Semantic Relation Extraction from a Cultural Database

Canasai KRUENGKRAI Virach SORNLERTLAMVANICH
Watchira BURANASING Thatsanee CHAROENPORN
National Electronics and Computer Technology Center
Thailand Science Park, Klong Luang, Pathumthani 12120, Thailand
{canasai.kru,virach.sor,watchira.bur,thatsanee.cha}@nectec.or.th

ABSTRACT
Semantic relation extraction aims to extract relation instances from natural language texts. In this paper, we propose a semantic relation extraction approach based on simple relation templates that determine relation types and their arguments. We attempt to reduce semantic drift of the arguments by using named entity models as semantic constraints. Experimental results indicate that our approach is very promising. We successfully apply our approach to a cultural database and discover more than 18,000 relation instances with expected high accuracy.

KEYWORDS: semantic relation extraction, cultural database.

Proceedings of the 3rd Workshop on South and Southeast Asian Natural Language Processing (SANLP), pages 15–24,
COLING 2012, Mumbai, December 2012.

1 Introduction

In this paper, we are interested in extracting certain basic facts from a cultural database derived from the Thai Cultural Information Center website[1]. The size of this cultural database has gradually increased to around 80,000 records (from November 2010 to October 2012). Each record contains a number of fields describing a specific cultural object. Figure 1 shows an excerpt of the front-end web page of the record no. 35860, which is about the Mid-River Pagoda. The content includes four main components: (1) cover image and thumbnails, (2) title, (3) description and (4) domain. We need to extract facts (hereafter referred to as relation instances) from the description. One can view relation instances as formal meaning representations of corresponding texts. These relation instances are useful for question answering and other applications. Using this record as an example, we could extract a relation instance ISLOCATEDAT(เจดีย์กลางน้ำ, ตำบลปากน้ำ) from the first text segment:

เจดีย์กลางน้ำตั้งอยู่ที่ตำบลปากน้ำ

(The *Mid-River Pagoda* is located at *Tambon Paknam*)

Recent research in semantic relation extraction has shown the possibility to automatically find such relation instances. Some approaches rely on high-quality syntactic parsers. For example, DIRT (Lin and Pantel, 2001) and USP (Poon and Domingos, 2009) discover relation instances based on the outputs from dependency parsers. Such parsers and annotated training corpora are difficult to obtain in non-English languages. Pattern-based approaches (Agichtein and Gravano, 2000; Pantel and Pennacchiotti, 2006; Banko et al., 2007) seem to be more practical for languages with limited NLP resources. For example, TEXTRUNNER (Banko et al., 2007) can efficiently extract relation instances from a large-scale Web corpus with minimal supervision. It only requires a lightweight noun phrase chunker to identify relation arguments. More advanced approaches like SNE (Kok and Domingos, 2008), RESOLVER (Yates and Etzioni, 2009) and SHERLOCK (Schoenmackers et al., 2010) exploit the outputs of TEXTRUNNER for learning.

Our cultural database allows us to make two assumptions:

(A1) Each record belongs to only one main cultural domain.
(A2) Each record has only one subject of relations.

The assumption (A1) seems to hold for most of records. We adopt the assumption (A2) from (Hoffmann et al., 2010) that try to extract infobox-like relations from Wikipedia. Also, the assumption (A2) seems to hold for our data since the description provides the details about one cultural object whose name is expressed in the record title.

Based on the above two assumptions, we propose our strategy to semi-automatically extract relation instances from the cultural database. We focus on unary relation extraction similar to (Hoffmann et al., 2010; Chen et al., 2011). We assume that the subject of the relation is the record title.[2] Each relation remains only one argument to be extracted. We describe our relation templates (Section 2.1) and how to effectively find relation texts in a large database (Section 2.2). We use named entities to reduce semantic drift of the target arguments (Section 2.3). We examine the effect of the distances between the relation surfaces and the target arguments (Section 3.1) and provide preliminary results of our experiments (Section 3.2). The results indicate that our strategy of semantic relation extraction is very promising for real-world applications.

[1] http://m-culture.in.th/

[2] In Figure 1, although the surface words of the *Mid-River Pagoda* are slightly different ("พระเจดีย์กลางน้ำ" vs. "เจดีย์กลางน้ำ"), they convey the same meaning and this assumption still holds.

Figure 1: An excerpt of the front-end web page of the record about the Mid-River Pagoda.

Domain	Relation	Surface	Argument
Cultural attraction	ISLOCATEDAT	ตั้งอยู่ที่	LOC
	ISBUILTIN	สร้าง(ขึ้น)*ใน สร้าง(ขึ้น)*เมื่อ ตั้ง(ขึ้น)*เมื่อ	DATE
	ISBUILTBY	สร้าง(ขึ้น)*โดย ตั้ง(ขึ้น)*โดย	PER, ORG
	HASOLDNAME	เดิมชื่อ ชื่อเดิม	LOC, ORG
Cultural person	MARRIEDWITH	สมรสกับ	PER
	HASFATHERNAME	บิดาชื่อ	PER
	HASMOTHERNAME	มารดาชื่อ	PER
	HASOLDNAME	เดิมชื่อ ชื่อเดิม	PER
	HASBIRTHDATE	เกิด(เมื่อ)*	DATE
	BECOMEMONKIN	อุปสมบทเมื่อ	DATE
Cultural artifact	ISMADEBY	ผลิต(ขึ้น)*โดย ทำ(ขึ้น)*โดย ผลงานโดย	PER, ORG
	ISSOLDAT	จำหน่ายที่	LOC, ORG

Table 1: Our relation templates.

2 Approach

2.1 Designing relation templates

Table 1 shows our relation templates. There are five main cultural domains in the database, and each main cultural domain has several sub-domains. In our work, we focus on three cultural domains, including attraction, person and artifact, as shown in the first column. Based on these cultural domains, we expect that the subject of relations in each record (i.e., the record title) should be a place, a human or a man-made object, respectively. As a consequence, we can design a set of relations that correspond to the subject. For example, if the subject is a place, we may need to know *where* it is, *when* it was built and *who* built it. We can formally write these expressions by ISLOCATEDAT, ISBUILTIN and ISBUILTBY. The second column shows our relations that are associated with the subject domains. The third column shows relation surfaces used for searching

relation texts in which arguments may co-occur. The word in parentheses with an asterisk indicates that it may or may not appear in the surface.

The answers to *where*, *when* and *who* questions are typically short and expressed in the form of noun phrases. Using noun phrases as relation arguments can lead to high recall but low precision. For example, the noun phrase occurring after the relation ISBUILTIN could be a place (*is built in the area of* . . .) or an expression of time (*is built in the year of* . . .). In our case, we expect the answer to be the expression of time, and hence returning the place is irrelevant. This issue can be thought of as semantic drift. Here, we attempt to reduce semantic drift of the target arguments by using named entities as semantic constraints. The forth column shows named entity types[3] associated with the subject domains and their relations.

2.2 Searching relation texts

Searching text segments containing a given relation surface (e.g., "สร้างโดย" (*is built by*)) in a large database is not a trivial task. Here, we use Apache Solr[4] for indexing and searching the database. Apache Solr works well with English and also has extensions for handling non-English languages. To process Thai text, one just enables ThaiWordFilterFactory module in schema.xml. This module invokes the Java BreakIterator and specifies the locale to Thai (TH). The Java BreakIterator uses a simple dictionary-based method, which does not tolerate word boundary ambiguities and unknown words. For example, the words "สร้าง" and "ก่อสร้าง" occur in the Java's system dictionary. Both convey the same meaning (*to build*). We can see that the first word is a part of the second word. However, these two words are indexed differently. This means if our query is "สร้าง", we cannot retrieve the records containing "ก่อสร้าง". In other words, the dictionary-based search returns results with high precision but low recall.

In our work, we process Thai text in lower units called character clusters. A character cluster functions as an inseparable unit which is larger than (or equal to) a character and smaller than (or equal to) a word. Once the character cluster is produced, it cannot be further divided into smaller units. For example, we can divide the word "ก่อสร้าง" into 5 character clusters like "ก่-อ-ส-ร้า-ง". As a result, if our query is "สร้าง", we can retrieve the records containing "ก่อสร้าง". We refer to (Theeramunkong et al., 2000) for more details about character cluster based indexing. In our work, we implement our own ThaiWordTokinizeFactory module and plug it into Apache Solr by replacing the default WhitespaceTokenizerFactory. Our character cluster generator class is based on the spelling rules described in (Kruengkrai et al., 2009).

In Thai, sentence boundary markers (e.g., a full stop) are not explicitly written. The white spaces placing among text segments can function as word, phrase, clause or sentence boundaries (see the "รายละเอียด" section in Figure 1 for example). To obtain a relation text, which is not too short (one text segment) or too long (a whole paragraph), we proceed as follows. After finding the position of the target relation surface, we look up at most $\pm$ 4 text segments to generate relation texts. This length should be enough for morphological analyzer and named entity recognizer.

2.3 Learning named entities

We control semantic drift of the target arguments using named entities. We build our named entity (NE) recognizer from an annotated corpus developed by (Theeramunkong et al., 2010). The origi-

[3] We use four main named entity types: PER = persons, ORG = organizations, LOC = locations, DATE = dates (expressions of time).

[4] http://lucene.apache.org/solr/

(I): word 1,2 grams + label bigrams $\langle w_j \rangle, j \in [-2, 2] \times y_0$ $\langle w_j, w_{j+1} \rangle, j \in [-2, 1] \times y_0$ $\langle y_{-1}, y_0 \rangle$	**(III): (II) + POS 3 grams** $\langle p_j, p_{j+1}, p_{j+2} \rangle, j \in [-2, 0] \times y_0$
(II): (I) + POS 1,2 grams $\langle p_j \rangle, j \in [-2, 2] \times y_0$ $\langle p_j, p_{j+1} \rangle, j \in [-2, 1] \times y_0$	**(IV): (III) + k-char prefixes/suffixes** $\langle P_k(w_0) \rangle, k \in [2, 3] \times y_0$ $\langle S_k(w_0) \rangle, k \in [2, 3] \times y_0$ $\langle P_k(w_0), S_k(w_0) \rangle, k \in [2, 3] \times y_0$

Table 2: Our NE features.

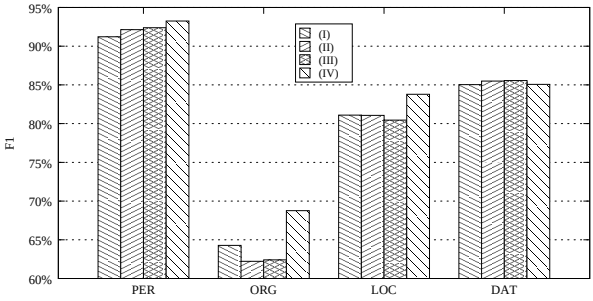

Figure 2: F1 results for our NE models.

nal contents are from several news websites. The corpus consists of 7 NE types. We focus on 4 NE types according to our relation templates in Table 1. Once we obtained the NE corpus, we checked it and found several issues as follows:

1. Each NE tag contains nested NE tags. For example, the person name tag contains the forename and surename tags.
2. The corpus does not provide gold word boundaries and POS tags.
3. Each NE type is annotated separately.

For the first issue, we ignored the nested NE tags and trained our model with top NE tags (PER, ORG, LOC, DATE). For the second issue, we used a state-of-the-art Thai morphological analyzer (Kruengkrai et al., 2009) to obtain word boundaries and POS tags. In this work, we trained the morphological analyzer using ORCHID corpus (Sornlertlamvanich et al., 1997) and TCL's lexicon[5] (Charoenporn et al., 2004). We then converted the corpus format into the IOB tagging style for NE tags. Thus, the final form of our corpus contains three columns (word, POS tag, NE tag), where the first two columns are automatically generated and of course contain a number of errors. For the third issue, we trained the model separately for each NE type. We obtained 33231, 20398, 8585, 2783 samples for PER, ORG, LOC, DATE, respectively.

To ensure that our NE models work properly, we split samples into 90%/10% training/test sets and conducted some experiments. We trained our NE models using k-best MIRA (Margin Infused Relaxed Algorithm) (Crammer et al., 2005). We set $k = 5$ and the number of training iterations to

[5]http://www.tcllab.org/tcllex/

Relation	Argument	Distance					
		0	1	2	3	4	5
Cultural attraction							
isLocatedAt	LOC	356	574	591	624	678	757
isBuiltIn	DATE	3825	11487	11538	11573	11633	11667
isBuiltBy	PER, ORG	131	202	218	234	249	257
hasOldName	LOC, ORG	0	9	21	26	27	29
Cultural person							
marriedWith	PER	132	177	177	177	177	177
hasFatherName	PER	120	372	372	373	373	373
hasMotherName	PER	97	383	383	383	383	383
hasOldName	PER	51	259	273	277	277	283
hasBirthDate	DATE	4122	4745	4801	4947	4966	5075
becomeMonkIn	DATE	346	435	435	436	436	436
Cultural artifact							
isMadeBy	PER, ORG	62	107	109	125	129	130
isSoldAt	LOC, ORG	31	31	56	59	62	64

Table 3: Numbers of relation instances when the distances are varied.

10. We denote the word by w, the k-character prefix and suffix of the word by $P_k(w)$ and $S_k(w)$, the POS tag by p and the NE tag by y. Table 2 summarizes all feature combinations used in our experiments. Our baseline features (I) include word unigrams/bigrams and NE tag bigrams. Since we obtained the word boundaries and POS tags automatically, we introduced them gradually to our features (II, III, IV) to observe their effects.

Figure 2 shows F1 results for our NE models. We used the `conlleval` script[6] for evaluation. We observe that PER is easy to identify, while ORG is difficult. Prefix/suffix features dramatically improve performance on ORG. Using all features (IV) gives best performance on PER (93.24%), ORG (68.75%) and LOC (83.78%), while slightly drops performance on DATE (85.06%). Thus, our final NE models used in relation extraction are based on all features (IV). Although these results are from the news domain, we could expect similar performance when applying the NE models to our cultural domains.

2.4 Summary

We summarize our strategy as follows. After selecting the subject domain, we send its relation surfaces (shown in the 3rd column of Table 1) to Apache Solr. We then trim the resulting record descriptions to obtain the relation texts (described in Section 2.2). Next, we perform word segmentation and POS tagging simultaneously using our morphological analyzer and feed the results into our NE models (described in Section 2.3). We invoke the appropriate NE model based on our relation templates (described in Section 2.1). Finally, our system produces outputs in the form of RELATION(a, b), where a is a record title, and b is an argument specified by its NE type in the templates.

[6] Available at http://www.cnts.ua.ac.be/conll2000/chunking/conlleval.txt.

Relation	Argument	# Samples	# Correct	# Incorrect	Accuracy
Cultural attraction					
ISLOCATEDAT	LOC	50	49	1	98%
ISBUILTIN	DATE	50	48	2	96%
ISBUILTBY	PER, ORG	50	48	2	96%
HASOLDNAME	LOC, ORG	27	23	4	85%
Cultural person					
MARRIEDWITH	PER	50	49	1	98%
HASFATHERNAME	PER	50	48	2	96%
HASMOTHERNAME	PER	50	49	1	98%
HASOLDNAME	PER	50	47	3	94%
HASBIRTHDATE	DATE	50	48	2	96%
BECOMEMONKIN	DATE	50	50	0	100%
Cultural artifact					
ISMADEBY	PER, ORG	50	44	6	88%
ISSOLDAT	LOC, ORG	50	49	1	98%

Table 4: Performance of our relation extraction.

3 Experiments

3.1 Effect of the distances between relation surfaces and arguments

In this section, we examine the number of extracted instances for each relation (without considering its accuracy). Our assumption is that the target argument tends to be relevant if it is adjacent (or close) to the relation surface. The relevance weakens with the distance. In our first example, the target argument "ตำบลปากน้ำ" (*Tambon Paknam*, a subdistrict name) is adjacent (distance = 0) to the relation surface "ตั้งอยู่ที่" (*is located at*). This target argument is relevant. Suppose there are intervening words[7] between them. The relevance tends to decrease. However, if we only select adjacent named entities to be the target arguments, the coverage may be limited. In our experiments, we varied the distances from 0 to 5 intervening words for observation.

Table 3 shows the numbers of relation instances when the distances are varied. For all relations, we observe that the numbers of relation instances do not significantly change after one word distance[8]. For example, we cannot extract more relation instances for MARRIEDWITH + PER, even we increased the distance. This indicates that using named entities helps to bound the number of possible arguments.

3.2 Preliminary results

To inspect the quality of relation instances extracted by our strategy, we randomly selected at most 50 instances of each relation for evaluation. Our evaluation procedure is as follows. Based on the assumptions (A1) and (A2), we expect that the subject (record title) of an instance should be relevant to its domain. We ignored instances whose subject is irrelevant. For example, the subject of the record no. 8026 is a person, but the volunteer assigned it to the cultural artifact domain. Note that this case rarely occurs, but exists. Next, a relation instance is considered to be correctly

[7]The intervening words include whitespace and punctuation tokens.

[8]This single word tends to be a whitespace token.

Record no.	Relation instance
Cultural attraction	
38481	ISLOCATEDAT(วัดโพธิ์ศรี, บ้านโพธิ์ศรี ต.อินทร์บุรี)
114585	ISBUILTIN(วัดเขาวงกฏ, ประมาณปี พ.ศ.2471-2573)
114333	ISBUILTBY(วัดปิตุลาธิราชรังสฤษฎิ์, กรมหลวงรักษ์รณเรศร์)
61446	HASOLDNAME(วัดหนองกันเกรา, วัดหนองตะเกรา)
Cultural person	
14125	MARRIEDWITH(นายเนาวรัตน์ พงษ์ไพบูลย์, นางประคองกูล อิศรางกูร ณ อยุธยา)
32530	HASFATHERNAME(พระครูประยุตนวการ, นายเหยม เดชมาก)
45389	HASMOTHERNAME(หลวงพ่อลั้ง สุทสฺสโน, นางพริ้ง แก้วแดง)
144574	HASOLDNAME(พระครูมงคลวรวัฒน์, สวัสดิ์ บุพศิริ)
145771	HASBIRTHDATE(อาจารย์ธนิสร์ ศรีกลิ่นดี, วันจันทร์ที่ 23 มกราคม 2494)
123678	BECOMEMONKIN(พระครูพิจิตรสิทธิคุณ, วันที่ ๑๖ เมษายน พ.ศ. ๒๕๒๘)
Cultural artifact	
160974	ISMADEBY(หนังสือประวัติคลองตำเนินสะดวก, พระครูสิริวรรณวิวัฒน์)
94286	ISSOLDAT(ข้าวเกรียบปากหม้อ, ตลาดเทศบาลพรานกระต่าย)

Table 5: Relation instances produced by our system.

extracted if its argument exactly matches the fact. For example, if our system only extracts the first name while the fact is the whole name, then we consider this instance to be incorrect. Finally, we set the maximum distance between the relation surface and its argument to 5. Table 4 shows the performance of our relation extraction. The overall results are surprisingly good, except those of HASOLDNAME and ISMADEBY. Table 5 shows some samples of relation instances produced by our system.

4 Related work

Named entity recognition has been applied to relation extraction. Hasegawa et al. (2004) propose an approach that discovers relations between two named entity types. Their approach clusters pairs of named entities using the similarity of context words intervening between them and assigns labels using frequent context words. In the Thai writing style, sentence boundary markers are absent, and subjects are often omitted. These two issues make it difficult to obtain two named entities in the same sentence. Our approach only considers one named entity and its preceding context words and uses simple templates to determine relation types.

Relation extraction can be simplified by focusing on unary relations. Hoffmann et al. (2010) present LUCHS, a self-supervised system that learns a large number of relation-specific extractors. Each extractor is trained according to an attribute of Wikipedia's infoboxes. Training data are created by matching attribute values with corresponding sentences. Their approach requires an article classifier to reduce the number of extractors to be invoked for prediction. The overall strategy fits well with Wikipedia data. Unfortunately, resources like infoboxes are not available in our data. Chen et al. (2011) propose an approach that learns relation types by using declarative constraints. The constraints capture regularities of relation expressions at various levels of linguistic structure,

including lexical, syntactic, and discourse levels. To learn a model, their approach requires a constituent-parsed corpus, which is generated automatically using the Stanford parser (de Marneffe and Manning, 2008). Such a high-quality parser is difficult to obtain in languages with limited NLP corpora and tools like Thai.

5 Conclusion

We successfully applied our approach to a cultural database and could discover more than 18,000 relation instances with expected high accuracy. The outputs of our relation extraction can be useful for other applications such as question answering or suggesting related topics based on semantic relations.

In future work, we plan to extract more relations, especially in the cultural artifact domain. We are interested in some relations like ISMADEOF which requires the NE type like materials. However, this NE type is not available in the current NE corpus. We will explore other techniques to constrain the noun phrases to prevent the semantic drift problem.

References

Agichtein, E. and Gravano, L. (2000). Snowball: Extracting relations from large plain-text collections. In *In Proceedings of ICDL*, pages 85–94.

Banko, M., Cafarella, M. J., Soderl, S., Broadhead, M., and Etzioni, O. (2007). Open information extraction from the web. In *Proceedings of IJCAI*, pages 2670–2676.

Charoenporn, T., Kruengkrai, C., Sornlertlamvanich, V., and Isahara, H. (2004). Acquiring semantic information in the tcl's computational lexicon. In *Proceedings of the Fourth Workshop on Asia Language Resources*.

Chen, H., Benson, E., Naseem, T., and Barzilay, R. (2011). In-domain relation discovery with meta-constraints via posterior regularization. In *Proceedings of ACL-HLT*, pages 530–540.

Crammer, K., McDonald, R., and Pereira, F. (2005). Scalable large-margin online learning for structured classification. In *Proceedings of NIPS Workshop on Learning With Structured Outputs*.

de Marneffe, M.-C. and Manning, C. D. (2008). The stanford typed dependencies representation. In *Proceedings of the workshop on Cross-Framework and Cross-Domain Parser Evaluation*, pages 1–8.

Hasegawa, T., Sekine, S., and Grishman, R. (2004). Discovering relations among named entities from large corpora. In *Proceedings of ACL*.

Hoffmann, R., Zhang, C., and Weld, D. S. (2010). Learning 5000 relational extractors. In *In ACL*.

Kok, S. and Domingos, P. (2008). Extracting semantic networks from text via relational clustering. In *Proceedings of ECML-PKDD*, pages 624–639.

Kruengkrai, C., Uchimoto, K., Kazama, J., Torisawa, K., Isahara, H., and Jaruskulchai, C. (2009). A word and character-cluster hybrid model for thai word segmentation. In *Proceedings of InterBEST: Thai Word Segmentation Workshop*.

Lin, D. and Pantel, P. (2001). Dirt-discovery of inference rules from text. In *Proceedings of KDD*, pages 323–328.

Pantel, P. and Pennacchiotti, M. (2006). Espresso: leveraging generic patterns for automatically harvesting semantic relations. In *Proceedings of ACL*, pages 113–120.

Poon, H. and Domingos, P. (2009). Unsupervised semantic parsing. In *Proceedings of EMNLP*, pages 1–10.

Schoenmackers, S., Etzioni, O., Weld, D. S., and Davis, J. (2010). Learning first-order horn clauses from web text. In *Proceedings of EMNLP*, pages 1088–1098.

Sornlertlamvanich, V., Charoenporn, T., and Isahara, H. (1997). *ORCHID: Thai Part-Of-Speech Tagged Corpus*. Technical Report TR-NECTEC-1997-001, NECTEC.

Theeramunkong, T., Boriboon, M., Haruechaiyasak, C., Kittiphattanabawon, N., Kosawat, K., Onsuwan, C., Siriwat, I., Suwanapong, T., and Tongtep, N. (2010). Thai-nest: A framework for thai named entity tagging specification and tools. In *Proceedings of CILC*.

Theeramunkong, T., Sornlertlamvanich, V., Tanhermhong, T., and Chinnan, W. (2000). Character cluster based thai information retrieval. In *Proceedings of IRAL*, pages 75–80.

Yates, A. and Etzioni, O. (2009). Unsupervised methods for determining object and relation synonyms on the web. *Journal of Artificial Intelligence Research*.