ECTI Transactions on Computer and Information Technology
Journal homepage: https://ph01.tci-thaijo.org/index.php/ecticit/
Published by the ECTI Association, Thailand, ISSN: 2286-9131

Ontology Generation and Instance Extraction of Medicine Information from Thai Semi-structured Data

Taneth Ruangrajitpakorn[1], Thepchai Supnithi[2] and Rachada Kongkachandra[3]

ABSTRACT

An ontology is a widely used knowledge base for representing domain knowledge. Developing a knowledge-representing ontology is difficult, as it requires both domain and engineering expertise. Yet, such ontologies are essential for enabling intelligent systems to comprehend real-world knowledge through structured concept networks. In the Thai context, ontology research remains limited due to the scarcity of structured resources, standardized schemas, and annotated corpora for automatic knowledge extraction. This study addresses this gap by proposing a pattern-based methodology for ontology generation and instance extraction from Thai semi-structured medicine data, providing an alternative to resource-intensive deep-learning methods. The proposed approach identifies patterns of collocated Thai text and builds a collocation tree of word sequences, in which shared sequences represent ontological properties and variable sequences represent instance values. The method was applied to two complementary Thai medicine datasets, namely I-Med (a hospital dispensing-record database) and Pobpad (a public health-information website), to generate and integrate ontology components. These templates were transformed into ontological properties and converted into RDF/OWL format to produce a standard ontology usable for querying and reasoning. The generated ontology achieved high performance (Precision = 0.97, Recall = 0.90, F1 = 0.91) and received favorable assessments from domain experts. The results indicate that the proposed approach can effectively extract structured knowledge from Thai semi-structured text and produce a reliable ontology suitable for medical knowledge representation, providing a data-driven foundation for future Thai intelligent systems.

Article information:

Keywords: Ontology Generation, Instance Extraction, Thai Semi-structured Data, Medicine Information, Knowledge Representation

Article history:
Received: August 20, 2025
Revised: October 16, 2025
Accepted: December 15, 2025
Published: January 10, 2025 (Online)

DOI: 10.37936/ecti-cit.2026201.263623

1. INTRODUCTION

In computer science and information science, an ontology is a formal knowledge model that captures concepts, relationships, and definitions within one or more domains [1–6]. Ontologies have been developed in many fields to organise information into explicit structures that limit complexity and enable shared understanding. A well-designed ontology facilitates knowledge exchange between humans and systems, separates domain knowledge from operational procedures, and provides the foundation for intelligent applications such as semantic web systems, expert systems, and question-answering platforms [1][2][7]. Because of these benefits, domain experts in various fields have continually sought to develop ontologies for use as core knowledge bases.

Ontology engineering can be broadly categorised into two approaches: manual and automatic development. Manual ontology engineering relies on the collaboration of domain experts and ontology engineers to ensure semantic accuracy and domain cover-

[1] The author is with the Department of Computer Science, Faculty of Science and Technology, Thammasat University, Thailand, Email: taneth.ruan@gmail.com
[2] The author is with the Artificial Intelligence Research Group, National Electronics and Computer Technology Center, Thailand, Email: thepchai.sup@nectec.or.th
[3] The author is with the Data Science and Innovation Program, College of Interdisciplinary Studies, Thammasat University, Thailand, Email: krachada@staff.tu.ac.th
[3] Corresponding author: krachada@staff.tu.ac.th

age. While reliable, this process is labour-intensive, subject to expert interpretation, and difficult to scale. Moreover, manual ontologies often serve only as abstract schemas that require additional instantiation to become usable in practical applications.

Automatic ontology generation, on the other hand, replaces expert-driven modelling with information extraction from textual or structured sources [14]. Most existing automatic approaches construct lightweight hierarchical structures that express taxonomic relations but contain limited semantic properties [14–19]. Consequently, such ontologies are suitable for tasks like classification or navigation but insufficient for reasoning or knowledge-based applications that require rich relational semantics. Despite progress in automation, developing a knowledge-representing ontology that captures both conceptual hierarchies and interrelated properties remains a challenge, particularly in low-resource languages such as Thai. Structured knowledge sources and annotated corpora are scarce, and linguistic ambiguity further complicates text-based extraction. In recent years, several studies have shifted focus from full ontology construction to ontology enrichment, where existing schemas are expanded through automatic instance or relation extraction. These approaches help improve ontology usability but still rely on predefined structures and extensive labelled data. To address these challenges, this research proposes a pattern-based methodology for generating ontology schema and instances from semi-structured Thai textual data. The method aims to reduce expert workload by automatically deriving both conceptual relations and individual instances from public Thai medicine datasets, providing a foundation for scalable and semantically rich knowledge representation in the medical domain.

2. BACKGROUND AND REVIEW

Ontology construction has evolved through two primary lines of research: manual ontology engineering and automatic ontology generation. Manual ontology engineering, supported by editors such as Protégé [8][9] and reasoning tools [10][11], has long been the dominant approach because it ensures semantic precision and domain validity, resulting in what is often termed a knowledge-representing or heavy-weight ontology [5][12][13]. However, the process is costly, time-consuming, and highly dependent on human expertise. Differences in how experts conceptualise domains often lead to multiple versions of ontologies that are inconsistent or difficult to reuse. Consequently, researchers have explored automation to increase scalability and consistency in ontology creation.

2.1 Automatic Ontology Generation

Automatic ontology generation replaces manual modelling with algorithmic extraction of concepts, relations, and instances from unstructured or semi-structured data [14]. Early studies typically focused on producing lightweight ontologies, represented as hierarchical trees of parent–child relations. Such structures enable classification and navigation but contain few semantic properties beyond taxonomic links [14–19]. Although this form of ontology is sufficient for indexing or clustering tasks, it lacks the expressive relations required for inference or intelligent reasoning in knowledge-based systems.

Some large-scale projects, such as YAGO (Yet Another Great Ontology) [20] and BabelNet [21], have demonstrated that richer, property-based ontologies can be automatically produced by exploiting structured web content. YAGO integrates lexical hierarchies from WordNet with factual data from Wikipedia infoboxes, generating millions of entities and relations in a knowledge-representing ontology. The success of YAGO depends on predefined extraction rules and the availability of structured patterns within Wikipedia, which limits its generalisability to other domains. The produced ontology has been adopted in several intelligent applications, including the DeepQA project [22], natural-language question-answering systems [23], and personalised search platforms [24]. Similarly, BabelNet combines multilingual lexical resources and encyclopaedic data to form a semantic network connecting words and concepts across languages. Both approaches rely on structured, high-resource corpora that are not readily available in many specialised or local domains.

2.2 Ontology Enrichment and Modern Approaches

As research advanced, attention shifted from building complete ontologies to ontology enrichment, in which existing schemas are expanded by automatically extracting new instances or relations. Enrichment aims to maintain consistency with the base ontology while increasing coverage and usability. In parallel, deep-learning-based ontology extraction methods [25–27] have been developed to learn latent semantic structures from large text corpora. These models can detect hierarchical relations and hidden dependencies between concepts but typically require extensive annotated data and domain-specific corpora to train effectively.

More recently, large language models (LLMs) have shown potential in supporting ontology enrichment and relation extraction. For example, LLMs4OL [28] evaluates several large language models on zero-shot ontology-learning tasks such as taxonomy discovery and relation extraction. Although LLMs can assist ontology engineers, they still require large annotated corpora and careful fine-tuning to achieve reliable results. Similarly, [29] introduced a semi-automatic methodology for constructing medical ontologies with limited expert input, demonstrating that functional

structures can emerge even when expertise is scarce.

In the Thai-language context, [30] applied large language models to extract entities and relations from Thai corporate documents for enriching a financial-product ontology. Their method successfully identified class members and property values based on predefined prompts but assumed that an ontology schema already existed. These works illustrate that ontology enrichment has become an important direction in knowledge engineering, yet they also reveal the persistent difficulty of applying such techniques in low-resource languages.

2.3 Challenges in the Thai Context

Automatic ontology generation and enrichment remain particularly challenging in the Thai language. The scarcity of structured Thai knowledge graphs and standard ontologies in RDF or OWL format limits the use of existing embedding or completion models. Moreover, Thai exhibits linguistic characteristics such as the absence of explicit word boundaries, high ambiguity, and flexible syntax, all of which complicate reliable relation extraction. Prior studies have therefore focused mainly on entity recognition or instance-level extraction rather than complete ontology construction.

A further difficulty lies in the available Thai data, which are mostly unorganised, inconsistent in format, and often unverified by domain specialists. Such conditions increase the risk of incorporating incomplete or inaccurate information into a knowledge-representing ontology. When extraction processes rely directly on semi-structured or user-generated web content, errors or redundancies can propagate into the generated schema or instances. These issues underline the importance of human validation and controlled data selection when applying automatic ontology generation in the Thai context.

From these observations, it is evident that although ontology generation and enrichment have advanced globally, most existing frameworks depend on high-resource environments and structured data. For Thai and other low-resource languages, there remains a need for a more practical approach capable of building relation-rich, data-driven ontologies directly from semi-structured textual resources. The present work addresses this gap by introducing a pattern-based method that learns conceptual structures and property relations from publicly available Thai medical data, enabling the automatic creation of both ontology schema and populated instances without extensive human intervention.

3. A METHODOLOGY FOR ONTOLOGY AND INSTANCE EXTRACTION USING TEXT PATTERN

This work proposes a methodology that can extract knowledge from semi-structured information given in a knowledge-providing semi-structured text documents and portal web into a form of knowledge-representing ontology. Since an ontology alone is a schema representing blueprint of knowledge to define concepts and relation, it does not contain real data and cannot be solely used in an application. Hence, instances (aka. individuals) are required to extract and aligned to the ontology to represent real data or world object of the concepts. This work is designed to include developing a knowledge-representing ontology and extracting instances and their properties. To form a proper ontology, the main components to collect are concepts as ontological classes, axioms (in a form of relation constraints between classes to denote real world knowledge) and instances. Each component is handled separately with different methods and is combined for a complete ontology. An overview of the methodology is illustrated in Figure 1.

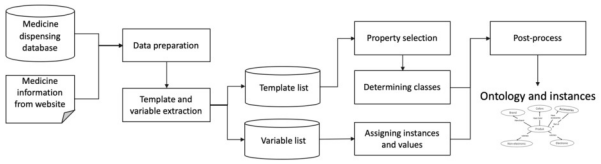

Fig.1: *The workflow of the ontology generation and instance extraction.*

The proposed workflow mirrors the reasoning process typically performed in human knowledge engineering, where experts analyse information patterns, identify conceptual relations, and formalise them into structured ontology components. In this study, the term semi-structured data refers to descriptive textual information that exhibits recurring linguistic or sectional patterns but lacks a formal schema—typified by informational entries such as medicine descriptions, agricultural guidelines, or public advisory sheets. These data often contain implicit relationships between concepts (for example, “ ยา – การใช้ – สรรพคุณ” (drug – usage – therapeutic effect), “ พืช – วิธีปลูก” (crop – cultivation method)) presented in short, labelled sections. The method targets this category of pattern-regular, semi-structured descriptive data rather than formally structured (e.g., XML or JSON) or completely unstructured free text.

The data preparation stage standardises the textual input and removes redundant symbols. Pattern extraction then identifies recurring lexical and positional structures that express semantic relations. These patterns are analysed in property detection and classification, where the system distinguishes between object properties (relations between concepts) and data properties (attributes with literal values). Class generation and instance assignment formalise the recognised entities into ontology schema and populated instances, while post-processing ensures structural consistency and removes duplicates.

This modular design allows each component to be

reused or extended for other Thai semi-structured descriptive data that share similar linguistic and structural characteristics. The method operates on recurrent linguistic and structural patterns rather than on predefined domain vocabularies, allowing it to be adapted to other Thai semi-structured datasets with similar textual characteristics. The data sources shown in Figure 1 (I-Med and Pobpad) represent the applied domain used in this study to demonstrate the method. They serve as examples of semi-structured Thai text. Comparable datasets with similar structural characteristics could replace them without altering the underlying workflow.

3.1 Data Preparation

This work applies a Thai semi-structured dataset as an input to generate ontological components. The semi-structure in this work refers to a text loosely organized by a topic. Still, the information within is an open text without structure and predetermined annotation, such as text information in a table. The data can be in the form of a roll-based table or a type-based table.

In this research, the selected data sources are medication data from medicine dispensing record database in a hospital (I-med) and content from Pobpad website [31]. The data from the I-med is in the roll-based table style as shown in Figure 2-Top, while the Pobpad provides the data in type-based table style as illustrated in Figure 2-Bottom. The content data are mainly given in Thai, but the name of the drug is given in English in the I-med and English with Thai in parentheses (or inverse as Thai with English in parentheses) in Pobpad. For content, the I-med gives information about drug instruction in a phrase-based consecutive Thai text in which provides to how-to consume or use the drug effectively. The information in the I-med includes administration route, unit-consumption amount, time and duration, and condition for using. On the other hand, the information from the Pobpad data set gives separated information in each field, but the information may contain several information chunks without notation and symbol to boundary the chunks. There are many types of information provided in Pobpad data, but most of the data are unstructured data; thus, we select the specific data provided in the table form to be used in this research.

The data are thus processed for preparation towards text processing as follows. Initially, text cleansing is processed to handle text data. In this part, the aim to cleanse the marking or symbol with little to none meaning off the data so the data can be processed easier in the later parts. To make the precise results, we split the data of the drug name and content data for tackling differently. The drug names are mostly given in English while some may attach with Thai name in parentheses. Non-words that have no semantic meaning including under-scroll are replaced with the white-space, and trademark and symbolic notations are removed. The capitalised letter in a drug name is transformed into lower case for consistency. With the different nature in information, drug names of I-Med are not purely the generic name of a medicine but also include their dose and extra information such as abbreviation of dosage form, brand of the drug provider, and incomprehensible marker. For example, the drug name 'DoxyCycline 20_mg (สีชมพู) cap _ก' contains a drug generic name 'doxycycline' with dosage of '20 milligram (mg)', and extra information including 'cap' referring capsule dosage form, 'สีชมพู' and '_ก' for incomprehensible marker. On the other hand, the drug name given in the Pobpad provides only the generic name of a medicine. We then decide to keep the name and dosage amount of the data from I-med since other information parts are not consistent and referable. In terms of dosage, several representations of the same measurement are seen throughout the data.

	Administration route & unit-consumption amount	Time	Duration	Condition
Original	รับประทานครั้งละ 1 เม็ด	ก่อนอาหารครึ่งชั่วโมง	เช้า	เมื่อมีไข้
Lit. Eng Translation	(consume 1 tablet)	(consume 1 tablet 30 minutes before meal)	(in the morning)	(when having a fever)
Original	ฉีดเข้าเส้นเลือดดำ 2 กรัม	วันละ 1 ครั้ง	ก่อนนอน	เวลานอนไม่หลับ
Lit. Eng Translation	(intravenous injection 2 gram)	once a day	(before bed)	(when having insomnia)

	Drug group	Type	Usage
Original	ยาปฏิชีวนะ	ยาตามใบสั่งแพทย์	รักษาการติดเชื้อแบคทีเรีย
Lit. Eng Translation	(anit-biotic)	(doctor prescription drug)	(to cure bacterial infection)
Original	ยาระงับปวด	ยาอันตราย	ลดอาการปวดระดับปานกลางจนถึงระดับรุนแรง
Lit. Eng Translation	(Analgesic)	(dangerous drug)	(to relieve medium to severe pain)

Fig.2: *Explanation of chunking of information with literal English translation; top is from the hospital database and bottom is from the PobPad website.*

For content data, symbols for separation including comma (,), vertical bar (|), and slash (/) are unified into the single symbol as commas. Misspellings and typos are handled manually. In addition, the drug entries with unreadable content and blank content are removed from the data set. The content part in both data sets composes of Thai text in a form of phrases or consequence of words. So, it requires applying automated text tokenisation to define a word boundary. This work applies an automatic word segmentation tool [32] from the Language and Semantic Technology Laboratory, NECTEC. The tool was applied with a customized dictionary containing medical and pharmaceutical terminology to improve segmentation accuracy and reduce out-of-vocabulary issues. The segmentation parameters were set to use the default dictionary-based matching depth, with a maximum word length of 20 characters, and forced dictionary entries were added for medical dosage units (e.g., "มก.", "มิลลิกรัม", "แคปซูล", "เม็ด") and common

instruction verbs. The 20-character threshold is a default parameter of the Thai word-segmentation tool. It defines the maximum character length that the segmenter treats as a single lexical unit during tokenisation, since Thai words are typically short. The parameter can be adjusted if applied to other languages or segmentation tools with different lexical-length distributions. Manual post-validation was performed to correct boundary errors and compound-word missegmentations. Among many available tools, this segmentation tool is suitable for this task since it is implemented for concept-based words rather than short word forms and allows additional terms to be inserted as a forced dictionary to prevent unknown-word errors. Furthermore, manual validation is performed to correct misspelling and incorrect word segmentation results. Last, unification data is a process to find common individuals of the different data style. In this part, the name of the given drug is matched into one pool to contain data from both sources. The name of the drug is the main indicator for unifying the data. Since most of the data share English name of the drug, the English name is used for matching from the two datasets. However, the dose information attached after the name is ignored in this matching process. After all processes, the data from the two datasets can be mapped into one larger dataset and are ready to be extracted for instance and ontology generation as demonstrated in Table 1.

Table 1: *Examples of integrated data from the two datasets to extend medication information.*

Drug name	Drug instruction from I-Med	Drug group from Pobpad	Type from Pobpad	Usage from Pobpad
acetylcysteine_200mg	1 ซองผสมน้ำครึ่งแก้ว 1 ซองวันละ 3 ครั้ง หลังอาหาร เช้า, กลางวัน, เย็น (1 pack mixing with half-glass of water, 3 times a day, post-meal at morning, noon, dinner.)	ยาละลายเสมหะ (Mucolytics)	ยาตามใบสั่งแพทย์ (Prescribed medicine)	ละลายเสมหะ สลายมูกหนียวข้นให้เบาบางลง
amoxycillin_500mg	รับประทานครั้งละ 2 เม็ด หลังอาหาร วันละ 2 ครั้ง เช้า, เย็น (consume 2 pills, post-meal, 2 times per day at morning and dinner)	ยาปฏิชีวนะ (antibiotic)	ยาตามใบสั่งแพทย์ (Prescribed medicine)	รักษาการติดเชื้อแบคทีเรีย
ibuprofen_100_mg/5ml	รับประทานครั้งละ 1 เม็ด หลังอาหารทันที วันละ 3 ครั้ง เช้า,กลางวัน,เย็น เวลาปวด (consume 1 pill, post-meal, 3 times a day, at morning, noon, dinner.)	ยาต้านอาการอักเสบที่ไม่ใช่สเตียรอยด์ (NSAID)	ยาตามใบสั่งแพทย์ (Prescribed medicine) , ยาหาซื้อได้เอง (over-the-counter: OTC)	ลดอาการปวด ลดไข้ และแก้อักเสบ
metoclopramide_10mg/2ml	ฉีดเข้าเส้น 10 มิลลิกรัม ทุก 6 ชั่วโมง เวลามีอาการ (intravenous injection for 10 mg, every 6 hours, when having a condition)	ยาแก้คลื่นไส้อาเจียน (Antiemetics)	ยาตามใบสั่งแพทย์ (Prescribed medicine)	รักษาอาการคลื่นไส้อาเจียน และภาวะกรดไหลย้อน
omeprazole_20_mg	รับประทานครั้งละ 1 เม็ด ก่อนอาหารครึ่งชั่วโมง วันละครั้ง เช้า เวลามีอาการ (consume 1 pill, before-meal, 1 time a day, at morning when having a condition.)	ยายับยั้งการหลั่งกรด (Proton Pump Inhibitors)	ยาตามใบสั่งแพทย์ (Prescribed medicine) , ยาหาซื้อได้เอง (OTC)	รักษาโรคกรดไหลย้อน โรคหลอดอาหารอักเสบ และแผลในกระเพาะอาหาร

4. DATA-DRIVEN APPROACH TO ONTOLOGY GENERATION USING TEMPLATE

The proposed method focuses on finding the patterns of Thai text word sequences and extract the information regarding sharing words and unique words. The sharing word sequences are considered as templates while the unique words which are possibly varied from instances are kept as variables. For forming into an ontology, the obtained templates are transformed into ontological properties indicating attributive relations of concepts, and the variables are considered for the value of each relation to the given individuals.

4.1 Template and Variation Extraction

A crucial component in a functional ontology to represent domain knowledge is the properties of a concept. There can be many details given in texts. From observation, same types of detail are commonly expressed similarly with certain clue words or pattern of words. This can be used to denote a property of an ontology class if the detail is given to several concepts under the same class. In this process, we want to find properties and concept range of properties from text. The main idea of the method to find a list of templates that are commonly shared in data entries. Templates are a sequence of words with some different lexical values among them. Template [33] is a technique to find common word sequences in a natural language for a task of finding word pattern. For example, please consider the given data in Figure 3.

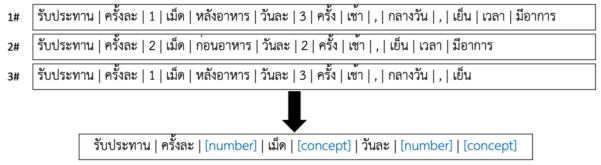

Fig.3: *Sharing and different patterns in a similar text.*

The example shows that there are some common lexical items with some different lexical items among the data entries. The same content thus can be used as a pattern template, and the difference is what makes data individually unique. Based on the idea of an ontology, this identifies the property and its individual value of the property of concepts they belong to. Furthermore, the conceptual entities that share the same set of properties can be considered as they are in the same conceptual class. Hence, we can take the idea to determine ontology properties accordingly.

The algorithm GatherColloationTree takes a set of words from all text data in the same field to construct

```
GatherColloationTree(WordSequences) -> LeafNodeList
1   PatternTree = new Pattern() ;
2   For each item W in WordSequences
3               Pattern = new Pattern (W)
4               If AddToPatternTree(Pattern, PatternTree)
5                       Do nothing
6               Else
7                       AddChildTree(Pattern, PatternTree)
8   LeafNodeList = AddLeaves(PatternTree)
9   Filter(LeafNodeList)
10  OrderAscensionWordSequences (LeafNodeList)
11  Return LeafNodeList
```

```
AddToPatternTree(Pattern, PatternTree)-> True | False
1   For each child P of PatternTree
2                       Sequence(Pattern, P)
3               TestForSplitTree(Pattern, PatternTree)
4               If any Pattern added to Children of PatternTree
5                       Return True
6               Else
7                       Return False
```

```
Sequence(Pat1,Pat2)-> True | False
1   If Intersection (IDs(Pat1), IDs(Pat2)) ≥ x
2               If AddToPatternTree(Pat1),(Pat2)
3                       Do nothing
4               Else
5
6               AddPatternTree(CombinePatterns(Pat1,Pat2), (Pat2)
7   Else
8               Return False
```

```
CombinePatterns(Pat1,Pat2)-> NewPattern
1   NewPattern = new Pattern()
2       NewPattern.SetIDs (Intersection(IDs(Pat1),IDs(Pat2))
3       NewPattern.SetWordSequences(Pattern(Pat1),Pattern(Pat2))
4   Return NewPattern
5
```

a tree of word collocations based on the given word sequences by adding more words of the same pattern to the tree. The result of the collocation tree is a list of filtered words in collocation. The collocation word is considered with the attempt of the possible daughter words using the function AddToPatternTree. The possible alternatives of words from other text data are tested. If the words cannot be added to the existing tree branch, the new branch is generated for the new daughter words of that parent node using the function AddChildTree. The words are kept in order as they are in the original text data. The result will be a set of the longest recurrent word sequences. In the function AddToPatternTree, the current words in consideration are tested for adding to the tree Pattern by combining with each child of the current node of the tree. The function Sequence performs the combination test between the two sequences for possible combination. The Pattern considers the current word to be subsequent daughters or add a new daughter. Hence, each of collocation is recursively tested for combination with the nodes of the tree and filtered from the root node to the leaf branches. Last, the function CombinePatterns creates and returns a new node by combining the words from both the collocations to be added.

For examples, please consider the following Thai word sequences.

- รับประทาน|ครั้งละ| 1 |เม็ด|หลังอาหาร| เช้า,กลางวัน,เย็น
- รับประทาน|ครั้งละ| 1 |เม็ด|หลังอาหาร| เช้า,เย็น
- รับประทาน|ครั้งละ| 1 |เม็ด|ก่อนอาหาร| เช้า,เย็น
- รับประทาน|ครั้งละ| 1 |เม็ด| ทุก|4|ชั่วโมง
- รับประทาน|ครั้งละ| 2 |เม็ด|หลังอาหาร เช้า,กลางวัน,เย็น
- รับประทาน|ครั้งละ| 1 |เม็ด| ทุก|12|ชั่วโมง

The algorithms then generate pattern trees of word sequences of these six texts are given in Figure 4.

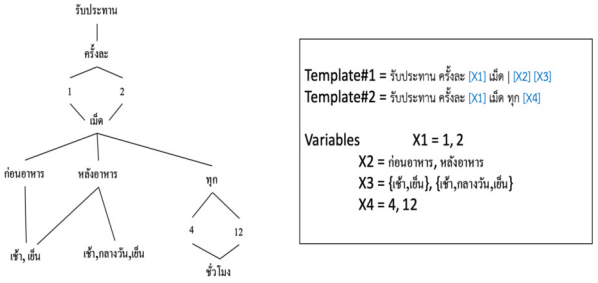

Fig.4: *Generated Pattern Tree and resulting templates and variables.*

The trees take words in collocation to generate branches once there are different lexical forms in the same sequence. The frequency of the appearance is also recorded. The method takes the word sequence from left to right. Thus, the generated trees are based on the initial words of the sentence. The found patterns then are considered as the templates. The words in the branches of these patterns are kept as variables. The templates and variables are the core information gained from the data. These templates then can be used to generate a property for an ontology, and the variables which are individually related to instances can be used as property values regarding template-based property.

4.2 Detecting Property

To recognise ontological properties at the schema level, the obtained templates are considered. The obtained templates then are listed and grouped based on similarity of the templates. The number of allowed variables is the number of properties to be given to the item.

To group the templates, we consider the core verb of the template. Words in templates are then tagged with Thai simple Part-of-speech (POS) [34]. Since this work focuses on the verb, only verbs in the template are tagged. The most left verb then is considered as the core verb of the text. To semantically handle synonymous verbs, Lexitron [35]: an electronic Thai-English dictionary is used as the reference dictionary to provide a list of synonyms to align the same meaning verbs and nouns. The words with the same meaning thus are equivalently treated as the same word. However, the dictionary may not contain all the existing words in the dataset, especially those compound words; hence, such unknown words

are added to include them in alignment. With the synonym sets, the templates can be grouped based on the same meaning of the core action. To further the grouping, the commonness within templates is considered. For this case, if templates are a subset of another template, they are merged into the longer one.

After grouping the similar templates, number of templates should significantly be reduced to represent only different conceptual meanings. For this part, we begin to divide information into template groups as a possible property of the concept. The number of variable slots in a template then is considered as a number of assigned properties, for example the template ‘ รับประทานครั้งละ [x1] เม็ด [x2] [x3]’ which has 3 variable slots will produce 3 properties as follows.

- property♯ 1: รับประทานครั้งละ [x1] เม็ด
- property♯ 2: รับประทาน [x2]
- property♯ 3: รับประทาน [x3]

As an ontology standard, each property must have a different unique name for a referring namespace. With that in mind, we take words in templates into account for naming the property name. The core verb of the template and the referred variable slot with left unique words are taken as the temporary property name. The generated temporary name may or may not be understandable since it only takes surrounding word to form a name. Still, it should serve the purposed on differentiating the referring name of an ontology component for later usage. In fact, the temporary name should be properly named later by human for better understanding of the given relation. Unfortunately, the proper automatic naming method that can represent meaning of the relation is very hard even for a human.

Once the properties are obtained, another essential factor of property relations is their type. According to common ontology components, there are two types of ontological property including object property (aka part-of relation) and data property (aka attribute-of relation. Although these two property types are both relations to connect classes across the tree, they have a different specification setting and usage.

The object property (OP) is a relation that relates two classes with the roles of domain and range. Based on the aforementioned triple in Section 3.1, the domain role plays the subject role of the relation, and the range role plays the object role of the relation. In the case of the OP, the allowed component is only ontological class. the range class can be a class in any hierarchical level in an ontology tree including universal root node, intermediate node, and leaf node. The designated node in a non-leaf level thus includes all classes from its child nodes as possible range of the OP. However, setting the range class to the universal root node is not common and favoured in an ontology design because it can be inferred that anything in the ontology can be a range of the relation in which is not sensible for knowledge representation.

For data property (PP), the relation is to relate a class as a domain to data. The data can also be unspecific or specified using the standard data type such as string, integer, float, boolean, and date/time according to W3C standard [36][37]. The PP is a relation that aims to link a class to non-semantic concept information, namely date and time, measurable value and individual given name. The value of PP hence can be indefinite unlike range of OP that can be limitedly defined into hierarchical concepts. With the specification of data type, the data value can be used accordingly such as number comparison for the numerical data type including integer and float, and regular expression matching for the string data type. However, in case of specified data type and the data value are not compatible, it will result in error reasoning or reasoning shut down when applying to inference engine due to finding inconsistency in specified data type.

With the above specification, deciding the property type can take the given variables into account. There are three aspects for consideration. First is the data type of variables itself if they are a group of short text information, long text information, numerical value, and mix of several types. Second is the definiteness of the variables if the possible values are greater than certain number of variables. Third, the frequency of words in a text is amendable. The decision to assign a property type can be visualised as shown in Figure 5.

The decision is made based on the decision node. For determining short or long text, the count of strings of the text is used. However, Thai text includes several string styles including the lower and upper symbol to denote tone and vowel. Thus, the counting excludes such alphabet and only count for consonant alphabets only. This work applies the count of 20 strings as criterion for those lower than 20 string is short text and the reset as long text. Moreover, for word frequency for word grouping, the frequency of words is counted to find if the words are statistically distributed for the same variable slot. If the words are common, they can be used as ontological classes representing concepts of entities. With the entire processes in this part, property lists with a specific type and domain-range relation are assigned. These properties then will help to determine classes and hierarchical tree.

4.3 Determining Class

An ontological class plays the main role of a core component in an ontology. It is a component that is formed into a hierarchical tree and where properties belong to. Moreover, it is a conceptual representation of instances. For a man-made ontology, classes are determined by gathering relevant concepts in a domain and judged by knowledge engineer experts

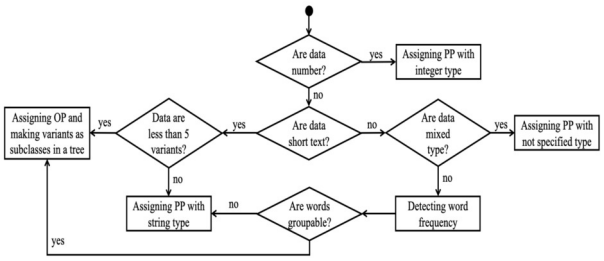

Fig.5: *Decision flowchart to assign property type.*

to determine hierarchical tree by which concept is generalisation for superclass and which concepts are specification for subclass. Unfortunately, this work which is in an automated data-driven approach cannot achieve that.

To determine classes and class hierarchy, there are two cases in this work. First, the classes obtained through the object property decision process. Second, the class is needed for instances to attach to. For the former, the obtained classes from the previous processes are those from the object property (OP) assignment. The range classes from template-based property are initially differentiated from variable slots. Please consider the example in Figure 6.

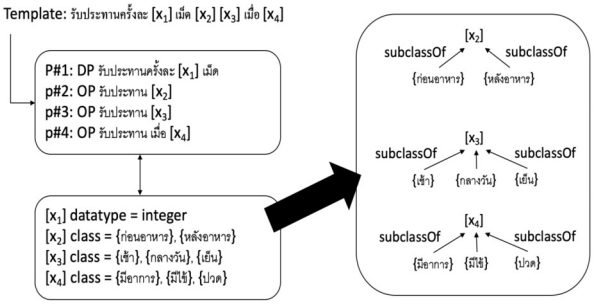

Fig.6: *Examples of class grouping for subclass assignment.*

For the latter, instances (each roll of the data in which are the list of individual drugs in this work) require the class to attach too. Initially, the class is generated with temporary name to keep the instances. Additionally, if the source dataset provides the categorisation of the instances, the categorisation can be applied to create a list of subclasses following the provided tree. Unfortunately, the selected datasets in this work do not give the such tree-based categorization; thus, only single class is created in this process. Once the classes from both cases are generated, a universal root node called 'entity' to connect all classes in generated to relate all orphan classes (class that has no parent class assigned to it) as their superclass.

4.4 Assigning Instances and Their Value

Instances are necessary for an ontology to be practically usable. Instances are mapped to a class to inherit their property for values of each attribute to be provided. In this work, an instance is a unique input item of the extracted information. From the combined dataset, the instances of this work are drug name with dose. The drug name is used for an instance name representing drug individual. The instances are assigned to the 'Class to collect instances', and they inherit properties of both OP and PP from the generated schema. Since each instance has its own property value, variables from its own data are then assigned for the value of property accordingly.

Since the 'Class to collect instances' has many properties from all the templates to serve for all possible details of the instances, it is allowed to have missing property values and to keep only the existing details from the data in this stage. The value from the variable then is assigned according to its variable slot. If there are more than 1 variables for the slot, duplication of the property is allowed to separately store the value with the same property relation. Let take 'amoxycillin' and 'omeprazole' drug as an example from Table 1. The drugs inherit property schema and get the values according to data as shown in Figure 7.

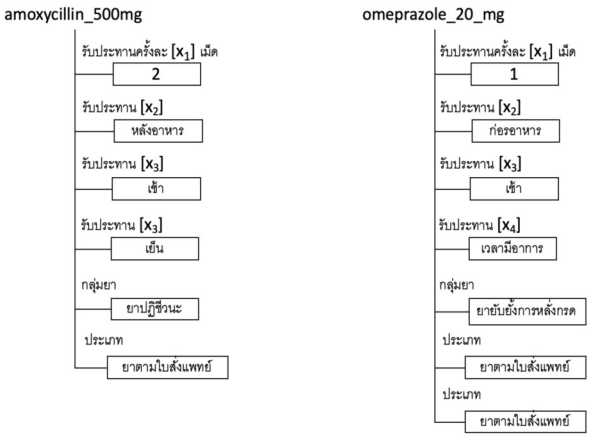

Fig.7: *Instances and their property value from detected variables.*

According to the data, some properties have 2 values including 'รับประทาน [x3]' of the instance 'amoxycillin_500mg', and 'ประเภท' of the instance 'omeprazole_20_mg'. Hence, the properties are doubled to keep the values separately. This assists on making the instances more suitable for query and using with reasoning/inference since the split data are not needed to be processed again using string matching or regular expression to tackle the data. Furthermore, they are the range class of object property relation, and they can be easily counted for statistics.

With values, instances thus show attributive differences between them. The more properties they have,

the more informative they become. It is possible that some properties of an instance are missing, and the information may be not given from the source since the proposed method is a data-driven approach and take information solely from the text data. In such case, the property and its value are not stored within the instance and may affect ontology performance.

4.5 Post-process for Improving Ontology Comprehensibility and Usage

At the current state, there are some temporary names in the generated ontology schema. The temporary names though do not affect the structural representation of the ontology or its usage in application. However, the temporary naming does not incur the real meaning of the classes and the relations, and they are not interoperable to humans who may use and validate the ontology. In addition, after converting the name into standard ontology formats such as OWL [36] and RDF [37], the names are thus used across the ontology for a URI reference and visualisation. Thus, it is necessary to handle the naming properly in this state.

The name of classes and relations is supposed to be unique and semantically represent the actual meaning. This however cannot be handled automatically and requires great understanding of the domain knowledge. Hence, manual naming should be done to fulfill the part. The name change must be unique among all ontological components including class, relation, and instance.

Last, the ontology components are in triple form; thus, we apply OWL API [38] to convert the generated ontology into OWL/RDF format for standardising and usability in ontology editor tools such as Protégé. In the ontology editor, it allows the ontology to be queried using SPARQL and inferred using SWRL for testing the ontology.

5. RESULTS

This section provides results of the proposed method. First, the generated ontology and its related components are reported. Second, an evaluation using ontology for querying is performed.

5.1 Result of Ontology Generation Method

The Thai semi-structured text for extracting ontology and instances was collected from two sources: I-Med, which provides hospital prescription and dispensing records, and Pobpad, which provides public health information about medicines. A total of 456 medicines were found to appear in both sources and were used as raw data for ontology and instance extraction.

After training the ontology model using the proposed pattern-based method, the resulting ontology was generated and analysed in terms of its schema components and instances. According to standard ontology components, the produced ontology contains six main class trees representing the core conceptual groups in the medicine domain. These classes cover medicine (ยา), usage condition (ข้อแม้), time (ช่วงเวลา), administration time (เวลา), administration (การใช้), and drug type (ประเภทยา). Figure 8 shows the tree structure of these classes.

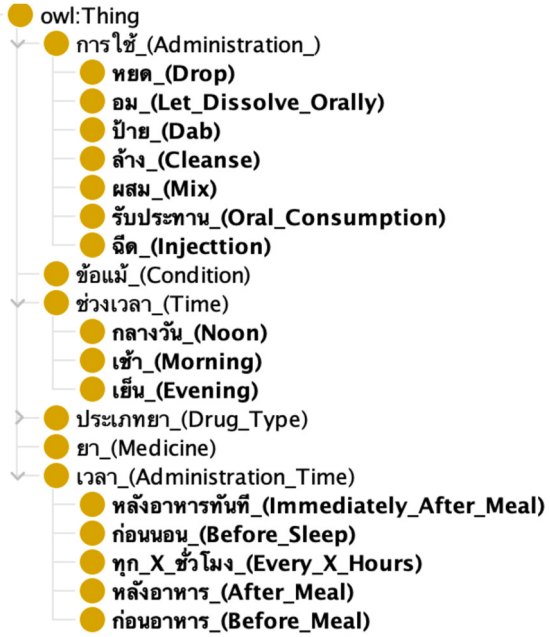

Fig.8: *Tree structure of classes in the obtained ontology.*

With the classes, the properties get the object to belong to. Object properties are to connect two classes as domain and range. The domain refers to a class that owns the property relation (subject) while the range refers to a class and its subclass to be an object of the property relation. Aside from an object property connecting classes, a data property to connect a class as subject to a datatype as object is generated. Both object and data properties belong to the main class 'ยา' which is the class for instances to attach to. Hence, the properties are inheritable by the instances for storing the value gained as variables. The generated ontology includes five object properties and eleven data properties that together describe the relations between medicines and their attributes.

The object properties define conceptual relations among ontology classes, including:

1. [ยา] subject การใช้งาน(hasAdministration) [การใช้] object
2. [ยา] subject ช่วงเวลาการใช้งาน(usagePeriod) [ช่วงเวลา] object
3. [ยา] subject เวลา (timeOfAdministration) [เวลา] object
4. [ยา] subject ข้อแม้ใช้งาน (conditionOfUse) [ข้อแม้] object
5. [ยา] subject มีประเภทยา (hasDrugType) [ประเภทยา] object

The data properties capture literal or quantitative attributes attached to medicine instances, including:

1. [ยา] subject รับประทานครั้งละx เม็ด (take x tablets per dose) [integer] object

2. [ยา] subject รับประทานครั้งละx แคปซูล (take x capsules per dose) [integer] object
3. [ยา] subject หยอด x หยด (instill x drops) [integer] object
4. [ยา] subject รับประทานครั้งละ x ช้อนชา (take x teaspoons per dose) [integer] object
5. [ยา] subject รับประทานทุกx ชั่วโมง (take every x hours) [string] object
6. [ยา] subject x ซองผสมน้ำ (mix x sachets with water) [integer] object
7. [ยา] subject ผสมน้ำ x แก้ว (mix with x glasses of water) [integer] object
8. [ยา] subject ฉีดครั้งละ x ซีซี (inject x cc per dose) [integer] object
9. [ยา] subject ฉีดครั้งละ x ซีซี (inject x units per dose) [integer] object
10. [ยา] subjectรับประทานครั้งละx มิลลิกรัม (take x milligrams per dose) [integer] object
11. [ยา] subject มีกลุ่มยา (hasTherapeuticGroup) [string] object

For the instances, we exemplify the extracted detail to show the property values in a form of OWL/RDF which includes marked-up annotation. The example of the RDF formatted instance of the 'paracetamol 500 mg' tablet from the dataset is given in Figure 9.

```
<owl:NamedIndividual rdf:about="http://druginstruction#paracetamol_500_mg ">
    <rdf:type rdf:resource="http://druginstruction#ยา/">
    <di:การใช้งาน rdf:resource="http://druginstruction#รับประทาน/">
    <di:ช่วงเวลาใช้งาน rdf:resource="http://druginstruction#ทุก 4 ถึง 6 ชั่วโมง/">
    <di:รับประทานครั้งละ_X_เม็ด rdf:datatype=" /XMLSchema#integer">1</di:
รับประทานครั้งละ_X_เม็ด>
    <di:มีกลุ่มยา=" /XMLSchema#string">ยาระงับปวดและลดไข้</di:มีกลุ่มยา>
    <di:ข้อแม้ rdf:resource="http://druginstruction#เวลามีอาการ/">
    <di:ประเภทยา rdf:resource="http://druginstruction#ยาตามใบสั่งแพทย์/">
    <di:ประเภทยา rdf:resource="http://druginstruction#ยาหาซื้อได้เอง/">
</owl:NamedIndividual>
```

Fig.9: *Obtained Instances representing 'paracetamol 500 mg' tablet from the dataset.*

In this example, the instance paracetamol 500 mg is linked through the object property การใช้งาน(hasAdministration) to รับประทาน (oral use), indicating the administration method. The property ช่วงเวลาใช้งาน(usagePeriod) connects the instance to ทุก 4 ถึง 6 ชั่วโมง (every 4 to 6 hours) under the class ช่วงเวลา, describing the dosage interval. The data property รับประทานครั้งละX เม็ด (take X tablets per dose) carries an integer value of 1, meaning that one tablet is taken each time. Another data property, มีกลุ่มยา (hasTherapeuticGroup), provides a literal string value " ยาระงับปวดและลดไข้ (pain relief and fever reduction group)" to indicate its therapeutic category. The instance also connects via the object property ข้อแม้ (conditionOfUse) to เวลามีอาการ (when symptoms appear), specifying the usage context. Notably, the property ประเภทยา (hasDrugType) is linked to two object values, ยาตามใบสั่งแพทย์(prescription drug) and ยาหาซื้อได้เอง (over-the-counter drug). In RDF representation, a property that has multiple objects is recorded as two separate property relations, each linking the same subject to a different object, to maintain clarity and reasoning accuracy.

5.2 Evaluation of using the ontology in querying

We asked three pharmacist experts to provide searching conditions as queries for the ontology after introducing its structure and properties so that they understood which attributes could be used as criteria. Although the number of experts was limited to three, they were carefully selected for their dual expertise in pharmacy and data-driven knowledge systems, which is rare in the Thai context. Each expert is highly experienced in the pharmaceutical domain, well-familiar with the detailed structure of the underlying data sources and possesses a clear understanding of what an ontology is and how ontology-based reasoning operates. This combination of qualifications ensured that the evaluation focused on the correctness, completeness, and usability of the generated ontology rather than on statistical generalization. The three experts collaboratively designed a total of twenty representative queries to cover different usage scenarios of medicine retrieval, and each expert then evaluated the query results independently, providing separate precision, recall, and F1-score assessments for objectivity. Examples of SPARQL queries used in the evaluation are shown below.

```
SELECT ?drug ?type
WHERE {
   ?drug di:มีประเภทยา di:ยาตามใบสั่งแพทย์;
         di:มีกลุ่มยา ?group .
   FILTER (STR(?group) = "ยายับยั้งการหลั่งกรด" )
}
```

The SPARQL query shown above aims to retrieve all medicines that belong to the "ยาตามใบสั่งแพทย์" (prescription drugs) and have the therapeutic property of " ยายับยั้งการหลั่งกรด " (proton pump inhibitor) group. In this query, the variable ?drug represents the medicine instances to be retrieved, while ?type refers to their corresponding drug type. The triple pattern specifies that the queried drugs must have an object property linking them to the individual : ยาตามใบสั่งแพทย์ under the class tree of drug types. The subsequent triple associates the same drug with its therapeutic group through a data property, which stores literal string values such as " ยายับยั้งการหลั่งกรด ". Consequently, the query is a multi-criteria retrieval for selecting prescription-only drugs within a therapeutic property of proton pump inhibitor.

For measuring the query performance, a confusion matrix was applied to calculate Precision (P), Recall (R), and F-measure (F1). In the confusion matrix, a true positive (TP) represents a retrieved medicine that correctly matches the query, a false positive (FP)

represents a retrieved medicine that does not belong to the query, and a false negative (FN) represents a medicine that should be retrieved but is missing from the result. The values of P, R, and F1 were then computed from these counts to evaluate retrieval accuracy. The evaluation was conducted using twenty representative queries (n = 20), and the results, grouped by ontology aspect composition, are summarized in Table 2.

Table 2: *Evaluation results of twenty SPARQL queries grouped by ontology aspect composition.*

Aspect / Combination	n	Precision (P)	Recall (R)	F1-score (F1)
Administration	2	1.00	0.97	0.98
Time	4	1.00	0.96	0.98
Condition	2	1.00	0.99	0.99
Frequency	1	1.00	0.89	0.94
Group	1	1.00	1.00	1.00
Administration + Time	4	0.97	0.96	0.96
Group + Time	2	0.91	0.63	0.74
Group + Condition	2	0.90	0.70	0.79
Administration + Condition + Frequency	2	0.97	0.90	0.93
Average		0.97	0.90	0.91

Overall, the ontology achieved an average Precision of 0.97, Recall of 0.90, and F1-score of 0.91, demonstrating high retrieval accuracy and consistent reasoning performance. Queries involving a single ontology aspect such as administration, time, or condition produced the best performance, with F1-scores between 0.98 and 0.99. These results confirm that individual relations are well defined and reliably extracted. Multi-aspect queries that combine two or more properties, such as administration with time or group with condition, showed lower recall due to overlapping expressions and deeper reasoning paths across ontology branches. Some complex queries also suffered from a higher number of false negatives, which occurred when the requested criteria were not explicitly provided in the source data, resulting in incomplete retrieval and lower recall scores. Despite these variations, precision remained high across all aspect combinations, indicating that the ontology structure effectively captures semantic relationships and supports reliable retrieval even for complex query compositions.

6. DISCUSSION

6.1 Implications of Data-Driven Ontology Generation

This study is primarily demonstrative in nature, aiming to show that a knowledge-representing ontology can be generated directly from semi-structured Thai text without relying on annotated corpora or predefined schema. Most existing ontology-generation frameworks, such as Text2Onto, OntoMiner, or YAGO, were developed for structured or resource-rich environments and are designed either to construct lightweight taxonomies or to enrich existing ontologies. Their objectives, data formats, and processing assumptions therefore differ fundamentally from those of the present study. Consequently, the proposed methodology is not directly comparable to existing frameworks, as their inputs and intended outputs operate at different ontology levels and desired outputs. Instead, this work is a methodological demonstration showing that pattern-based extraction can automatically derive both ontology schema and instances from Thai semi-structured descriptive data, illustrating the applicability of the approach within comparable contexts.

The proposed pattern-based approach demonstrates that ontology schema and instances can be generated effectively from semi-structured Thai textual data. By relying on the lexical method syntactic patterns instead of annotated corpora, the method automatically constructs a relation-rich ontology suitable for domain analysis and semantic querying. The obtained ontology illustrates how semi-structured content can be transformed into formal knowledge without relying on deep linguistic resources.

The results indicate that pattern-based generation achieves a practical level of relational coverage, capturing both conceptual hierarchies and instance-level properties that reflect actual usage in the dataset. The observed performance supports the viability of pattern-based modelling as a resource-efficient alternative to machine-learning-based ontology extraction, particularly for low-resource languages. However, the automatically generated ontology still functions as a preliminary model rather than a fully expert-validated one. Its primary value lies in providing a structured foundation that can be expanded, merged, or refined through subsequent human supervision.

6.2 Limitations and Data Dependency

As a data-driven process, the ontology's completeness and accuracy depend on the quality and representativeness of the source data. The extraction rules can only capture relations explicitly stated in the input; therefore, knowledge that does not appear in the data remains absent. This limitation was reflected in the uneven distribution of extracted relations and the occasional sparsity of certain classes. When larger and more diverse data were provided, the ontology became richer but also more prone to inconsistency, revealing the trade-off between coverage and precision.

Another important limitation lies in Thai word segmentation, which remains inherently ambiguous

in compound expressions and domain-specific terminology. Although this study employed Lexeme Tokenizer with a customised medical dictionary and manual verification to mitigate such errors, segmentation quality still affects template extraction accuracy. Alternative approaches that minimise reliance on explicit segmentation—such as character-based or subword neural models—have shown promise in recent research. However, these methods typically require large, annotated corpora and substantial computational resources, making them less compatible with the semi-structured and low-resource nature of the data targeted in this study. Consequently, the current dictionary-based approach remains practical for this domain, while future work may explore hybrid or lightweight neural models once suitable Thai resources become available.

Another concern arises from the characteristics of Thai descriptive data. Publicly available Thai texts are often unorganised, lack a consistent structure, and are rarely verified by domain experts. Consequently, errors, redundancy, or colloquial phrasing may propagate into the generated ontology. Thai linguistic ambiguity further complicates the process, as many words share identical forms across different meanings. Such ambiguity affects both class naming and property extraction, occasionally resulting in overlapping or generalised relations. The pattern-based method mitigates some of these issues through contextual rule design, yet semantic drift still occurs when terms are polysemous or loosely used in the source material.

6.3 Human Post-Processing and Validation

Although the ontology can be generated automatically, human involvement remains essential for achieving semantic completeness and domain accuracy. The current methodology cannot fully substitute the analytical reasoning and contextual judgement provided by ontology engineers and medical specialists. Expert review is required to refine class hierarchies, adjust property naming, and verify the logical consistency of relations. This step is particularly important for medical information, where subtle differences in terminology may alter interpretation or usage.

Human post-processing also plays a crucial role in transforming the automatically generated structure into a truly knowledge-representing ontology. Through manual validation, ambiguous or duplicated elements can be resolved, and concept definitions can be standardised to improve interoperability with existing ontologies. In this respect, automation and expertise are complementary rather than competing forces. The proposed method significantly reduces the initial workload by producing an initial ontology from semi-structured data, while experts focus their effort on semantic refinement instead of exhaustive manual construction. Once validated, the generated ontology provides a reliable base schema for further ontology enrichment, which may be performed either manually by experts or automatically when appropriate resources are available. This flexibility ensures that the ontology can continue to evolve through successive layers of enrichment—whether by domain-driven expansion or algorithmic extraction—while maintaining a coherent structure and semantic accuracy. To further mitigate the problem of missing or incomplete relations, future work may explore ontology-completion or knowledge-graph-completion techniques to enrich the dataset automatically when suitable external resources become available.

7. CONCLUSIONS

This work aims to assist and partially automate the knowledge-extraction process to reduce the workload of human experts. Since domain-specific textual data are often organized in table-like structures with recurring linguistic patterns, this research exploits these repetitive word patterns to detect both shared and unique components in Thai semi-structured text for ontology generation. The central idea is that shared linguistic patterns form property templates, while variable parts become instance values, allowing automatic derivation of both conceptual properties and individual instances. For computational use, post-processing steps including semantic naming of classes and relations and conversion to RDF/OWL syntax were applied for standardization and interoperability.

The targeted data in this study are Thai drug instructions and usage information. Two complementary datasets, I-Med (clinical prescription records) and Pobpad (public drug information), were integrated for ontology and instance extraction. The resulting ontology comprises six main concept trees, five object properties, eleven data properties, and 456 instances. When evaluated through SPARQL queries designed by domain experts, the ontology achieved an average Precision of 0.97, Recall of 0.90, and F1-score of 0.91, confirming its accuracy and usability.

The main contribution of this research is a pattern-based method for ontology generation and instance extraction from Thai semi-structured medicine data. The findings demonstrate that data-driven template extraction can effectively produce a reliable ontology without heavy manual modeling. The method reduces expert workload and provides a scalable foundation for semantic querying and knowledge retrieval in Thai medical and health information systems.

DISCLAIMER

The ontology and instance extraction methodology presented in this study were developed for research purposes to demonstrate automated knowledge modeling in the medical domain. The generated ontology

does not serve as a substitute for professional medical advice, diagnosis, or treatment. All outputs and inferences derived from the ontology must be verified and approved by qualified medical experts or relevant authorities before any clinical or operational application. The authors assume no liability for any misuse or interpretation of the ontology beyond its intended research scope.

AUTHOR CONTRIBUTIONS

Conceptualization, T.R. and R.K.; methodology, T.R. and R.K.; software, T.R.; validation, T.R., R.K. and T.S.; formal analysis, T.R., R.K. and T.S.; investigation, T.R.; data curation, T.R.; writing—original draft preparation, T.R.; writing—review and editing, T.R., R.K. and T.S.; visualization, T.R.; supervision, R.K.; funding acquisition, T.R. All authors have read and agreed to the published version of the manuscript.

References

[1] R. Studer, R. Benjamins and D. Fensel, "Knowledge engineering: Principles and methods," *Data & Knowledge Engineering*, vol. 25, no. 1–2, pp. 161–198, 1998.

[2] W. Swartout and A. Tate, "Ontologies," in *IEEE Intelligent Systems and their Applications*, vol. 14, no. 1, pp. 18-19, Jan.-Feb. 1999

[3] N. F. Noy and D. L. McGuinness, "Ontology development 101: A guide to creating your first ontology," *Stanford Knowledge Systems Laboratory Technical Report KSL-01-05*, 2001.

[4] B. Smith, "Beyond concepts: Ontology as reality representation," in *Formal Ontology in Information Systems – Proc. 3rd Int. Conf. (FOIS 2004)*, A. C. Varzi and L. Vieu, Eds. Amsterdam: IOS Press, pp. 73–85, 2004.

[5] R. Mizoguchi, "Tutorial on ontological engineering—Part 1: Introduction to ontological engineering," *New Generation Computing*, vol. 21, no. 4, pp. 365–384, 2003.

[6] J. Davies, "Lightweight ontologies," in *Theory and Applications of Ontology: Computer Applications*, pp. 197–229, 2010.

[7] T. R. Gruber, "Towards principles for the design of ontologies used for knowledge sharing," in *Formal Ontology in Conceptual Analysis and Knowledge Representation*, N. Guarino and R. Poli, Eds. Deventer, The Netherlands: Kluwer Academic Publishers, 1993.

[8] Protégé, "A free, open-source ontology editor and framework for building intelligent systems," [Online]. Available: `https://protege.stanford.edu`. [Accessed: Jan. 6, 2020].

[9] K. Kozaki, Y. Kitamura, M. Ikeda and R. Mizoguchi, "Hozo: An environment for building/using ontologies based on a fundamental consideration of 'role' and 'relationship'," in *Proc. 13th Int. Conf. Knowledge Engineering and Knowledge Management (EKAW 2002)*, Sigüenza, Spain, pp. 213–218, Oct. 1–4, 2002.

[10] B. Motik, R. Shearer and I. Horrocks, "Hypertableau reasoning for description logics," *Journal of Artificial Intelligence Research*, vol. 36, pp. 165–228, 2009.

[11] V. Haarslev and R. Möller, "Racer: A core inference engine for the semantic web," in *Proc. 2nd Int. Workshop on Evaluation of Ontology-based Tools (EON2003)*, held at ISWC 2003, Oct. 20, 2003.

[12] N. Chalortham, P. Leesawat, T. Ruangrajitpakorn and T. Supnithi, "A framework of ontology-based tablet production supporting system for a drug reformulation," *IEICE Trans. Inf. & Syst.*, vol. E94-D, no. 3, pp. 448–455, 2011.

[13] T. Ruangrajitpakorn, C. Prombut and T. Supnithi, "A development of an ontology-based personalised web from rice knowledge website," in *Proc. 13th Int. Conf. on Knowledge, Information and Creativity Support Systems (KICSS 2018)*, pp. 1–6, 2018.

[14] R. Navigli and S. Ponzetto, "BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network," *Artificial Intelligence*, vol. 193, pp. 217–250, 2012.

[15] M. Sabou, C. Wroe, C. Goble and G. Mishne, "Learning domain ontologies for web service descriptions: An experiment in bioinformatics," in *Proceedings of the 14th international conference on World Wide Web*, Chiba, Japan, pp. 190-198, 2005.

[16] P. Cimiano, A. Hotho and S. Staab, "Learning concept hierarchies from text corpora using formal concept analysis," *Journal of Artificial Intelligence Research*, vol. 24, pp. 305–339, 2005.

[17] P. Cimiano and J. Völker, "Text2Onto: A framework for ontology learning and data-driven change discovery," in *Proc. 10th Int. Conf. on Applications of Natural Language to Information Systems (NLDB)*, pp. 227-238, 2005.

[18] H. Davalcu, S. Vadrevu, S. Nagarajan and I. V. Ramakrishnan, "OntoMiner: bootstrapping and populating ontologies from domain-specific Web sites," in *IEEE Intelligent Systems*, vol. 18, no. 5, pp. 24-33, Sept.-Oct. 2003.

[19] E. Maier, S. Streit, T. Diggelmann and M. Hoffleisch, "Learning a lightweight ontology for semantic retrieval in patient-centered information systems," *Int. Journal of Knowledge Management*, vol. 7, no. 3, 2011.

[20] F. Suchanek, G. Kasneci and G. Weikum, "Yago: A core of semantic knowledge," in *Proceedings of the 16th international conference on World Wide*

Web, pp. 697–706, 2004.

[21] BabelNet, [Online]. Available: https://babelnet.org.

[22] D. Ferrucci *et al.*, "Building Watson: An overview of the DeepQA project," *AI Magazine*, vol. 31, no. 3, pp. 59–79, 2010.

[23] T. Boiński and A. Ambrożewicz, "DBpedia and YAGO as knowledge base for natural language based question answering—The evaluation," in *Proc. Int. Conf. on Man-Machine Interactions 5 (ICMMI 2017)*, pp. 251-260, 2018.

[24] S. Calegari and G. Pasi, "Personal ontologies: Generation of user profiles based on the YAGO ontology," *Information Processing & Management*, vol. 49, no. 3, pp. 640–658, 2013.

[25] A. Amalki, K. Tatane and A. Bouzit, "Deep learning-driven ontology learning: A systematic mapping study," *Engineering, Technology & Applied Science Research*, vol. 15, no. 1, pp. 9461–9468, 2025.

[26] J. Chen, O. Mashkova, F. Zhapa-Camacho, R. Hoehndorf, Y. He and I. Horrocks, "Ontology Embedding: A Survey of Methods, Applications and Resources," in *IEEE Transactions on Knowledge and Data Engineering*, vol. 37, no. 7, pp. 4193-4212, July 2025.

[27] K. Yang, C. Huo, Y. Zhang, Q. Feng, and Y. Song, "TransBox: EL++-closed ontology embedding," in *Proc. 33rd ACM Int. Conf. on Information and Knowledge Management (CIKM '24)*, ACM, 2024. [Online]. Available: https://arxiv.org/abs/2410.14571.

[28] S. B. Giglou, J. D'Souza, and S. Auer, "LLMs4OL: Large language models for ontology learning," in *Proc. 22nd Int. Semantic Web Conf. (ISWC 2023)*, CEUR-WS, pp. 1–13, 2023.

[29] M. Azzi, "A methodology for building a medical ontology with limited domain experts' involvement," *Digital Health and AI Ethics*, vol. 5, no. 2, pp. 18–32, 2025.

[30] C. Saetia *et al.*, "Financial product ontology population with large language models," in *Proc. TextGraphs-17: Graph-based Methods for Natural Language Processing*, Bangkok, Thailand, 2024, pp. 31–40. [Online]. Available: https://aclanthology.org/2024.textgraphs-1.4.

[31] PobPad Website, "Medicine information," (in Thai). [Online]. Available: https://www.pobpad.com/ยา-a-z. [Accessed: Aug. 18, 2021].

[32] National Electronics and Computer Technology Center, "Thai Lexeme Tokenizer," [Online]. Available: http://www.sansarn.com/lexto/. [Accessed: Dec. 8, 2019].

[33] V. A. Nunez, B. A. Hong and E. Ong, "Automatically extracting templates from examples for NLP tasks," in *Proc. 22nd Pacific Asia Conf. on Language, Information and Computation*, Cebu, Philippines, pp. 452–459, 2008.

[34] T. Ruangrajitpakorn, W. na Chai, P. Boonkwan, M. Boriboon and T. Supnithi, "The design of lexical information for Thai to English MT," in *Proc. SNLP 2007*, Pattaya, Thailand, 2007.

[35] P. Palingoon, P. Chantanapraiwan, S. Theerawattanasuk, T. Charoenporn and V. Sornlertlumvanich, "Qualitative and quantitative approaches in bilingual corpus-based dictionary," in *Proc. 5th Symp. on Natural Language Processing & Oriental COCOSDA Workshop 2002*, 2002.

[36] P. Patel-Schneider, P. Hayes and I. Horrocks, "OWL web ontology language semantics and abstract syntax," *W3C Recommendation*, Feb. 2004. [Online]. Available: https://www.w3.org/TR/2004/REC-owl-semantics-20040210.

[37] G. Klyne and J. Carroll, "Resource description framework (RDF): Concepts and abstract syntax," *W3C Recommendation*, Feb. 2004. [Online]. Available: https://www.w3.org/TR/2004/REC-rdf-concepts-20040210.

[38] OWL API, version 4.5.9. [Online]. Available: https://github.com/owlcs/owlapi. [Accessed: Jan. 22, 2021].

[39] The Gene Ontology Consortium, "Gene ontology: tool for the unification of biology," *Nature Genetics*, vol. 25, no. 1, pp. 25-29, May 2000.

[40] M. Feld and C. Müller, "The Automotive Ontology: Managing knowledge inside the vehicle and sharing it between cars," in *Proc. 10th Int. Conf. Automotive User Interfaces and Interactive Vehicular Applications (Auto-UI 2010)*, Pittsburgh, USA, Nov. 2010.

Taneth Ruangrajitpakorn is a Ph.D. candidate in Computer Science at Thammasat University, Thailand. He currently works at the Artificial Intelligence Research Group (AINRG), National Electronics and Computer Technology Center (NECTEC), where he conducts research and development in AI technologies, especially for Thai language and government applications. His primary research interests include knowledge graphs, ontology, Thai natural language processing, and artificial intelligence. His work spans machine translation, knowledge engineering, and applied artificial intelligence, with a focus on recent methods such as zero-shot large language model classification and deep learning tailored for governmental and public sector challenges in Thailand. He has been involved in projects that integrate semantic technologies with practical AI deployments for language understanding and real-world system use.

Thepchai Supnithi is a principal researcher at the Artificial Intelligence Research Group (AINRG), National Electronics and Computer Technology Center (NECTEC), Thailand, where he previously served as the Director of the group. His research interests include natural language processing, applied artificial intelligence, and knowledge graphs. He is actively involved in advancing AI research and applications in Thailand and the Asia-Pacific region. His main research covers machine translation, text summarization, corpus construction, and applied artificial intelligence across diverse domains such as culture, education, and medical applications. In addition to his research activities, he serves on the executive committees of several international organizations, including the Asia-Pacific Society for Computers in Education (APSCE), the Association for Computational Linguistics – Asia Chapter (AACL), and the International Joint Conference on Natural Language Processing (IJCNLP). He is currently the President of the Artificial Intelligence Association of Thailand (AIAT).

Rachada Kongkachandra is an Assistant Professor in the Data Science and Innovation Program, College of Interdisciplinary Studies, Thammasat University, Thailand. She earned her B.Sc. in Computer Science from Thammasat University in 1990, M.Sc. in Computer Technology from the Asian Institute of Technology in 1991, and Ph.D. in Electrical and Computer Engineering from King Mongkut's University of Technology in 2005. Her research interests span AI Ethics, AI Governance, Data Governance, and Social Data Science, with a focus on integrating responsible AI principles and data-driven methods to address societal issues such as education and aging populations. As an IEEE CertifAIEd^{TM} Authorized Assessor, she contributes to developing and assessing frameworks for trustworthy and socially impactful AI. With over three decades of experience, she engages in interdisciplinary projects bridging technology, governance, and societal needs.