PAPER Special Section on Knowledge Discovery, Data Mining and Creativity Support System

Extracting Chemical Reactions from Thai Text for Semantics-Based Information Retrieval

Peerasak INTARAPAIBOON[†a)], ***Student Member***, **Ekawit NANTAJEEWARAWAT**[†b)], ***and*** **Thanaruk THEERAMUNKONG**[†c)], ***Members***

SUMMARY Based on sliding-window rule application and extraction filtering, we present a framework for extracting multi-slot frames describing chemical reactions from Thai free text with unknown target-phrase boundaries. A supervised rule learning algorithm is employed for automatic construction of pattern-based extraction rules from hand-tagged training phrases. A filtering method is devised for removal of incorrect extraction results based on features observed from text portions appearing between adjacent slot fillers in source documents. Extracted reaction frames are represented as concept expressions in description logics and are used as metadata for document indexing. A document knowledge base supporting semantics-based information retrieval is constructed by integrating document metadata with domain-specific ontologies.

key words: *information extraction, semantics-based information retrieval, ontology, description logics, automated reasoning*

1. Introduction

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 information extraction (IE) technology and recent development of machine-processable ontology languages, such as OWL [1], will contribute significantly to realization of such semantics-based information retrieval.

In this paper, we present a framework for extracting multi-slot frames describing chemical reactions from chemistry thesis abstracts written in Thai. From input thesis abstracts, partially annotated with entity classes in a preprocessing phase, extractions are made based on inductively learned patterns of triggering entity tags and triggering plain words. A well-known supervised rule learning algorithm, called WHISK [11], is used as the core algorithm for constructing extraction rules.

Pattern-based IE rules do not have ability to automatically segment input documents so that they can be applied only to 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 [4] and the notion of a sentence is unclear [2]. To apply IE rules without predetermining the boundaries of sentences and potential target phrases, rule application using sliding windows (RAW) is introduced. Using sliding windows, IE rules are often instantiated across or outside the boundaries of target text portions and, therefore, tend to make many false positive extractions. A filtering module is proposed for removal of incorrect slots in an extracted frame based on features observed from text portions appearing between adjacent slot fillers in their source document.

Extracted frames 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.

The remainder of the paper proceeds as follows: Section 2 describes our IE framework. Section 3 presents IE experiments. Section 4 explains construction of a document knowledge base and demonstrates semantics-based document retrieval. Section 5 discusses related works. Section 6 concludes the paper.

2. Extracting Reaction Frames: Framework

Our target phrase is a chemical-reaction description containing at least two of the following components: reaction name, reaction product(s), reactant(s), and catalyst(s). A framework for extracting chemical-reaction frames from Thai free-text thesis abstracts, outlined in Fig. 1, is described below.

Manuscript received June 2, 2010.
Manuscript revised October 1, 2010.

[†]The authors are with the School of Information and Computer Technology, Sirindhorn International Institute of Technology, Thammasat University, Thailand.
a) E-mail: ipeerasak@siit.tu.ac.th
b) E-mail: ekawit@siit.tu.ac.th
c) E-mail: thanaruk@siit.tu.ac.th

DOI: 10.1587/transinf.E94.D.479

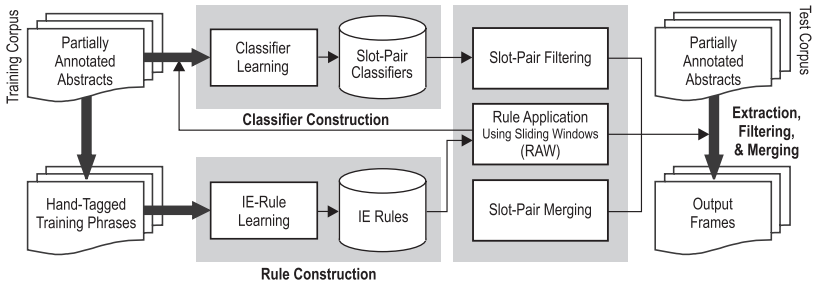

Fig. 1 An overview of the presented IE framework.

. . . |จาก|การทดลอง|∼|ผลิตภัณฑ์|หลัก|ที่|ได้|จาก|ปฏิกิริยา|∼|[rac ออกซิเดชัน]|∼|ของ|∼|[sub เอทานอล]|
∼|คือ|∼|[sub อะเซทาดีไฮด์]|∼|กับ|∼|[sub คาร์บอนไดออกไซด์]|∼|นอกจากนั้น|[sub โพรพิโอนาลดีไฮด์]|∼|
จะ|ได้|จาก|ปฏิกิริยา|∼|[rac ออกซิเดชัน]|∼|ของ|∼|[sub 1-โพรพานอล]|∼|ผล|การทดลอง|แสดง|ว่า|. . .

Fig. 2 A portion of a partially annotated word-segmented abstract.

. . . *From the experiment, the main products obtained from the* [rac *oxidation*] *reaction of* [sub *ethanol*]
are [sub *acetaldehyde*] *and* [sub *carbon dioxide*]. *Moreover,* [sub *propionaldehyde*] *is obtained from the*
[rac *oxidation*] *reaction of* [sub *1-propanol*]. *The experimental results show that* . . .

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

Extracted frame: {RNM [rac ออกซิเดชัน]}{PDT [sub โพรพิโอนาลดีไฮด์]}{RCT [sub 1-โพรพานอล]}
English translation: {RNM [rac *oxidation*]}{PDT [sub *propionaldehyde*]}{RCT [sub *1-propanol*]}

Fig. 4 A frame extracted from the second target phrase in Fig. 2.

Pattern: *(sub)*ได้*จาก*(rac)*ของ*(sub)
Output template: {RNM $2}{PDT $1}{RCT $3}

Fig. 5 An IE rule example.

2.1 Preprocessing, Extracted Frames, and IE Rules

Word segmentation is applied to all collected abstracts as part of a preprocessing step. Predefined lexicons of chemical reaction names and chemical substances are then employed to partially annotate word-segmented text with entity tags. Figure 2 illustrates a portion of an obtained word-segmented and partially annotated abstract, where '|' indicates a word boundary, '∼' signifies a space, and the tags "rac" and "sub" denote "reaction name" and "substance," respectively. The portion contains two target phrases, which are underlined in the figure. Figure 3 provides a literal English translation of this abstract portion; translations of the two target phrases are also underlined. Figure 4 shows the frame required to be extracted from the second target phrase in Fig. 2. It contains three slots with the role names RNM, PDT, and RCT, which stand for "reaction name," "product," and "reactant," respectively. Figure 5 gives a typical example of an IE rule. Its pattern part contains three triggering class tags, three triggering plain words, and six instantiation wildcards. The three triggering class tags also serve as *slot markers*—the terms into which they are instantiated are taken as fillers of their respective slots in the resulting extracted frame. When instantiated into the second target phrase in Fig. 2, this rule yields the frame in Fig. 4.

2.2 Rule Learning and Rule Application Using Sliding Windows

WHISK [11] uses a covering algorithm to construct a set of multi-slot extraction rules. It takes a corpus of training instances that are hand-tagged with desired extraction outputs to guide rule creation. The algorithm induces rules top-down, starting from the most general rule that covers all training instances, and then specializing the initial rule by adding triggering terms one at a time in order to prevent rule application with incorrect extractions. Reasons for selecting WHISK include not only its previous success in English-text IE applications, but also its capability to generate multi-slot extraction rules, which enable extracted slots to be semantically connected, e.g., reactants and products in a reaction. Other rule learning algorithms with performance comparable to WHISK, e.g., RAPIER [3] and SRV [5], can generate

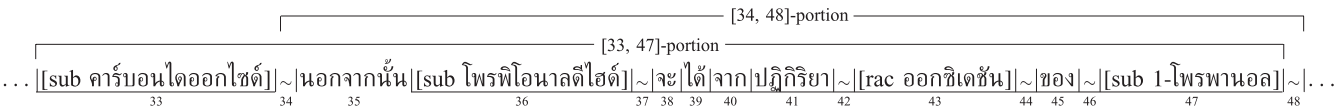

Fig. 6 Text portions from which extractions are made when the rule in Fig. 5 is applied to the text string in Fig. 2 using a 15-word sliding window.

Table 1 Frames extracted from the text portions in Fig. 6 by the rule in Fig. 5.

Portion	Extracted frame	Correctness
[33, 47]	{RNM [rac ออกซิเดชัน]}{PDT [sub คาร์บอนไดออกไซด์]}{RCT [sub 1-โพรพานอล]}	Incorrect
[34, 48]	{RNM [rac ออกซิเดชัน]}{PDT [sub โพรพิโอนาลดีไฮด์]}{RCT [sub 1-โพรพานอล]}	Correct

only single-slot (individual-field) extraction rules and do not suit our requirements.

WHISK rules are usually applied to individual sentences. In the Thai writing system, however, the end point of a sentence is usually not specified [4]. To apply IE rules to free text with unknown boundaries of sentences and potential target text portions, *rule application using sliding windows (RAW)* is introduced. Using a k-word sliding window, a rule r is applied to each k-word portion of a document one-by-one sequentially. More precisely, assume that a document d consisting of n words is given and that for any l, m such that $1 \leq l \leq m \leq n$, the $[l, m]$-*portion* of d is the portion beginning at the lth word position and ending at the mth word position of d. Then r is applied to the $[i, i+k-1]$-portion of d for each i such that $1 \leq i \leq n-k+1$. Using RAW, IE rules are often instantiated across or outside the boundaries of target text portions and an effective extraction filtering method is necessary.

When a WHISK rule is applied, triggering class tags and words in its pattern part match those appearing in a target text portion one-by-one from left to right and an extraction is made from the instantiated slot markers determined by the first successful matching. Figure 6 illustrates the application of the rule in Fig. 5 to the text portion in Fig. 2 using a 15-word sliding window, under the assumption that the first word appearing in Fig. 6 is the 33rd word in the abstract being processed. Extractions are made from the [33, 47]-portion and the [34, 48]-portion of the abstract. The resulting frames are shown in Table 1. When the rule is applied to the [33, 47]-portion, the slot filler taken through the first slot marker of the rule, i.e., "sub," does not belong to the reaction phrase containing the fillers taken through the second and the third slot markers of it, i.e., "rac" and "sub," whence an incorrect extraction occurs. Using RAW, rules are often instantiated across or outside the boundaries of target text portions and an effective extraction filtering method is necessary.

2.3 Extraction Filtering

Our proposed method for filtering out incorrect extractions will now be described. In the rest of this section, suppose that d is a document (i.e., a thesis abstract), $\textsc{Req}(d)$ is the set of all frames required to be extracted from d, i.e., the set of frames used for judging the correctness of extractions made from d, and $\textsc{Raw}(d)$ is the set of all frames extracted from d by using RAW. Given a frame $f \in \textsc{Req}(d) \cup \textsc{Raw}(d)$, let $slot(f)$ denote the set of all slots in f and for any $s \in slot(f)$, let $role(s)$ denote the role name of s and $loc(s)$ the location (word position) in d of the slot filler of s. It is assumed that

- target phrases in d do not overlap, i.e., for any frames $f, f' \in \textsc{Req}(d)$, if $f \neq f'$, then for any $s \in slot(f)$ and any $s' \in slot(f')$, $loc(s) \neq loc(s')$, and
- for any frame $f \in \textsc{Req}(d) \cup \textsc{Raw}(d)$, $|slot(f)| > 1$.

This assumption holds for most multi-slot IE applications, including extraction of chemical-reaction frames discussed in this paper.

Let f be a frame in $\textsc{Raw}(d)$. f is a *false positive* frame if for any $f' \in \textsc{Req}(d)$, $slot(f) \not\subseteq slot(f')$, i.e., f always contains some irrelevant slot when it is compared with each individual frame in $\textsc{Req}(d)$. A *slot pair* in f is a pair $\langle s, s' \rangle \in slot(f) \times slot(f)$ such that $loc(s) < loc(s')$. It is called an *adjacency slot pair* if there exists no $s'' \in slot(f)$ such that $loc(s) < loc(s'') < loc(s')$. A slot pair $\langle s, s' \rangle$ in f is *correct* if there exists $f' \in \textsc{Req}(d)$ such that $\{s, s'\} \subseteq slot(f')$, and it is *incorrect* otherwise.

Proposition 1: For any frame $f \in \textsc{Raw}(d)$, f is false positive if and only if there exists an incorrect adjacency slot pair in f.

Proof See Appendix A. ■

Proposition 1 suggests that a filtering method can be devised based on removal of incorrect adjacency slot pairs. Predicting whether an adjacency slot pair is incorrect can be regarded as a binary classification problem. Classifiers for making such prediction are constructed as follows: Given a frame $f \in \textsc{Raw}(d)$ and a slot pair $\langle s, s' \rangle$ in f, let the *text portion covered by* $\langle s, s' \rangle$ be defined as the $[loc(s), loc(s')]$-portion of d. Given role names r and r', a slot pair $\langle s, s' \rangle$ is said to be *of type* $\langle r, r' \rangle$ if $role(s) = r$ and $role(s') = r'$. Then for each pair $\langle r, r' \rangle$ of role names, a classifier is constructed based on text portions covered by adjacency slot pairs of type $\langle r, r' \rangle$ observed when RAW is applied to training data. Table 2 shows the adjacency slot pairs occurring in the two extracted frames in Table 1, along with the text portions covered by them, their types, and their correctness.

Table 2 Adjacency slot pairs in the extracted frames in Table 1.

Portion	Adjacency slot pair $\langle s, s' \rangle$	Portion covered by $\langle s, s' \rangle$	Type of $\langle s, s' \rangle$	Correctness of $\langle s, s' \rangle$
$[33, 47]$	{PDT [sub คาร์บอนไดออกไซด์]}{RNM [rac ออกซิเดชัน]}	$[33, 43]$	$\langle$PDT, RNM$\rangle$	Incorrect
	{RNM [rac ออกซิเดชัน]}{RCT [sub 1-โพรพานอล]}	$[43, 47]$	$\langle$RNM, RCT$\rangle$	Correct
$[34, 48]$	{PDT [sub โพรพิโอนาลดีไฮด์]}{RNM [rac ออกซิเดชัน]}	$[36, 43]$	$\langle$PDT, RNM$\rangle$	Correct
	{RNM [rac ออกซิเดชัน]}{RCT [sub 1-โพรพานอล]}	$[43, 47]$	$\langle$RNM, RCT$\rangle$	Correct

For classifier learning, the adjacency slot pairs in the first and the third rows (respectively, those in the other two rows) are included in a training set for constructing a model for classifying adjacency slot pairs of type $\langle$PDT, RNM$\rangle$ (respectively, type $\langle$RNM, RCT$\rangle$). Features representing text portions covered by adjacency slot pairs are described in Sect. 3.

2.4 Slot-Pair Merging

The patterns of target phrases in a test set may not be covered by those in a training set. As a result, a single IE rule alone may extract only some part of a target phrase. To obtain more complete extraction results, some adjacency slot pairs should be combined although they are taken from different extracted frames. Assume that $AP(d)$ is the set of all adjacency slot pairs in frames belonging to RAW(d) and $\widehat{AP}(d)$ is the set obtained from $AP(d)$ by removing all slot pairs that are filtered out. Adjacency slot pairs in $\widehat{AP}(d)$ are merged based on a binary relation $\bowtie$ defined as follows: For any slot pairs $p = \langle s_1, s_2 \rangle$ and $p' = \langle s'_1, s'_2 \rangle$ in $\widehat{AP}(d)$, $p \bowtie p'$ if $loc(s_i) = loc(s'_j)$ for some $i, j \in \{1, 2\}$, i.e., p and p' have overlapping slot fillers. Now let $\bowtie^+$ be the transitive closure of $\bowtie$. Since $\bowtie$ is reflexive and symmetric, $\bowtie^+$ is an equivalence relation. It follows that:

Proposition 2: For each equivalence class $[p]$ in the quotient set $\widehat{AP}(d)/\bowtie^+$, if all adjacency slot pairs in $[p]$ are correct, then there exists $f \in$ REQ(d) such that all slots occurring in $[p]$ belong to $slot(f)$.

Proof See Appendix B. ■

By Proposition 2, if all incorrect adjacency slot pairs are filtered out, then all slots occurring in the same equivalence class in $\widehat{AP}(d)/\bowtie^+$ always belong to the same frame in REQ(d) (although they may belong to different frames in RAW(d)) and should therefore be merged together into one output frame.

3. Extracting Reaction Frames: Experiments

From Thai dissertation and thesis on-line database[†] provided by Technical Information Access Center (TIAC), 220 chemistry thesis abstracts related to chemical reactions were collected. They were randomly divided into two data sets, referred to as D1 and D2; each of them was once used as a training set and once as a test set. Table 3 provides some characteristics of the two data sets, e.g., abstract and target-phrase length (in words) and the number of annotated words.

3.1 Experimental Schema, Rule Learning, and Classifier Learning

Using our implementation of WHISK, 53 and 56 rules were generated when D1 and D2, respectively, were used as training sets. For each test set, two experiments, called 1W- and 2W-experiments, were conducted: in the first experiment, the length of the longest target phrase observed when a rule made correct extractions on training data was taken as the window size for the rule, and the window size was doubled in the second experiment. For constructing slot-pair classifiers, two kinds of features were used for representing text portions covered by slot pairs: first, the number of spaces, the number of plain words, and the number of annotated words occurring in a covered text portion; and secondly, the presence or absence of certain specific terms and entity tags. The principal component analysis (PCA) was used for feature selection. On average, 29.21% and 24.24% of observed features were selected in the 1W-experiment and the 2W-experiment, respectively.

The Weka machine learning suite was employed for classifier learning and evaluation, using its default parameters. Three standard models were used, i.e., Decision Tree (DT) using C4.5, k-Nearest Neighbor (kNN), and Support Vector Machine (SVM) based on the RBF kernel. As observed during the learning process, 3NN performed slightly better than 1NN and 5NN in both D1 and D2, and was chosen as a representative of kNN.

3.2 Experimental Results

Recall and precision were used as performance measures; the former is the proportion of correct slot fillers to relevant slot fillers and the latter is that of correct slot fillers to all obtained slot fillers. We evaluated our IE framework in comparison with known-boundary extraction. For known-boundary extraction, we manually located all target phrases in each test set and applied the rules obtained from WHISK directly to these manually identified text portions. Table 4 shows the evaluation results when DT, kNN, and SVM classifiers were used in our filtering module; recall and precision are given in percentage. The table shows that in 2W-experiments the performance of our framework is close to

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

Table 3 Data set characteristics.

Data set	No. of abstracts	Avg. abstract length	Avg. no. of annotated words per abstract	Avg. no. of target phrases per abstract	Avg. target-phrase length
D1	110	278.29	12.72	0.98	10.64
D2	110	270.93	12.45	1.94	8.97

Table 4 Evaluation results.

Test set	Known-boundary extraction		Window size	DT		kNN		SVM	
	Recall	Precision		Recall	Precision	Recall	Precision	Recall	Precision
D1	87.30	95.04	1W	72.96	96.55	72.96	96.55	72.96	94.12
			2W	84.69	96.30	84.69	96.30	84.69	93.19
D2	88.57	97.34	1W	78.68	99.17	78.68	99.17	78.02	96.73
			2W	86.37	97.52	86.37	97.04	84.84	93.69

that of known-boundary extraction in terms of both recall and precision. On closer examination, DT and kNN yield similar filtering performance; both of them perform slightly better than SVM.

Target phrases from which RAW fails to make any extraction, i.e., false negatives, can be divided into two types: phrases that match the pattern part of some existing rule, and those that do not. Extractions can be made from false negatives of the first type if the window size used by RAW is sufficiently large. For known-boundary extraction, since a WHISK rule is applied directly to target phrases without any restriction on phrase length, there is no false negative of the first type. Recall obtained from known-boundary extraction therefore provides an upper bound of recall possibly achieved using WHISK-based extraction. Using our framework, when the window size is doubled (i.e., 2W-experiment), resulting recall is already close to that obtained from known-boundary extraction in both D1 and D2. It is thus expected that no significant improvement of recall would be gained by further extension of the window size.

3.3 Comparison with Extraction Using Automatically Identified Sentence Boundaries

Adopting the idea of predicting sentence boundaries described in [7], the performance of applying WHISK rules to text segments that were separated by predicted sentence-break spaces was evaluated. For sentence boundary prediction, a conditional random field (CRF) tagging model, learned from the whole ORCHID Thai part-of-speech tagged corpus [12], was used to classify white spaces into 2 types: sentence-break spaces and non-sentence-break spaces. The CRF++ toolkit[†] was employed (using its default parameters) for constructing the CRF tagging model. The obtained model was then applied to both D1 and D2. To evaluate the model, predicted sentence-break spaces occurring within target phrases were observed. 19 and 22 target phrases in D1 and D2, respectively, were broken by incorrectly predicted sentence-break spaces. When the rules constructed in the experiments in Sect. 3.1 were applied to each individual text portion appearing between predicted sentence-break spaces in D1, the recall and precision of 74.27% and 57.72%, respectively, were obtained. When applied to D2, the obtained recall and precision were 79.34% and 75.68%. Compared with the results of the 2W-experiments shown in Table 4, the performance of extraction through such automatic sentence boundary detection is significantly lower than that of our proposed IE framework.

4. Semantics-Based Information Retrieval

4.1 Document Representation and Integration with Background Knowledge

Each extracted frame is represented in description logics (DL) as a concept expression, which is used as metadata for document indexing and can directly be encoded in OWL. Figure 7 illustrates the concept expression representing the frame in Fig. 4. 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

$$C_d \equiv \mathsf{Doc} \sqcap \exists \textsc{HasIndex}.C_1 \sqcap \cdots \sqcap \exists \textsc{HasIndex}.C_n,$$

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[††] and Rex ontology,[†††] 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

[†]Available at http://crfpp.sourceforge.net.
[††]Available at http://ontology.dumontierlab.com.
[†††]Available at http://onto.eva.mpg.de/obo.

ChemReaction ⊓ ∃HasRnm.Oxidation
⊓ ∃HasPdt.Propionaldehyde
⊓ ∃HasRct.1-Propanol

Fig. 7 A concept expression representing the frame in Fig. 4.

Table 5 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

1-Propanol ⊑ Alcohol (1)
Propionaldehyde ⊑ Aldehyde (2)
Aldehyde ≡ OrganicCompound ⊓ ∃HasPart.AldehydeGroup (3)
OrganicCompound ⊑ Compound (4)
AldehydeGroup ⊑ CarbonAtom ⊓ ∃HasBondWith.OxygenAtom ⊓ ∃HasBondWith.HydrogenAtom (5)
HasPdt ⊑ HasParticipant (6)
OrganicReaction ≡ ChemReaction ⊓ ∃HasParticipant.OrganicCompound (7)

Fig. 8 Part of background knowledge.

C_{q_1}: Doc ⊓ ∃HasIndex.(ChemReaction ⊓ ∃HasRct.Alcohol)
C_{q_2}: Doc ⊓ ∃HasIndex.OrganicReaction
C_{q_3}: Doc ⊓ ∃HasIndex.(ChemReaction ⊓ ∃HasPdt.(Compound ⊓ ∃HasPart.CarbonAtom))

Fig. 9 Query representation.

taxonomies of chemical reactions. Table 5 gives some characteristics of these two ontologies and Fig. 8 shows some background-knowledge axioms they provide.

Using the FaCT OWL-DL reasoner in the Protégé ontology editor, it takes around 30 seconds on a standard machine with Intel Core2 processor 1.6 GHz and 1.0 GB RAM for classifying the concepts defined in these two ontologies along with all document metadata extracted from the 220 thesis abstracts in the data sets D1 and D2 used in Sect. 3.

4.2 Document Retrieval: Examples

To demonstrate semantics-based information retrieval in the obtained document knowledge base, assume that d_0 is a document containing the second target phrase in Fig. 3, i.e.,

> "propionaldehyde is obtained from the oxidation reaction of 1-propanol," (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 $C_{d_0} \equiv$ Doc ⊓ ∃HasIndex.C, where C is the concept expression in Fig. 7, which represents the frame extracted from Statement (8) (i.e., the frame given in Fig. 4),
- 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. 9, and
- the background-knowledge axioms in Fig. 8 are employed.

A DL-based reasoner then infers that C_{q_1} subsumes C_{d_0} in one inference step using Axiom (1), infers that C_{q_2} subsumes C_{d_0} in four steps using Axioms (2), (3), (6), and (7), and infers that C_{q_3} subsumes C_{d_0} in four steps using Ax-

ioms (2), (3), (4), and (5). Accordingly, each of q_1, q_2, and q_3 retrieves d_0.

It is noteworthy that the concept expression shown in Fig. 7, the background knowledge axioms shown in Fig. 8, and the expressions representing queries in Fig. 9 can all be formalized using the lightweight description logic $\mathcal{EL}$ [6], for which polynomial-time reasoners (e.g., CEL[†]) are available. However, the Chemical Complex ontology, which is used as part of our document knowledge base (see Sect. 4.1), 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.

5. Related Works

Very few works on IE from Thai text were reported in the literature. Sukhahuta and Smith [13] 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 [13] itself, one main cause of errors comes from the ambiguity of the sentence structure. Only hand-crafted triggering-term patterns were considered in [13]; extraction-pattern learning was not discussed. Narupiyakul et al. [8] 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 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 entity classes. Moreover, individual-field extraction, such as that in [8], has a serious limitation for a significant number of applications, in particular, when a document contains fillers of more than one frame, e.g., it cannot relate a reactant and a product involved in a particular chemical reaction when a document describes several reactions.

As reported in [9], reaction-related roles, e.g., reactants and products, have been used for indexing individual substances occurring in research abstracts in bibliographic databases provided by Chemical Abstracts Service (CAS). Such role indexing appears to be useful for making keyword-based search more precise. Role assignment in CAS databases is, however, performed manually and, unlike our multi-slot-frame approach, no semantic relation is made between substances involving the same reaction. Sankar and Aghila [10] developed XML-based ontologies for representing chemical reactions. Their ontologies describe taxonomies of organic reactions, organic compounds, and reagents, along with binary relations between them. Based on these ontologies, a reaction representation system was introduced. Although a retrieval model based on XML node-based search was described, only keyword-based search was discussed in [10].

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

6. Conclusions

Using our implementation of WHISK, IE rules are created from hand-tagged chemical-reaction phrases in a training corpus. To apply the obtained rules to free text without predetermining target-phrase boundaries, rule application using sliding windows is introduced. A filtering method is proposed for removal of false positive slot fillers. Based on our experimental results, when the window size is sufficiently large, the performance of our IE framework is close to that of rule application with manually located target phrases. Extraction results are used as metadata for document indexing. Using domain-specific ontologies as background knowledge, semantics-based document retrieval is demonstrated.

Acknowledgements

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

References

[1] G. Antoniou and F.V. Harmelen, "Web ontology language: OWL," in Handbook on Ontologies, ed. S. Staab and R. Studer, pp.67–92, Springer, 2004.
[2] W. Aroonmanakun, "Thoughts on word and sentence segmentation in Thai," Proc. 7th International Symposium on Natural Language Processing, pp.85–90, Pattaya, Thailand, 2007
[3] M.E. Califf and R.J. Mooney, "Bottom-up relational learning of pattern matching rules for information extraction," J. Machine Learning Research, vol.4, pp.177–210, 2003.
[4] N. Danvivathana, The Thai Writing System, Helmut Buske Verlag, 1987.
[5] D. Freitag, "Machine learning for information extraction in informal domains," Mach. Learn., vol.39, no.2-3, pp.169–202, 2000.
[6] C. Lutz and F. Wolter, "Conservative extensions in the lightweight description logic $\mathcal{EL}$," Proc. 21st International Conference on Automated Deduction, Automated Deduction, Bremen, Germany, Lecture Notes in Artificial Intelligence, vol.4603, pp.84–99, 2007.
[7] P. Mittrapiyanuruk and V. Sornlertlamvanich, "The automatic Thai sentence extraction," Proc. 4th Symposium on Natural Language Processing, pp.23–28, Chiang Mai, Thailand, 2000.
[8] L. Narupiyakul, C. Thomas, N. Cercone, and B. Sirinaovakul, "Thai syllable-based information extraction using hidden Markov models," Computational Linguistics and Intelligent Text Processing, Lect. Notes Comput. Sci., vol.2945, pp.537–546, 2004.
[9] D.D. Ridley, "Strategies for chemical reaction searching in sciFinder," J. Chemical Information and Computer Sciences, vol.40, pp.1077–1084, 2000.
[10] P. Sankar and G. Aghila, "Design and development of chemical ontologies for reaction representation," J. Chemical Information and Modeling, vol.46, pp.2355–2368, 2006.
[11] S. Soderland, "Learning information extraction rules for semi-structured and free text," Mach. Learn., vol.34, no.1-3, pp.233–272, 1999.

[12] V. Sornlertlamvanich, N. Takahashi, and H. Isahara, "Building a Thai part-of-speech tagged corpus (ORCHID)," J. Acoust. Soc. Jpn, vol.20, no.3, pp.189–198, 1999.
[13] R. Sukhahuta and D. Smith, "Information extraction strategies for Thai documents," International Journal of Computer Processing of Oriental Languages, vol.14, pp.153–172, 2001.

Appendix

This part provides the proofs of Propositions 1 and 2 (in Sects. 2.3 and 2.4).

Appendix A: Proof of Proposition 1

Let f be a frame in $\textsc{Raw}(d)$. Suppose first that f is false positive. Let $s \in slot(f)$. There are two cases:

Case 1: There exists $f_s \in \textsc{Req}(d)$ such that $s \in slot(f_s)$. Since f is false positive and $|slot(f)| > 1$, there exists $s' \in slot(f)$ such that $loc(s) \neq loc(s')$ and $s' \notin slot(f_s)$. Suppose that $loc(s) < loc(s')$. Then there exists a sequence $s_1, \ldots, s_n$ of slots in f, where $n > 1$, such that $s_1 = s$, $s_n = s'$, and for any $i \in \{1, \ldots, n-1\}$, $\langle s_i, s_{i+1} \rangle$ is an adjacency slot pair in f. Since $s \in slot(f_s)$ and $s' \notin slot(f_s)$, there exists $j \in \{1, \ldots, n-1\}$ such that $s_j \in slot(f_s)$ and $s_{j+1} \notin slot(f_s)$. Since target phrases in d do not overlap, $s_j \notin slot(f')$ for any $f' \in \textsc{Req}(d) - \{f_s\}$. Then for any $f'' \in \textsc{Req}(d)$, $\{s_j, s_{j+1}\} \not\subseteq slot(f'')$. Thus $\langle s_j, s_{j+1} \rangle$ is an incorrect adjacency slot pair. It can be shown in a similar way that if $loc(s) > loc(s')$, then an incorrect adjacency slot pair in f also exists.

Case 2: There exists no $f_s \in \textsc{Req}(d)$ such that $s \in slot(f_s)$. Since $|slot(f)| > 1$, there exists an adjacency slot pair p in f such that p contains s. It follows readily that p is an incorrect adjacency slot pair.

Conversely, suppose that $\langle s, s' \rangle$ is an incorrect adjacency slot pair in f. Then for any $f' \in \textsc{Req}(d)$, $\{s, s'\} \not\subseteq slot(f')$, whence $slot(f) \not\subseteq slot(f')$. Therefore f is false positive. ■

Appendix B: Proof of Proposition 2

Proposition 3 is used for proving Proposition 2.

Proposition 3: Let $p = \langle s_1, s_2 \rangle$ and $p' = \langle s'_1, s'_2 \rangle$ be adjacency slot pairs in $\widehat{AP}(d)$ such that $p \bowtie p'$. If p and p' are correct, then there exists a unique frame $f \in \textsc{Req}(d)$ such that $\{s_1, s_2, s'_1, s'_2\} \subseteq slot(f)$.

Proof Suppose that p and p' are correct. Then $\{s_1, s_2\} \subseteq slot(f)$ and $\{s'_1, s'_2\} \subseteq slot(f')$ for some frames $f, f' \in \textsc{Req}(d)$. Since $p \bowtie p'$ and target phrases in d do not overlap, $f = f'$ and for any frame $f'' \in \textsc{Req}(d)$ such that $f \neq f''$, $\{s_1, s_2, s'_1, s'_2\}$ and $slot(f'')$ are disjoint. ■

Proof of Proposition 2 Let $[p]$ be an equivalence class in the quotient set $\widehat{AP}(d)/\bowtie^+$. Suppose that all adjacency slot pairs in $[p]$ are correct. Since p is correct and target phrases in d do not overlap, there exists a unique frame $f \in \textsc{Req}(d)$ such that each slot in p belongs to $slot(f)$. Now let $p' \in [p]$. Then there exists a sequence $p_1, \ldots, p_n$ of adjacency slot pairs in $\widehat{AP}(d)$, where $n > 1$, such that $p_1 = p$, $p_n = p'$, and for any $i \in \{1, \ldots, n-1\}$, $p_i \bowtie p_{i+1}$. It follows from Proposition 3 that for any $j \in \{2, \ldots, n\}$, each slot in p_j belongs to $slot(f)$. So each slot in p' belongs to $slot(f)$. ■

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

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

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