



## EXTRACTING CHEMICAL INFORMATION FROM THAI UNSTRUCTURED TEXT WITH UNKNOWN PHRASE BOUNDARIES

Peerasak Intarapaiboon

Department of Mathematics and Statistics, Faculty of Science and Technology, Thammasat University, Pathumthani 12121, Thailand

\*e-mail: [peerasak@mathstat.sci.tu.ac.th](mailto:peerasak@mathstat.sci.tu.ac.th)

---

### Abstract

Due to the limitations of language-processing tools for the Thai language, pattern-based information extraction from Thai documents requires supplementary techniques. Based on sliding-window rule application and extraction filtering, we present a framework for extracting multi-slot frames describing chemical reactions and those describing chemical syntheses from Thai unstructured 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. The experimental results show that the filtering components improve precision while preserving recall satisfactorily.

---

**Keywords:** information extraction, rule learning, binary classification

### Introduction

Multi-slot extraction is a task concerning with extracting facts (slot fillers) from text and linking them into case frames. It is particularly useful when a textual information entry contains more than one target event; for example, from a text portion “Acetaldehyde is obtained from the oxidation reaction of ethanol, while propionaldehyde is obtained from the oxidation reaction of 1-propanol.” It is often desirable to identify not only that acetaldehyde and propionaldehyde are reaction products and ethanol and 1-propanol are reactants, but also that acetaldehyde and ethanol participate in one reaction whereas the other two substances participate in the another one.

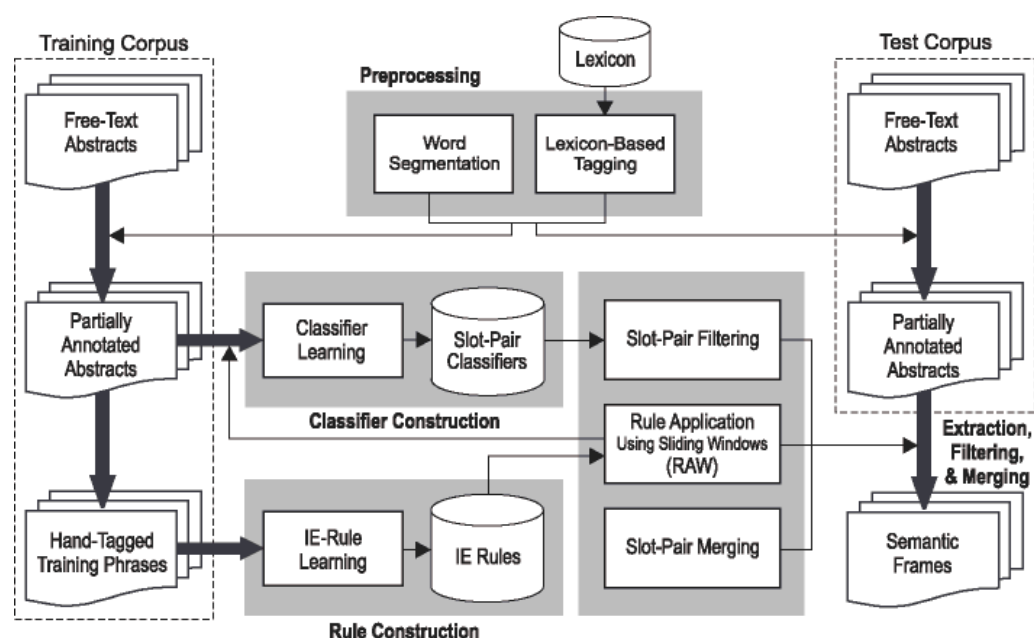
In this paper, we present a framework for extracting multi-slot frames describing chemical reactions and chemical syntheses from chemistry thesis abstracts written in Thai. From input thesis abstracts, partially annotated with entity classes in a preprocessing phase, extractions are made based on inductively learned patterns of triggering entity tags and triggering plain words. A well-known supervised rule learning algorithm, called WHISK (Soderland 1999), 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 unstructured 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 (Danvivathana 1987) and the notion of a sentence is unclear (Aroonmanakun 2007). 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.

## Framework

A framework for extracting chemical-information frames from Thai unstructured-text thesis abstracts, outlined in Figure 1, is described below.



**Figure 1** An overview of the proposed 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|...

**Figure 2** A portion of a partially annotated word-segmented abstract

### 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 Figure 2. It contains three slots with the role names RNM, PDT, and RCT, which stand for “reaction name,” “product,” and “reactant,” respectively.

*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 . . .*

**Figure 3** A literal English translation of the partially annotated Thai text in Figure 2

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

**Figure 4** A frame extracted from the second target phrase in Figure 2

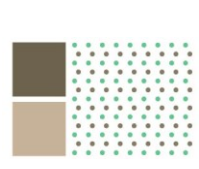
**Figure 5** An IE rule example

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 Figure 2, this rule yields the frame in Figure 4.

#### Rule Learning and Rule Application using Sliding Windows:

WHISK (Soderland 1999) 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 (Califf and Mooney 2007) and SRV (Freitag 2000), can generate 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 (Danvivathana 1987). To apply IE rules to unstructured 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  $l$ th word position and ending at the  $m$ th 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.

#### 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,  $\mathbf{FREQ}(d)$  is the set of all frames required to be extracted from  $d$ , and  $\mathbf{FEXT}(d)$  is the set of all frames extracted from  $d$  by using RAW. Given a frame  $f$  in  $\mathbf{FREQ}(d)$  or in  $\mathbf{FEXT}(d)$ , let  $\mathbf{slot}(f)$  denote the set of all slots in  $f$  and for any  $s$  in  $\mathbf{slot}(f)$ , let  $\mathbf{role}(s)$  denote the role name of  $s$  and  $\mathbf{loc}(s)$  the location (word position) in  $d$  of the slot filler of  $s$ . It is assumed that target phrases in  $d$  do not overlap and for any frame  $f$  in  $\mathbf{FREQ}(d) \cup \mathbf{FEXT}(d)$ ,  $|\mathbf{slot}(f)| > 1$ . This assumption holds for most multi-slot IE applications, including extraction of chemical-information frames discussed in this paper.

Let  $f$  be a frame in  $\mathbf{FEXT}(d)$ .  $f$  is a *false positive* frame if for any  $f'$  in  $\mathbf{FREQ}(d)$ ,  $\mathbf{slot}(f) \not\subseteq \mathbf{slot}(f')$ , i.e.,  $f$  always contains some irrelevant slot when it is compared with each individual frame in  $\mathbf{FREQ}(d)$ . A *slot pair* in  $f$  is a pair  $\langle s, s' \rangle \in \mathbf{slot}(f) \times \mathbf{slot}(f)$  such that  $\mathbf{loc}(s) < \mathbf{loc}(s')$ . It is called an *adjacency slot pair* if there exists no  $s'' \in \mathbf{slot}(f)$  such that  $\mathbf{loc}(s) < \mathbf{loc}(s'') < \mathbf{loc}(s')$ . A slot pair  $\langle s, s' \rangle$  in  $f$  is *correct* if there exists  $f' \in \mathbf{FREQ}(d)$  such that  $\{s, s'\} \subseteq \mathbf{slot}(f')$ , and it is *incorrect* otherwise. By the slot pair meaning, we can alternatively say that an extracted frame  $f$  is false positive if and only if there exists an incorrect adjacency slot pair in  $f$ .

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  $f \in \mathbf{FEXT}(d)$  and an adjacency slot pair  $\langle s, s' \rangle \in f$ , let the *text portion enclosed by*  $\langle s, s' \rangle$  be defined as the  $[\mathbf{loc}(s)+1, \mathbf{loc}(s')-1]$ -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  $\mathbf{role}(s) = r$  and  $\mathbf{role}(s') = r'$ . Then for each pair  $\langle r, r' \rangle$  of role names, a classifier is constructed based on text portions enclosed by adjacency slot pairs of type  $\langle r, r' \rangle$  observed when RAW is applied to training data. Features representing enclosed text portions are described in the experiment section.

#### 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  $\mathbf{AP}(d)$  is the set of all adjacency slot pairs in frames belonging to  $\mathbf{FEXT}(d)$  and  $\mathbf{AP}^*(d)$  is the set obtained from  $\mathbf{AP}(d)$  by removing all slot pairs that are filtered out. Adjacency slot pairs in  $\mathbf{AP}^*(d)$  are merged based on a binary relation  $\otimes$  defined as follows: For any slot pairs  $p = \langle s_1, s_2 \rangle$  and  $p' = \langle s_1', s_2' \rangle \in \mathbf{AP}^*(d)$ ,  $p \otimes p'$  if  $\mathbf{loc}(s_i) = \mathbf{loc}(s'_j)$  for some  $i, j \in \{1, 2\}$ , i.e.,  $p$  and  $p'$  have overlapping slot fillers.

## Experiments

The proposed framework is evaluated on two corpora with different types of target phrases in the chemistry domain. The corpora were constructed from Thai dissertation and thesis on-line database provided by Technical Information Access Center (TIAC). The first corpus, referred to as CR, consists of 220 thesis abstracts containing descriptions of chemical reactions, while

the second one, referred to as CS, consists of 141 thesis abstracts containing those of chemical syntheses. A target phrase in the CR corpus is a chemical-reaction description containing at least two of the following components: reaction name, reaction products, reactants, and catalysts. A target phrase in the CS corpus is a chemical-synthesis description containing at least one reaction product along with at least one of reaction name, reaction products, reactants, and catalysts. Each corpus was randomly divided into two data sets, referred to as DCR1 and DCR 2 for CR and as DCS1 and DCS 2 for CS; each of them was once used as a training set and once as a test set. Table 1 provides some characteristics of target phrases in the obtained data sets.

**Table 1** Target phrase information

Domain	Data set	# Target phrases	Target-phrase length			# Target phrases per an abstract		
			Max.	Avg.	Min.	Max.	Avg.	Min.
CR	DCR1	122	41	10.6	3	8	1.0	0
CR	DCR2	188	22	10.0	3	9	1.9	0
CS	DCS1	124	22	10.6	5	7	1.8	0
CS	DCS2	128	29	10.9	4	6	1.8	0

Using our implementation of WHISK, 53, 56, 53, and 59 rules were generated when DCR1, DCR2, DCS1, and DCS2, 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 enclosed by slot pairs: first, the number of spaces, the number of plain words, and the number of annotated words occurring in an enclosed text portion; and secondly, the presence or absence of certain specific terms. A principal component analysis (PCA) algorithm available in the Weka machine learning suite is 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, for chemical-reaction extraction; 33.14% and 28.82% of observed features were selected in the 1W-experiment and the 2W-experiment, respectively, for chemical-synthesis extraction.

Weka was also 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 ( $k$ NN), and Support Vector Machine (SVM) based on the RBF kernel. As observed during the learning process, 3NN performed slightly better than 1NN and 5NN, and was chosen as a representative of  $k$ NN.

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 2 shows the evaluation results when DT,  $k$ NN, 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 that of known-boundary extraction in terms of both recall and precision. On closer examination, DT and  $k$ NN yield similar filtering performance; both of them perform slightly better than SVM.



**Table 2** Evaluation results

Test set	Known-boundary extraction		Window size	DT		kNN		SVM	
	Recall	Precision		Recall	Precision	Recall	Precision	Recall	Precision
DCR1	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
DCR2	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
DCS1	88.50	96.52	1W	85.30	94.35	85.30	93.36	83.71	87.33
			2W	88.50	93.90	85.62	91.47	82.43	83.77
DCS2	83.14	97.91	1W	76.92	97.01	76.92	97.01	72.49	93.16
			2W	83.14	95.25	83.14	95.25	74.56	84.56

## Related works

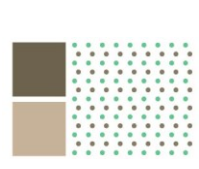
Very few works on IE from Thai text were reported in the literature. Sukhahuta and Smith (2001) proposed strategies for Thai-text IE using corpus-based syntactic surface analysis based on predefined context-free grammar rules. The extraction precision of their developed system is still relatively low; as pointed out by the authors, one main cause of errors comes from the ambiguity of the sentence structure. Only hand-crafted triggering-term patterns were considered in (Sukhahuta and Smith 2001); extraction-pattern learning was not discussed. Narupiyakul et al. (2004) 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 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.

## 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 (RAW) 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. Further works include extension of the types of target phrases.

## References

1. Soderland S (1999) Learning Information Extraction Rules for Semi-Structured and Free Text. *Machine Learning* 34:233—272.
2. Danvivathana N (1987) *The Thai Writing System*. Helmut Buske Verlag, Hamburg.
3. Aroonmanakun W (2007) Thoughts on Word and Sentence Segmentation in Thai. In *Proc. 7th International Symposium on Natural Language Processing*. Chonburi, Thailand pp.85—90.

- 
4. Califf M E and Mooney R J (2003) Bottom-up Relational Learning of Pattern Matching Rules for Information Extraction. *Journal of Machine Learning Research* 4:177—210.
  5. Freitag D (2000) Machine Learning for Information Extraction in Informal Domains. *Machine Learning* 39:169—202.
  6. Sukhahuta R and Smith D (2001) Information Extraction Strategies for Thai Documents. *International Journal of Computer Processing of Oriental Languages* 14:153—172.
  7. Narupiyakul L, Thomas C, Cercone N, and Sirinaovakul B (2004) Thai Syllable-Based Information Extraction Using Hidden Markov Models. In *Proc. 5th International Conference on Intelligent Text Processing and Computational Linguistics*. Seoul, Korea, *Lecture Notes in Computer Science* 2945:537—546.