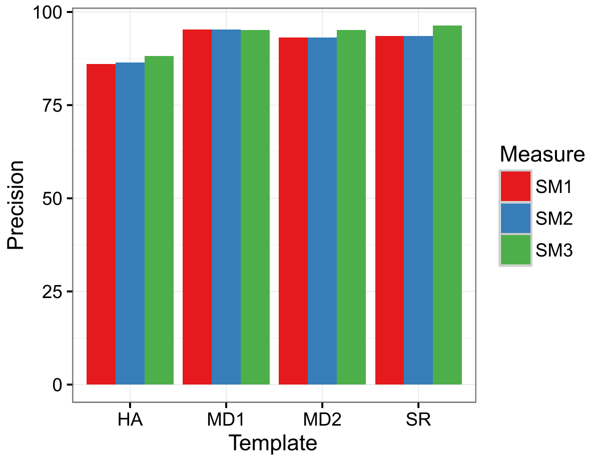

Fig. 10 Precision comparison of similarity measures using 2W.

and 10 show those with 2W. One can see that none of the three measures performs obviously better than another one. SM1 and SM2 show the same accuracy, while SM3 produces slightly better performance than the others.

To compare the results of different neighborhood sizes, Figs. 11 and 12 depict the average recall and precision over similarity measures when 1W was used. Likewise, Figs. 13 and 14 depict those when 2W was applied. We then see that the overall profile of $k = 3$ is a little higher that of the others.

6.3.1 Comparison with Extraction with Known Boundaries

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

Table 10 Comparison with rule application to manually identified target phrases

Method	Template Type							
	MD1		MD2		HA		SR	
	R	P	R	P	R	P	R	P
Known Boundary	88.41	97.86	100.00	100.00	97.69	97.69	95.12	86.67
RAW+SM1	85.99	95.23	98.04	93.11	93.45	93.46	91.54	86.00
RAW+SM2	85.99	95.23	97.86	93.09	93.45	93.46	91.54	86.39
RAW+SM3	86.63	95.07	99.64	95.07	94.61	96.28	91.71	88.14

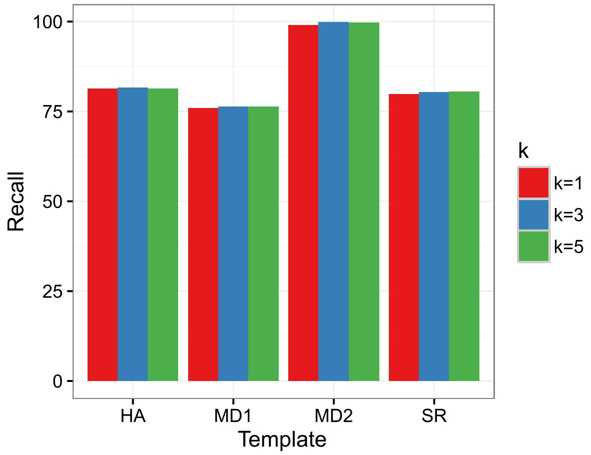

Fig. 11 Recall comparison of neighborhood sizes using 1W.

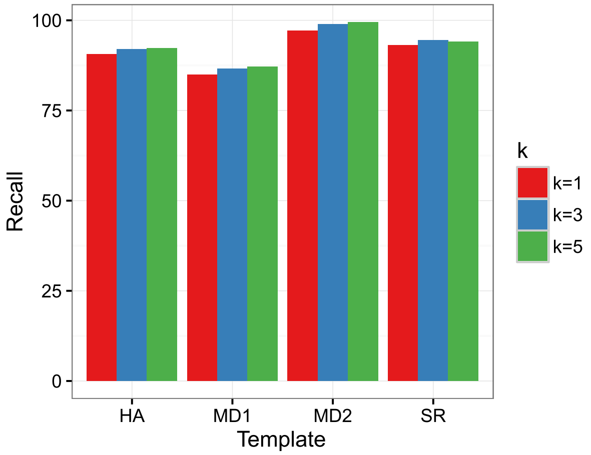

Fig. 13 Recall comparison of neighborhood sizes using 2W.

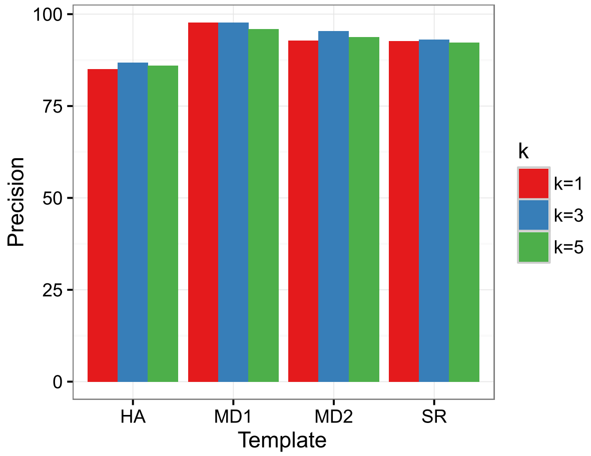

Fig. 12 Precision comparison of neighborhood sizes using 1W.

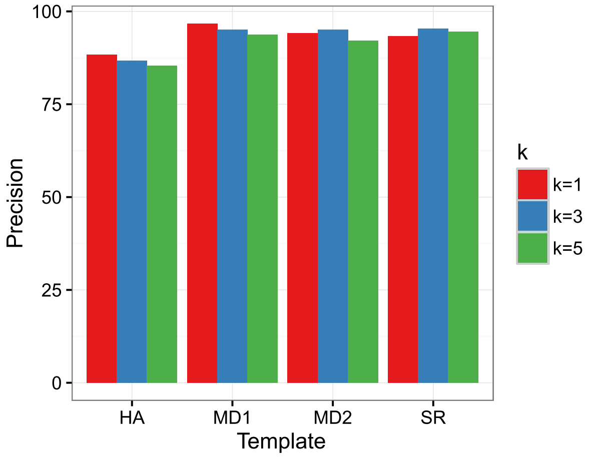

Fig. 14 Precision comparison of neighborhood sizes using 2W.

sets and apply the rules obtained from WHISK directly to these manually identified text portions. Table 10 compares the evaluation results obtained from such direct rule application to the average results over *k* of our framework using 2W. In the MD domain, the performance obtained from the proposed method is close to that of known-boundary extraction. However, in the SR and HA domains, where target phrases are longer, the recalls of our method are relatively lower than those of the baseline, while the precisions of our method and the baseline are comparable. It is noteworthy that although the basic idea behind WIF is to detect rule application across a target-phrase boundary, wildcard-instantiation-based filtering may also improve the precision of a rule for known-boundary extraction itself, for example, as seen in the last row of Table 10, RAW+SM3 improves the precision of known-boundary extraction from 86.67 to to 88.14 for Type-SR.

6.3.2 Comparison with Extraction with Other Filtering Techniques

The proposed framework is also compared with the other baseline framework in which filtering techniques are used. In the IE framework for Thai text [5], two extraction filtering modules, called wildcard-instantiation filtering (WIF) and overlapping-frame filtering (OFF), are proposed for remov-

Table 11 Comparison with other filtering techniques

Method	Template Type							
	MD1		MD2		HA		SR	
	R	P	R	P	R	P	R	P
Baseline(SVM)	**86.96**	94.74	97.86	**95.31**	93.64	92.57	90.73	85.32
Baseline (kNN)	85.99	93.68	96.26	94.24	91.33	90.80	90.24	83.71
Baseline (NB)	85.51	94.15	95.72	93.72	91.33	91.33	89.27	83.18
Baseline (DT)	**86.96**	94.24	97.86	**95.31**	93.64	93.64	91.22	85.78
RAW+SM1	85.99	**95.23**	98.04	93.11	93.45	93.46	91.54	86.00
RAW+SM2	85.99	**95.23**	97.86	93.09	93.45	93.46	91.54	86.39
RAW+SM3	86.63	95.07	**99.64**	95.07	**94.61**	**96.28**	**91.71**	**88.14**

ing incorrect extractions. The first module uses a binary classifier for prediction of rule application across a target-phrase boundary; the second one uses weighted classification confidence to resolve conflicts arising from overlapping extractions. More precisely, in the first module, four standard models are used, i.e., Support Vector Machine (SVM) based on the RBF kernel, k-Nearest Neighbor (kNN), Naive Bayes (NB), and Decision Tree (DT) using C4.5. The four models are constructed from the vectors corresponding to the extractions from the training corpus (cf. Sect. 5.2.1). The second module, i.e. OFF, is derived from the fact that one target phrase is independent of another target 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.

In an NB model, a confidence value is the predicted-class conditional probability for the feature vector of an instance being classified. For a DT model, prediction confidence is normally calculated from class distribution of leaf nodes into which an instance is classified. For kNN and SVM models, prediction confidence is obtained from calibrating a classifier score, i.e., transforming a classifier score into a class membership probability.

Table 11 compares the proposed framework[†] with the baseline when SVM, kNN, NB, and DT classifiers are used and the 2W is made. The results reveal that the IFS-based framework performs better than the baseline, especially for the HA and SR templates whose the target phrase lengths and the dimensions of feature vectors[††] are relatively higher than those of MD1 and MD2. In both medical templates, it is imprecise to decide which framework outperforms the other owning to the trade-off between recall and precision. For example, for MD1, the baseline using SVM produces higher recall, but the proposed framework does higher precision.

7. Conclusions and Future Works

From a set of manually collected target phrases, IE rules are created using WHISK. To apply the obtained rules to unstructured text without predetermining target-phrase boundaries, rule application using sliding windows is introduced. It tends to produce many unwanted extractions, especially when IE-rules are applied across target-phase boundaries. An IFS-based filtering technique is proposed for removal of those false extractions. The experimental results show that the technique improves extraction precision while satisfactorily preserving recall. Further works include extension of the types of target phrases and empirical investigation of framework application in different data domains as well as different similarity measures.

Acknowledgments

This work has been supported by the Thailand Research Fund (TRF), under Grant No. MRG5980067 and partially supported by TRF, under Grant No. RTA6080013.

References

[1] S. Soderland, "Learning Information Extraction Rules for Semi-Structured and Free Text," Machine Learning, vol.34, no.1–3, pp.233–272, 1999.

[2] Q.L. Nguyen, D. Tikk, and U. Leser, "Simple tricks for improving pattern-based information extraction from the biomedical literature," Journal of Biomedical Semantics, vol.1, no.1, pp.1–17, 2010.

[3] Q. Liua, Z. Gaoa, B. Liuc, and Y. Zhang, "Automated rule selection for opinion target extraction," Knowl-Based. Syst., vol.104, pp.74–88, 2016.

[4] E. Kim, Y. Song, C. Lee, K. Kim, G.G. Lee, B.-K. Yi, and J. Cha, "Two-phase learning for biological event extraction and verification," ACM Transactions on Asian Language Information Processing, vol.5, no.1, pp.61–73, 2006.

[5] P. Intarapaiboon, E. Nantajeewarawat, and T. Theeramunkong, "Extracting Semantic Frames from Thai Medical-Symptom Unstructured Text with Unknown Target-Phrase Boundaries," IEICE Trans. Inf. & Syst., vol.E94.D, no.3, pp.465–478, 2011.

[6] I. Spasić, F. Sarafraz, J.A. Keane, and G. Nenadić, "Medication information extraction with linguistic pattern matching and semantic rules," Journal of the American Medical Informatics Association, vol.17, no.5, pp.532–535, 2010.

[7] J. Zhang and N.M. El-Gohary, "Semantic NLP-Based Information Extraction from Construction Regulatory: Documents for Automated Compliance Checking," J. Comput. Civil Eng., vol.30, no.2, pp.1–14, 2016.

[8] K.T. Atanassov, "Intuitionistic Fuzzy Sets," Fuzzy. Set. Syst., vol.20, no.1, pp.87–96, 1986.

[9] L.A. Zadeh, "Fuzzy Sets," Inform. Comput., vol.8, no.3, pp.338–353, 1965.

[†] The average performance over the *k* values is shown and it is similar to that in Table 10.

[††] The dimension of feature vectors for each IE rule is 4 times higher than the number of the internal wildcard of the rule.

[10] L. Dengfeng and C. Chuntian, "New Similarity Measures of Intuitionistic Fuzzy Sets and Application to Pattern Recognition," Pattern. Recogn. Lett., vol.23, no.1-3, pp.221–225, 2002.

[11] Z. Liang and P. Shi, "Similarity Measures on Intuitionistic Fuzzy Sets," Pattern. Recogn. Lett., vol.24, no.15, pp.2687–2693, 2003.

[12] H.B. Mitchell, "On the Dengfeng-Chuntian Similarity Measure and Its Application to Pattern Recognition," Pattern. Recogn. Lett., vol.24, no.16, pp.3101–3104, 2003.

[13] W.-L. Hung and M.-S. Yang, "Similarity Measures of Intuitionistic Fuzzy Sets Based on Hausdorff Distance," Pattern. Recogn. Lett., vol.25, no.14, pp.1603–1611, 2004.

[14] Z. Xu, "Some Similarity Measures of Intuitionistic Fuzzy Sets and Their Applications to Multiple Attribute Decision Making," Fuzzy Optimization and Decision Making, vol.6, no.2, pp.109–121, 2007.

[15] V. Khatibi and G.A. Montazer, "Intuitionistic Fuzzy Set VS. Fuzzy Set Application in Medical Pattern Recognition," Artif. Intell. Med., vol.47, no.1, pp.43–52, 2009.

[16] J. Ye, "Cosine Similarity Measures for Intuitionistic Fuzzy Sets and Their Applications," Math. Comput. Model., vol.53, no.1-2, pp.91–97, 2011.

[17] J. Zhang, Y. Sun, H. Wang, and Y. He, "Calculating Statistical Similarity between Sentences," Journal of Convergence Information Technology, vol.6, no.2, pp.22–34, 2011.

[18] T. Kenter and M. de Rijke, "Short Text Similarity with Word Embeddings," Proc. 24th ACM International on Conference on Information and Knowledge Management (CIKM '15), Melbourne, Australia, pp.1411–1420, 2015.

[19] W. Ma and T. Suel, "Structural Sentence Similarity Estimation for Short Text," In Proceedings of the 29th International Florida Artificial Intelligence Research Society Conference (FLAIRS 2016), Florida, USA, pp.232–237, 2016.

[20] T. Theeramunkong, P. Iamtana-anan, C. Nattee, A. Suriyawongkul, E. Nantajeewarawat, and P. Aimmanee, "A Framework for Constructing a Thai Medical Knowledge Base," In Proceedings of the 2nd International Conference on Knowledge, Information and Creativity Support Systems, Ishikawa, Japan, pp.45–50, 2007.

[21] T. Suwanapong, T. Theeramunkong, and E. Nantageewarawat, "The vector space models for finding co-occurrence names as aliases in thai sports news," In Proceedings of the 2nd Asian Conference on Intelligent Information and Database Systems, Lecture Notes in Computer Science, vol.5990, pp.122–130, Hue City, Vietnam, 2010.

Peerasak Intarapaiboon received his Ph.D. in Information Technology from the School of Information, Communication, and Computer Technology, Sirindhorn International Institute of Technology, Thammasat University. His research interests include knowledge representation, fuzzy sets, information extraction, automated reasoning, and machine learning.

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.