Final Report

Thai Sentence Similarity Measure Based on Intuitionistic Fuzzy Sets with Application on Information Extraction

Peerasak Intarapaiboon

October/ 2018

Contract No.MRG5980067

Final Report

Thai Sentence Similarity Measure Based on Intuitionistic Fuzzy Sets
with Application on Information Extraction

Peerasak Intarapaiboon
Faculty of Science and Technology, Thammasat University

Project Granted by the Thailand Research Fund

Abstract

Project Code: MRG5980067

Project Title: Thai Sentence Similarity Measure Based on Intuitionistic Fuzzy Sets with Application on Information Extraction

Investigator: Asst. Prof. Peerasak Intarapaiboon, Ph.D.

E-mail Address: ipeeasa@tu.ac.th

Project Period : 2 years

Abstract: 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 project 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 of not. From the experiment on various domains, the proposed technique improves extraction precision while satisfactorily preserving recall.

Keywords: Information extraction, natural language processing, fuzzy sets, similarity measures.

Contents

Chapter Title **Page**

List of Figures

Figure **Page**

List of Tables

Chapter 1

Executive summary

1.1 Motivation

Standard knowledge representation languages for the Semantic Web, such as RDF [46] and OWL [75], have been recently developed and are now in place. They have been evolved from traditional markup languages in order to represent the meaning of data, i.e., *metadata*, in a machine-understandable form. With the emergence of such languages, *keyword-based information retrieval* in the Syntactic Web era is being replaced with a more powerful search, namely *semantics-based information retrieval*.

To illustrate, suppose that one wants to retrieve Web pages concerning "a chemical reaction that produces carbon dioxide from ethanol." Using a keyword-based query, the user's information need is represented as a set of keywords, such as "carbon dioxide" and "ethanol", and web pages are recognized as relevant if they contain such specified keywords. Accordingly, a Web page containing the text portion

"*During combustion ethanol reacts with oxygen and produces carbon dioxide*" (1.1)

and that containing the text portion

"*A standard biochemical pathway is followed, from which yeast digest glucose and release ethanol and carbon dioxide*" (1.2)

are both retrieved. The second one is, however, not in the range of user interest, since it refers to ethanol as a product, rather than a reactant. By contrast, based on Description Logic (DL), which is a logical formalism underlying OWL, the above query is represented by the concept expression

$$Q:\ \mathsf{Reaction} \sqcap \exists \textsc{hasPdt}.\mathsf{CarbonDioxide} \sqcap \exists \textsc{hasRct}.\mathsf{Ethanol},$$

and Statements (1.1) and (1.2) are partially represented, respectively, by the concept expressions

$$C1:\ \mathsf{Reaction} \sqcap \exists \textsc{hasRct}.\mathsf{Ethanol} \sqcap \exists \textsc{hasRct}.\mathsf{Oxygen} \sqcap \exists \textsc{hasPdt}.\mathsf{CarbonDioxide},$$

$$C2:\ \mathsf{Reaction} \sqcap \exists \textsc{hasRct}.\mathsf{Glucose} \sqcap \exists \textsc{hasPdt}.\mathsf{Ethanol} \sqcap \exists \textsc{hasPdt}.\mathsf{CarbonDioxide}.$$

REACTANT:	*ethanol*
REACTANT:	*oxygen*
PRODUCT:	*carbon dioxide*

(a)

REACTANT:	*glucose*
PRODUCT:	*ethanol*
PRODUCT:	*carbon dioxide*

(b)

Figure 1.1 Examples of multi-slot frames.

Using $C1$ as metadata describing the first Web page and $C2$ as that describing the second one, subsumption checking in DL can then be employed as a document retrieval mechanism. Since Q subsumes $C1$, but not $C2$, only the first Web page is retrieved.

Moreover, domain experts can make use of such Semantic Web languages to represent their background knowledge, enabling deduction of implicit information that can be useful for semantics-based retrieval. Suppose, for example, that one wants to find Web pages concerning "a chemical reaction that produces carbon dioxide from an alcohol." A semantics-based search engine would retrieve a Web page containing Statement (1.1), although it does not contains the term "alcohol" explicitly, provided that the assertion "ethanol is an alcohol" can be derived from the background knowledge.

To realize the above vision of information retrieval, a crucial question still remains: how will document metadata be automatically or semi-automatically created? It is anticipated that *information extraction (IE)* technologies will contribute significantly to realization of metadata creation. IE is a process of identifying and extracting desired pieces of information. Based on output representation (target structure), IE frameworks are divided into two categories: *single-slot extraction* and *multi-slot extraction*. The former category focuses on extracting individual pieces of information of a certain specified type, while the latter one on extracting related pieces of information and connecting them in a form of multiple-field relational records.

From the text portions in Statements (1.1) and (1.2), for example, the frames describing chemical reactions in Fig. 1.1 can be extracted using IE techniques. From these frames, the concept expressions $C1$ and $C2$ above can be directly constructed. In order to relate reactants with products participating in the same chemical reaction, multi-slot extraction appears to be more suitable than single-slot extraction, in particular when input text contains more than one target chemical reaction description.[1]

A well-known supervised rule learning algorithm, called WHISK [85], is wildly used for multi-slot extraction from structured to free text. The algorithm uses a covering learning algorithm to generate regular expression patterns. However, Pattern-based IE rules do not have ability to automatically segment input documents so that they can be applied only to

[1]When a textual document contains multiple chemical reaction descriptions, single-slot IE extracts individual reactants and individual products separately, without relating reactants and products involving the same reaction.

relevant text portions. When applied to free text, a rule is usually applied to each individual sentence one by one. Identifying the boundary of a Thai sentence is, however, problematic. In Thai, there is no explicit end-sentence punctuation [18].

1.2 Problem Statement

In this project, we aim to introduce methods to calculate a relevant score when a typical IE rule is applicable to a text portion. The extractions with low scores will be eliminated. In these calculating methods, similarity or distance measures between sentences play an important role. Although there are several algorithms for determining a degree of similarity in textual level and a sentence can be considered as a (short) document, it is difficult to adopt those algorithms for our problem, i.e., sentential level. The main reason is that such algorithms for the textual level are designed to deal with long documents rather than short ones. Hence, a representation form for capturing hidden semantics of sentences and similarity measures on the representation form are the main challenge of the research.

Recently, intuitionistic fuzzy set (IFS) [3] has been much explored in both theory and application. Differing from representation of a fuzzy set (FS) [107], 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 paid their focus. Many IFS-based frameworks for solving various problems yield satisfactory results. For example, [Khatibi and G. 2009] conducted experiments for bacterial classification using three similarity measures: one for fuzzy sets, and two for IFSs. The results evidenced that the both measures for IFSs outperformed the other one for fuzzy sets. In [39], as another example, an IFS-based classification framework was proposed and the framework accuracy outperformed some traditional classification methods. By the success of research in IFS, especially in similarity measurement, it is anticipated that IFS technologies will contribute to this project.

1.3 Results

An IFS-based filtering technique is proposed for removal of those false extractions. The experimental results on documents related with various domains such as medical, news, and chemical, show that the technique improves extraction precision while satisfactorily preserving recall. When comparing to other classification models, our proposed filtering method produce relatively better results.

Chapter 2

Background

In general, IE is aimed at extracting relevant information from a huge amount of data, which could be textual documents, images, or even signals. In this dissertation, IE refers to information extraction from text. Diverse types of input and output are considered in IE systems. They are characterized in Section 2.1. Section 2.2 describes processes usually involved in a generic framework for event extraction from unstructured text. Section 2.3 characterizes the Thai language and reviews current works concerning natural language processing (NLP) and information extraction in Thai.

Intuitionistic fuzzy sets and their similarity measures that are used in the proposed framework (Chapter 3) are explained in Section 2.4.

2.1 Input and Output Types for Information Extraction

Input documents for an IE system can be classified into two text genres, i.e., structured text and unstructured text (or free text). Structured documents consist of information entries that are organized in a rigid format, such as bibliographies, telephone dictionaries, and automatically-created Web pages. Unstructured-text documents are written in natural languages, such as news stories and scientific reports. Fig. 2.1 and Fig. 2.2 provide examples of structured text and unstructured text, respectively.

There are various forms of IE output, depending upon target IE tasks. As proposed by two primary programs, i.e., Message Understanding Conference (MUC)[1] and Automatic Content Extraction (ACE), IE tasks can be decomposed into several subtasks. MUC-6 [29], for example, separates an IE task into name entity recognition, coreference resolution, template element recognition, relation extraction task, and scenario template task.

[1]MUCs were initiated and funded by DARPA (Defense Advanced Research Projects Agency) to encourage the development of new and better methods of information extraction. Running from 1987 through 1997, these competition-based conferences provided challenges for IE researchers in different domains such as naval operations messages (MUCK-I and MUCK-II), terrorism (MUC-3 and MUC-4), and business (MUC-5, MUC-6, and MUC-7), and also in different output formats.

```
@PHDTHESIS{Califf:98,
author = {M. E. Califf},
title = {Relational Learning Techniques for Natural Language Extraction.},
school = {Computer and Information Science, University of Texas at Austin},
year = {1998},
}
@CONFERENCE{Chieu:02,
author = {H. L. Chieu, and H. T. Ng},
title = {A Maximum Entropy Approach to Information Extraction from
        Semi-Structured and Free Text},
booktitle = {Proceedings of the 8th National Conference on Artificial Intelligence},
year = {2002},
pages = {786–791},
address = {Alberta, Canada},
}
@BOOK{Feldman:06,
title = {WordNet: An Electronic Lexical Database},
publisher = { MIT Press},
year = {1998},
author = {C. Fellbaum},
}
@ARTICLE{Lavelli:08,
author = {A. Lavelli, and M. E. Califf, and F. Ciravegna, and D. Freitag, and
          C. Giuliano, and N. Kushmerick, and N. Ireson},
title = {Evaluation of Machine Learning-Based Information Extraction Algorithms:
        Criticisms and Recommendations},
journal = {Language Resources and Evaluation},
year = {2008},
volume = {42},
pages = {361–393},
}
@ARTICLE{Soderland:99,
author = {S. Soderland},
title = {Learning Information Extraction Rules for Semi-Structured and Free Text},
journal = {Machine Learning},
year = {1999},
volume = {34},
pages = {233–272},
number = {1–3},
}
```

Figure 2.1 An example of structured text.

The ACE program, a follow-up to the MUC program, aiming to develop core technologies for extracting information from multimedia sources in different languages [22], classifies IE subtasks as follows:

- **Entity detection and tracking (EDT)** is aimed to identify all mentions of entities, whether a name or a pronoun, and to group them based on objects to which they refer. For instance, given a sentence "***Nicola Hanania****, a researcher at the asthma clinical research center at Baylor College of Medicine, says* ***he*** *is very interested in the technology,*" it is expected that an EDT system identifies the mention "***he***" with the mention "***Nicola Hanania***." The current EDT task in the ACE program identifies seven types of entities, i.e., Person, Organization, Location, Facility, Weapon, Vehicle, and Geo-Political Entity. Each type is moreover divided into subtypes, for instance, Organization subtypes are Government, Commercial, Educational, Non-profit, etc.

- **Relation detection and characterization (RDC)** involves predicting whether a relation exists between a pair of entities and assigning to it one of predefined types.

McCann has initiated a new so-called global collaborative system, composed of world-wide account directors paired with creative partners. In addition, Peter Kim was hired from WPP Group's J. Walter Thompson last September as vice chairman, chief strategy officer, world-wide.

Figure 2.2 An example of unstructured text.

TYPE:	Create
OBJECT:	*The Eiffel Tower*
TIME:	*1889*
PLACE:	*Paris*

Figure 2.3 An example of an event template.

There are five general types of relations, i.e., Role, At, Part, Near, and Social. Each type is also divided into more specific subtypes; for instance, Social subtypes include Parent, Sibling, Spouse, etc. If the LOCATEDAT relation is of interest, for example, then the information to be extracted from the sentence "*Established on April 1, 1976 at California, Apple Inc. is an American multinational corporation that designs and markets consumer electronics, computer software, and personal computers*" is LOCATEDAT(Apple Inc., California).

- **Event detection and characterization (EDC)** discovers and characterizes types of events in which EDT entities participate. General event types include Destroy, Create, Transfer, Move, and Interact. Fig. 2.3 exemplifies an output template of the Create type generated from the sentence "*In 1889, the Eiffel Tower was built as the centrepiece of a giant fair in Paris.*" The template consists of four slots, namely "TYPE," "OBJECT," "TIME," and "PLACE," and their slot fillers are "Create," "*The Eiffel Tower,*" "*1889,*" and "*Paris,*" respectively.

- **Temporal expression recognition and normalization (TERN)** detects temporal expressions in text and normalizes them into ISO formats. This task is straightforward when absolute temporal expressions (e.g, *October 22, 1981*) are found. However, temporal expressions appearing in a natural language may be vague; they include indexical expressions (e.g., *yesterday*, *next week*, *Monday*) and relative expressions (e.g., *three minutes after the President arriving*). Apart from detection and normalization, TERN systems should have ability to associate absolute meanings with those vague expressions.

This dissertation focuses mainly on EDC, which will be detailed in the next section.

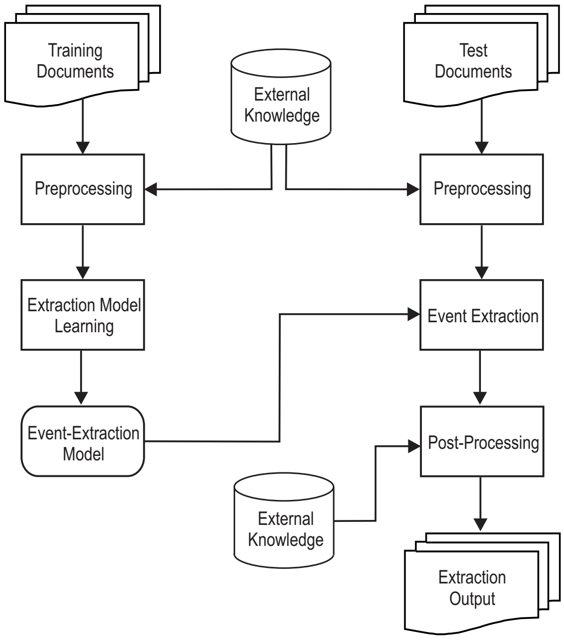

Figure 2.4 An overview of a basic event-extraction framework.

2.2 A Basic Architecture for Event Extraction

In this section, common components used in most event extraction systems based on learning approaches are explained. A general architecture of such systems in shown in Fig. 2.4, where a rectangular box represents a process, a cylindrical shape represents an external resource, a box with rounded corners represents an extraction model, and an arrow specifies process order or an artifact used or produced by a process. Like trainable frameworks for other purposes, event-extraction systems consist two phases: a training phase and a test phase. During the former phase, training documents are firstly preprocessed by several useful linguistic components, such as tokenization, part-of-speech tagging, and syntactic analysis. Learning methods are then selected and applied in order to construct extraction models. During the latter phase, after preprocessing test documents, the extraction models are then deployed and, in some case, postprocessing modules are used for extraction refinement. Some processes in the architecture are supported by external knowledge sources, for example, dictionaries, and domain-specific ontologies. Basic components of the architecture are further discussed below.

2.2.1 Preprocessing

Several types of preprocessing are usually applied and appropriate preprocessing techniques are chosen depending on text analysis algorithms that are intended to be used. Preprocessing components typically extract or label additional information about words or text to reveal syntactic and semantic information. They include:

- **Sentence segmentation** is aimed at dividing a word stream into grammatical sentences. This preprocessing task plays an important role in systems that only consider relations among entities within the same sentence, e.g., [101; 28]. For many European languages, some punctuation marks, particularly the full stop character, explicitly indicate sentence boundaries. Nevertheless, due to the use of such punctuation marks for other purposes (e.g., abbreviation, decimal representation), it is often not trivial to make sentence segmentation. Techniques for sentence boundary disambiguation usually use syntactic and semantic information of tokens around a punctuation mark as features for predicting whether the mark indicates a sentence boundary [43; 73; 81].

- **Tokenization** is a process that breaks a textual document into small units depending on information levels of interest, such as characters and words. In most text processing systems, documents are broken into word tokens, since a word is a fundamental unit that carries meaning [96; 15] and, moreover, several natural language processing algorithms work at the level of word tokens. For languages in which words are consecutively written without delimiters, such as Chinese, Japanese, Korean, and Thai, word tokenization (or word segmentation) is a nontrivial task. Word segmentation methods are categorized into dictionary-based methods [105; 102], machine-learning-based methods [74; 32; 103; 94; 27], and hybrid methods [77; 96; 92; 67].

  When using dictionaries, only words appearing in the dictionaries are identified and, consequently, resulting performance depends greatly upon the quality of dictionaries in use. If the coverage of the dictionaries are not sufficiently high, a great number of out-of-vocabulary (OOV) words may be obtained, leading to low segmentation accuracy [95]. In a machine-learning-based approach, a textual document is first decomposed into smaller units such as characters, syllables, and *n*-grams, and a learned model is then applied to predict whether two contiguous units belong to the same word. A hybrid approach attempts to get the best of the two previous ones. For instance, [96] proposed a dictionary-based statistical system for Myanmar word segmentation. The system begins with separating documents into syllables using linguistic rules. For each text portion divided by punctuation marks and spaces, all possible combinations of merged words are then generated by dictionary-based matching. The combination containing the minimum number of words is taken as the word-segmentation result of the portion. When there are two or more combinations containing the same minimum number of words, one of them is selected based on scores derived using mutual information.

- **Part-of-speech (POS) tagging** is a task of assigning an appropriate POS tag to each word based on the context in which it appears. In order to reduce human effort, many researchers develop both semi-automatic and automatic POS taggers using corpus-based machine-learning approaches. When a POS tagger is trained and evaluated on the same domain, high satisfactory accuracy can be obtained [36].

However, POS-tagging accuracy normally significantly decreases when training and test domains are different. An evidence can be seen in [16], where three Hidden Markov Model (HMM) taggers were learnt from different sources, consisting of a general English corpus, i.e., Penn Treebank-2 [55], and two medical corpara, i.e., GENIA and MED. It was shown that, when evaluated on each medical corpus, the accuracy of the tagger trained on Penn Treebank-2 was substantially lower than that of the tagger generated from part of the evaluated corpus itself. Similar empirical evidences were also reported in [8; 47; 60].

- **Syntactic parsing**, from a linguistic point of view, is a method for analyzing a sentence to determine its grammatical structure [2]. Parsing can be classified by several criteria such as syntactic representation and complexity.

  Syntactic parsing can be characterized based on the forms of target output into two types: constituency and dependency. In the first type, a sentence is decomposed into constituents or phrases, and a phrase structure tree representing relationships between words and phrases is created. By contrast, in the second type, the goal of parsing a sentence is to create a dependency graph consisting of lexical nodes linked by binary relations called dependencies. More comparative discussion between these two approaches can be found in [82]. Based on parsing complexity, parsing can be divided into two types: full parsing and shallow parsing. Full parsing aims to provide a thorough sentence structure, while shallow parsing (or chunking) aims to identify sentence constituents (e.g., noun groups, verb groups, etc.) without specifying their internal structure.

  For IE tasks, syntactic parsing often provides useful features for classifier learning and consequently improves the extraction performance [25; 41; 78; 65; 66; 63]. Many experiments have been conducted to compare accuracy of IE systems with different types of parsing. Such experiments are, for instance, [91; 35; 41; 78; 63]. However, what type of parsing is most appropriate for IE is still a controversial issue.

- **Name entity recognition (NER)** seeks to detect and classify atomic elements in textual documents into predefined classes such as the person names, organizations, locations, quantities, monetary values, percentages, etc. As pointed out in [89], NER and single-slot extraction look similar at first glance. However, as clarified in [50], although single-slot extraction may use the results of NER, it usually makes use of further contextual information to distinguish, for example, the speaker of a seminar from other people mentioned in a seminar announcement.

  There are two main approaches to construct NER systems, i.e., handcrafted-rule approach and statistical learning approach. The former approach normally requires knowledge engineers to build grammatical and semantic rules. This approach usually consumes a lot of manual work to establish well-performing rules through several steps such as rule construction, rule refinement, and rule selection, etc. On the other hand, the latter one requires a large corpus to (semi-)automatically generate NER rules. When a sufficiently large corpus is not available, the handcrafted-rule approach is often employed, e.g., [24; 38; 80].

Results from many experiments investigating the impact of different features on extracting information, such as [100; 30; 11], show that semantic features, e.g., named entities, hypernyms, and WordNet synsets, satisfactorily improve the extraction accuracy. Named entity annotation is usually included as a preprocessing step in most IE systems, especially in domain-specific applications.

2.2.2 Event Extraction

There are two basic approaches to event extraction: single-slot extraction and multi-slot extraction.

Single-Slot Extraction

Single-slot extraction assumes that each information entry contains at most one event of interest and extracts fillers for different slots in an event frame independently. The assumption guarantees that, for each information entry, if individual slots are correctly extracted, then an answer template constructed by combining them also correctly describes a target event.

As an example, [26] defines an event template, which consists of four slots, i.e., *speaker*, *start-time*, *end-time*, and *location*, in order to store information extracted from a set of seminar announcements. A rule set for each template slot is separately learnt from a training corpus and then applied to a test corpus. A comprehensive review of single-slot extraction and empirical comparisons can be found in several articles, such as [12; 50; 69; 83].

Multi-Slot Extraction

Multi-slot extraction is aimed not only at identifying slot fillers but also at connecting slot fillers taking part in the same event. It is applicable even when an information entry contains several target events. Two extraction strategies are commonly used for multi-slot extraction: (i) binary relation extraction with relation merging, and (ii) direct multi-slot extraction.

In the first strategy, event frames are typically constructed by extracting pairs of related slot fillers, i.e., *binary relations*, and then merging those participating in the same event into a more complete frame. Machine-learning paradigms, i.e., supervised learning, semi-supervised learning, and unsupervised learning, can be employed for binary relation extraction. Binary relation extraction based on supervised learning can be categorized into a pattern-based approach, a feature-based approach, and a kernel-based approach. Table 2.1 briefly describes key characteristics along with example frameworks of each binary relation extraction approach.

For merging extracted relations participating in the same event, various techniques ranging from simple methods to complex methods are proposed. A relatively simple method can be found in [14; 58], where a set of extracted binary relations is represented as an undirected graph, with slot fillers being vertices and their relations being edges, and each maximal clique in the graph is transformed into an event template. As an alternative to a simple method, statistics-based approaches are employed in [56; 64], where classification models are constructed to predict whether two extracted relations refer to the same event.

Direct multi-slot extraction attempts to extract all slot fillers of an event simultaneously. For example, from the sentence "*John and Jane are CEOs at Inc. Corp. and Biz. Corp. respectively,*" two frames representing ternary relations (*John, CEO, Inc. Corp.*) and (*Jane, CEO, Biz. Corp.*) should be produced. Classification-based methods are often used for direct multi-slot extraction. In [58], for example, all possible candidates are generated from the above example statement; they are (*John, CEO, Inc. Corp.*), (*John, CEO, Biz. Corp.*), (*Jane, CEO, Inc. Corp.*), (*Jane, CEO, Biz. Corp.*), (*John, CEO,* $\perp$), (*Jane, CEO,* $\perp$), (*John,* $\perp$*, Inc. Corp.*), (*John,* $\perp$*, Biz. Corp.*), (*Jane,* $\perp$*, Inc. Corp.*), (*Jane,* $\perp$*, Biz. Corp.*), ($\perp$*, CEO, Inc. Corp.*), and ($\perp$*, CEO, Biz. Corp.*), where $\perp$ denotes a missing argument. A classification model is then applied to predict whether each of them is a target event. Using this approach, however, a huge amount of candidates are often generated and it is often difficult to manage incomplete but correct instances, e.g., missing-slot extractions such as (*Jane, CEO,* $\perp$), in a classifier learning process.

Table 2.1 Characteristics of binary relation extraction approaches.

Approach	Key characteristics	Example frameworks
Pattern-based	Based on assumption that pairs of terms sharing similar linguistic contexts are connected by similar semantic relations, patterns for capturing relation are mainly constructed in terms of syntactic and semantic constraints. Patterns are normally expressed in forms of regular expressions over triggering words and POS tags.	[5; 17; 33]
Feature-based	Each instance is represented as a feature vector in an n-dimensional space, where n is the number of features used to represent instances. A learning algorithm is used to induce a classifier from feature vectors representing training instances. Feature selection plays an important role in order to reduce complexity of learning in a high dimensional space.	[9; 23]
Kernel-based	A kernel function, a symmetric and positive-semidefinite function, is used for estimating instance similarity. A test instance that is similar to training instances in a certain class C with respect to the kernel function in use is labeled with C.	[79; 108; 111]
Semi-supervised	This paradigm is usually used for learning a classifier when a small amount of labeled data is available in a training corpus. Popular techniques for combining labeled and unlabeled data for training a classifier are, for example, bootstrapping and label propagation.	[7; 13; 51; 110]
Unsupervised	Most unsupervised systems are based on co-occurrence analysis by assuming that two entities are related if they frequently appear together in the same document. Unsupervised relation extraction systems usually aim at detecting the existence of a relation, rather than determining its type.	[1; 10; 21; 76; 52; 97; 99; 104]

Another widely used method for multi-slot extraction is *pattern-based matching*. Extraction patterns are often represented in terms of regular expressions or rules. Manual creation of extraction patterns is an expensive task. Several research efforts focus on automatically (or at least semi-automatically) creating extraction rules. Major systems for learning multi-slot extraction patterns are WI [49], CRYSTAL [84], LIEP [37], and WHISK [85][2]. Table 2.2 characterizes those pattern-based learning systems in terms of their input text styles.

Table 2.2 Characteristics of some systems for learning multi-slot extraction patterns.

System	Text style	
	Structure	Unstructure
WI	✓	–
CRYSTAL	✓	✓
LIEP	✓	✓
WHISK	✓	✓

2.2.3 Post-Processing

Extracted results of IE systems often contain some incorrect extractions, i.e., false positives, leading to low extraction accuracy, especially, precision. To increase the precision, a post-processing phase, normally concerning with filtering those false positives, is implemented in several IE systems. Existing filtering techniques are based on syntactic information, domain-specific knowledge, and statistical learning.

An example of syntax-based filtering can be found in [57], where genetic (gene-gene) relations and etiologic (gene-disease) relations are extracted using an NLP-based IE system, namely SemGen. After applying their IE system to a corpus of Medline citations on diabetes, a distance-based filtering procedure is employed to remove extracted relations that are likely to be incorrect. A relation is filtered out if the number of tokens in the text portion enclosed by the slot fillers of the relation is greater than a predefined threshold.

For filtering driven by domain-specific knowledge, filter conditions, e.g., keywords and rules, are created by domain experts. As an example, [71] proposed an NLP-based system for extracting chemical-enzyme interactions from chemical abstracts, and applied a set of hand-crafted rules for filtering extracted relations.

From a statistical-learning viewpoint, a filtering task can be reduced to a binary classification problem. A classification can be constructed to predict whether extractions are correct. In [45], biological events, each of which consists of three slots, i.e., one interaction type, one effecter, 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

[2]WHISK is a successor of CRYSTAL.

classifier is employed to assign a class to each of its slot fillers,[3] using features such as POS tags, semantic annotations of neighboring words, and distance between consecutive slot fillers. When the class assigned to a slot filler is inconsistent with its extracted type, a frame containing the slot filler is discarded.

2.2.4 External Knowledge

There are various external knowledge sources ranging from general purpose resources applicable in many domains to narrow-scope resources relying on a particular domain. One of the most famous unrestricted domain resources is an English lexical database called WordNet [61]. WordNet groups English words into sets of synonyms, called synsets, and connects syssets using various semantic relations, e.g., Antonymy, Hyponymy, and Meronymy. [87] proposed a framework for learning IE patterns using information of a synset taxonomy (e.g., path lengths between synsets, depths of synsets, etc.) to measure semantic similarity of patterns. From a set of all possible IE patterns, their framework begins with manually selecting a few patterns that are highly relevant to an application domain and adding them to a group of accepted patterns. A remaining pattern is chosen as a newly accepted pattern when it is semantically similar to existing accepted patterns.

Among knowledge resources that are applicable to IE systems, a special attention is recently paid to the role of ontologies. An ontology is the specification of the key concepts in a given domain and the relations that exist among these concepts. In the simplest case, an ontology can be represented as a concept hierarchy, i.e., only *is-a* relation is decribed. For more complex ontologies, other relations are defined and domain-specific axioms are formulated. Ontologies can be adopted in several components of IE systems, such as annotation, pattern generalization, and slot-attribute specification.

2.3 Thai Information Extraction

2.3.1 Thai Writing System

In the Thai writing system, words are consecutively written without delimiters. Spaces are only occasionally presented between phrases or words within sentences—there is no standard rule of how to use spaces in the Thai language. The grammar of the Thai language is considerably simpler than the grammar of most Western languages, and for many foreigners learning Thai, this simplicity compensates for the additional difficulty of learning a variety of voice tones. It is a “Subject + Verb + Object” language with no definite or indefinite article, no verb conjugation, no noun declension, and no object pronouns. Words are not

[3]Candidate classes considered in [45] are interaction type, effecter, and reactant.

modified or conjugated for tense, number, gender, or subject-verb agreement. Tenses, levels of politeness, and verb-to-noun conversion are accomplished the simple addition of various modifying words (called “particles”) to the basic subject-verb-object format. Table 2.3 compares Thai to English.

Table 2.3 Comparing Thai to English.

Features	English	Thai
Word boundary indicated by spaces	Yes	No
An explicit mark (e.g. a full stop) at the end of a sentence	Yes	No
Capitalized letters	Yes	No
Writing left to right	Yes	Yes
Conjugation of verbs	Yes	No
Subject-verb agreement	Yes	No
Use of articles (definite/indefinite)	Yes	No
Pronominal form of social position	No	Yes
Noun classifier	No	Yes

2.3.2 Thai Natural Language Processing

The section presents a review of researches on Thai language processing.

- **Word segmentation:**

  The Thai language belongs to the class of non-segmenting languages. Since 1986, many researchers have been attempting to develop word-segmentation systems for Thai. Their proposed techniques can be categorized into dictionary-based methods, machine-learning-based methods, and hybrid methods.

  In [34], eight word-segmentation methods, four of which are based on dictionaries and the other on machine learning, were evaluated on the ORCHID corpus [86]. The four dictionary-based methods were distinguished by matching strategies[4] and dictionaries in use, i.e., longest matching with a general dictionary, longest matching with a general dictionary augmented with domain-specific words, maximal matching with a general dictionary, and maximal matching with a general dictionary and domain-specific words. They yielded the F-measure values of 87.55%, 91.75%, 87.86%, and 91.98%, respectively. For the machine-learning approach, four well-known classifiers, i.e., Naïve Bayes, Decision Tree, Support Vector Machine, and Conditional Random Field, are evaluated and the F-measure values of 64.90%, 77.50%, 90.74%, and 95.38%, respectively, were reported.

[4] Two matching strategies used in [34] are longest matching and maximal matching. The first strategy attempts to find the longest possible segmentation, while the second one attempts to maximize the number of segmentations that match known words given in a predefined dictionary.

In [4], six word-segmentation methods are evaluated on the legal documents: a dictionary-based methods; three machine-learning-based methods, using 3-state and 6-state Hidden Markov Models (HMMs), and a decision tree; and two hybrid methods, i.e., a 3-state HMM augmented with a dictionary and a decision tree, and a 6-state HMM with a dictionary and a decision tree. The F-measures of 25.77%, 59.37%, 39.18%, 42.15%, 50.59%, and 64.59%, respectively, were reported.

Although the results of several word-segmentation methods show satisfactory performance, different methods may not be compared directly due to non-unifying test sets and different evaluation methodologies in use. Benchmark for Enhancing the Standard of Thai language processing (BEST)—a series of contests on Thai language processing—was settled in 2009. The scope of this series covers several important NLP modules, such as Thai word segmentation, Thai named-entity recognition, compound word and phrase recognition, clause and sentence segmentation, Tree-bank construction, and text summarization. However, its current focus is word segmentation. In BEST-2010, many participants achieved the F-measure of more than 90%, while the winner yielded 94.24%.

Currently available tools for Thai word segmentation include CUWS[5][72], CTTEX[6], SWATH[7][59], and ThaiWordSeg.[8]

- **POS-tagging:**

  Like POS tagging in many other languages, Thai POS taggers are usually trained using supervised machine-learning algorithms, e.g., [68], and accordingly their tagging accuracy depends on the quality of training POS corpora. Currently, only one Thai POS corpus, i.e., ORCHID, is publicly available. ORCHID is a relatively small corpus compared with POS corpora such as the Brown corpus and Penn Chinese Treebank. The source documents in ORCHID are technical papers in the proceedings of the National Electronics and Computer Technology Center (NECTEC) annual conferences.

  As a preliminary study, we conducted an experiment comparing the performance of a POS tagger learned from ORCHID when tested on ORCHID itself and when tested on a collection of thesis abstracts in the chemistry domain. A conditional random field (CRF) model constructed using the CRF++ toolkit[9] was employed in the experiment. When applied to documents in ORCHID, tagging accuracy of the obtained model was approximately 91%. When applied to those in the chemistry domain, the accuracy dropped to 68%. A similar evidence was shown in [88], where the average POS tagging accuracy of 60% was reported when a trigram POS tagging model learned from

[5] Available at http://oracle.cp.eng.chula.ac.th/me/cuws.
[6] Available at http://www.mm.co.th/pub/firefox-thai.
[7] Available at http://www.cs.cmu.edu/ paisarn/software.html.
[8] Available at http://thaiwordseg.sourceforge.net.
[9] Available at http://crfpp.sourceforge.net.

ORCHID was tested in the Thai import and export domain. In [88], in order to use POS information for Thai IE, POS tagging results produced by the learned model were manually corrected.

Domain adaptation methods have been applied to text processing in several languages when a training set and a test set are obtained from different domains (or from different probability distributions) [19; 6; 48]. For example, for English POS tagging, [48], proposed a domain adaptation framework using prior knowledge involving POS tags analyzed from both a training domain and a test domain, e.g., a business and financial domain and a biomedical domain. However, application of domain adaptation to construction of Thai POS taggers has not been reported in the literature.

- **Other linguistic analysis:**

  Other NLP components, such as parsing and sentence segmentation, normally require POS information. Only a few works have been reported on deep linguistic analysis. [98] proposed a machine-learning-based parser for dependency parsing. Given an input sentence, their parser first estimates the probability for a term to be the root node of a parse tree. An SVM model is then used to derive dependency relations, and a beam search algorithm is employed to find the best dependency structure. For sentence-boundary detection, [62] used a part-of-speech trigram model to locate sentence boundaries by classifying white spaces appearing in a paragraph into 2 types: sentence-break spaces and non-sentence-break spaces. The model was trained and evaluated on subsets of the ORCHID corpus and around 80% break detection and 9% false-break rates were achieved.

2.3.3 Information Extraction System for Thai Text

[88] 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 relatively low. As pointed out in [88] 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 [88]; extraction-pattern learning was not discussed.

[70] introduced a method for automated 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,

Table 2.4 Some similarity measures between IFSs.

Author	Expression
Dengfeng[20]	$SM1(A,B) = 1 - \frac{1}{\sqrt[p]{h}} \sqrt[p]{\sum_{i=1}^{h} \lvert\varphi_A(i) - \varphi_B(i)\rvert^p}$ where $\varphi_k(i) = (\mu_k(x_i) + 1 - \nu_k(x_i))/2, k = \{A,B\}$, and $p = 1,2,3,\ldots$
Mitchell[H. 2003]	$SM2(A,B) = \frac{1}{2}\left(\rho_\mu(A,B) + \rho_f(A,B)\right)$ 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[106]	$SM3(A,B) = \frac{1}{h}\sum_{i=1}^{h} \frac{\mu_A(x_i)\mu_B(x_i) + \nu_A(x_i)\nu_B(x_i)}{\sqrt{\mu_A^2(x_i)+\nu_A^2(x_i)}\sqrt{\mu_B^2(x_i)+\nu_B^2(x_i)}}$

such as that in [70], has a serious limitation for a significant number of applications, in particular, when an information entry contains fillers of more than frame, e.g., it cannot relate a particular person with his address when an information entry contains several person names and addresses.

2.4 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 [3] defined an intuitionistic fuzzy set A in $IFS(X)$ as follows:

$$A = \{\langle x, \mu_A(x), \nu_A(x)\rangle | x \in X\} \tag{2.1}$$

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

$$\mu_A : X \to [0,1], \tag{2.2}$$

$$\nu_A : X \to [0,1], \tag{2.3}$$

such that

$$0 \le \mu_A(x) + \nu_A(x) \le 1, \forall x \in X. \tag{2.4}$$

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

$$\pi_A(x) = 1 - \mu_A(x) - \nu_A(x). \tag{2.5}$$

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

A similarity measure S for $IFS(X)$ is a real function $S : IFS(X) \times IFS(X) \to [0,1]$, 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 2.4 highlights some similarity measures between IFSs. SM1 and SM2 are distance-based measures, while SM3 is cosine-based measures.

Chapter 3

Framework

3.1 Information Extraction from Thai Texts

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

3.1.1 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. Fig. 3.1 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, '$\sim$' 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. Fig. 3.2 provides a literal English translation of this text portion; the translations of the three target phrases are also underlined. Fig. 3.3 shows the frame required to be extracted from the second underlined symptom phrase in Fig. 3.1. It contains three slots, i.e., SYM, LOC, and PER, which stand for "symptom," "location," and "period," respectively.

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|. . .

Figure 3.1 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*. . .

Figure 3.2 A literal English translation of the partially annotated Thai text in Fig. 3.1

3.1.2 IE Rules and Rule Application

A well-known supervised rule learning algorithm, called WHISK [85], is used as the core algorithm for constructing extraction rules. Figure 3.4 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. 3.3, 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. 3.5, when the rule in Fig. 3.4 is applied to the information entry in Fig. 3.1 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 3.6 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.

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*]}

Figure 3.3 A target phrase and an extracted frame

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

Figure 3.4 An IE rule example

Table 3.1: Instantiation of the internal wildcards of the rule in Fig.3.4 into the information entry in Fig.3.1.

Extraction	Text portion	Internal wildcard 1st	Internal wildcard 2nd	Internal wildcard 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]

3.2 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.

3.2.1 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. 3.6), 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 charac-

[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

Figure 3.5: Text portions from which extractions are made when the rule in Fig. 3.4 is applied to the information entry in Fig. 3.1 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

Figure 3.6 Frames extracted from the text portions in Fig. 3.5 by the rule in Fig. 3.4

teristics 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 [40; 109; 42; 54].

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:

3.2.2 Preprocessing

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,[1] into which each internal wildcard is instantiated. (All wildcards except the first one are called an *internal wildcard*.)

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

Table 3.2 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	$[1,0,0,0,1,0,0,0,1,1,0,0]$

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 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 3.2.1. *This example illustrates the vector-based representation process. Suppose, in a training corpus, the rule shown in Fig.3.4 can produce extractions only when it is applied to the information entry in Fig.3.1. Then solely three extractions (cf. Fig.3.5 and 3.6) are made. Table 3.1 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

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

It is worthy to emphasis that, for tokens with semantic tags, we collect only their tags. For example, “sym” in W^1_{inc} is the class tag of $w_3$6. Following the (a3) step, we can construct a vector representation corresponding to each extraction as depicted in Table 3.2. □

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

$$X = \{x^1_1, x^1_2, x^1_3, x^1_4 \ldots, x^n_1, x^n_2, x^n_3, x^n_4\}.$$

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

Example 3.2.2. *Let consider the output the feature vectors from Example 3.2.1, 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 $f^1_{2,1} > f^1_{1,1} > f^1_{3,1}$, 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 $f^s_{i,j}$ in the vector-based representation of the i-th extraction is then converted in terms of the three degrees of x^s_j as the following steps:

(b1) $f^s_{i,j}$ is normalized by:

$$z^s_{i,j} = \begin{cases} \frac{f^s_{i,j} - \overline{X}^k_j}{sd^s_j}, & sd^s_j \neq 0 \\ 0 & sd^s_j = 0 \end{cases}, \tag{3.1}$$

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

$$\overline{X}^s_j = \frac{\sum_{i=1}^{|E_r|} f^s_{i,j}}{|E_r|}, \tag{3.2}$$

and

$$sd^s_j = \left(\frac{\sum_{i=1}^{|E|} (f^s_{i,j} - \overline{X}^s_j)^2}{|E_r|} \right)^{1/2}. \tag{3.3}$$

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

$$\mu_i(x^s_j) = r^s_j \frac{1}{1+e^{-z^s_{i,j}}}, \tag{3.4}$$

where $0 < r^s_j \leq 1$ is a weight for x_j.

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

$$\nu_i(x^s_j) = \overline{r}^s_j \frac{1}{1+e^{z^s_{i,j}}}, \tag{3.5}$$

where $0 < \overline{r}^s_j \leq 1$ is a weight for x_j.

(b4) Denoted by $\pi_i(x^s_j)$, the hesitancy degree of the document i with respect to x^s_j is calculated by (2.5), i.e.,

$$\pi_i(x^s_j) = 1 - \mu_i(x^s_j) - \nu_i(x^s_j).$$

Example 3.2.3. *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 3.2.2. For convenience, the vectors are represented in terms of the matrix E shown in Table 3.3. 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^s_j *and* $\overline{r}^s_j$ *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,*

Table 3.3 An example of the proposed IFS-based representation from Example 3.2.3.

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	$\begin{bmatrix} 0.51 & 0.45 & 0.18 & 0.60 & 0.18 & 0.45 & 0.18 & 0.45 & 0.45 & 0.45 & 0.45 & 0.45 \\ 0.19 & 0.45 & 0.60 & 0.18 & 0.60 & 0.45 & 0.60 & 0.45 & 0.45 & 0.45 & 0.45 & 0.45 \\ 0.67 & 0.45 & 0.60 & 0.60 & 0.60 & 0.45 & 0.60 & 0.45 & 0.45 & 0.45 & 0.45 & 0.45 \end{bmatrix}$

gathering the first row of the matrices, we can form an IFS, namely IFS_1 corresponding to $\vec{V}_1$:

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

□

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

Chapter 4

Experimental Results

4.1 Data Sets, Output Templates, and Training Process

4.1.1 Data set preparation

The proposed framework is evaluated in three different domains of Thai text: *medical-symptom descriptions (MD), soccer match reports (SR), soccer player transferring (PT), housing advertisements (HA), stock prices (SP), company dividends (CD),* and *chemical-reaction descriptions (CR).* 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[93]. An information entry in the SR and PT data sets 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. The information entries for the business domains, i.e. SP and CD, are collected from on-line newspapers and the last domain from Thai dissertation and thesis on-line database[1] provided by Technical Information Access Center (TIAC).

As results, 173, 294, 448, 130, 191, 245, and 220 information entries with the average length of 45.0, 68.6, 64.3, 94.7, 62.1, 72.6, and 275.2 words per entry in the MD, SR, PT, HA, SP, CD, and CR 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[93]. A lexicon containing soccer player names and soccer team names, collected as part of a project on developing an alias extraction system [90], is used for semantic annotation in the domains SR and PT, while a lexicon containing city names is used for HA. Moreover, regular expression based semi-automatic annotation

[1] Available at `http://thesis.stks.or.th`.

Table 4.1 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.
PT	[Cl1 $C1$][Cl2 $C2$][Fee F] [Ply P]	A club $C1$ pay a club $C2$ a transfer fee F for a player P.
HA	[Area A][Bdr N][Rsr M]	A house of area size A has N bedrooms and M rest rooms.
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.
CR	[PDT P][RNM R][RCT T] [CAT C]	A substance P is obtained from a chemical reaction R using a substance T as a reactant and a substance C as a catalyst.

is applied for tagging quantity information, e.g. “Period of Time,” “Minute,” and “Price.”

4.1.2 Output templates

Seven 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* and one from each of the rest domains, referred to as *Type-SR*, *Type-PT*,*Type-HA*, *Type-SP*, *Type-CD*, and *Type-CR*. Table 4.1 gives the output-template forms for the seven 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. 3.1 (also in Fig. 3.2) 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. The slot DIF in the SP template and the slot TOT in the CD template are optional. A target phrase in the CR domain contains at least two of the following components: reaction

Table 4.2 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	A-MD1	90	11	3.5	2	7	3.6	1
MD1	B-MD1	136	8	3.3	2	11	2.9	1
MD2	A-MD2	80	15	4.1	2	3	1.4	0
MD2	B-MD2	66	8	3.9	2	5	1.2	0
SR	A-SR	93	28	8.0	3	6	2.0	2
SR	B-SR	156	21	6.4	3	6	3.9	2
PT	A-PT	90	64	26.5.0	11	2	0.5	0
PT	B-PT	108	57	29.1	12	2	0.4	0
HA	A-HA	87	37	16.1	7	2	1.2	1
HA	B-HA	113	34	15.2	7	2	1.3	1
SP	A-SP	109	74	18.0	5	6	1.4	0
SP	B-SP	122	49	13.0	8	8	2.3	0
CD	A-CD	106	73	19.7	5	7	1.2	0
CD	B-CD	117	66	18.1	5	7	1.2	0
CR	A-CR	122	41	10.6	3	8	1.0	0
CR	B-CR	188	22	10.0	3	9	1.9	0

name (RMN), reaction products (PDT), reactants (RCT), and catalysts (CAT). An output frame in the CR domain may contain more than one slot of the same type.[2]

4.1.3 Rule learning

For each template, the collected information entries are randomly divided into two data sets, of the same size. The obtained data sets are referred to as A-MD1 and B-MD1 for MD1, A-MD2 and B-MD2 for MD2 A-SR and B-SR for SR A-PT and B-PT for PT A-HA and B-HA for HA A-SP and B-SP for SP A-CD and B-CD for CD A-CR and B-CR for CR Each of them is once used as a training set and once as a test set. Table 4.2 characterizes target phrases in the obtained data sets

Using our implementation of WHISK, IE rules are automatically generated. Table 4.3 summarizes information of the rule set for each template about the numbers of generated IE

[2]For instance, from the text "The main products obtained from the oxidation reaction of ethanol are acetaldehyde and carbon dioxide," the output frame consisting of one reaction-name slot, one reactant slot, and two product slots should be generated.

Table 4.3 IE-rule characteristics for each data set

Data set	No. of rules
A-MD1	15
B-MD1	17
A-MD2	11
B-MD2	14
A-SR	7
B-SR	8
A-PT	20
B-PT	18
A-HA	9
B-HA	9
A-SP	18
B-SP	18
A-CD	5
B-CD	8
A-CR	54
B-CR	47

rules. Examples of rules learned from these training data sets are shown in Appendix A.

4.2 Parameter setting

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

- The weights r_j^s and $\bar{r}_j^s$ are based on statistical characteristics of feature type by

$$r_j^s = \bar{r}_j^s = \frac{|1 - sd_j^s|}{|1 + sd_j^s|}.$$

- The neighborhood size k, in this experiment, is varied as 1, 3, and 5.

Table 4.4 Evaluation results using the base window size (1W)

Method	k	Template Type															
		MD1		MD2		SR		PT		HA		SP		CD		CR	
		R	P	R	P	R	P	R	P	R	P	R	P	R	P	R	P
RAW	-	76.33	83.60	100.00	39.12	81.50	55.29	81.95	48.07	82.93	40.57	85.42	45.13	85.75	58.49	60.83	68.68
RAW+SM1	1	75.85	97.52	98.93	92.50	79.77	92.62	80.02	85.78	81.46	85.20	83.32	92.30	79.53	89.90	60.10	93.40
	3	76.33	97.53	100.00	94.92	80.35	92.67	81.01	90.20	81.46	86.53	84.01	93.00	83.45	93.45	59.57	95.12
	5	76.33	95.76	99.47	93.47	80.92	92.11	80.56	91.45	80.98	86.01	81.13	95.70	82.40	94.10	59.57	94.35
RAW+SM2	1	75.85	97.52	98.93	92.50	79.77	92.62	79.89	85.78	80.49	84.18	81.56	90.08	81.10	92.15	58.76	91.30
	3	76.33	97.53	99.47	95.38	80.35	92.67	80.35	94.50	81.46	86.53	84.01	93.60	80.05	92.23	60.10	90.44
	5	76.33	95.76	99.47	93.47	79.77	91.39	80.24	91.45	80.98	85.57	84.01	93.02	79.78	94.67	57.55	89.74
RAW+SM3	1	75.85	98.13	98.93	93.43	79.77	92.62	80.02	90.35	81.95	85.71	82.33	94.71	80.20	93.02	59.08	93.27
	3	76.33	98.14	100.00	95.90	80.35	93.92	81.01	93.40	81.95	87.50	80.17	95.63	83.51	95.12	60.10	95.12
	5	76.33	96.34	100.00	94.44	80.92	93.33	80.56	91.28	81.95	86.60	84.20	94.08	84.23	94.08	59.74	94.08

Table 4.5 Evaluation results using the double base window size (2W)

Method	k	Template Type															
		MD1		MD2		SR		PT		HA		SP		CD		CR	
		R	P	R	P	R	P	R	P	R	P	R	P	R	P	R	P
RAW	-	87.44	61.36	100.00	34.00	95.95	59.71	92.59	34.73	92.68	35.32	93.87	43.15	95.93	51.76	71.18	54.29
RAW+SM1	1	84.54	97.22	96.26	93.75	92.49	91.43	92.30	85.20	90.24	88.10	91.43	90.52	91.23	90.60	67.34	85.40
	3	86.47	94.71	98.40	94.85	94.22	94.77	90.45	86.53	92.20	85.52	94.87	92.78	93.50	91.08	70.06	83.67
	5	86.96	93.75	99.47	90.73	93.64	94.19	91.63	87.24	92.20	84.38	93.50	91.18	92.54	92.15	69.45	85.84
RAW+SM2	1	84.54	97.22	96.26	93.75	92.49	91.43	90.45	85.10	90.24	88.52	91.43	90.45	90.15	90.25	67.34	83.55
	3	86.47	94.71	98.40	94.85	94.22	94.77	92.01	86.53	92.20	85.52	91.43	89.07	93.50	91.18	70.06	84.32
	5	86.96	93.75	98.93	90.69	93.64	94.19	91.63	85.57	92.20	85.14	92.13	91.26	94.04	89.75	70.75	82.68
RAW+SM3	1	85.51	95.68	98.93	94.87	94.22	97.02	92.01	85.71	91.22	88.63	90.69	93.09	92.54	90.38	68.05	83.01
	3	86.96	95.74	100.00	95.41	94.80	96.47	90.45	87.50	91.71	89.10	94.87	94.77	93.50	87.45	70.06	86.32
	5	87.44	93.78	100.00	94.92	94.80	95.35	91.02	86.60	92.20	86.70	94.87	94.10	93.50	91.64	71.07	85.28

4.3 Experimental results

The proposed framework is evaluated using the four test data sets for their respective template types (cf. Table 4.2). 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. 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 4.4 and 4.5 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-PT, Type-HA, and Type-SP 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 4.4, 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.

Table 4.6 Comparison with rule application to manually identified target phrases

Method	Template Type															
	MD1		MD2		SR		PT		HA		SP		CD		CR	
	R	P	R	P	R	P	R	P	R	P	R	P	R	P	R	P
Known Boundary	88.41	97.86	100.00	100.00	97.69	97.69	92.59	89.08	95.12	86.67	95.10	94.97	96.63	94.16	79.44	87.99
RAW+SM1	85.99	95.23	98.04	93.11	93.45	93.46	91.46	86.32	91.54	86.00	93.27	91.49	92.42	91.28	68.95	84.97
RAW+SM2	85.99	95.23	97.86	93.09	93.45	93.46	91.36	85.73	91.54	86.39	91.66	90.26	92.56	90.39	69.38	83.52
RAW+SM3	86.63	95.07	99.64	95.07	94.61	96.28	91.16	86.60	91.71	88.14	93.48	93.99	93.18	89.82	69.73	84.87

4.3.1 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 sets and apply the rules obtained from WHISK directly to these manually identified text portions. Table 4.6 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, HA, and CR 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 4.6, RAW+SM3 improves the precision of known-boundary extraction from 86.67 to to 88.14 for Type-HA.

Table 4.7 Comparison with other filtering techniques

Method	Template Type															
	MD1		MD2		SR		PT		HA		SP		CD		CR	
	R	P	R	P	R	P	R	P	R	P	R	P	R	P	R	P
RAW+SM1	85.99	95.23	98.04	93.11	93.45	93.46	91.46	86.32	91.54	86.00	93.27	91.49	92.42	91.28	68.95	84.97
RAW+SM2	85.99	95.23	97.86	93.09	93.45	93.46	91.36	85.73	91.54	86.39	91.66	90.26	92.56	90.39	69.38	83.52
RAW+SM3	86.63	95.07	99.64	95.07	94.61	96.28	91.16	86.60	91.71	88.14	93.48	93.99	93.18	89.82	69.73	84.87
RAW+SVM	89.18	92.89	85.03	91.89	92.69	92.18	89.82	82.56	89.91	83.90	91.68	86.01	92.72	89.34	65.09	78.04
RAW+kNN	88.48	93.85	85.50	91.55	90.87	92.04	85.19	79.01	87.13	85.24	90.73	83.31	92.81	89.75	62.82	84.56
RAW+NB	83.24	96.22	86.89	87.10	89.56	95.11	87.50	81.47	87.13	85.81	92.03	85.54	91.26	90.45	63.62	80.55
RAW+DT	88.28	94.77	87.36	93.38	90.35	93.84	84.26	76.16	87.13	86.42	91.14	83.35	89.40	85.85	63.95	81.85

4.3.2 Comparison with Extraction with Other Filtering Techniques

The proposed framework is also compared with the other filtering techniques are used. 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, for prediction of rule application across a target-phrase boundary. The four models are constructed from the vectors corresponding to the extractions from the training corpus (cf. Table 3.2 in Sec.3.2.2).

Table 4.7 compares the proposed framework[3] 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, SR, and CR templates whose the target phrase lengths 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.

[3] The average performance over the k values is shown and it is similar to that in Table4.6.

Chapter 5

Application to Semantics-Based Information Retrieval

In traditional keyword-based information retrieval systems, retrieval results are determined solely by appearance of query keywords in documents or in document indexes. In domain-specific applications, however, it is often desirable to describe an information need more precisely by specifying required relations between domain concepts. A user in the chemistry domain, for example, may wish to search for a document concerning

"*a chemical reaction that produces a compound containing a carbon atom.*"

With the background knowledge that "propionaldehyde has some carbon atom as its component," the same user may furthermore expect the retrieval results to include a document containing a statement such as

"*propionaldehyde is obtained from the oxidation reaction of 1-propanol,*"

which looks very different syntactically from the search condition specified above. It is anticipated that IE technology and recent development of machine-processable ontology languages, such as OWL [75], will contribute significantly to realization of such semantics-based information retrieval.

This chapter illustrates application of our IE framework to semantics-based information retrieval. Frames extracted from chemical-reaction descriptions are represented as concept expressions in description logics (DL), which can readily be encoded in OWL, and are used as metadata for document indexing. To support semantics-based document retrieval, they are integrated with existing OWL chemical-substance and chemical-reaction ontologies, which provide domain-specific background knowledge.

5.1 Document Representation and Integration with Background Knowledge

Fig. 5.1 illustrates the concept expression representing the second chemical-reaction statement above. Assuming that d is a document from which n extracted frames, say $f_1, \ldots, f_n$, are obtained, d is then represented by a concept C_d defined by the equality axiom

$$C_d \equiv \mathsf{Doc} \sqcap \exists \text{HASINDEX}.C_1 \sqcap \cdots \sqcap \exists \text{HASINDEX}.C_n,$$

$$\mathsf{ChemReaction} \sqcap \exists \textsc{HasRnm}.\mathsf{Oxidation} \sqcap \exists \textsc{HasPdt}.\mathsf{Propionaldehyde} \sqcap \exists \textsc{HasRct}.\mathsf{1\text{-}Propanol}$$

Figure 5.1 A concept expression representing the second chemical statement.

Table 5.1 Ontology characteristics.

Ontology	No. of concepts	No. of leaf concepts	Ontology depth			No. of roles	No. of existential restrictions	No. of universal restrictions
			Max.	Avg.	Min.			
Chem. Complex	791	694	10	6.30	3	74	354	77
Rex	546	289	14	6.33	1	5	1	0

where Doc is a primitive concept denoting the set of all documents and $C_1, \ldots, C_n$ are concept expressions representing the frames $f_1, \ldots, f_n$, respectively.

A document knowledge base is constructed by integrating axioms describing documents with domain-specific ontologies. Two existing OWL ontologies, Chemical Complex ontology[1] and Rex ontology,[2] were used in our exploratory study. The former ontology describes both chemical substances (including atoms, molecules, and organic compounds) and reactions using various restrictions on role fillers, while the latter one focuses mainly on classification taxonomies of chemical reactions. Table 5.1 gives some characteristics of these two ontologies and Fig. 5.2 shows some background-knowledge axioms they provide.

$$\mathsf{1\text{-}Propanol} \sqsubseteq \mathsf{Alcohol} \tag{5.1}$$

$$\mathsf{Propionaldehyde} \sqsubseteq \mathsf{Aldehyde} \tag{5.2}$$

$$\mathsf{Aldehyde} \equiv \mathsf{OrganicCompound} \sqcap \exists \textsc{HasPart}.\mathsf{AldehydeGroup} \tag{5.3}$$

$$\mathsf{OrganicCompound} \sqsubseteq \mathsf{Compound} \tag{5.4}$$

$$\mathsf{AldehydeGroup} \sqsubseteq \mathsf{CarbonAtom} \sqcap \exists \textsc{HasBondWith}.\mathsf{OxygenAtom} \sqcap \exists \textsc{HasBondWith}.\mathsf{HydrogenAtom} \tag{5.5}$$

$$\textsc{HasPdt} \sqsubseteq \textsc{HasParticipant} \tag{5.6}$$

$$\mathsf{OrganicReaction} \equiv \mathsf{ChemReaction} \sqcap \exists \textsc{HasParticipant}.\mathsf{OrganicCompound} \tag{5.7}$$

Figure 5.2 Part of background knowledge.

[1] Available at http://ontology.dumontierlab.com.

[2] Available at http://onto.eva.mpg.de/obo.

C_{q_1}: $\mathsf{Doc} \sqcap \exists \text{HASINDEX}.(\mathsf{ChemReaction} \sqcap \exists \text{HASRCT}.\mathsf{Alcohol})$

C_{q_2}: $\mathsf{Doc} \sqcap \exists \text{HASINDEX}.\mathsf{OrganicReaction}$

C_{q_3}: $\mathsf{Doc} \sqcap \exists \text{HASINDEX}.(\mathsf{ChemReaction} \sqcap \exists \text{HASPDT}.(\mathsf{Compound} \sqcap \exists \text{HASPART}.\mathsf{CarbonAtom}))$

Figure 5.3 Query representation.

5.2 Document Retrieval: Examples

To demonstrate semantics-based information retrieval in the obtained document knowledge base, assume that d_0 is a document containing the following phrase, i.e.,

"propionaldehyde is obtained from the oxidation reaction of 1-propanol," (5.8)

and consider the following three queries:

q_1: Find documents that discuss a chemical reaction involving an alcohol as a reactant.

q_2: Find documents that discuss an organic reaction.

q_3: Find documents that discuss a reaction producing a compound containing a carbon atom.

Knowing that (i) 1-propanol is a kind of alcohol, (ii) propionaldehyde is an organic compound and a reaction involving an organic compound is called an organic reaction, and (iii) propionaldehyde has some carbon atom as its component, one would expect that each of q_1, q_2, and q_3 retrieves d_0. Such semantics-based retrieval requires domain-specific background knowledge and an inference mechanism, which can be realized using subsumption reasoning in DL.

Using subsumption reasoning, a document d is retrieved by a query q if the concept expression representing d is subsumed by that representing q with respect to background-knowledge axioms. Suppose that

- the document d_0 mentioned above is represented by the concept C_{d_0} defined by the axiom $C_{d_0} \equiv \mathsf{Doc} \sqcap \exists \text{HASINDEX}.C$, where C is the concept expression in Fig. 5.1, which represents the frame extracted from Statement (5.8),
- the queries q_1, q_2, and q_3 are represented by the concept expressions C_{q_1}, C_{q_2}, and C_{q_3}, respectively, in Fig. 5.3, and
- the background-knowledge axioms in Fig. 5.2 are employed.

A DL-based reasoner then infers that C_{q_1} subsumes C_{d_0} in one inference step using Axiom (5.1), infers that C_{q_2} subsumes C_{d_0} in four steps using Axioms (5.2), (5.3), (5.6), and (5.7), and infers that C_{q_3} subsumes C_{d_0} in four steps using Axioms (5.2), (5.3), (5.4), and (5.5). Accordingly, each of q_1, q_2, and q_3 retrieves d_0.

It is noteworthy that the concept expression shown in Fig. 5.1, the background knowledge axioms shown in Fig. 5.2, and the expressions representing queries in Fig. 5.3 can all be formalized using the lightweight description logic $\mathcal{EL}$ [53], for which polynomial-time reasoners (e.g., CEL[3]) are available. However, the Chemical Complex ontology, which is used as part of our document knowledge base (see Section 5.1), contains some axioms that are constructed using concept constructors such as cardinality restriction, universal restriction, and union, which are not provided by $\mathcal{EL}$. All axioms in our document knowledge base can be formalized in the $\mathcal{SHOIN}(\mathbf{D})$ description logic, which is the underlying formalism of OWL-DL.

[3]Available at http://lat.inf.tu-dresden.de/systems/cel.

Chapter 6

Conclusions

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. The primary purpose of this project is to provide a framework for event extraction from Thai unstructured text.

From a set of manually collected target phrases, IE rules are created using our implementation of WHISK. To apply the obtained rules to unstructured-text information entries with unknown target-phrase boundaries, rule application using sliding windows (RAW) is introduced. The IFS-based filtering technique is proposed for the removal of false positives resulting from rule application across target-phrase boundaries. The experimental results obtained from evaluation on six domains, i.e., medical-symptom descriptions (MD), soccer match reports (SM), soccer player transfers (PT), house advertising (HA) stock prices (SP), company dividends (CD), and chemical-reaction descriptions (CR), show that these filtering methods improve precision and satisfactorily preserve high recall of RAW. The proposed framework outperforms the baseline method using the four classification models, i.e. SVM, kNN, naive bayes, and decision trees.

To demonstrate how the proposed IE framework facilitates semantics-based information retrieval, extraction results in the domain of chemical-reaction descriptions are represented as concept expressions in description logics and used as metadata for document indexing. Using domain-specific ontologies as background knowledge, semantics-based document retrieval is demonstrated. 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.

Bibliography

[1] Abulaish, M. and Dey, L. (2007). Biological relation extraction and query answering from medline abstracts using ontology-based text mining. *Data & Knowledge Engineering*, 61:228–262.

[2] Allen, J. F. (1987). *Natural Language Understanding*. Benjamin Cummings, Menlo Park, California, first edition edition.

[3] Atanassov, K. (1986). Intuitionistic fuzzy sets. *Fuzzy Set and Systems*, 20:87–96.

[4] Bheganan, P., Nayak, R., and Xu, Y. (2009). Thai word segmentation with hidden markov model and decision tree. In *Proceedings of the 13th Pacific-Asia Conference on Knowledge Discovery and Data Mining*, volume 5476 of *Lecture Notes in Computer Science*, pages 74–85, Bangkok, Thailand.

[5] Blaschke, C. and Valencia, A. (2002). The frame-based module of the suiseki information extraction system. *IEEE Intelligent Systems*, 17(2):14–20.

[6] Blitzer, J. (2007). *Domain Adaptation of Natural Language Processing Systems*. PhD thesis, Computer and Information Science.

[7] Blum, A. and Mitchell, T. (1998). Combining labeled and unlabeled data with co-training. In *Proceedings of the 11th Annual Conference on Computational learning theory*, pages 92–100, Wisconsin, USA.

[8] Brants, T. (2000). Tnt: A statistical part-of-speech tagger. In *Proceedings of the 6th Conference on Applied Natural Language Processing*, pages 224–231, Seattle, Washington.

[9] Bunescu, R., Ge, R., Kate, R. J., Marcotte, E. M., Mooney, R. J., Ramani, A. K., and Wong, Y. W. (2005). Comparative experiments on learning information extractors for proteins and their interactions. *Artificial Intelligence in Medicine*, 33(2):139–155.

[10] Bunescu, R., Mooney, R., Ramani, A., and Marcotte, E. (2006). Integrating co-occurrence statistics with information extraction for robust retrieval of protein interactions from medline. In *Proceedings of the HLT-NAACL Workshop on Linking Natural Language Processing and Biology: Towards deeper biological literature analysis*, pages 49–56, New York City, NY.

[11] Cao, Y.-G., Cimino, J. J., Ely, J., and Yu, H. (2010). Automatically extracting information needs from complex clinical questions. *Journal of Biomedical Informatics*, 43:962–971.

[12] Chang, C.-H., Kayed, M., Girgis, M. R., and Shaalan, K. F. (2006). A survey of web information extraction systems. *IEEE Transactions on Knowledge and Data Engineering*, 18(10):1411–1428.

[13] Chen, J., Ji, D., Tan, C. L., and Niu, Z. (2006). Semi-supervised relation extraction with label propagation. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the ACL*, pages 25–28, New York, USA. Association for Computational Linguistics.

[14] Chieu, H. L. and Ng, H. T. (2002). A maximum entropy approach to information extraction from semi-structured and free text. In *Proceedings of the 8th National Conference on Artificial Intelligence*, pages 786–791, Alberta, Canada. American Association for Artificial Intelligence.

[15] Choi, K.-S., Isahara, H., Kanzaki, K., Kim, H., Pak, S. M., and Sun, M. (2009). Word segmentation standard in chinese, japanese and korean. In *Proceedings of the 7th Workshop on Asian Language Resources*, pages 179–186, Stroudsburg, PA, USA. Association for Computational Linguistics.

[16] Coden, A. R., Pakhomov, S. V., Ando, R. K., Duffy, P. H., and Chute, C. G. (2005). Domain-specific language models and lexicons for tagging. *Journal of Biomedical Informaiton*, 38:422–430.

[17] Corney, D. P. A., Buxton, B. F., Langdon, W. B., and Jones, D. T. (2004). Biorat: Extracting biological information from full-length papers. *Bioinformatics*, 20(17):3206–3213.

[18] Danvivathana, N. (1987). *The Thai Writing System*. Helmut Buske Verlag, Hamburg.

[19] Daume, H. and Marcu, D. (2006). Domain adaptation for statistical classifiers. *Journal of Artificial Intelligence Research*, 26:101–126.

[20] Dengfeng, L. and Chuntian, C. (2002). New similarity measures of intuitionistic fuzzy sets and application to pattern recognition. *Pattern Recognition Letters*, 23:221–225.

[21] Ding, J., Berleant, D., Nettleton, D., and Wurtele, E. (2002). Mining medline: Abstracts, sentences, or phrases? In *Pacific Symposium on Biocomputing*, pages 326–337, Lihue, Hawaii.

[22] Doddington, G., Mitchell, A., Przybocki, M., Ramshaw, L., Strassel, S., and Weischedel, R. (2004). The automatic content extraction (ace) program–tasks, data, and evaluation. In *Proceedings of LREC 2004*, pages 837–840.

[23] Donaldson, I., Martin, J., Bruijn, B., Wolting, C., Lay, V., Tuekam, B., Zhang, S., Baskin, B., Bader, G., Michalickova, K., Pawson, T., and Hogue, C. (2003). Prebind and

textomy–mining the biomedical literature for protein-protein interactions using a support vector machine. *BMC Bioinformatics*, 4.

[24] Farmakiotou, D., Karkaletsis, V., Koutsias, J., Sigletos, G., Spyropoulos, C. D., and Stamatopoulos, P. (2000). Citeseerx rule-based named entity recognition for greek texts. In *Proceedings of the Workshop on Computational lexicography and Multimedia Dictionaries (COMLEX 2000)*, pages 75–78.

[25] Feldman, R. and Sanger, J. (2006). *The Text Mining Handbook: Advanced Approaches in Analyzing Unstructured Data*. Cambridge University Press.

[26] Freitag, D. (1998). *Machine Learning for Information Extraction in Informal Domain*. PhD thesis, Carnegie Mellon University.

[27] Gao, J., Li, M., Wu, A., and Huang, C.-N. (2006). Chineseword segmentation and named entity recognition: A pragmatic approach. *Computational Linguistics*, 31(4):531–574.

[28] Giuliano, C., Lavelli, A., and Romano, L. (2007). Relation extraction and the influence of automatic named-entity recognition. *ACM Transactions on Speech and Language Processing*, 5(1):2:1–2:26.

[29] Grishman, R. and Sundheim, B. (1996). Message understanding conference - 6: A brief history. In *Proceedings of the 16th International Conference on Computational Linguistics*, pages 466–471, Copenhagen, Denmark.

[30] GuoDong, Z. and Min, Z. (2007). Extracting relation information from text documents by exploring various types of knowledge. *Information Processing and Management*, 43:969–982.

[H. 2003] H. B. M. (2003). On the dengfeng-chuntian similarity measure and its application to pattern recognition. *Pattern Recognition Letters*, 24:3101–3104.

[32] Ha, L. (2003). A method for word segmentation in vietnamese. In *Proceedings of the Corpus Linguistics 2003 Conference*, pages 282–287, Lancaster, UK.

[33] Hao, Y., X. Zhu, M. H., and Li, M. (2005). Discovering patterns to extract proteinprotein interactions from the literature: Part ii. *Bioinformatics*, 21(15):3294–3300.

[34] Haruechaiyasak, C., Kongyoung, S., and Dailey, M. (2008). A comparative study on thai word segmentation approaches. In *Proceedings of ECTI-CON 2008*, pages 125–128.

[35] Hearne, M., Ozdowska, S., and Tinsley, J. (2008). Comparing constituency and dependency representations for smt phrase-extraction. In *Actes de la 15e Conférence Annuelle sur le Traitement Automatique des Langues Naturelles*, TALN ’08, Avignon, France. ATALA.

[36] Huang, L., Peng, Y., Wang, H., and Wu, Z. (2006). Statistical part-of-speech tagging for classical chinese. In *Text, Speech and Dialogue*, volume 2448 of *Lecture Notes in Computer Science*, pages 296–311.

[37] Huffman, S. B. (1996). Learning information extraction patterns from examples. In *Proceedings of Connectionist, Statistical, and Symbolic Approaches to Learning for Natural Language Processing*, volume 1040 of *Lecture Notes in Computer Science*, pages 246–260, London, UK.

[38] Humphreys, K., Gaizauskas, R., Azzam, S., Huyck, C., Mitchell, B., Cunningham, H., and Wilks, Y. (1998). University of sheffield: Description of the lasie-ii system as used for muc-7. In *Proceedings of the Seventh Message Understanding Conferences (MUC-7)*. Morgan.

[39] Intarapaiboon, P. (2016). Text classification using similarity measures on intuitionistic fuzzy sets. *ScienceAsia*, 42:52–60.

[40] Intarapaiboon, P., Nantajeewarawat, E., and Theeramunkong, T. (2012). Extracting semantic frames from thai medical-symptom unstructured text with unknown target-phrase boundaries. *IEICE Transactions on Information and Systems*, E94.D:465–478.

[41] Johansson, R. and Nugues, P. (2008). The effect of syntactic representation on semantic role labeling. In *Proceedings of the 22nd International Conference on Computational Linguistics - Volume 1*, COLING '08, pages 393–400, Manchester, United Kingdom. Association for Computational Linguistics.

[42] Kenter, T. and Rijke, M. (2015). Short text similarity with word embeddings. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, pages 1411–1420, Melbourne, Australia.

[43] Khalifa, I., Feki, Z. A., and Farawila, A. (2011). Arabic discourse segmentation based on rhetorical methods. *International Journal of Electric & Computer Sciences*, 11(1):10–15.

[Khatibi and G. 2009] Khatibi, V. and G. A. M. (2009). Intuitionistic fuzzy set vs. fuzzy set application in medical pattern recognition. *Artificial Intelligence in Medicine*, 47:43–52.

[45] Kim, E., Song, Y., Lee, C., Kim, K., Lee, G., and Yi, B.-K. (2006). Two-phase learning for biological event extraction and verification. *ACM Transactions on Asian Language Information Processing*, 5(1):61–73.

[46] Klyne, G. and Carroll, J. J. (2004). Resource description framework (rdf): Concepts and abstract syntax.

[47] Kuba, A., A.Hócza, and Csirik, J. (2004). Pos tagging of hungarian with combined statistical and rule-based methods. In *Text, Speech and Dialogue*, volume 3206 of *Lecture Notes in Computer Science*, pages 113–120.

[48] Kundu, G., Chang, M., and Roth, D. (2011). Prior knowledge driven domain adaptation. In *ICML 2011 Workshop on Combining Learning Strategies to Reduce Label Cost*.

[49] Kushmerick, N., Weld, D. S., and Doorenbos, R. (1997). Wrapper induction for information extraction. In *Proceedings of International Joint Conferences on Artificial Intelligence*, pages 729–737, Nagoya, Japan.

[50] Lavelli, A., Califf, M. E., Ciravegna, F., Freitag, D., Giuliano, C., Kushmerick, N., Romano, L., and Ireson, N. (2008). Evaluation of machine learning-based information extraction algorithms: Criticisms and recommendations. *Language Resources and Evaluation*, 42:361–393.

[51] Li, H., Matsuo, Y., and Ishizuka, M. (2010). Semantic relation extraction based on semi-supervised learning. In *Proceedings of the 6th Asia Information Retrieval Societies Conference*, volume 6458 of *Lecture Notes in Computer Science*, pages 270–279, Taipei, Taiwan.

[52] Louis, A. L. and Engelbrecht, A. P. (2010). Unsupervised discovery of relations for analysis of textual data. *Digital Investigation*, In Press.

[53] Lutz, C. and Wolter, F. (2007). Conservative extensions in the lightweight description logic $\mathcal{EL}$. In *Proceedings of the 21st International Conference on Automated Deduction: Automated Deduction*, volume 4603 of *Lecture Notes in Artificial Intelligence*, pages 84–99, Bremen, Germany. Springer-Verlag.

[54] Ma, W. and Suel, T. (2016). Structural sentence similarity estimation for short text. In *Proceedings of the 29th International Florida Artificial Intelligence Research Society Conference*, pages 232–237, Florida, USA.

[55] Marcus, M. P., Marcinkiewicz, M. A., and Santorini, B. (1994). Building a large annotated corpus of english: The penn treebank. *Computational Linguistics*, 19(2):313–330.

[56] Maslennikov, M. and Chua, T.-S. (2010). Combining relations for information extraction from free text. *ACM Transactions on Information Systems*, 28:14:1–14:35.

[57] Masseroli, M., Kilicoglu, H., Lang, F.-M., and Rindflesch, T. C. (2006). Argument-predicate distance as a filter for enhancing precision in extracting predications on the genetic etiology of disease. *BMC Bioinformatics*, 7:291–302.

[58] McDonald, R., Pereira, F., Kulick, S., Winters, S., Jin, Y., and White, P. (2005). Simple algorithms for complex relation extraction with applications to biomedical ie. In

Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics, pages 491–498, Ann Arbor, Michigan. Association for Computational Linguistics.

[59] Meknavin, S., Charoenpornsawat, P., and Kijsirikul, B. (1997). Feature-based thai word segmentation. In *Proceedings of the Natural Language Processing Pacific Rim Symposium*, Phuket, Thailand.

[60] Miguel, D. and Roxas, R. (2007). Comparative evaluation of tagalog part of speech taggers. In *Proceedings of the 4th National Natural Language Processing Research Symposium*, pages 74–76.

[61] Miller, G. A. (1995). Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.

[62] Mittrapiyanuruk, P. and Sornlertlamvanich, V. (2000). The automatic thai sentence extraction. In *Proceedings of the 4th Symposium on Natural Language Processing*, Chiang Mai, Thailand.

[63] Miwa, M., Pyysalo, S., Hara, T., and Tsujii, J. (2010a). A comparative study of syntactic parsers for event extraction. In *Proceedings of the 2010 Workshop on Biomedical Natural Language Processing*, pages 37–45, Uppsala, Sweden. Association for Computational Linguistics.

[64] Miwa, M., Saetre, R., Kim, J.-D., and Tsujii, J. (2010b). Event extraction with complex event classification using rich features. *Journal of Bioinformatics and Computational Biology*, 8(1):131–146.

[65] Miwa, M., Sætre, R., Miyao, Y., and Tsujii, J. (2009). Proteinprotein interaction extraction by leveraging multiple kernels and parsers. *International Journal of Medical Informatics*, 78(12):e39–e46.

[66] Miyao, Y., Sagae, K., Sætre, R., Matsuzaki, T., and Tsujii, J. (2009). Evaluating contributions of natural language parsers to proteinprotein interaction extraction. *Bioinformatics*, 25(3):394–400.

[67] Mukund, S., Srihari, R., and Peterson, E. (2010). An information-extraction system for urdua resource-poor language. *ACMTransactions on Asian Language Information Processing*, 9(4):15:1–15:43.

[68] Murata, M., Ma, Q., and Isahara, H. (2002). Comparison of three machine-learning methods for thai part-of-speech tagging. *ACM Transactions on Asian Language Information Processing*, 1:145–158.

[69] Muslea, I. (1999). Extraction patterns for information extraction tasks: A survey. In *AAAI-99 Workshop on Machine Learning for Information Extraction*, pages 1–6, Orlando, Florida.

[70] Narupiyakul, L., Thomas, C., Cercone, N., and Sirinaovakul, B. (2004). Thai syllable-based information extraction using hidden markov models. In *Proceedings of the 5th International Conference on Intelligent Text Processing*, volume 2945 of *Lecture Notes in Computer Science*, pages 537–546, Seoul, Korea.

[71] Nguyen, Q. L., Tikk, D., and Leser, U. (2010). Simple tricks for improving pattern-based information extraction from the biomedical literature. *Journal of Biomedical Semantics*, 1(1).

[72] Niennattrakul, V., Leelaphattarakij, P., and Srisawat, J. (2009). Cuws: Thai word segmentation software. http://oracle.cp.eng.chula.ac.th/me/cuws.

[73] Palmer, D. D. and Hearst, M. A. (1997). Adaptive multilingual sentence boundary disambiguation. *Computational Linguistics*, 23(2):241–267.

[74] Papageorgiou, C. P. (1994). Japanese word segmentation by hidden markov model. In *Proceedings of the Human Language Technologies Workshop*, pages 283–288.

[75] Patel-Schneider, P. F., Hayes, P., and Horrocks, I. (2004). Owl web ontology language: Semantics and abstract syntax.

[76] Perez-Iratxeta, C., Bork, P., and Andrade, M. A. (2001). Xplormed: a tool for exploring medline abstracts. *TRENDS in Biochemical Sciences*, 26(9):573–575.

[77] Phuong, L. H., Huyên, N. T. M., Roussanaly, A., and Vinh, H. T. (2008). A hybrid approach to word segmentation of vietnamese texts. In *Proceedings of the 2nd International Conference on Language and Automata Theory and Applications*, volume 5196 of *Lecture Notes in Computer Science*, pages 240–249.

[78] Punyakanok, V., Roth, D., and Yih, W. (2008). The importance of syntactic parsing and inference in semantic role labeling. *Computational Linguistics*, 34(2):257–287.

[79] Reichartz, F., Korte, H., and Paass, G. (2009). Dependency tree kernels for relation extraction from natural language text. In *Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases: Part II*, volume 5782 of *Lecture Notes in Artificial Intelligence*, pages 270–285, Bled, Slovenia.

[80] Riaz, K. (2010). Rule-based named entity recognition in urdu. In *Proceedings of the 2010 Named Entities Workshop, ACL 2010*, pages 126–135, Uppsala, Sweden. Association for Computational Linguistics.

[81] Riley, M. D. (1989). Some applications of tree-based modelling to speech and language. In *Proceedings of the workshop on Speech and Natural Language*, pages 339–352, Cape Cod, Massachusetts.

[82] Schneider, G. (1998). A linguistic comparison of constituency, dependency and link grammar.

[83] Siefkes, C. and Siniakov, P. (2005). An overview and classification of adaptive approaches to information extraction. *Journal on Data Semantics*, IV:172–212.

[84] Soderland, S. (1997). [pdf] learning to extract text-based information from the world wide web. In *Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining*, pages 251–254.

[85] Soderland, S. (1999). Learning information extraction rules for semi-structured and free text. *Machine Learning*, 34(1–3):233–272.

[86] Sornlertlamvanich, V., Takahashi, N., and Isahara, H. (1999). Building a thai part-of-speech tagged corpus (orchid). *Journal Acoustical Society of Japan*, 20(3):189–198.

[87] Stevenson, M. and Greenwood, M. A. (2006). Learning information extraction patterns using wordnet. In *Proceedings of the 5th International Conference on Language Resources and Evaluations*, pages 95–102.

[88] Sukhahuta, R. and Smith, D. (2001). Information extraction strategies for thai documents. *International Journal of Computer Processing of Oriental Languages*, 14(2):153–172.

[89] Sun, Z., Lim, E.-P., Chang, K., Ong, T.-K., and Gunaratna, R. K. (2005). Event-driven document selection for terrorism information extraction. In *Intelligence and Security Informatics*, volume 3495 of *Lecture Notes in Computer Science*, pages 211–223. Springer Berlin / Heidelberg.

[90] Suwanapong, T., Theeramunkong, T., and Nantageewarawat, E. (2010). 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, volume 5990 of Lecture Notes in Computer Science*, pages 122–130, Hue City, Vietnam.

[91] Swanson, R. and Gordon, A. S. (2006). A comparison of alternative parse tree paths for labeling semantic roles. In *Proceedings of the COLING/ACL on Main conference poster sessions*, pages 811–818, Sydney, Australia. Association for Computational Linguistics.

[92] Techo, J., Nattee, C., and Theeramunkong, T. (2009). A corpus-based approach for automatic thai unknown word recognition using boosting techniques. *IEICE Transactions on Information and Systems*, E92-D(12):2321–2333.

[93] Theeramunkong, T., Iamtana-anan, P., Nattee, C., Suriyawongkul, A., Nantajeewarawat, E., and Aimmanee, P. (2007). A framework for constructing a thai medical knowledge base. In *Proceedings of the 2nd International Conference on Knowledge, Information and Creativity Support Systems*, pages 45–50, Ishikawa, Japan.

[94] Theeramunkong, T. and Tanhermhong, T. (2004). Pattern-based features vs. statistical-based features in decision trees for word segmentation. *IEICE Transactions on Information and Systems*, E87-D(5):1254–1260.

[95] Theeramunkong, T. and Usanavasin, S. (2001). Non-dictionary-based thai word segmentation using decision trees. In *Proceedings of The First International Conference on Human Language Technology Research*, pages 1–5, San Diego, USA. Association for Computational Linguistics.

[96] Thet, T. T., Na, J.-C., and Ko, W. K. (2008). Word segmentation for the myanmar language. *Journal of Information Science*, 34(5):688–704.

[97] Tian, G. and Qian, M. (2009). Research on social relation extraction of web persons. In *Proceedings of the 2009 Eighth IEEE International Conference on Dependable, Autonomic and Secure Computing*, pages 606–610, Washington, DC, USA. IEEE Computer Society.

[98] Tongchim, S., Altmeyer, R., Sornlertlamvanich, V., and Isahara, H. (2008). A dependency parser for thai. In *Proceedings of the 6th International Language Resources and Evaluation*, pages 136–139, Marrakech, Morocco.

[99] Wang, G., Yu, Y., and Zhu, H. (2007a). Pore: Positive-only relation extraction from wikipedia text. In *Proceedings of the 6th international The semantic web and 2nd Asian conference on Asian semantic web conference*, volume 4825 of *Lecture Notes in Computer Science*, pages 580–594, Busan, Korea. Springer-Verlag.

[100] Wang, T., Bontcheva, K., Li, Y., and Cunningham, H. (2005). D2.1.2 / ontology-based information extraction (obie) v.2. Technical report, SEKT: Semantically Enabled Knowledge Technologies.

[101] Wang, T., Li, Y., Bontcheva, K., Cunningham, H., and Wang, J. (2006). Automatic extraction of hierarchical relations from text. In *Proceedings of the 3rd European Semantic Web Conference*, volume 4011 of *Lecture Notes in Computer Science*, pages 215–229, Budva, Montenegro.

[102] Wang, X.-J., Liu, W., and Qin, Y. (2007b). A search-based chinese word segmentation method. In *Proceedings of the 16th international conference on World Wide Web*, pages 1129–1130, Banff, Alberta, Canada. ACM.

[103] Xue, N. (2003). Chinese word segmentation as character tagging. *Computational Linguistics and Chinese Language Processing*, 8(1):29–48.

[104] Yan, Y., Okazaki, N., Matsuo, Y., Yang, Z., and Ishizuka, M. (2009). Unsupervised relation extraction by mining wikipedia texts using information from the web. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th*

International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2, pages 1021–1029, Suntec, Singapore. Association for Computational Linguistics.

[105] Yao, Y. and Lua, K. T. (1998). Splitting-merging model of chinese word tokenization and segmentation. *Natural Language Engineering*, 4(4):309–324.

[106] Ye, J. (2011). Cosine similarity measures for intuitionistic fuzzy sets and their applications. *Mathematical and Computer Modelling*, 53:91–97.

[107] Zadeh, L. (1965). Fuzzy sets. *Information Control*, 8:338–353.

[108] Zelenko, D., Aone, C., and Richardella, A. (2003). Kernel methods for relation extraction. *Journal of Machine Learning Research*, 3:1083–1106.

[109] Zhang, J., Sun, Y., Wang, H., and He, Y. (2011). Calculating statistical similarity between sentences. *Journal of Convergence Information Technology*, 6:22–34.

[110] Zhou, D., He, Y., and Kwoh, C. K. (2007). Semi-supervised learning of the hidden vector state model for extracting protein–protein interactions. *Artificial Intelligence in Medicine*, 41:209–222.

[111] Zhou, G., Qian, L., and Fan, J. (2010). Tree kernel-based semantic relation extraction with rich syntactic and semantic information. *Information Sciences*, 180:1313–1325.

Appendix A

Examples of IE Rules

This appendix shows examples of IE rules learned using our implementation of WHISK in the experiments reported in Chapters 4 and **??**.

Table A.1 Examples of rules in the medical domain.

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)*ที่*(org)	{Obs \$2}{Attr \$1}
MD1	*(ch)*บริเวณ*(org)	{Obs \$2}{Attr \$1}
MD1	*(ch)*ตาม*(org)	{Obs \$2}{Attr \$1}
MD1	*(wst)*(vmq)*(numtime)	{Obs \$1}{Attr \$2}{Per \$3}
MD2	*(org)*ตาม*(sym)	{Sym \$2}{Loc \$1}
MD2	*(sym)*(org)	{Sym \$1}{Loc \$2}
MD2	*(sym)*บริเวณ*(org)	{Sym \$1}{Loc \$2}
MD2	*(sym)*ใน*(org)	{Sym \$1}{Loc \$2}
MD2	*(org)*มี*(sym)*(ptime)	{Sym \$2}{Loc \$1}{Per \$3}
MD2	*(sym)*(org)*นาน*(ptime)	{Sym \$1}{Loc \$2}{Per \$3}
MD2	*(sym)*(org)*เป็น*(ptime)	{Sym \$1}{Loc \$2}{Per \$3}

Table A.2 Examples of rules in the soccer match report domain.

Pattern	Output template
*(actG)*ของ*(per)*(mint)	{Act \$1}{Ply \$2}{Mint \$3}
(mint)(actG)*ของ*(per)	{Mint \$1}{Act \$2}{Ply \$3}
(mint)(per)*(actG)	{Mint \$1}{Ply \$2}{Act \$3}
(mint)(per)*หลบ*per*(actG)	{Mint \$1}{Ply \$2}{Act \$3}
*(mint)*ทำฟาวล์*(per)*(actG)	{Mint \$1}{Ply \$2}{Act \$3}
*(mint)*ให้*(per)*(actG)	{Mint \$1}{Ply \$2}{Act \$3}
(per)(actG)*(mint)	{Ply \$1}{Act \$2}{Mint \$3}

Table A.3 Examples of rules in the soccer player transfer domain.

Pattern	Output template
*(team)*คว้า*(per)*(team)*(prc)	{TEAM2 $1}{PLY $2}{TEAM1 $3}{PRICE $4}
*(per)*จาก*(team)*(team)*(prc)	{PLY $1}{TEAM1 $2}{TEAM2 $3}{PRICE $4}
(team)(prc)*(per)*(team)	{TEAM2 $1}{PRICE $2}{PLY $3}{TEAM1 $4}
*(team)*ซื้อ*(per)*(team)*(prc)	{TEAM2 $1}{PLY $2}{TEAM1 $3}{PRICE $4}
(team)(per)*(team)*(prc)	{TEAM2 $1}{PLY $2}{TEAM1 $3}{PRICE $4}
*(team)*จ่าย*(prc)*(team)*(per)	{TEAM2 $1}{PRICE $2}{TEAM1 $3}{PLY $4}
(team)(per)*(prc)*(team)	{TEAM2 $1}{PLY $2}{PRICE $3}{TEAM1 $4}
(team)(prc)*ให้*(team)*(per)	{TEAM2 $1}{PRICE $2}{TEAM1 $3}{PLY $4}
*(team)*บรรจุ*(team)*(per)*(prc)	{TEAM2 $1}{TEAM1 $2}{PLY $3}{PRICE $4}
*(team)*ขาย*(per)*(team)*(prc)	{TEAM1 $1}{PLY $2}{TEAM2 $3}{PRICE $4}
(team)(per)*ให้*(team)*(prc)	{TEAM1 $1}{PLY $2}{TEAM2 $3}{PRICE $4}
(per)(team)*(team)*(prc)	{PLY $1}{TEAM1 $2}{TEAM2 $3}{PRICE $4}
(team)(prc)*ของ*(team)*(per)	{TEAM1 $1}{PRICE $2}{TEAM2 $3}{PLY $4}
*(team)*หัน*(per)*(prc)*(team)	{TEAM2 $1}{PLY $2}{PRICE $3}{TEAM1 $4}
*(per)*ย้าย*(team)*(prc)*(team)	{PLY $1}{TEAM1 $2}{PRICE $3}{TEAM2 $4}
*(team)*ชิว*(per)*(team)*(prc)	{TEAM2 $1}{PLY $2}{TEAM1 $3}{PRICE $4}
(team)(team)*(prc)*(per)	{TEAM1 $1}{TEAM2 $2}{PRICE $3}{PLY $4}
*(per)*เป็นสมาชิก*(team)*(prc)	{PLY $1}{TEAM2 $2}{TEAM1 $3}{PRICE $4}
*(team)*ยอมรับ*(per)*(team)*(prc)	{TEAM1 $1}{PLY $2}{TEAM2 $3}{PRICE $4}
(per)(team)*ทุ่ม*(prc)*(team)	{PLY $1}{TEAM2 $2}{PRICE $3}{TEAM1 $4}

Table A.4 Examples of rules in the stock price domain.

Pattern	Output template
(com)(prc)	{COM $1}{PRC $2}
*(com)*ที่*(prc)ราคา	{COM $1}{PRC $2}
*(com)*ปรับ*ปิด*(prc)	{COM $1}{PRC $2}
*(com)*ปิด*ที่*(prc)	{COM $1}{PRC $2}
*(com)*หุ้น*ไป*(prc)*(prc)	{COM $1}{DIF $2}{PRC $3}
*(com)*ปิด*(prc)*(prc)	{COM $1}{PRC $2}{DIF $3}
*(com)*ปิด*(prc)*(prc)	{COM $1}{PRC $2}{DIF $3}
(com)(prc)*(prc)	{COM $1}{DIF $2}{PRC $3}
(com)(prc)*ที่*(prc)	{COM $1}{DIF $2}{PRC $3}
*(com)*จะมา*(prc)*(prc)	{COM $1}{PRC $2}{DIF $3}
*(com)*ที่*(prc)*(prc)	{COM $1}{PRC $2}{DIF $3}

Table A.5 Examples of rules in of the company dividend domain.

Pattern	Output template
(com)(prc)	{COM $1}{DIV $2}
*(com)*ปันผล*(prc)	{COM $1}{DIV $2}
*(com)*หุ้น*(prc)	{COM $1}{DIV $2}
(com)(prc)*(prc-m)	{COM $1}{DIV $2}{PRICE $3}
(com)(prc-m)*หุ้น*(prc)	{COM $1}{PRICE $2}{DIV $3}

Table A.6 Examples of rules in the chemical reaction domain.

Pattern	Output template
ผลพลอยได้(rac)* คือ*(sub)	{RNM $1}{PDT $2}
(sub) เพื่อ*ตัวเร่งปฏิกิริยา*(rac)	{CAT $1}{RNM $2}
การทำปฏิกิริยา(sub)* ได้*(sub)	{RCT $1}{PDT $2}
ปฏิกิริยา(sub)* กับ*(sub)	{RCT $1}{RCT $2}
(sub) เร่ง*ปฏิกิริยา*(sub)* ได้*(sub)	{CAT $1}{RCT $2}{PDT $3}
(sub) เร่ง*ปฏิกิริยา*(sub)* เป็น*(sub)	{CAT $1}{RCT $2}{PDT $3}
ตัวเร่งปฏิกิริยา(sub)*(rac)* ของ*(sub)	{CAT $1}{RNM $2}{RCT $3}
ตัวเร่งปฏิกิริยา(sub)* บน*(rac)* ของ*(sub)	{CAT $1}{RNM $2}{RCT $3}
การทำปฏิกิริยา(sub)* กับ*(sub)* ให้เป็น*(sub)	{RCT $1}{RCT $2}{PDT $3}
(rac)(sub)* ใช้*(sub)* ตัวเร่งปฏิกิริยา	{RNM $1}{RCT $2}{CAT $3}
(rac) ของ*(sub)* เป็น*(sub)	{RNM $1}{RCT $2}{PDT $3}
(rac) ของ*(sub)* ได้*(sub)	{RNM $1}{RCT $2}{PDT $3}
(rac)(sub)* กับ*(sub)	{RNM $1}{RCT $2}{RCT $3}
(rac) ของ*(sub)*(sub)* ริเริ่ม	{RNM $1}{RCT $2}{RCT $3}
(sub) จาก*(sub)* กับ*(sub)	{PDT $1}{RCT $2}{RCT $3}
(sub) เตรียม*(sub)* กับ*(sub)	{PDT $1}{RCT $2}{RCT $3}
(sub) เตรียม*(rac)* ของ*(sub)	{PDT $1}{RNM $2}{RCT $3}
ปฏิกิริยา(sub)*(sub)* ให้เป็น*(sub)	{RCT $1}{RCT $2}{PDT $3}
(sub) ทำปฏิกิริยา*(sub)*กลายเป็น*(sub)	{RCT $1}{RCT $2}{PDT $3}
(sub) ทำปฏิกิริยา*(rac)* กับ*(sub)	{RCT $1}{RNM $2}{RCT $3}
(rac)(sub)* ตัวเร่งปฏิกิริยา*(sub)	{RNM $1}{RCT $2}{CAT $3}
(rac)(sub)* ผลิตภัณฑ์*(sub)	{RNM $1}{RCT $2}{PDT $3}
(rac)(sub)* กับ*(sub)* ได้*(sub)	{RNM $1}{RCT $2}{RCT $3}{PDT $4}
ตัวเร่งปฏิกิริยา(sub)*(rac)*(sub)* ไป*เป็น*(sub)	{CAT $1}{RNM $2}{RCT $3}{PDT $4}
(rac)(sub)* เป็น*(sub)* ตัวเร่งปฏิกิริยา*(sub)	{RNM $1}{RCT $2}{PDT $3}{CAT $4}
ผลิตภัณฑ์(rac)*(sub)* คือ*(sub)*(sub)	{RNM $1}{RCT $2}{PDT $3}{PDT $4}
(rac)(sub)* กับ*(sub)* ใช้*(sub)* ตัวเร่งปฏิกิริยา	{RNM $1}{RCT $2}{RCT $3}{CAT $4}
(rac)(sub)* กับ*(sub)* ใช้*(emz)* ตัวเร่งปฏิกิริยา	{RNM $1}{RCT $2}{RCT $3}{CAT $4}
(rac) ด้วย*(sub)* เปลี่ยน*(sub)* เป็น*(sub)	{RNM $1}{RCT $2}{RCT $3}{PDT $4}
(rac)(sub)* กับ*(sub)* เป็น*(sub)	{RNM $1}{RCT $2}{RCT $3}{PDT $4}

Appendix B

List of Publications

International Journals

- Intarapaiboon, P. and Theeramunkong, T. (2018): An Application of Intuitionistic Fuzzy Sets to Improve Information Extraction from Thai Unstructured Text. IEICE Transactions on Information and Systems. vol. E101-D(9), 2334–2345.

Lecture Note

- Intarapaiboon, P. and Theeramunkong, T.: An Improvement of Pattern-Based Information Extraction Using Intuitionistic Fuzzy Sets. In: The 10th Multi-disciplinary International Workshop on Artificial Intelligence (MIWAI2016). LNAI Vol. 10053. Springer-Verlag (2016). pp.63–75.

An Improvement of Pattern-Based Information Extraction Using Intuitionistic Fuzzy Sets

Peerasak Intarapaiboon[1(✉)] and Thanaruk Theeramunkong[2]

[1] Deportment of Mathematics and Statistics, Faculty of Science and Technology, Thammasat University, Pathum Thani, Thailand
peerasak@mathstat.tu.ac.th
[2] School of Information and Computer Technology, Sirindhorn International Institute of Technology, Thammasat University, Pathum Thani, Thailand
thanaruk@siit.tu.ac.th

Abstract. Multi-slot information extraction (IE) 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 unknowingness 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 the obstacle, but it raises another problem that is incorrect (unwanted) extractions. This paper aims to present a method for removing incorrect extractions. In the method, extractions are represented as intuitionistic fuzzy sets (IFSs), 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 of not. From the preliminary experiment on a medical domain, the proposed technique improves extraction precision while satisfactorily preserving recall.

1 Introduction

Information extraction (IE) is a process of identifying and extracting desired pieces of information. Multi-slot IE is a special task of IE that extract related pieces of information simultaneously and connecting them in a form of multiple-field relational records. Most IE systems usually involve rule-based approaches, which an IE rule is often represented in terms of a regular expression, e.g., WI [1], CRYSTAL [2], LIEP [3], and WHISK [4]. 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. As such, several IE frameworks come up with components to alleviate the detriment suffered by the aforementioned issue. One approach to overcome the problem is removing inefficient rules [5,6]. An alternative approach uses the all IE rules and then eliminates unwanted extractions [7–10].

C. Sombattheera et al. (Eds.): MIWAI 2016, LNAI 10053, pp. 63–75, 2016.
DOI: 10.1007/978-3-319-49397-8_6

Recently, intuitionistic fuzzy set (IFS) [11] has been much explored in both theory and application. Differing from representation of a fuzzy set (FS) [12], 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 and Chuntian [13] gave the axiomatic definition of similarity measures between IFSs, various similarity measures have been proposed continuously [14–19]. One of most applications of IFS similarity measures is classification problems. Khatibi and Montazer [18] conducted experiments for bacterial classification using similarity measures for FSs and IFSs. The results indicated that each measure for IFSs outperformed that for FSs. Ye [19] cosine and weighted cosine similarity measures for IFSs were proposed and applied to a small medical diagnosis problem.

By the success of research in IFS, especially similarity measurement, 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) how to represent an extracted frame in terms of an IFS and (ii) how to apply 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 [7], 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, Intarapaiboon [8] proposed an pattern-based IE framework to extract multi-slot frames. 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 [9], 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

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

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

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.

3.2 IE Rules and Rule Application

A well-known supervised rule learning algorithm, called WHISK [Sodeland, 1999], 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.

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

Fig. 4. An IE rule example

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 k-word portion of an information entry one-by-one sequentially, where the window size, k, 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. Table 1 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.

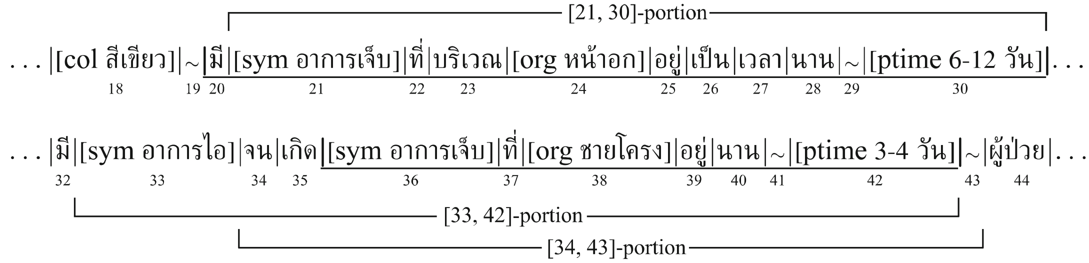

Fig. 5. Text portions from which extractions are made when the rule in Fig. 4 is applied to the information entry in Fig. 1 using a 10-word sliding window

Table 1. Frames extracted from the text portions in Fig. 5 by the rule in Fig. 4

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

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 [11] 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

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

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

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

Definition 1 [15]. *A similarity measure S for $IFS(X)$ is a real function $S : IFS(X) \times IFS(X) \to [0,1]$, which satisfies the following properties:*

P1: $0 \leq S(A,B) \leq 1$,
P2: $S(A,B) = S(B,A), \forall A, B \in IFS(X)$,

Table 2. Some similarity measures between IFSs.

Author	Expression
Dengfeng and Chuntian [13]	$S_d^p(A,B) = 1 - \frac{1}{\sqrt[p]{h}} \sqrt[p]{\sum_{i=1}^{h} \lvert\varphi_A(i) - \varphi_B(i)\rvert^p}$ where $\varphi_k(i) = (\mu_k(x_i) + 1 - \nu_k(x_i))/2, k = \{A, B\}$, and $p = 1, 2, 3, \ldots$
Mitchell [15]	$S_m^p(A,B) = \frac{1}{2}\left(\rho_\mu(A,B) + \rho_f(A,B)\right)$ 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 [19]	$S_C(A,B) = \frac{1}{h}\sum_{i=1}^{h} \frac{\mu_A(x_i)\mu_B(x_i)+\nu_A(x_i)\nu_B(x_i)}{\sqrt{\mu_A^2(x_i)+\nu_A^2(x_i)}\sqrt{\mu_B^2(x_i)+\nu_B^2(x_i)}}$

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 2 highlights some similarity measures between IFSs.

5 The Proposed Technique—*IFS-Based Extraction Filtering*

As the example shown in Sect. 3, RAW probably produces false extractions. Hence, to improve the extraction accuracy, a method for removing unwanted extractions is necessary. The idea behind our method for removing incorrect extractions is based on the fact that if an internal wildcard[1] of a rule is instantiated across a target-phrase boundary, then an incorrect extraction is made. Predicting whether an internal wildcard is instantiated across a target-phrase boundary can be regarded as a binary classification problem.

In our technique, an intuitionistic fuzzy set will be generated for each extracted frame. Like k-NN, to determine whether an extraction E is correct or not, a majority vote among the k nearest neighbors of the IFS corresponding to E is applied, where a distance is calculated by an IFS similarity measure. Given an IE rule r with n internal wildcards, the precise steps of the proposed method are detailed as follows:

5.1 Preprocessing

Vector-Based Document Representation.

(a1) Apply the rule r into all information entries in the training corpus, whence semantic frames are obtained.

[1] A wildcard occurs between the first and the last slot markers of a rule, called an *internal wildcard*.

(a2) For each internal wildcard, observe plain words in which the wildcard instantiates during Step (a1). These words are separated to 2 sets: one containing different words only when correct extractions are made; and the other containing those only when incorrect ones are made. For convenience, W^k_{cor} and W^k_{inc} are referred to the former set and the latter set, respectively, of the k-th internal wildcard.

(a3) Construct a feature vector corresponding to each extracted frame. Denoted by

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

a feature vector observed when the i-th frame is extracted where $\boldsymbol{v}_i^k$ is a 4-dimensional feature vector corresponding to the instantiation of the k-th internal wildcard in the rule pattern, and '$\parallel$' refers to vector concatenation. A feature vector of k-th internal wildcard is defined as:

$$\boldsymbol{v}_i^k = [f^k_{i,1}, f^k_{i,2}, f^k_{i,3}, f^k_{i,4}],$$

where $f^k_{i,1}$, $f^k_{i,2}$, $f^k_{i,3}$, and $f^k_{i,4}$ are the length of tokens[2], the number of spaces, the number of plain words in W^k_{cor}, and the number of plain words in W^k_{inc} observed from the text portion into which the wildcard is instantiated.

IFS-Based Document Representation. To convert a feature vector for an extraction to an IFS, we propose one method which its conceptual idea is explained as follows: Suppose $A_i = \{\langle HF^k_j, \mu_i(HF^k_j), \nu_i(HF^k_j)\rangle, \}$ is an IFS for the vector V_i, when j and k are indexes for feature types and internal wildcards, respectively. In this work, $\mu_i(HF^k_j)$ presents a confidential level to say that $f^k_{i,j}$ 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, k, in the other feature vectors. In contrast, $\nu_i(HF^k_j)$ does a confidential level to say that $f^k_{i,j}$ in the i-th feature vector is not relatively high. The next example gives more details.

Example 1. Assume a considered rule has two internal wildcard and there are three extractions made by the rule. Let the feature vectors for these extractions be

$$\boldsymbol{V}_1 = [5,2,1,3,1,1,0,0],\ \ \boldsymbol{V}_2 = [1,0,1,0,1,0,1,0],\ \ \boldsymbol{V}_3 = [2,1,0,1,3,1,0,1].$$

To interpret this situation, for the first extraction, the first internal wildcard instantiates into a five-token-long text portion in which two tokens are white spaces, one token is in W^1_{cor}, three tokens are in W^1_{inc}. It is worthy to note that the other token in the portion is in either $W^1_{cor} \cap W^1_{inc}$ or $(W^1_{cor} \cup W^1_{inc})^c$. Since $f^1_{1,1} > f^1_{3,1} > f^1_{2,1}$, the confidential level to say that the first internal wildcard matches with a longer text portion for the first extraction than those for the rest extractions. Hence, $\mu_1(HF^1_1) > \mu_3(HF^1_1) > \mu_2(HF^1_1)$ and $\nu_1(HF^1_1) < \nu_3(HF^1_1) < \nu_2(HF^1_1)$. □

[2] A token might be a word, a white space, or a symbol.

Based on the idea discussed above, the process of transformation will be formally explained. Given the universe of discourse

$$X = \{HF_1^1, HF_2^1, HF_3^1, HF_4^1 \ldots, HF_1^n, HF_2^n, HF_3^n, HF_4^n\}.$$

Every value $f_{i,j}^k$ in the vector-based representation of the i-th extraction is then converted in terms of the three degrees of HF_j^k as the following steps:

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

$$z_{i,j}^k = \frac{f_{i,j}^k - \overline{X}_j^k}{s_j^k}, \tag{6}$$

where $\overline{X}_j^k$ and s_j^k are the mean and the standard deviation, respectively, of the feature type j for the wildcard k over extractions. More precisely, if E is the set of extractions made by the r,

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

and

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

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

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

where $0 < r_j^k \leq 1$ is a weight for HF_j.

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

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

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

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

$$\pi_i(HF_j^k) = 1 - \mu_i(HF_j^k) - \nu_i(HF_j^k).$$

Example 2. Assume a considered rule has two internal wildcard and there are only three extractions made, i.e., $\boldsymbol{V_1}$, $\boldsymbol{V_2}$, and $\boldsymbol{V_3}$ as shown in Example 1. For convenience, the extractions are gathered and represented in terms of the matrix as below:

$$E = \begin{bmatrix} 5 & 2 & 1 & 3 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 2 & 1 & 0 & 1 & 3 & 1 & 0 & 1 \end{bmatrix}.$$

Next, we compute the mean and the standard deviation for each feature type of each wildcard, then the results are presented as the row matrices M and SD:

$$M = \begin{bmatrix}2.67\ 1.00\ 0.67\ 1.33\ 1.33\ 0.67\ 0.33\ 0.33\end{bmatrix},$$
$$SD = \begin{bmatrix}2.08\ 1.00\ 0.58\ 1.53\ 1.53\ 0.58\ 0.58\ 0.58\end{bmatrix}.$$

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 containing the normalizing values:

$$Z = \begin{bmatrix} 1.12 & 1.00 & 0.58 & 1.09 & -0.58 & 0.58 & -0.58 & -0.58 \\ -0.80 & -1.00 & 0.58 & -0.87 & -0.58 & -1.15 & 1.15 & -0.58 \\ -0.32 & 0.00 & -1.15 & -0.22 & 1.15 & 0.58 & -0.58 & 1.15 \end{bmatrix}.$$

Suppose that the weights r_j^k and $\bar{r}_j^k$ 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 following two matrices, respectively:

$$D_\mu = \begin{bmatrix} 0.60 & 0.58 & 0.51 & 0.60 & 0.29 & 0.51 & 0.29 & 0.29 \\ 0.25 & 0.22 & 0.51 & 0.24 & 0.29 & 0.19 & 0.61 & 0.29 \\ 0.34 & 0.40 & 0.19 & 0.36 & 0.61 & 0.51 & 0.29 & 0.61 \end{bmatrix},$$
$$D_\nu = \begin{bmatrix} 0.22 & 0.24 & 0.32 & 0.23 & 0.58 & 0.32 & 0.58 & 0.58 \\ 0.62 & 0.66 & 0.32 & 0.63 & 0.58 & 0.68 & 0.22 & 0.58 \\ 0.52 & 0.45 & 0.68 & 0.50 & 0.22 & 0.32 & 0.58 & 0.22 \end{bmatrix}.$$

Finally, we can convert the feature vectors $\boldsymbol{V_1}$, $\boldsymbol{V_2}$, and $\boldsymbol{V_3}$ 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 $\boldsymbol{V_1}$:

$$\begin{aligned} IFS_1 = \{&\langle HF_1^1, 0.60, 0.22\rangle\langle HF_2^1, 0.58, 0.24\rangle, \langle HF_3^1, 0.51, 0.32\rangle, \langle HF_4^1, 0.60, 0.23\rangle, \\ &\langle HF_1^2, 0.29, 0.58\rangle, \langle HF_2^2, 0.51, 0.32\rangle, \langle HF_3^2, 0.29, 0.58\rangle, \langle HF_4^2, 0.29, 0.58\rangle\}. \end{aligned}$$

□

5.2 Extraction Classification

Recalling again that E 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, …, $IFS_{|E|}$.

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 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 nearest to IFS_t, where a distance is measured by an IFS similarity measure.

6 Experimental Results

6.1 Data Sets and Output Templates

Information Entries. We constructed the corpus by gathering medicinal and pharmaceutical web sites from 2759 URLs. The obtained data covers 474 diseases and 770 medicinal chemical substances, with approximately 6600 and 3350 information entries, respectively. Disease information entries were divided into 3 data sets, i.e., D1, D2, and D3, based on their disease groups. D1 comprises distinct 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 diseases. The collected information entries were preprocessed using a word segmentation program, called CTTEX, developed by NECTEC, and were then partially annotated with semantic class tags using a predefined ontology lexicon. Table 3 summarizes the characteristics of the three data sets. The second column shows the number of information entries in each data set. It is followed by a column group showing the maximum number, the average number, and the minimum number of words per information entry in each data set. The last two column groups of this table characterize the three data sets in terms of the number of symptom phrases and their occurrences.

Table 3. Data set characteristics

Data set	No. of info. entries	No. of words per info. entry			No. of distinct symptom phrases		No. of symptom phrase occurrences	
		Max.	Avg.	Min.	MD1	MD2	MD1	MD2
D1	59	130	44	9	179	77	213	84
D2	56	146	45	7	136	66	160	69
D3	58	140	55	8	161	65	210	73

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

Table 4. 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 T
MD2	{SYM S}{LOC P}{PER T}	A primitive named symptom S appears at a human-body part P for a time period T

Table 5. 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

as *MD1* and *MD2*, are considered in our experiments. Table 4 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. Table 5 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.

6.2 Experimental Results

D1 was used as training set. All MD1 and MD2 symptom phrases occurring in D1 were manually tagged with desired output frames and were used for rule learning. The length of the longest symptom phrase observed when a rule yields correct extractions on the training set is taken as the window size for the rule. By applying the obtained rules to the information entries in D1 using RAW, an IFS-based representation for each extraction was constructed when r_j^k and $\bar{r}_j^k$ in Eqs. (9) and (10) were set based on statistical characteristics of the corresponding wildcard instantiation by:

$$r_j^k = \bar{r}_j^k = \left| \frac{1 - s_j^k}{1 + s_j^k} \right|,$$

where s_j^k is the standard deviation for the feature type j of the wildcard k (see Eq. (8)).

The proposed framework was evaluated on D2 and D3. 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 the proportion of correct extractions to all obtained extractions. Table 6 shows the evaluation results obtained from using RAW without any extraction filtering and RAW with the proposed filtering method using the similarity measures in Table 2, i.e., $\text{RAW} + S_d^p$, $\text{RAW} + S_m^p$, and $\text{RAW} + S_C$. In the table, 'R' and 'P' stand for recall and precision, which are given in percentage. Compared to the results obtained using RAW alone, regardless of which similarity measure is used, the IFS-based

Table 6. Evaluation results

Type	Data set	RAW		RAW + S_d^p		RAW + S_m^p		RAW + S_C	
		R	P	R	P	R	P	R	P
MD1	D2	88.1	60.3	88.1	93.4	86.3	93.9	86.9	97.9
	D3	89.0	55.3	89.0	93.0	88.6	94.9	88.6	96.9
MD2	D2	100.0	37.5	98.6	86.1	98.6	84.4	97.1	86.4
	D3	98.6	31.9	98.6	83.2	94.5	83.6	97.3	87.6

filtering module improves precision while satisfactorily preserving recall for all template types and all test sets. Among the three measures, it is clear that S_C outperforms the others. On further analysis, we found that the precision values for MD2 are significantly lower than those for MD1 because the variety of the structures for the MD2 template is more than that for the MD1 template. There are two optional slots for MD2, but only one for MD1, see their descriptions in Sect. 6.1.

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. An IFS-based filtering technique is proposed for removal of false positives resulting from rule application across target-phrase boundaries. 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.

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

References

1. Kushmerick, N., Weld, D.S., Doorenbos, R.: Wrapper induction for information extraction. In: Proceedings of International Joint Conferences on Artificial Intelligence, Nagoya, Japan, pp. 729–737 (1997)
2. Soderland, S.: Learning to extract text-based information from the world wide web. In: Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining, pp. 251–254 (1997)
3. Huffman, S.B.: Learning information extraction patterns from examples. In: Wermter, S., Riloff, E., Scheler, G. (eds.) IJCAI 1995. LNCS, vol. 1040, pp. 246–260. Springer, Heidelberg (1996). doi:10.1007/3-540-60925-3_51
4. Soderland, S.: Learning information extraction rules for semi-structured and free text. Mach. Learn. **34**(1–3), 233–272 (1999)

5. Nguyen, Q.L., Tikk, D., Leser, U.: Simple tricks for improving pattern-based information extraction from the biomedical literature. J. Biomed. Semant. **1**(9), 1–17 (2010)
6. Liua, Q., Gaoa, Z., Liuc, B., Zhang, Y.: Automated rule selection for opinion target extraction. Knowl.-Based Syst. **104**, 74–88 (2016)
7. Kim, E., Song, Y., Lee, C., Kim, K., Lee, G., Yi, B.-K.: Two-phase learning for biological event extraction and verification. ACM T. Asian Lang. Inf. Process. **5**(1), 61–73 (2006)
8. Intarapaiboon, P., Nantajeewarawat, E., Theeramunkong, T.: Extracting semantic frames from Thai medical-symptom unstructured text with unknown target-phrase boundaries. IEICE Trans. Inf. Syst. **E94.D**(3), 465–478 (2012)
9. Spasić, I., Sarafraz, F., Keane, J.A., Nenadic, G.: Medication information extraction with linguistic pattern matching and semantic rules. J. Am. Med. Inform. Assoc. **17**(5), 532–535 (2010)
10. Zhang, J., El-Gohary, N.: Semantic NLP-based information extraction from construction regulatory documents for automated compliance checking. J. Comput. Civ. Eng. **30**(2), 1–14 (2014)
11. Atanassov, K.: Intuitionistic fuzzy sets. Fuzzy Set Syst. **20**, 87–96 (1986)
12. Zadeh, L.A.: Fuzzy sets. Inf. Control **8**, 338–353 (1965)
13. Dengfeng, L., Chuntian, C.: New similarity measures of intuitionistic fuzzy sets and application to pattern recognition. Pattern Recogn. Lett. **23**, 221–225 (2002)
14. Liang, Z., Shi, P.: Similarity measures on intuitionistic fuzzy sets. Pattern Recogn. Lett. **24**, 2687–2693 (2003)
15. Mitchell, H.B.: On the Dengfeng-Chuntian similarity measure and its application to pattern recognition. Pattern Recogn. Lett. **24**, 3101–3104 (2003)
16. Hung, W.-L., Yang, M.-S.: Similarity measures of intuitionistic fuzzy sets based on Hausdorff distance. Pattern Recogn. Lett. **25**, 1603–1611 (2004)
17. Xu, Z.: Some similarity measures of intuitionistic fuzzy sets and their applications to multiple attribute decision making. Fuzzy Optim. Decis. Making **6**, 109–121 (2007)
18. Khatibi, V., Montazer, G.A.: Intuitionistic fuzzy set vs. fuzzy set application in medical pattern recognition. Artif. Intell. Med. **47**, 43–52 (2009)
19. Ye, J.: Cosine similarity measures for intuitionistic fuzzy sets and their applications. Math. Comput. Model. **53**, 91–97 (2011)