PAPER

Discovery of Predicate-Oriented Relations among Named Entities Extracted from Thai Texts

Nattapong TONGTEP^{†a)}, *Student Member* and Thanaruk THEERAMUNKONG^{†b)}, *Member*

SUMMARY Extracting named entities (NEs) and their relations is more difficult in Thai than in other languages due to several Thai specific characteristics, including no explicit boundaries for words, phrases and sentences; few case markers and modifier clues; high ambiguity in compound words and serial verbs; and flexible word orders. Unlike most previous works which focused on NE relations of specific actions, such as *work_for*, *live_in*, *located_in*, and *kill*, this paper proposes more general types of NE relations, called predicate-oriented relation (PoR), where an extracted action part (verb) is used as a core component to associate related named entities extracted from Thai Texts. Lacking a practical parser for the Thai language, we present three types of surface features, i.e. punctuation marks (such as token spaces), entity types and the number of entities and then apply five alternative commonly used learning schemes to investigate their performance on predicate-oriented relation extraction. The experimental results show that our approach achieves the F-measure of 97.76%, 99.19%, 95.00% and 93.50% on four different types of predicate-oriented relation (action-location, location-action, action-person and person-action) in crime-related news documents using a data set of 1,736 entity pairs. The effects of NE extraction techniques, feature sets and class unbalance on the performance of relation extraction are explored.

key words: relation extraction, named entity, surface feature, information extraction

1. Introduction

Recently several information extraction (IE) approaches have been proposed to transform an unstructured text into knowledge base, such as those in [1]–[3]. The extracted knowledge is often used in question-answering systems for response to 4W1H (who, what, when, where and how) questions. A number of named entity (NE) extraction methods have been proposed to find a part of words, such as a person name, a location and a time expression in several languages, such as English [4]–[8], Japanese [9]–[15] and Chinese [16], [17]. For more effective usage of extracted NEs, it is very useful to find relations among those entities.

Towards discovery of relations among NEs, named entity extraction as well as other preprocesses such as tokenization, sentence splitting, part-of-speech tagging and lemmatization, are usually applied [18]–[21]. As an early work on relation extraction, Ferrández et al. [4] extracted relations based on clause splitting of documents. In addition, this method also provided a resolution of anaphora

phenomenon between these entities using natural language processing (NLP) techniques. To discover relations among two named entities, a number of works [2], [3], [22] proposed methods to identify relations using context words between them. In [23], Agichtein and Cucerzan claimed that relation extraction from text documents was a harder task than named entity recognition. They proposed a general language modeling method for quantifying the difficulty of information extraction by predicting performance of named entity recognition such as location, organization, person name and miscellaneous named entities, and relation extraction such as birth dates, death dates and invention name. Zelenko et al. [24] proposed kernel methods with support vector machines (SVMs) for extracting relation among person-affiliation and organization-location. Culotta et al. [21] experimented on the Automatic Content Extraction (ACE) corpus using different features such as WordNet, parts of speech and entity types. Their results showed that the dependency tree kernel achieved a 20% F1 improvement over a “bag-of-words” kernel. As an early work on NE applications for biomedical domain, Hobbs [6] proposed a method called FASTUS (Finite-State Automation Text Understanding System) to extract information from biomedical texts using a finite-state transducer. Later Takeuchi and Collier [5] used SVMs to extract named entities without use of a large-scale domain-dependent dictionary and hand-made patterns but a small corpus of a text containing entities tagged by a domain expert. Recently some works have applied NE extraction to discover hidden pattern in social networks. Zhu et al. [25] presented a so-called CORDER system to find relations among entities in an organization’s documents on a social network. The mined knowledge was in the form of who works with whom, on which projects and with which customers, using strength measured for each co-occurring NE based on its co-occurrences and distances with the target. The CORDER comprised the steps of data selection, named entity recognition and ranking by relation strengths. As an integrated community project, tasks of entity and relation extraction from English, Chinese and Arabic texts were conducted in the Automatic Content Extraction (ACE) program*, including three sets of annotation tasks; Entity Detection and Tracking (EDT), Relation Detection and Characterization (RDC), and Event Detection and Characterization (EDC) [26]. Three main EDT tasks were the detection of entities mentioned in a document, the tracking of entities

Manuscript received July 21, 2011.

Manuscript revised February 2, 2012.

[†]The authors are with School of Information, Computer and Communication Technology, Sirindhorn International Institute of Technology, Thammasat University, Thailand.

a) E-mail: nattapong@siit.tu.ac.th

b) E-mail: thanaruk@siit.tu.ac.th

DOI: 10.1587/transinf.E95.D.1932

*<http://projects.ldc.upenn.edu/ace/>

across documents, and the detection of entities pre-defined in a database. In each task, it was necessary to recognize entities, entity attributes, entity mentions, mention roles and mention extents [26]. In this project, seven types of entities (person, organization, location, facility, weapon, vehicle and geo-political entity (GPEs)) and their subtypes (e.g., organization subtypes include government, commercial, educational, non-profit, other) were identified and then physical relations (e.g., Located, Near and Part-Whole) among entities were considered in Relation Detection and Characterization (RDC) task. Interaction, movement, transfer, creation and destruction events were five types of events identified and characterized in the Event Detection and Characterization (EDC) task.

In previous works [2], [3], [17], [18], [21], the detected NE relations are mostly of specific actions, such as *work_for* (person–organization), *live_in* (person–location) and *kill* (person–person). NE extraction or NE relation extraction from texts in English and other languages with sufficient processing tools, syntactic information can be provided to analyze the sentence structure and extract NEs, and semantic information is usually applied to detect relations among such NEs. However, named entity extraction in Thai is much more difficult than in other languages since a Thai text has no explicit boundary for words, phrases or even sentences. Even now we still lack practical syntactic tools, such as parsers to analyze the Thai texts. In Thai texts, there are punctuation marks, mostly spaces, but they are occasionally inserted between words, phrases or sentences. Without explicit boundaries among units, especially sentence boundaries, high difficulty in analyzing Thai texts is triggered in both syntactic and semantic level since it requires semantic, discourse or even pragmatic information to chunk the running text into a set of meaningful units. Since there is no sentence marker in the Thai language, a paragraph contains chunks of phrases which may relate loosely with each other at semantic level. Moreover, for some sentences, main subjects, verbs, or objects can be omitted or their orders can be swapped with valid meaning. This structural ambiguity seriously hinders any automated process on the Thai language. Because of these properties, it is extremely hard to create a general parser for the Thai language and then we lack tools for syntactic analysis. Several works [27]–[33] used clue words, space and discourse contexts to discover compound nouns and/or proper names, some of which are named entities. Kawtrakul et al. [34] summarized a state of the art of Thai language resources (e.g. corpus, lexicon and language analysis tools), Thai language behavior analysis (word, new word formation and sentence) and computational models (unknown word extraction, named entity identification, new word generation and noun phrase recognition). To the best of our knowledge, there are few works on extraction of relations among named entities in Thai and no intensive studies on which feature are useful. Moreover, since a predicate part (a verb or an action word) represents a central meaning of a chunk of texts, discovery of relations among such predicates and named entities becomes an im-

portant process towards the understanding of a text.

To this end, this paper studies how effective surface features can be useful to detect predicate-oriented relations among named entities in Thai crime-related news documents. The surface information, including the number of punctuations, the number of entities and the types of entities between a pair of entities, are applied as features to determine the existence of relation among the pair of entities. The effectiveness of features on relation extraction is investigated with the variety of supervised learning schemes. The rest is organized as follows. Section 2 describes named entities in Thai and other related entities. Our approach including types of relations, the framework and surface features are presented in Sect. 3. Section 4 describes experimental settings. In Sect. 5, the experimental results are discussed. Finally, a conclusion is illustrated in Sect. 6.

2. Thai Named Entities and Other Related Entities

Words in the Thai language are multi-syllabic and they can be loosely combined to form a new word. Since Thai has no inflection and no word delimiters, morphological processing is to recognize word boundaries, without recognizing a lexical form from a surface form, like in English. Moreover, in Thai there is no change in tense forms for verbs, singular or plural forms for both verbs and nouns. Uppercase and lowercase characters are unavailable. Some certain vowels, all tone marks and diacritics are written above and below the main consonant, but some vowels are written before and after the main consonant. Frequently it is possible to invent new words by means of compounding words or adding a prefix or suffix to a word. Like other languages, proper names such as person names and place names, occur frequently in a text, especially in news documents. Most Thai transliterated words come from English but some are derived from other European languages such as German. It seems trivial for Thai people to recognize a new word, a proper name or a transliterated word in a running text (a text without space or any delimiters) since we may use semantic when we interpret them. Anyway it seems extremely hard for a machine to detect such names or words if they are not in a dictionary. In Thai news documents, it is quite common to have proper names and transliterated words. Typically a news document constitutes headline (a top of the news document), leads (first paragraph after headline), neck (a connector between leads and body) and body (news detailed content) form a typical news structure as shown in Fig. 1.

In news documents, there are several types of named entities such as date, time, location, person name, product name, plant name and disease name, depending on their domains or categories. In our work, we focus on named entities involving with an event, such as person name (PER), location (LOC) including action (ACT), where each type of named entities can be viewed as a chunk of information. Moreover, all expressions except ACT, PER and LOC types can be defined as other entities (OTH) in our work.

<headline>ตร.เครียดจัด!! มาตัวตาย</headline>
Distressed Pattaya Police Officer Commits Suicide

<lead>ตั้งสนั่นโรงพัก ดาบตำรวจ เจ้าหน้าที่งานการเงิน สภ. เมืองพัทยา เครียดจัด!!
 ใช้อาวุธปืนระเบิดสมองตัวเอง เผยมีปัญหารื่องการเงินในระบบการเงินของโรงพัก ไปใช้ส่วนตัว
 แต่ไม่มีเงินนำมาคืน ล่าสุด ดาบตำรวจทนพิษบาดแผลไม่ไหว สิ้นใจที่โรงพยาบาล</lead>
*A financial police officer has committed suicide whilst on duty at Pattaya Police Station by shooting his head.
 It is believed he was severely depressed due to financial problems. He was brought to a hospital immediately but finally died.*

<body> พัทยา- วันนี้ (5 พ.ย. 51) เมื่อเวลา 11.00 น. พ.ต.ต.สุริยา วรวิงษ์ สารวัตรเวร สภ.เมืองพัทยา
 ได้รับแจ้งมีดาบตำรวจแผนกธุรการ ใช้อาวุธปืนยิงตัวเอง ภายในห้องทำงานฝ่ายการเงิน จึงรายงานให้
 พ.ต.อ.นภดล วงษ์นอม ผกก. พ.ต.ท.สุกซ์ชัย เปี่ยมมนัส รอง ผกก.(ป.) รับทราบแล้วรีบไปตรวจสอบ...
 </body>
*On November 5 2008 at 11.00 am. Police Major Suriya Wariwong was informed that a police officer had shot himself at
 the Pattaya police station. He notified Pattaya Superintendent Pol. Col. Noppadol Wongnorn and
 Deputy Superintendent Pol. Lt. Col. Supachuch Piammanus, before rushing to where the incident occurred...*

Fig. 1 An example news in Thai.

3. The Framework

The explanation has been done in two stages; training stage and testing stage (see Fig. 2). In the training stage, we applied a set of preparatory patterns to obtain possible named entities automatically (NE Extraction) from news documents and manually corrected their boundaries and types (NE Correction). After we marked a set of named entities with their assigned types in the news documents, we scanned throughout these documents to generate a set of possible pairs of named entities (Relation Candidate Generation). By a set of simple rules, we automatically filtered out some naively meaningless candidate relations (Relation Candidate Filtering). Thereafter, the remaining candidate relations are assigned with a class by a number of experts (Relation Candidate Labelling). Relation classes are of two types; *related* stands for relation existence and *unrelated* for relation non-existence. For each candidate relation, besides the relation class, a number of surface features are extracted (Features Extraction). The extracted surface features and the relation class are used as an input to learn a model for classification (RE Learning).

On the other hand, in the testing stage, a set of named entities with their types are extracted from a test news document. Preliminarily, they are manually corrected and used as an input to the system. Candidate relations are extracted and classified by a model learned from the training stage. For the purpose of evaluation, five-fold cross validation with stratified sampling technique is applied.

3.1 NE Extraction

Compared to other languages, named entities in Thai are difficult to extract due to some major causes as follows. Firstly, the Thai language lacks an explicit word boundary that can be used as a clue to detect named entities and sometimes a named entity is a string composed of substrings which are short words. Secondly, the Thai language has no explicit phrase and sentence boundaries which make us have

Fig. 2 The proposed framework.

no clue to help word segmentation to recognize words, including named entities. Thirdly, some components such as main subjects, verbs, or objects can be omitted from a sentence and/or their positions can be swapped, making us hard to use some kinds of grammar rules to analyze and obtain named entities. Lacking the explicit boundary markers and possessing of flexible language structure seriously hinders any automated process on the Thai language. To extract Thai NEs, the most common approach is to use a set of rules (patterns), which define NE structures. However, the order in applying these rules triggers ambiguity in NE extraction. To solve this, we developed a technique [35] to give priority to patterns based on longest word matching as well as pattern categorization and ordering, where the length of patterns and the popularity of patterns are considered, without assistance of word segmentation and part-of-speech tagging. While the reader can find the details from [35], we provide a brief description as follows.

For the longest word matching, if two possible patterns can be applied, we prefer the one that matches longer words to the one that outputs shorter ones. For the pattern categorization and prioritization, first we group patterns by NE

Table 1 Some examples of NE patterns and their instances.

Type	Sub type	Example of Patterns	Examples
LOC	Organizational	(สำนักงาน มหาวิทยาลัย วิทยาลัย)(...)	มหาวิทยาลัยธรรมศาสตร์ (Thammasat University)
LOC	Political	(จังหวัด อำเภอ)(...)	จังหวัดปทุมธานี (Pathum Thani Province)
LOC	Natural	(แม่น้ำ เกาะ ทะเลสาบ)(...)	เกาะสมุย (Samui Island)
PER	-	(นาย นางสาว รศ.)(...)	นายจอห์น (Mr. John)
ACT	>2 syllables	(กระ วีก ระ วาด ละ ลาน ตา สง เงียบ)	กระวีกระวาด (to hurry; four syllables)

types, second we sort NE types ascending by their ambiguity level, i.e., LOC, ACT, and PER, and last for patterns of each NE type, we order these patterns descending by their lengths. For the ambiguity level in NE type, NE type with more clue word is less ambiguous than those with fewer clue words or without clue words. Besides the above pattern prioritization, constraints of “minimum number of syllables” and “Thai character clusters (TCCs)” are applied. These constraints are considered when each pattern is applied to extract NEs from a running Thai text. For the first constraint, the resultant NEs are required to have more than two syllables (especially for ACT NEs). When a pattern is applied on a running text in order to extract an NE and the extracted NE is too short, in several cases it is likely to be a substring inside some regular words. This observation motivates towards the setting of this constraint. The second constraint is to apply rules to detect TCCs, each of which is an ambiguous and inseparable unit, usually smaller than a word. The detail of TCCs can be found in [36]. With assist of these two constraints, a set of patterns are constructed in the form of regular expressions for detecting named entities. Although there are various types of NEs, this work focuses on extracting named entities of four types, i.e., person (PER), location (LOC), action (ACT) and other entities (OTH). The pattern-based NE extraction algorithm including examples of regular expression in the set of patterns is discussed in detail in [35].

To detect NEs, we prepare a set of patterns for matching a free text and extracting NEs. Two types of patterns are used in this case. The first one is to use the direct pattern to extract, for example, กระวีกระวาด (to hurry), จังหวัดกรุงเทพ (Bangkok), บาร์ค โอบามา (Barack Obama). The second one is to use a clue for extracting, for example, นายxxx (Mr. xxx), บริษัท yyy จำกัด (YYY Co. Ltd.), ร้านอาหารzzz (zzz restaurant), where xxx, yyy, and zzz are named entities. As our preliminary experiments, we found out that 100% of LOC have clues, 50% of PER have clues, and 83% of ACT can be detected. Table 1 shows some examples of patterns and their instances applied to detect LOC, PER, and ACT.

The following indicates the steps to obtain patterns used for NE extraction.

- (1) We create an initial set of extraction patterns by manual. Since the patterns are quite limited. It is not difficult to construct those patterns.
- (2) We collect location/organization names and person names from some online sources, such as Thai

Table 2 List of dictionaries and word lists.

NE Type & Phrase	Information Type	No. of Records
PER	Title	555
PER	Title Abbreviation	340
PER	Title Exception	198
PER	Position	244
LOC	Amphoe	924
LOC	Province	76
LOC	Thai government department	2,449
LOC	Thai government organization	309
LOC	Common country name	192
LOC	Full country name	192
LOC	Capital city name	194
DAT-TIM	Time unit	26
DAT-TIM	Season	17
DAT-TIM	Number	30
DAT-TIM	Month	24
DAT-TIM	Era	16
DAT-TIM	Day	14
DAT-TIM	Year	24
PREP	Preposition for location	63
ACT	Verb phrase with three or more syllables	2,767
ACT	Verb	13,027
QUE	Question phrase with three or more syllables	6
CONJ	Conjunction phrase with three or more syllables	88
ADV	Adverb phrase with three or more syllables	912

Wikipedia[†], The Royal Institute^{††}, The Government Information System^{†††}. We collect a set of verbs with more than two syllables from YAI TRON dictionary^{††††}.

- (3) We apply the extraction names and patterns to tag names and actions in a set of news documents.
- (4) We check the output to find errors and then add more patterns or names to the list.
- (5) Repeat (3) and (4) until there is no change in the list.

With these steps, the final name list and pattern list can be summarized as shown in Table 2 and Table 3.

As stated above, the order of applying patterns may affect the extraction result, especially the ending or starting

[†]<http://th.wikipedia.org>

^{††}<http://www.royin.go.th>

^{†††}<http://www.oic.go.th/ginfo>

^{††††}<http://th.lug.wikia.com/wiki/YAI TRON> (English-Thai dictionary)

point of the matching patterns. Towards this, we apply the pattern in the order of small ambiguity first. For the ambiguity level in NE type, NE type with more clue word is less ambiguous than those with fewer clue words or without clue words. Figure 3 shows an example of applying patterns in different orders and their characteristics. Here, L is location, P is person, and A is action (verb with more than two syllables). By the order of LAP , we can get the correct extraction of named entities. Here, it is also possible to obtain incorrect extraction in the case of LAP while some orders may find suitable NEs. However, there are fewer mistakes for this order, compared to the other ones.

To extract name entities by patterns, we apply a set of basic rules to enumerate patterns in the order that we can efficiently acquire named entities. The basic rules are (1) to order the pattern by pattern ambiguity level in NE type, that is $LOC > ACT (> 2 \text{ syllables}) > PER$ and (2) to order the patterns in the order of their lengths. Intuitively, NE type with more clue word is less ambiguous than those with fewer clue words or without clue words.

Figure 4 shows the algorithm to extract NEs using our proposed NE extraction method. Given a set of news documents $D = \{d_1, d_2, d_3, \dots, d_{|D|}\}$ and a set of NE patterns or templates, $P = \{p_1, p_2, p_3, \dots, p_{|P|}\}$, the algorithm will output a set of named entities (NEs), $E = \{E_1, E_2, E_3, \dots, E_{|E|}\}$ grouped by their types. In this work, we focus on three main entity types ($|E| = 3$) where E_1 , E_2 , and E_3 correspond to the sets of extracted NEs of LOC, ACT and PER, respectively. Moreover, the algorithm occupies one main function namely *PatternOrdering* while *NETypeGrouping*, *GroupByPattern* including *OrderingByLength* and *ExtractByPattern* are four sub-functions. The *PatternOrdering* function is to group and reorder the set of patterns based on NE types (principally sorted by ascend-

ing order of ambiguity level of NE types), i.e., LOC, ACT, PER and OTH, and moreover for patterns of each NE type, ordered in descending manner of their lengths. The output is a set of $|E|$ final pattern groups, each of which is for

Pattern-based NE extraction algorithm

```

 $D = \{d_1, d_2, d_3, \dots, d_{|D|}\}$ 
# a set of news documents

 $P = \{p_1, p_2, p_3, \dots, p_{|P|}\}$ 
# a set of NE patterns or templates

 $E = \{E_1, E_2, E_3, \dots, E_{|E|}\}$ 
# a sets of extracted NEs from the document set  $D$ 
# grouped by the NE type

main(){
  if(updated( $P$ )) {
     $FPG = \text{PatternOrdering}(P)$ ;
  }
  foreach  $E_i \in E$  {
     $E_i = \phi$ ;
  }
  foreach  $d_l \in D$  {
    foreach  $FPG_i \in FPG$  {
      foreach  $FPG_{ij} \in FPG_i$  {
        foreach  $FPG_{ijk} \in FPG_{ij}$  {
           $NE_{ijkl} = \text{ExtractByPattern}(FPG_{ijk}, d_l)$ ;
           $E_i = E_i \cup \{NE_{ijkl}\}$ ;
        }
      }
    }
  }
}

function PatternOrdering( $P$ ) {
   $PG = \text{NETypeGrouping}(P)$ ;
   $FPG = \phi$ ;
  foreach  $PG_i \in PG$  {
     $PPG_i = \text{GroupByPattern}(PG_i)$ ;
     $FPG_i = \phi$ ;
    foreach  $PPG_{ij} \in PPG_i$  {
       $LPG_{ij} = \text{OrderingByLength}(PPG_{ij})$ ;
       $FPG_i = FPG_i \cup \{LPG_{ij}\}$ ;
    }
     $FPG = FPG \cup \{FPG_i\}$ ;
  }
  return  $FPG$ ;
}

```

Fig. 4 Our proposed pattern-based NE extraction algorithm.

Table 3 No. of patterns for each NE type and phrase.

NE Type & Phrase	No. of patterns
DAT-TIM	350
LOC	9,284
QUE	6
CONJ	88
ACT	2,767
ADV	912
PER	2,582
Total	15,077

Sentence	
Thai	[จังหวัดนครนายก] [นายฉลอง จงประดับ] [กำลังขับ] [รถเก๋งสีเทา...]
English	[Nakhon Nayok Province], [Mr. Chalong Chongpradab] [was driving] [a bronze car...]
Order	Result
LAP	[จังหวัดนครนายก] _L (นาย) _{clue} [ฉลอง จงประดับ] _P [กำลังขับ] _A รถเก๋งสีเทา...
LPA	[จังหวัดนครนายก] _L (นาย) _{clue} [ฉลอง] _P จงประดับ [กำลังขับ] _A รถเก๋งสีเทา...
PLA	จังหวัดนคร(นาย) _{clue} [ก] _P (นาย) _{clue} [ฉลอง] _P จงประดับ [กำลังขับ] _A รถเก๋งสีเทา...
PAL	จังหวัดนคร(นาย) _{clue} [ก] _P (นาย) _{clue} [ฉลอง] _P จงประดับ [กำลังขับ] _A รถเก๋งสีเทา...
APL	จังหวัดนคร(นาย) _{clue} [ก] _P (นาย) _{clue} [ฉลอง] _P จงประดับ [กำลังขับ] _A รถเก๋งสีเทา...
ALP	[จังหวัดนครนายก] _L (นาย) _{clue} [ฉลอง] _P จงประดับ [กำลังขับ] _A รถเก๋งสีเทา...

Fig. 3 An example of applying NE patterns in different orders.

Sentence
Thai: นายอาคมฆ่านายสมคิดที่รัชดาผับ หลังก่อเหตุ นายอาคมได้หลบหนีไปที่จังหวัดราชบุรี
English: Mr. Akom killed Mr. Somkid at Ratchada Pub. Thereafter Mr. Akom escaped to Ratchaburi Province.
Entity Identification
[PER01:นายอาคม][ACT01:ฆ่า][PER02:นายสมคิด][OTH01:ที่][LOC01:รัชดาผับ]<space>
[PER01:Mr. Akom][ACT01:killed][PER02:Mr. Somkid][OTH01:at][LOC01:Ratchada Pub]<space>
[OTH02:หลังก่อเหตุ]<space>[PER03:นายอาคม][ACT02:ได้หลบหนีไป][OTH03:ที่][LOC02:จังหวัดราชบุรี]
[OTH02:Thereafter]<space>[PER03:Mr. Akom][ACT02:escaped][OTH03:to][LOC02:Ratchaburi Province]

Fig. 5 Named entity identification.

Sentence: นายเอ ไปพบเจ้าหน้าที่ กระทรวงต่างประเทศ
[PER01:นายเอ] [ACT01:ไป][ACT02:พบ][OTH01:เจ้าหน้าที่ กระทรวงต่างประเทศ]
Mr. A went and met a staff member of foreign affairs.
[PER01:นายเอ] [ACT01:ไป][ACT02:พบ][OTH01:เจ้าหน้าที่] [LOC01:กระทรวงต่างประเทศ]
Mr. A went and met a staff at Ministry of Foreign Affairs.
[PER01:นายเอ] [ACT01:ไป][OTH01:พบ][OTH02:เจ้าหน้าที่ กระทรวงต่างประเทศ]
Mr. A went to meet a staff member of foreign affairs.
[PER01:นายเอ] [ACT01:ไป][OTH01:พบ][OTH02:เจ้าหน้าที่] [LOC01:กระทรวงต่างประเทศ]
Mr. A went to meet a staff at Ministry of Foreign Affairs.
[PER01:นายเอ] [ACT01:ไปพบ][OTH01:เจ้าหน้าที่ กระทรวงต่างประเทศ]
Mr. A met a staff member of foreign affairs.
[PER01:นายเอ] [ACT01:ไปพบ][OTH01:เจ้าหน้าที่] [LOC01:กระทรวงต่างประเทศ]
Mr. A met a staff at Ministry of Foreign Affairs.

Fig. 6 An example of ambiguity in named entity boundary.

Fig. 7 Predicate-oriented relation.

each entity type, $FPG = \{FPG_1, FPG_2, \dots, FPG_{|E|}\}$. The *NETypeGrouping* sub-function separates NE extraction patterns or templates into a number of groups based on their types. The *GroupByPattern* sub-function further groups NE patterns with the same type into a set of subgroups based on their pattern similarity. The *OrderingByLength* sub-function rearranges the order of NE patterns or templates in the same subgroup by the pattern lengths. The *ExtractByPattern* sub-function applies the NE patterns or templates (FPG_{ijk}) on a certain document in order to extract named entities from the document (NE_{ijkl}).

Figure 5 shows an example of a sentence in Thai and the results of entity identification. After named entities are extracted, relation candidates are generated by assuming a relation among any pair of named entities within the same paragraph.

3.2 NE Correction

Normally, the extracted NEs may not be always suitable since the Thai language itself has high structural ambiguity, seriously affecting the accuracy of tokenization of named entities (e.g. person name, location, action and other miscellaneous entities). When the set of NE-identification patterns is insufficient, some errors may occur in both NE boundary identification and NE type recognition. That is, some named entities are detected with shorter or longer strings, or even their boundaries are detected correctly, they may be classified incorrectly to another type (see Fig. 6). These errors trigger performance drop in relation extraction, the succeeding process. To solve this issue, we omit and/or repair incorrect named entities manually to create a corpus for training.

3.3 Predicate-oriented Relation Types

In the past, several works [2], [3], [17], [18], [21] focused on how to identify relevant entities (e.g., person, location and organization) and their relations (e.g., *work_for*, *live_in* and *kill*) such as *work_for* (person-organization), *live_in* (person-location) and *kill* (person-person). In contrast to these previous works, this work focuses on predicate-oriented relations among relevant entities (e.g., person, location and action) and their relationships (e.g., *related* and *unrelated*) are identified such as *related* (person-action), *unrelated* (person-action), *related* (action-person), *unrelated* (action-person), *related* (action-location) and *unrelated* (action-location). Suppose two related relations are *related* (person-action) and *related* (action-location), and action in both relations occur at the same position in a running text, then we can assume a relation for the pair of relations to be (person-action-location) (see Fig. 7). Unlike the previous works which consider only some specific actions (predicates) (i.e., work for, located in, live in, kill, and etc.), this work focuses on all possible kinds of verbs or actions in texts. They are treated as entities and their relationships will be *related* or *unrelated*. In summary, four types of predicate-oriented relations are: (1) person and action [PER-ACT], (2) action and location [ACT-LOC], (3) location and action [LOC-ACT], and (4) action and person [ACT-PER] (see Table 4).

The task of relation recognition between entities ([PER-ACT], [ACT-LOC], [LOC-ACT] and [ACT-PER]) in this work is similar to Semantic Role Labeling (SRL) (also called Predicate Argument Structure Analysis). In the SRL approach, for each sentence, verbal arguments are identi-

Table 4 Types of relations.

Type	Relation	Class	Relation Definition
ACT-LOC	action-location	<i>related, unrelated</i>	location (LOC) where the action (ACT) takes place.
LOC-ACT	location-action	<i>related, unrelated</i>	location (LOC) where the action (ACT) takes place.
ACT-PER	action-person	<i>related, unrelated</i>	person (PER) who is the theme of the action (ACT).
PER-ACT	person-action	<i>related, unrelated</i>	person (PER) who acts (ACT).

Table 5 An example of relation identification.

Example	Relation	Class
[PER01:นายอาคม][ACT01:ฆ่า] [PER01:Mr. Akom][ACT01:killed]	PER-ACT	<i>related</i>
[PER01:นายอาคม][ACT02:โดนหลบหนีไป] [PER01:Mr. Akom][ACT02:escaped]	PER-ACT	<i>related</i>
[PER02:นายสมคิด][ACT02:โดนหลบหนีไป] [PER02:Mr. Somkid][ACT02:escaped]	PER-ACT	<i>unrelated</i>
[PER03:นายอาคม][ACT02:โดนหลบหนีไป] [PER03:Mr. Akom][ACT02:escaped]	PER-ACT	<i>related</i>
[ACT01:ฆ่า][PER02:นายสมคิด] [ACT01:killed][PER02:Mr. Somkid]	ACT-PER	<i>related</i>
[ACT01:ฆ่า][PER03:นายอาคม] [ACT01:killed][PER03:Mr. Akom]	ACT-PER	<i>unrelated</i>
[ACT01:ฆ่า][LOC01:รัชดาภิเษก] [ACT01:killed][LOC01:Ratchada Pub]	ACT-LOC	<i>related</i>
[ACT01:ฆ่า][LOC02:จังหวัดราชบุรี] [ACT01:killed][LOC02:Ratchaburi Province]	ACT-LOC	<i>unrelated</i>
[ACT02:โดนหลบหนีไป][LOC02:จังหวัดราชบุรี] [ACT02:escaped][LOC02:Ratchaburi Province]	ACT-LOC	<i>related</i>
[LOC01:รัชดาภิเษก][ACT02:โดนหลบหนีไป] [LOC01:Ratchada Pub][ACT02:escaped]	LOC-ACT	<i>unrelated</i>

fied and other arguments are labeled as semantic roles (e.g., agent/patient). SRL treats all types of roles while our approach tries to focus on ACT, PER and LOC. One dominant feature is that this work focuses on the Thai language. In the Thai language, it is quite common that the main subject, object or even verb is omitted or their positions in a sentence are swapped. Therefore the two relations such as [LOC-ACT] and [ACT-LOC] need to be distinguished. In [LOC-ACT], LOC is the preceding entity (PE) and ACT is the succeeding entity (SE), while for [ACT-LOC], ACT is the preceding entity (PE) and LOC is the succeeding entity. From the example of named entities in Fig. 5, the relations among them can be investigated as shown in Table 5.

3.4 Relation Candidate Generation and Filtering

In a Thai news document, all four types of predicate-oriented relations are generated. A sentence marker in Thai is not available like a punctuation mark ‘full stop’ (.) in English, but a paragraph usually comprises one main idea. To simplify the task with a reasonable constraint, generated relation candidates are filtered out. Only pairs of named entities located in the same paragraph in a document are considered.

3.5 Relation Candidate Labelling

Towards the construction of ground truth data, we developed a system for relation class labelling as shown in Fig. 8.

Table 6 Surface features.

Type	Feature	Definition
eTYP	1L-PE	PE’s left-hand-side entity
	1R-PE	PE’s right-hand-side entity
	1L-SE	SE’s left-hand-side entity
	1R-SE	SE’s right-hand-side entity
SPC	Space(1L-PE,PE)	Existence of token space between 1L-PE and PE
	Space(1R-PE,PE)	Existence of token space between 1R-PE and PE
	Space(1L-SE,SE)	Existence of token space between 1L-SE and SE
	Space(1R-SE,SE)	Existence of token space between 1R-SE and SE
nENT	nEntity(PE,SE)	Number of entities between PE and SE

The system will guide a user by a set of instructions to label each element properly by indicating predicate-oriented relation types with their definitions as well as some examples. Users need to understand the type of relations and carefully read a news document, and then provide the correct answer for each relation class (*related* or *unrelated*). All pairs of relations with the labels answered by experts will be used together with surface features for learning process.

3.6 Feature Extraction

In the past, several works were proposed to exploit context (neighboring) words, relations in the WordNet, parts of speech and entity types as features, for detecting NE relations surrounding some specific actions (e.g., work for, located in, live in, and kill). Unfortunately, there have been no works focusing on extracting NE relations related to a general action and no intensive studies on which feature are useful. In this work, we explore a set of simple context features found among NE relation pairs, without considering additional linguistic resources such as WordNet or part of speech tags. In our approach, we focus on surface information and type of entities. Among a pair of entities, a set of surface features are extracted. These two entities are called the preceding entity (PE) and the succeeding entity (SE), respectively. To tag relations, we can exploit contexts in a text. This work exploits three sets of surface features; the existence of token space (SPC), the number of named entities (nENT) and entity type (eTYP) surrounding PE and SE. Furthermore, we determine five well-known classification models described in Sect. 4 to study the effectiveness of our surface features. Table 6 shows the definition of surface features. These surface features can be used to extract relations. Table 7 shows four example cases of extracting surface features.

Fig. 8 Our relation labelling system.

Table 7 Four cases of extracted surface features when considering whether the PE relates with the SE or not.

Entity	PER-ACT	ACT-PER	ACT-LOC	LOC-ACT
PE (Preceding Entity)	[PER01:นายอาคม]	[ACT01:ฆ่า]	[ACT01:ฆ่า]	[LOC01:รัชดาฯ]
SE (Succeeding Entity)	[ACT02:โดนหลบหนีไป]	[PER03:นายอาคม]	[LOC01:รัชดาฯ]	[ACT02:โดนหลบหนีไป]
Extracted Feature				
1L-PE	-	PER	PER	OTH
1R-PE	ACT	PER	PER	OTH
1L-SE	PER	OTH	OTH	PER
1R-SE	OTH	ACT	OTH	OTH
Space(1L-PE,PE)	0	0	0	0
Space(1R-PE,PE)	0	0	0	1
Space(1L-SE,SE)	0	1	0	0
Space(1R-SE,SE)	0	0	1	0
nEntity(PE,SE)	6	4	2	2
Class	<i>related</i>	<i>unrelated</i>	<i>related</i>	<i>unrelated</i>

3.7 Handling of Unbalanced Class Distribution

The original data set for the relation extraction task in this work faces with the problem of unbalanced class distribution. To deal with unbalanced class distribution problem, three possible solutions are (1) to assign distinct costs to the classification errors, (2) to resample the original a training set, either by over-sampling the minority class and/or under-sampling the majority class until the classes include an equal or almost equal number of samples, and (3) to include an internal process to bias the discrimination-based process so as to reimburse for the class imbalance.

Since our data set comprises 1,736 pairs of named en-

Table 8 Balancing class distribution (original vs. over-sampling).

Relation Type	Majority Class (<i>unrelated</i>) Original	Minority Class (<i>related</i>) Original	Minority Class (<i>related</i>) Over-sampling
ACT-LOC	346	109	327
LOC-ACT	450	39	429
ACT-PER	260	77	231
PER-ACT	333	122	244

tity with four types of relations in crime-related news documents (1,389 entity pairs for *unrelated* relation and 347 entity pairs for *related* relation) which is small, we focus on the second type of the solutions by over-sampling the mi-

nority class (see Table 8). The last column shows the over-sampling results when the factors are 3, 11, 3 and 2 for ACT-LOC, LOC-ACT, ACT-PER and PER-ACT, respectively.

4. Experimental Settings

This section shows the experimental settings for evaluating our predicate-oriented relation extraction among named entities. By preliminary experiments, we found out that the highest accuracy was obtained from the crime category, compared to other categories (i.e., economics, politics, foreign and sports). Moreover, news documents in this category are well written and their NEs seem easily to detect. By these reasons, this work focuses on extraction of predicate-oriented relations from Thai news documents in the category of crime. To do this, we collected crime-related news documents from three Thai news publishers: Kom Chad Luek[†], Daily News Online^{††} and Manager Online^{†††}. From these news documents, we exploit a number of patterns to detect well-formed named entities and then apply a classification method to identify predicate-oriented relations between NE pairs. With the limitation of time and labor, we have collected 1,736 pairs of relations in Thai news documents (crime) for our experiments.

In general, it is possible to use various classification models for detecting predicate-oriented relations. In this work, we compare five well-known classification models; decision tree, random forest, naïve Bayes, rule learner and support vector machines. Here, the features used are surface features as shown in Sect.3.6. A decision tree (DT: C4.5)[37] represents a natural and intuitive divide-and-conquer method to classify a pattern through a sequence of questions. Random forest (RF)[38] forms a supervised model using techniques of bagging ensembles of random trees. Naïve Bayes (NB)[39] is a probabilistic classifier using the normal distribution to model numeric attributes. Rule learner (RL)[40] or repeated incremental pruning to produce error reduction (RIPPER) uses a separate-and-conquer method to build a rule set by repeatedly adding rules to an empty rule set until all positive examples are covered. As the last model, support vector machines (SVMs)[41] apply a well-known kernel-based method that utilizes support vectors to maximize both classification accuracy and margin between the classes.

To evaluate our proposed method, we have performed a number of experiments with the five-fold cross validation and the stratified sampling technique. Four measures, i.e. precision (P), recall (R), F-measure (F) and accuracy (ACC) are used. Since we focus on detection of predicate-oriented relations among named entities, therefore the precision, recall and F-measure in the experiments are of the *related* class. In contrast, the accuracy indicates the combined classification performance of both *related* and *unrelated* classes.

5. Results and Discussion

In this section, the experimental results are discussed in

Fig. 9 The average F-measure of *related* class, comparing different sets of surface features. (averaged from four relation types and five learning schemes)

three aspects. For all investigations, we discuss both the unbalanced class distribution (original) and the balanced class distribution (over-sampling). Here, the over-sampling is done to increase the number of positive class (*related* class) to be comparable with the number of negative class (*unrelated* class). For investigation of the first aspect, we explore performance of surface features on predicate-oriented relation extraction and then investigate whether a reduced feature set still preserve performance of relation extraction or not. Here, the performance is calculated by averaging f-measures and accuracies obtained from four relations and five learning models. For exploration of the second aspect, the five well-known classifiers (i.e., DT, RF, NB, RL and SVMs) are compared to investigate their performances on predicate-oriented relation extraction, which is binary classification between *related* and *unrelated* classes. Moreover, the feature set used for comparison is the three surface features, i.e., SPC+eTYP+nENT (the full set) while four predicate-oriented relations focused are ACT-LOC, LOC-ACT, ACT-PER and PER-ACT. As for the last aspect, we discuss on the detailed results of both original dataset (an unbalanced dataset) and the over-sampling dataset (a balanced dataset) for all combinatorial cases.

5.1 Exploring Surface Feature Performance

In this experiment, we compare the performance of different sets of surface features for both environments of unbalanced and balanced class distribution. The performance of surface features is calculated as the average among four relation types and five learning models.

As the result in Fig. 9, only SPC seems insufficient for relation classification, especial for unbalanced class distribution (original). It gained 0% F-measure for the *related* class. The performance of a set of three surface features (SPC+eTYP+nENT) is higher than a set of two sur-

[†]<http://www.komchadluek.net>

^{††}<http://www.dailynews.co.th>

^{†††}<http://www.manager.co.th>

Fig. 10 The average accuracy, comparing different sets of surface features. (averaged from four relation types and five learning schemes)

face features (SPC+eTYP, SPC+nENT and eTYP+nENT) and a set of single surface feature (SPC, eTYP and nENT). Among four predicate-oriented relation types and five learning schemes using balanced class distribution, the lowest average F-measure, i.e., 27.53%, is obtained when only SPC is used. On the other hand, SPC+eTYP+nENT achieves the highest average F-measure of 80.97%. However, with balanced class distribution, SPC is highly significant to boost the average F-measure when using SPC as a surface feature together with eTYP (SPC+eTYP; from 64.55% to 72.64%) or nENT (SPC+nENT; from 59.48% to 69.97%). The combination of eTYP and nENT triggers the increment of the average F-measure over the single eTYP or the sole nENT (eTYP+nENT; from 64.55% and 59.48% to 75.32%). Focusing on a single surface feature, eTYP is the most effective surface feature to discover predicate-oriented relations (64.55%), followed by nENT (59.48%) and SPC (27.53%). With unbalanced class distribution, the lowest average F-measure is obtained from SPC and nENT while SPC+eTYP+nENT achieves the highest average F-measure (30.31%). However, the performance is very low compared to balanced class distribution.

Unlike F-measure, since the accuracy considers both *related* and *unrelated* classes, its value is relatively high, especially for the unbalanced class distribution (original) as shown in Fig. 10. The trend of F-measure and accuracy is similar when the distribution of class is balanced.

We conclude that the performance of a set of three surface features (SPC+eTYP+nENT) is the most effective surface feature set since it achieves the highest average F-measure and accuracy, followed by a set of two surface features (SPC+eTYP, SPC+nENT and eTYP+nENT) and then a set of single surface feature (SPC, eTYP and nENT).

5.2 Exploring Model Performance

In the second aspect, we compare the performance of predicate-oriented relation extraction gained by either of DT, RF, NB, RL or SVMs. Here all three surface features (SPC+eTYP+nENT) are assumed since this condition

Fig. 11 The F-measure of *related* class, comparing four different relation types. (Feature: SPC+eTYP+nENT, Data: Original)

seems suitable for learning a better classification model.

According to the results shown in Fig. 11, SVMs achieve the highest average F-measure (49.64%), followed by RL (39.67%), DT (38.86%), RF (38.02%) and NB (35.01%) for the unbalanced class distribution. Similar to several applications, SVMs seems the best classifier since it can catch the classification bias well. The others obtained comparable performance. However, with unbalanced class distribution, all learning models gain low average F-measure, usually less than 50%. For ACT-LOC, SVMs achieve the highest F-measure of 73.00% while RF obtains the lowest F-measure i.e., 48.73%. In LOC-ACT, surprisingly SVMs cannot detect the small positive class of the unbalanced class distribution. One reason is that the LOC-ACT is very unbalanced with the factor of 11 between *related* and *unrelated* class. The highest F-measure i.e., 34.92%, is obtained from NB. However, SVMs achieve the highest F-measure 59.83% and 65.71% for ACT-PER and PER-ACT, respectively. The lowest f-measures of ACT-PER and PER-ACT are 17.58% and 32.61%, respectively. They are obtained from NB. For the unbalanced class distribution, SVMs achieve the highest F-measure (73.00%) for ACT-LOC, followed by PER-ACT (65.71%) and ACT-PER (59.83%), in order.

For the result of sampling (balanced class distribution), shown in Fig. 12, SVMs achieve the highest average F-measure (96.36%), followed by RF (88.10%), DT (78.24%), RL (74.23%) and NB (67.90%), in order. For this case, SVMs still achieve the best performance but the F-measure is far better than the unbalanced case. The SVMs can catch characteristics of relations when they are balanced. RF is another good model for balanced class distribution due to its ensemble property. The highest F-measure of ACT-LOC is obtained from SVMs (97.76%) while the lowest F-measure (73.94%) is obtained from RL. In LOC-ACT, SVMs generate the highest F-measure (99.19%) while NB performed the worst with the lowest F-measure (79.14%). SVMs also achieve the highest F-measure of 95.00% and 93.50% for ACT-PER and PER-ACT, respectively. Moreover, NB obtained the lowest f-measures for ACT-PER and PER-ACT,

Fig. 12 The F-measure of *related* class, comparing four different relation types. (Feature: SPC+eTYP+nENT, Data: Sampling)

Fig. 13 The accuracy of four different relation types. (Feature: SPC+eTYP+nENT, Data: Original)

i.e., 63.25% and 55.01%. For the balanced class distribution, SVMs achieves the highest F-measure (99.19%) for LOC-ACT, followed by ACT-LOC (97.76%), ACT-PER (95.00%) and PER-ACT (93.50%), in order.

In Fig. 13, SVMs achieve the highest average accuracy (87.60%), followed by DT (83.16%), RL (82.92%), NB (80.74%) and RF (78.77%), in order, for the unbalanced class distribution. Moreover, SVMs achieve the highest accuracy for LOC-ACT (92.02%), followed by ACT-LOC (88.13%), ACT-PER (86.05%) and PER-ACT (84.18%), in order. RF generates the lowest accuracy for ACT-LOC, LOC-ACT and ACT-PER, i.e., 77.80%, 88.96% and 74.48% while NB gains the lowest accuracy of 72.75% for PER-ACT.

In Fig. 14, SVMs achieve the highest average accuracy (96.64%), followed by RF (88.37%), DT (79.63%), RL (77.68%) and NB (71.25%), in order, for the balanced class distribution. Furthermore, SVMs achieve the highest accuracy for all predicate-oriented relations while NB performed the worst with the lowest accuracy.

We conclude that with balanced class distribution, SVMs achieve the highest average F-measure and accuracy

Fig. 14 The accuracy of four different relation types. (Feature: SPC+eTYP+nENT, Data: Sampling)

among four predicate-oriented relations, followed by RF, DT, RL and NB, in order. Moreover, LOC-ACT is the easiest predicate-oriented relation type since it gains the highest F-measure and accuracy, followed by ACT-LOC, ACT-PER and PER-ACT, in order. The experimental results are the reflection of the structure of Thai news texts. For a paragraph in Thai news texts that includes a location followed by an action, the location and the action are likely to be related. An action usually followed by a person name or a location, but a person name is likely to be followed by other entities (e.g., person's position, age or a location).

5.3 Detailed Comparison

For the third aspect, we explore the detailed result of seven feature sets, four predicate-oriented relation types and five learning models. The experimental results of predicate-oriented relations for either unbalanced class distribution (original) or balanced class distribution (sampling) among surface features and classification schemes are compared in detail as shown in Table 9, Table 10, Table 11 and Table 12 for ACT-LOC, LOC-ACT, ACT-PER and PER-ACT, respectively.

Observations can be made as follow.

1. We can observe that some features are not effective to classify *related* and *unrelated* classes such as SPC and some conditions of nENT and eTYP+nENT.
2. SVMs perform the best in several cases, except LOC-ACT and ACT-PER for unbalanced class distribution.
3. It is difficult to classify relations in the situation of unbalanced class distribution. That is real practical situation. With sampling to form balanced class distribution, we can improve classification performance. Anyway, it is an artifact case.
4. The F-measure performance order of classifying *related* class in the unbalanced class distribution is ACT-LOC>PER-ACT>ACT-PER>LOC-ACT.

Table 9 The experimental results for the predicate-oriented relation of ACT-LOC.

Model	Feature	Original (Unbalanced)				Sampling (Balanced)			
		P	R	F	ACC	P	R	F	ACC
DT	SPC	0.00	0.00	N/A	76.04	54.60	54.43	54.51	55.87
	eTYP	49.06	23.85	32.10	75.82	76.19	68.50	72.14	74.30
	nENT	86.67	35.78	50.65	83.30	74.90	57.49	65.05	69.99
	SPC+eTYP	48.84	38.53	43.08	75.60	77.09	76.15	76.62	77.41
	SPC+nENT	77.36	37.61	50.61	82.42	76.00	81.35	78.58	78.45
	eTYP+nENT	86.67	35.78	50.65	83.30	74.90	57.49	65.05	69.99
RF	SPC+eTYP+nENT	75.93	37.61	50.30	82.20	76.25	88.38	81.87	80.98
	SPC	16.67	0.92	1.74	75.16	54.47	63.30	58.55	56.45
	eTYP	47.37	24.77	32.53	75.38	74.92	74.01	74.46	75.33
	nENT	75.93	37.61	50.30	82.20	71.80	58.41	64.42	68.65
	SPC+eTYP	43.18	34.86	38.58	73.41	77.81	84.71	81.11	80.84
	SPC+nENT	52.13	44.95	48.27	76.92	77.49	90.52	83.50	82.62
NB	eTYP+nENT	75.93	37.61	50.30	82.20	71.80	58.41	64.42	68.65
	SPC+eTYP+nENT	54.55	44.04	48.73	77.80	83.25	98.78	90.35	89.75
	SPC	0.00	0.00	N/A	76.04	54.40	52.91	53.64	55.56
	eTYP	52.33	41.28	46.15	76.92	74.46	63.30	68.43	71.63
	nENT	0.00	0.00	N/A	76.04	60.82	81.65	69.71	65.54
	SPC+eTYP	53.57	41.28	46.63	77.36	71.24	74.57	72.87	70.74
RL	SPC+nENT	52.38	20.18	29.14	76.48	63.02	74.01	68.07	66.28
	eTYP+nENT	0.00	0.00	N/A	76.04	60.82	81.65	69.71	65.54
	SPC+eTYP+nENT	63.10	48.62	54.92	80.88	72.38	76.15	74.22	74.29
	SPC	0.00	0.00	N/A	75.82	56.32	47.71	51.66	56.61
	eTYP	45.88	35.78	40.21	74.51	73.22	66.06	69.46	71.77
	nENT	77.78	38.53	51.53	82.64	76.13	56.57	64.91	70.29
SVMs	SPC+eTYP	51.00	46.79	48.80	76.48	73.77	73.09	73.43	74.29
	SPC+nENT	66.20	43.12	52.22	81.10	77.22	66.36	71.38	74.15
	eTYP+nENT	77.78	38.53	51.53	82.64	76.13	56.57	64.91	70.29
	SPC+eTYP+nENT	78.12	45.87	57.80	83.96	79.09	69.42	73.94	76.23
	SPC	0.00	0.00	N/A	76.04	54.95	79.82	65.09	58.40
	eTYP	80.00	3.67	7.02	76.70	81.91	76.15	78.92	80.24
	nENT	0.00	0.00	N/A	76.04	65.01	76.15	70.14	68.50
	SPC+eTYP	75.00	35.78	48.45	81.76	87.54	88.07	87.80	88.11
	SPC+nENT	65.93	55.05	60.00	82.42	83.12	79.82	81.44	82.32
	eTYP+nENT	0.00	0.00	N/A	76.04	88.95	93.58	91.21	91.23
	SPC+eTYP+nENT	80.22	66.97	73.00	88.13	95.61	100.00	97.76	97.77

Table 10 The experimental results for the predicate-oriented relation of LOC-ACT.

Model	Feature	Original (Unbalanced)				Sampling (Balanced)			
		P	R	F	ACC	P	R	F	ACC
DT	SPC	0.00	0.00	N/A	92.03	68.55	65.03	66.74	68.37
	eTYP	0.00	0.00	N/A	92.03	76.19	68.50	72.14	74.30
	nENT	0.00	0.00	N/A	92.03	68.13	65.27	66.67	68.14
	SPC+eTYP	0.00	0.00	N/A	92.03	83.33	90.91	86.96	86.69
	SPC+nENT	55.56	12.82	20.83	92.23	79.96	95.80	87.17	86.24
	eTYP+nENT	0.00	0.00	N/A	91.41	85.63	100.00	92.26	91.81
RF	SPC+eTYP+nENT	36.36	10.26	16.00	91.41	87.02	100.00	93.06	92.72
	SPC	0.00	0.00	N/A	92.03	65.79	64.10	64.93	66.22
	eTYP	37.50	7.69	12.76	91.62	72.26	92.31	81.06	78.95
	nENT	0.00	0.00	N/A	92.03	67.98	68.30	68.14	68.83
	SPC+eTYP	40.91	23.08	29.51	91.21	87.78	90.44	89.09	89.19
	SPC+nENT	26.92	17.95	21.54	89.57	83.37	99.30	90.64	89.99
NB	eTYP+nENT	32.14	23.08	26.87	89.98	88.82	100.00	94.08	93.86
	SPC+eTYP+nENT	25.81	20.51	22.86	88.96	92.26	100.00	95.97	95.91
	SPC	0.00	0.00	N/A	92.03	60.44	58.04	59.22	60.98
	eTYP	42.86	7.69	13.04	91.82	69.10	64.10	66.51	68.49
	nENT	0.00	0.00	N/A	92.03	62.50	80.42	70.34	66.89
	SPC+eTYP	54.55	15.38	23.99	92.23	72.53	63.40	67.66	70.42
RL	SPC+nENT	0.00	0.00	N/A	92.03	64.80	81.12	72.05	69.28
	eTYP+nENT	40.91	23.08	29.51	91.20	75.00	88.81	81.32	80.09
	SPC+eTYP+nENT	45.83	28.21	34.92	91.61	77.46	80.89	79.14	79.17
	SPC	0.00	0.00	N/A	92.03	68.06	64.57	66.27	67.92
	eTYP	0.00	0.00	N/A	91.21	71.82	73.66	72.73	73.03
	nENT	0.00	0.00	N/A	92.03	69.05	69.70	69.37	69.96
SVMs	SPC+eTYP	25.00	5.13	8.51	91.21	84.05	86.01	85.02	85.21
	SPC+nENT	50.00	17.95	26.42	92.02	71.50	100.00	83.38	80.54
	eTYP+nENT	91.98	99.33	95.51	91.41	76.74	100.00	86.84	85.21
	SPC+eTYP+nENT	30.00	7.69	12.24	91.20	83.95	100.00	91.27	90.67
	SPC	0.00	0.00	N/A	92.02	66.82	66.67	66.74	67.58
	eTYP	0.00	0.00	N/A	92.02	73.88	92.31	82.07	80.32
	nENT	0.00	0.00	N/A	92.02	64.20	76.92	69.99	67.80
	SPC+eTYP	0.00	0.00	N/A	92.02	89.95	89.74	89.85	90.10
	SPC+nENT	0.00	0.00	N/A	92.02	83.40	94.87	88.77	88.28
	eTYP+nENT	0.00	0.00	N/A	92.02	88.27	100.00	93.77	93.52
	SPC+eTYP+nENT	0.00	0.00	N/A	92.02	98.39	100.00	99.19	99.20

Table 11 The experimental results for the predicate-oriented relation of ACT-PER.

Model	Feature	Original (Unbalanced)				Sampling (Balanced)			
		P	R	F	ACC	P	R	F	ACC
DT	SPC	0.00	0.00	N/A	77.16	53.65	44.59	48.70	55.80
	eTYP	0.00	0.00	N/A	77.16	64.10	43.29	51.68	61.92
	nENT	71.43	25.97	38.09	80.72	65.54	50.22	56.87	64.16
	SPC+eTYP	0.00	0.00	N/A	77.16	67.81	68.40	68.10	69.87
	SPC+nENT	71.43	25.97	38.09	80.72	63.32	62.77	63.04	65.38
	eTYP+nENT	67.86	24.68	36.20	80.12	62.31	72.29	66.93	66.40
	SPC+eTYP+nENT	67.86	24.68	36.20	80.12	68.73	77.06	72.66	72.71
RF	SPC	0.00	0.00	N/A	77.16	52.80	57.14	54.88	55.80
	eTYP	36.84	9.09	14.58	75.68	60.92	62.77	61.83	63.55
	nENT	52.63	25.97	34.78	77.76	63.87	65.80	64.82	66.40
	SPC+eTYP	35.94	29.87	32.63	71.77	69.17	75.76	72.32	72.71
	SPC+nENT	34.55	24.68	28.79	72.13	70.68	81.39	75.66	75.35
	eTYP+nENT	45.31	37.66	41.13	75.39	73.03	84.42	78.31	78.00
	SPC+eTYP+nENT	42.11	31.17	35.82	74.48	79.50	95.67	86.84	86.36
NB	SPC	0.00	0.00	N/A	77.16	60.47	33.77	43.34	58.45
	eTYP	0.00	0.00	N/A	76.86	59.26	41.56	48.86	59.00
	nENT	0.00	0.00	N/A	77.16	62.62	58.01	60.23	63.96
	SPC+eTYP	25.00	3.90	6.75	75.38	61.34	63.20	62.26	63.96
	SPC+nENT	0.00	0.00	N/A	76.86	61.00	52.81	56.61	61.92
	eTYP+nENT	64.29	11.69	19.78	78.34	61.54	62.34	61.94	63.94
	SPC+eTYP+nENT	57.14	10.39	17.58	77.73	62.45	64.07	63.25	64.98
RL	SPC	0.00	0.00	N/A	77.16	53.17	47.19	50.00	55.58
	eTYP	33.33	1.30	2.50	76.86	61.02	46.75	52.94	60.90
	nENT	71.43	25.97	38.09	80.72	64.39	57.14	60.55	64.98
	SPC+eTYP	26.32	6.49	10.41	74.47	64.57	62.34	63.44	66.19
	SPC+nENT	63.64	18.18	28.28	78.92	66.82	61.04	63.80	67.42
	eTYP+nENT	68.97	25.97	37.73	80.42	65.52	65.80	65.66	67.62
	SPC+eTYP+nENT	63.64	27.27	38.18	79.82	71.20	58.87	64.45	69.45
SVMs	SPC	0.00	0.00	N/A	77.15	55.03	40.26	46.50	56.42
	eTYP	0.00	0.00	N/A	77.15	67.46	49.35	57.00	64.97
	nENT	0.00	0.00	N/A	77.15	67.24	50.65	57.78	65.17
	SPC+eTYP	0.00	0.00	N/A	77.15	78.57	85.71	81.99	82.28
	SPC+nENT	0.00	0.00	N/A	77.15	75.00	74.03	74.51	76.17
	eTYP+nENT	65.71	29.87	41.07	80.42	84.10	87.01	85.53	86.15
	SPC+eTYP+nENT	87.50	45.45	59.83	86.05	91.57	98.70	95.00	95.11

Table 12 The experimental result for the predicate-oriented relation of PER-ACT.

Model	Feature	Original (Unbalanced)				Sampling (Balanced)			
		P	R	F	ACC	P	R	F	ACC
DT	SPC	0.00	0.00	N/A	73.19	0.00	0.00	N/A	57.71
	eTYP	63.83	24.59	35.50	76.04	70.54	37.30	48.80	66.90
	nENT	57.45	22.13	31.95	74.73	71.84	30.33	42.65	65.50
	SPC+eTYP	69.57	26.23	38.10	77.14	68.94	37.30	48.41	66.39
	SPC+nENT	0.00	0.00	N/A	73.19	56.44	46.72	51.12	62.22
	eTYP+nENT	63.53	44.26	52.17	78.24	70.00	63.11	66.38	72.96
	SPC+eTYP+nENT	65.85	44.26	52.94	78.90	68.78	62.30	65.38	72.09
RF	SPC	0.00	0.00	N/A	72.97	43.08	22.95	29.95	54.59
	eTYP	53.16	34.43	41.79	74.29	71.26	50.82	59.33	70.54
	nENT	43.24	26.23	32.65	70.99	63.87	65.80	64.82	66.40
	SPC+eTYP	54.93	31.97	40.42	74.73	66.38	63.11	64.70	70.89
	SPC+nENT	35.24	30.33	32.60	66.37	56.83	64.75	60.53	64.31
	eTYP+nENT	46.53	38.52	42.15	71.65	70.50	80.33	75.09	77.48
	SPC+eTYP+nENT	51.61	39.34	44.65	73.85	75.28	83.61	79.23	81.46
NB	SPC	0.00	0.00	N/A	73.19	33.33	4.10	7.30	55.97
	eTYP	58.93	27.05	37.08	75.38	68.67	42.21	52.28	67.42
	nENT	0.00	0.00	N/A	73.19	70.00	5.74	10.61	59.10
	SPC+eTYP	51.67	25.41	34.07	73.63	61.70	47.54	53.70	65.34
	SPC+nENT	40.28	23.77	29.90	70.11	58.06	44.26	50.23	62.90
	eTYP+nENT	52.54	25.41	34.25	73.85	67.12	40.16	50.25	66.38
	SPC+eTYP+nENT	48.39	24.59	32.61	72.75	63.78	48.36	55.01	66.55
RL	SPC	0.00	0.00	N/A	73.19	44.12	6.15	10.80	57.02
	eTYP	65.22	24.59	35.71	76.26	67.05	48.36	56.19	68.12
	nENT	52.38	27.05	35.68	73.85	60.00	40.57	48.41	63.43
	SPC+eTYP	69.77	24.59	36.36	76.92	70.21	40.57	51.42	67.58
	SPC+nENT	57.14	16.39	25.47	74.29	76.83	25.82	38.65	65.33
	eTYP+nENT	63.22	45.08	52.63	78.24	71.50	58.61	64.42	72.62
	SPC+eTYP+nENT	58.70	44.26	50.47	76.70	73.08	62.30	67.26	74.36
SVMs	SPC	0.00	0.00	N/A	73.19	0.00	0.00	N/A	57.71
	eTYP	75.68	22.95	35.22	77.36	77.46	54.92	64.27	74.18
	nENT	0.00	0.00	N/A	73.19	71.56	31.97	44.19	65.86
	SPC+eTYP	74.03	46.72	57.29	81.32	80.37	72.13	76.03	80.76
	SPC+nENT	74.07	16.39	26.85	76.04	69.15	53.28	60.19	70.19
	eTYP+nENT	76.79	35.25	48.31	79.78	90.91	86.07	88.42	90.47
	SPC+eTYP+nENT	78.41	56.56	65.71	84.18	92.74	94.26	93.50	94.45

5. In the balanced class distribution situation, the F-measure performance order is LOC-ACT>ACT-LOC>ACT-PER≈PER-ACT.
6. We obtain higher accuracy for the unbalanced class distribution, compared to the balanced class distribution since the *unrelated* class (majority class) is relatively large.

In conclusion, we found that our approach with the proposed surface feature set of SPC+eTYP+nENT can achieve up to 99.19% F-measure for balanced class distribution and up to 73.00% for unbalanced class distribution.

6. Conclusion and Future Work

This paper presented an approach to extract entities and predicate-oriented relations among them from Thai texts. A predicate-oriented relation is the general type of NE relations, where an extracted action part is used as a core component to associate related named entities. Different surface feature sets (token space, the number of named entities and entity type) and five alternative commonly used learning schemes (decision tree, rule learner, random forest, naïve Bayes and support vector machines) were applied to investigate the performance on predicate-oriented relation extraction. Conducting the comparison of classification algorithms can be very useful data for NLP people who work on the Thai language. The experimental results showed that all predicate-oriented relations have the problem of unbalanced class distribution; handling of unbalanced class distribution must be done in order to achieve the better performance. The combination of three surface features (SPC+eTYP+nENT) is the most effective surface feature. SVMs outperform other models with the highest average F-measure and accuracy for balanced class distribution. Moreover, LOC-ACT is the easiest predicate-oriented relation type to detect. It gained the highest F-measure and accuracy, followed by ACT-LOC, ACT-PER and PER-ACT, in order. With SPC+eTYP+nENT, the support vector machines achieved the highest F-measure of 97.76%, 99.19%, 95.00% and 93.50% on four different types of predicate-oriented relations (action-location, location-action, action-person and person-action) in crime-related news documents. As our future work, we will investigate the predicate-oriented relation approach among other named entities using a larger dataset.

Acknowledgements

This work has been supported by the TRF Royal Golden Jubilee Ph.D. Program, under Grant No.PHD/0034/2550, the National Electronics and Computer Technology Center, Thailand, under Grant No.NT-B-22-KE-38-54-01, Government Research Fund via Thammasat University, Thailand, and the National Research University Project of Thailand Office of Higher Education Commission.

References

- [1] J. Zhu, A. Gonçalves, V. Uren, E. Motta, and R. Pacheco, "Corder: Community relation discovery by named entity recognition," Proc. 3rd Int'l Conference on Knowledge Capture (K-CAP '05), pp.219–220, New York, NY, USA, 2005.
- [2] T. Hasegawa, S. Sekine, and R. Grishman, "Discovering relations among named entities from large corpora," Proc. 42nd Annual Meeting on Association for Computational Linguistics (ACL '04), p.415, Morristown, NJ, USA, 2004.
- [3] B. Rosenfeld and R. Feldman, "Clustering for unsupervised relation identification," Proc. Sixteenth ACM Conference on Information and Knowledge Management (CIKM '07), pp.411–418, New York, NY, USA, 2007.
- [4] A. Ferrández, J. Martínez, and J. Peral, "An entity-relation approach to information retrieval," Proc. 1st Int'l Symposium on Information and Communication Technologies (ISICT '03), pp.285–290, Trinity College Dublin, 2003.
- [5] K. Takeuchi and N. Collier, "Bio-medical entity extraction using support vector machines," Artificial Intelligence in Medicine, vol.33, no.2, pp.125–137, 2005.
- [6] J.R. Hobbs, "Information extraction from biomedical text," J. Biomedical Informatics, vol.35, no.4, pp.260–264, 2002.
- [7] D. Nadeau, P.D. Turney, and S. Matwin, "Unsupervised named-entity recognition: Generating gazetteers and resolving ambiguity," 19th Canadian Conference on Artificial Intelligence, 2006.
- [8] D. Nadeau, Semi-supervised named entity recognition: learning to recognize 100 entity types with little supervision, Ph.D. thesis, Ottawa-Carleton Institute for Computer Science, School of Information Technology and Engineering, Ottawa, Ont., Canada, 2007.
- [9] T. Utsuro, M. Sassano, and K. Uchimoto, "Combining outputs of multiple japanese named entity chunkers by stacking," Proc. ACL-02 conference on Empirical methods in natural language processing (EMNLP '02), pp.281–288, Morristown, NJ, USA, 2002.
- [10] T. Kumano, H. Kashioka, H. Tanaka, and T. Fukusima, "Construction and analysis of japanese-english broadcast news corpus with named entity tags," Proc. ACL 2003 Workshop on Multilingual and mixed-language named entity recognition, pp.17–24, Morristown, NJ, USA, 2003.
- [11] H. Isozaki, "Japanese named entity recognition based on a simple rule generator and decision tree learning," Proc. 39th Annual Meeting on Association for Computational Linguistics (ACL '01), pp.314–321, Morristown, NJ, USA, 2001.
- [12] M. Asahara and Y. Matsumoto, "Japanese named entity extraction with redundant morphological analysis," Proc. 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology (NAACL '03), pp.8–15, Morristown, NJ, USA, 2003.
- [13] C. Nobata, S. Sekine, and J. Tsujii, "Difficulty indices for the named entity task in japanese," Proc. 38th Annual Meeting on Association for Computational Linguistics (ACL '00), pp.344–351, Morristown, NJ, USA, 2000.
- [14] M. Sassano and T. Utsuro, "Named entity chunking techniques in supervised learning for Japanese named entity recognition," Proc. 18th Conference on Computational Linguistics, pp.705–711, Morristown, NJ, USA, 2000.
- [15] S. Sekine and Y. Eriguchi, "Japanese named entity extraction evaluation: Analysis of results," Proc. 18th conference on Computational linguistics, pp.1106–1110, Morristown, NJ, USA, 2000.
- [16] T. Yao and H. Uszkoreit, "Chinese named entity and relation identification system," Proc. COLING/ACL on Interactive presentation sessions, pp.37–40, Morristown, NJ, USA, 2006.
- [17] Y. Zhang and J.F. Zhou, "A trainable method for extracting chinese entity names and their relations," Proc. Second Workshop on Chinese Language Processing, pp.66–72, Morristown, NJ, USA, 2000.

- [18] C. Giuliano, A. Lavelli, and L. Romano, "Relation extraction and the influence of automatic named-entity recognition," *ACM Trans. Speech Lang. Process.*, vol.5, no.1, pp.1–26, 2007.
- [19] S. Zhao, Information extraction from multiple syntactic sources, Ph.D. Thesis, Department of Computer Science, New York University, New York, NY, USA, 2005.
- [20] S. Zhao and R. Grishman, "Extracting relations with integrated information using kernel methods," *Proc. 43rd Annual Meeting on Association for Computational Linguistics (ACL '05)*, pp.419–426, Morristown, NJ, USA, 2005.
- [21] A. Culotta and J. Sorensen, "Dependency tree kernels for relation extraction," *Proc. 42nd Annual Meeting on Association for Computational Linguistics (ACL '04)*, p.423, Morristown, NJ, USA, 2004.
- [22] L.V. Lita and J. Carbonell, "Unsupervised question answering data acquisition from local corpora," *Proc. Thirteenth ACM int'l Conference on Information and Knowledge Management (CIKM '04)*, pp.607–614, New York, NY, USA, 2004.
- [23] E. Agichtein and S. Cucerzan, "Predicting accuracy of extracting information from unstructured text collections," *Proc. 14th ACM Int'l Conference on Information and Knowledge Management (CIKM '05)*, pp.413–420, New York, NY, USA, 2005.
- [24] D. Zelenko, C. Aone, and A. Richardella, "Kernel methods for relation extraction," *J. Mach. Learn. Res.*, vol.3, pp.1083–1106, 2003.
- [25] J.H. Zhu, A.L. Gonçalves, V.S. Uren, E. Motta, R. Pacheco, D.W. Song, and S. Rüger, "Community relation discovery by named entities," *Proc. 6th Int'l Conference on Machine Learning and Cybernetics*, pp.1966–1973, 2007.
- [26] G. Doddington, A. Mitchell, M. Przybicki, L. Ramshaw, S. Strassel, and R. Weischedel, "The automatic content extraction (ACE) program—tasks, data, and evaluation," *Proc. LREC 2004*, pp.837–840, 2004.
- [27] P. Charoenpornasawat, B. Kijisirikul, and S. Meknavin, "Feature-based proper name identification in thai," *Proc. National Computer Science and Engineering Conference: NCSEC'98*, 1998.
- [28] N. Chaicharoen and W. Aroonmanakun, Computerized integrated word segmentation and part-of-speech tagging of thai, Master's Thesis, Chulalongkorn University, Bangkok, Thailand, 2001.
- [29] A. Kawinpanithan and W. Aroonmanakun, A computational linguistic study of context clues of proper names in thai, Master's Thesis, Chulalongkorn University, Bangkok, Thailand, 2003.
- [30] H. Chanlekha and A. Kawtrakul, "Thai named entity extraction by incorporating maximum entropy model with simple heuristic information," *IJCNLP 2004. LNCS (LNAI) vol.3248*, pp.49–55, 2004.
- [31] N. Muanpai and A. Kawtrakul, "Enhancing thai document retrieval system performance with noun phrase analysis," *9th National Computer Science and Engineering Conference (NCSEC 2005)*, Oct. 2005.
- [32] N. Suwanno, Y. Suzuki, and H. Yamazaki, "Extracting thai compound nouns for paragraph extraction in thai text," *Proc. 2005 IEEE Int'l Conference (IEEE NLP-KE '05)*, pp.657–662, Nov. 2005.
- [33] K. Kriengkiet, K. Kosawat, and S. Anchaleenukul, "A computational linguistics study of compound nouns in Thai," *Proc. 7th Int'l Symposium on Natural Language Processing (SNLP2007)*, pp.31–36, Dec. 2007.
- [34] A. Kawtrakul, M. Suktarachan, P. Varasai, and H. Chanlekha, "A state of the art of thai language resources and thai language behavior analysis and modeling," *Proc. 3rd Workshop on Asian Language Resources and Int'l Standardization (COLING '02)*, pp.1–8, Morristown, NJ, USA, 2002.
- [35] N. Tongtep and T. Theeramunkong, "Pattern-based extraction of named entities in Thai news documents," *Thammasat International J. Science and Technology*, vol.15, no.1, pp.70–81, 2010.
- [36] T. Theeramunkong, V. Sornlertlamvanich, T. Tanhermhong, and W. Chinnan, "Character cluster based thai information retrieval," *IRAL '00: Proc. 5th Int'l workshop on on Information retrieval with Asian languages*, pp.75–80, New York, NY, USA, 2000.
- [37] J.R. Quinlan, *C4.5: Programs for Machine Learning*, Morgan Kaufmann Publishers, San Francisco, CA, USA, 1993.
- [38] L. Breiman, "Random forests," *Mach. Learn.*, vol.45, no.1, pp.5–32, 2001.
- [39] P. Domingos and M. Pazzani, "On the optimality of the simple bayesian classifier under zero-one loss," *Mach. Learn.*, vol.29, no.2–3, pp.103–130, 1997.
- [40] W.W. Cohen, "Fast effective rule induction," *Proc. 12th Int'l Conference on Machine Learning*, pp.115–123, 1995.
- [41] N. Cristianini and J. Shawe-Taylor, *An introduction to support Vector Machines: and other kernel-based learning methods*, Cambridge University Press, New York, NY, USA, 2000.

Nattapong Tongtep received his B.Sc. in Computer Science from Prince of Songkla University, Thailand. He is currently a student in combined M.Sc./Ph.D. program, School of Information, Computer and Communication Technology (ICT), Sirindhorn International Institute of Technology, Thammasat University, Thailand. His research interests are information extraction, natural language processing and data mining.

Thanaruk Theeramunkong received a bachelor degree in Electric and Electronics, and master and doctoral degrees in Computer Science from Tokyo Institute of Technology in 1990, 1992 and 1995, respectively. His current research interests include data mining, machine learning, natural language processing, and information retrieval.