Knowledge Graph for Deriving Insights on The Thai Government Dataset

K. Saratoon, A. Chutiporn, and S. Nuttapong

Abstract— **Natural language processing (NLP) is mandatory in working with text. There are many tools and applications that are based on it. However, most of those tools are often operated in, or only, the English language. In recent years, there has been a continuing development for NLP tools to support other languages or creating a specific tool for certain language for simple NLP tasks, but for some of the advanced tasks, the advancement is still behind that of the English language, Thai language is also one of them. So, in this research, the capabilities of the currently existing Thai NLP tools are explored and evaluated, with the tasks of extracting text from the Thai government dataset (eMENSCR) and creating the knowledge graph from it to improve data interpretability and gain more insight from the data, by utilizing queries that are exclusive, or less complex to execute, when the data is stored in the graph database such as performing a path traversal or relationship counting on the data. Natural language processing's part of speech tagging and named entity tagging is used to perform entity and relation extraction after filtering the unneeded data fields. Then the extracted information will be formulated into the format of "triple", which is in the form of (head, relation, tail). After the process of triple construction is finished, The triples are evaluated by Precision, recall, and F1 in order to measure the pipeline's performance and import to the Neo4j for query testing. The obtained results show that there is still room for improvement for both the tools and the methodology itself.**

***Keywords—Knowledge graph, NLP, Thai*.**

I. Introduction

Nowadays, most information is stored in a digitized format so that it can be easily looked up, modified, or updated. However, when the amount of information started to increase over time, it became hard for the user to search, utilize or analyzed it. There are many ways to deal with these problems such as choosing a suitable format to store those data (e.g., using a relational database to store records that value high consistency, using a document model to flexibly store various types of documents together) or changing the format of the existing data into another one to improve the data representation or other qualities. The Knowledge graph is one of those data formats.

A knowledge Graph (KG) is comprised of multiple triples linking together, the structure of each triple is <entity, relation, entity>, whereas the entities are real-world things and the relation represents how they are related to each other. One of the benefits of the knowledge graph is that, since the data is represented in the graph format, which is a visual format, it is easier to understand the data. Another benefit is that, the knowledge graph summarizes the flow of data, in other words, the relationship of each entity and how they impact each other, which can be useful for analysis or decision-making. So, overall, it can be said that a knowledge graph is useful for visualizing and summarizing the data, as well as increasing the interpretability of large amounts and/or complex data, which are the characteristics of the Electronic Monitoring and Evaluation System of National Strategy and Country Reform (eMENSCR) dataset.

eMENSCR dataset is JSON data that contains information on government public projects in the Thai language. However, due to its large amount of information and details, it's hard to gain insight into it. But if that information is extracted and represented in another format such as a graph, the data should be easier for understanding and analysis. For the information extraction method, there are various ways it could be done, one of them being natural language processing (NLP), which is one of the fields that receive attention from many researchers in recent years and is continuously improving, but most of them are usually in English language or other. There is not much research done on Thai language information extraction techniques, even less for constructing a Thai language knowledge graph using that information. So, the focus of this research is on a method to extract useful information in Thai language by using NLP techniques and represent that knowledge in the knowledge graph format to increase the interpretability and accessibility of the data.

This paper is organized as follows, Section 2 reviews related literature, Section 3 describes the proposed methodology, Section 4 discuss the result and evaluation Section 5 concludes and presents future research direction.

II. Background and Related Work

A. eMENSCR

Electronic Monitoring and Evaluation System of National Strategy and Country Reform (eMENSCR), is a platform that track the process and evaluate the performance of government projects in order to achieve the desired goal and improve the country as a whole according to the masterplan as well as reduce the existing limitations and drawbacks such as an employee working independently.

eMENSCR is a paperless system, so it is also eco-friendly and capable of making the progress report of the projects to the citizen, which led to the inclusion of people in tracking the project progress and its status. NSTDA intend to use eMENSCR as a centralized information database by integrating the data with others government sector to create a one-stop service where all information can be tracked in the same system in the future

Author K. Saratoon is with the ICT department, Asian Institute of Technology, Bangkok, Thailand (phone: 089-310-3310; e-mail: st122149@ait.asia).

Author A. Chutiporn. is with the ICT department, Asian Institute of Technology, Bangkok, Thailand (e-mail: chutiporn@ait.asia).

Author S. Nuttapong is with the *Strategic Analytics Networks with Machine Learning and AI (SAI), National Electronics and Computer Technology Center (NECTEC), Bangkok, Thailand* (e-mail: nuttapong.sanglerdsinlapachai@nectec.or.th).

979-8-3503-0219-6/23/$31.00 ©2023 IEEE

TABLE I. AN EXAMPLE OF EACH DATA FIELD'S VALUE

Name	Description	Values example	Translation
Name	Project name	โครงการพัฒนาศักยภาพผู้ประกอบการขนส่งไม่ประจำทางด้วยรถโดยสาร ปี 2563	transportations-related personnel improvement projects
Budgets	Project's budget	30000	-
Masterplan	Show which category of masterplan this project is under	masterplan:0701 y1_direction:07010005 y1_goal:070104	-
Rationale	The logic behind the project	กำกับดูแลการขนส่งรถโดยสารไม่ประจำทางให้มีมาตรฐานด้านความปลอดภัยที่ดีขึ้น	Oversees and improves the safety of transportation service
Objective	The objective of the project	1.ให้คำปรึกษาและรับฟังปัญหาผู้ประกอบการขนส่ง 2.ให้ความรู้เรื่องข้อควรปฏิบัติแก่ผู้ประกอบการขนส่ง	1. give advice to the service person as well as consult with their problems 2. Instruct the personnel with relevant knowledge.
Outcome	The outcome of the project	จัดอบรม สัมมนา แก่ผู้ให้บริการตามมาตรฐานที่กำหนด	Conduct seminar according to the agreed upon standards
Expected_results	The expected result of the project	1.การขนส่งมีคุณภาพมากขึ้น 2.ประชาชนมีความไว้วางใจในการใช้บริการ	1. improves the quality of transportation service 2. increase the customer's trust in a service
Project_gantt_chart	The procedure of the project	1.ประชุมเกี่ยวกับการเตรียมสัมมนา 2.แจ้งหน่วยงานที่เกี่ยวข้องให้ทราบ 3.จัดสัมมนา 4.ประเมินผลความเข้าใจของผู้เข้ารับการสัมมนา	1. Meeting about seminar details 2. inform related-personnel 3. conduct seminar 4. Evaluate the participant's knowledge
Area_description	The area where the project took place	กรุงเทพมหานคร	Bangkok
Target_group	The target group of the project	1.ผู้ประกอบการขนส่ง 2.ประชาชน	1. service person 2. citizen
Approvers	The approvers of the projects	1. อธิบดีกรมการขนส่งทางบก 2.รองปลัดกระทรวงคมนาคม ด้านการขนส่ง 3.ปลัดกระทรวงคมนาคม	1. Director-general of Department of Land Transport 2. Deputy Permanent Secretary of the Ministry of Transport 3. Deputy of the Ministry of Transport

How the real data looks in MongoDB can be seen in Figure 1

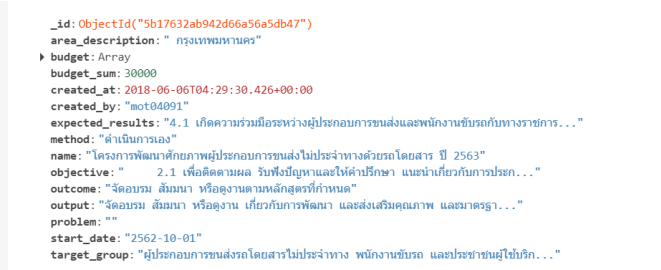

Figure 1. Data examples in MongoDB

B. *Knowledge graph*

A knowledge graph (KG) is a knowledge base that uses a graph-structured data model to integrate data. Knowledge graphs are often used to represent a connection of real-world entities such as objects, events, situations, or abstract concepts.. There are two main components for a knowledge graph 1.) The Node which represents the entity and 2.) The Edge represents the relationships between each entity.

KG-related research can be classified into two categories, which are research about KG construction techniques and KG application [1]. For the KG construction part, the crucial step lies in entity extraction and relation extraction. There are various ways how it can be done, depending on the purpose of how it is intended to be used. Some KG is constructed to serve as a large knowledge base for a user to look up information in them, by combining information from various sources such as DBpedia, YAGO, Wikidata, etc. Some KG is also used as a basis for practical applications such as Question answering systems, Recommendation systems, and information retrieval systems. Or some even constructed KG in order to perform in-depth analysis on the specific area/fields such as a domain-specific KG. An example in the business domain is [2], where they used a KG to perform fraud detection by using data about managers and customers, and in the engineering domain, [3], where they used a KG to compare different engineering models and try to derive new knowledge from them by constructing KG out of their dataset. This research purpose is the same as the latest one, to gain more insight into the dataset and also to perform consistency-checking.

In recent times, research has been focused on using a machine-learning model to perform information extraction rather than a rule-based method such as [4], where they used a constituency parse tree along with part of speech tagging for entity extraction part then use semantic role labeling (SRL) for relation extraction, and [5], where they used supervised sequence labeling model with bidirectional long short-term memory (BiLSTM) layer for entity extraction. For the relation extraction, they used word-level extraction to identify the verb then they clustered the verb with similar meanings together into the same categories (synset grouping). Another approach would be this, where they used the Bidirectional Encoder Representations from Transformer (BERT) model to perform entity extraction and use a dependency parser in SpaCy for detecting verbs in the process of relation extraction.

However, all the mentioned approaches operated on the "English" text, for Thai text, the tools or models or datasets for training are quite limited, so some methodologies are not possible with the currently available tools. According to [6], some of the tools for information extraction are AIforThai developed by NECTEC, PyThaiNLP developed by [7] and Spacy-thai developed by Koichi Yasuoka. This research will use the mentioned tools to perform information extraction.

C. *Natural language processing*

Natural is one of the branches of computer science, related to artificial intelligence (AI), focusing on developing the machines, such as computers, with the goal of making it possible for the machines to understand the word or text in human language. It combines many fields of knowledge in order to reach that goal such as, machine learning, statistical and computational linguistics. Some examples of NLP tasks are: Speech recognition: converting voice to text, Text extraction: identifies certain texts and extract them from a large amount of text, Machine translation: translate the text from one language to another language

1) *PyThaiNLP*

PyThaiNLP is a Python library for text processing with a focus on the Thai language. It provides standard NLP functions for Thai, for example, tokenization, parsing, part-of-speech tagging, etc. Some of these functions can also be used in a command-line interface [7]. In this research, the part of speech function (POS) and name-entity tagging (NE) will be the important step for information extraction.

2) *Part of speech tagging*

Part-of-speech (POS) tagging is a Natural Language Processing process that categorizes a word into specific tag according to the context of the sentences [8].

3) *Name-entity recognition*

Name-entity tagging or Name-entity recognition is the task of detecting and categorizing specific or important information in the text. Every detected entity is classified into a predetermined category [9] such as person names, organizations, and locations. The example of NER by PythaiNLP can be seen in Table 2, where the English text is just the translated version of Thai text.

TABLE II. NAME ENTITY RECOGNITION EXAMPLE

	Values
Input	มหาวิทยาลัยเทคโนโลยีราชมงคลธัญบุรีมีชื่อเสียงเป็นที่ยอมรับ
Translation of input in English	The Rajamangala University of Technology Thanyaburi is widely known and accepted
Result	<Organization>มหาวิทยาลัยเทคโนโลยีราชมงคลธัญบุรี</Organization> มีชื่อเสียงเป็นที่ยอมรับ
Translation of result in English	<Organization>The Rajamangala University of Technology Thanyaburi</Organization> is widely known and accepted

4) *Chunking*

Chunking or shallow parsing is a technique usually perform along with POS tagging for information extraction. It will group syntactically-related words into the same phrase such as a Noun phrase (NP) or a Verb phrase (VP), where the word in each phrase will not overlap with another phrase. It is useful when we want to extract specific phrases of our interest from a large amount of information.

III. METHODOLOGY

The overview of the process can be seen in Figure 2

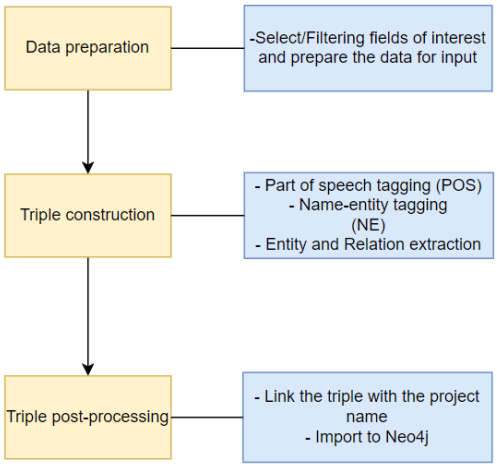

Figure 2. Process Overview

1) *Data preparation*

Firstly, the data needed to be prepared before we can use it such as filtering the unneeded data fields. This can be done by importing the data into MongoDB and exporting only the necessary data fields. In this research, the aim is to create a Knowledge graph triple out of a long text of strings so most of the data fields that will be used have a value as a string. The list of data fields used is the same as shown in Table 1.

2) *Triple construction*

For this part, the process is according to the diagram in Figure 3

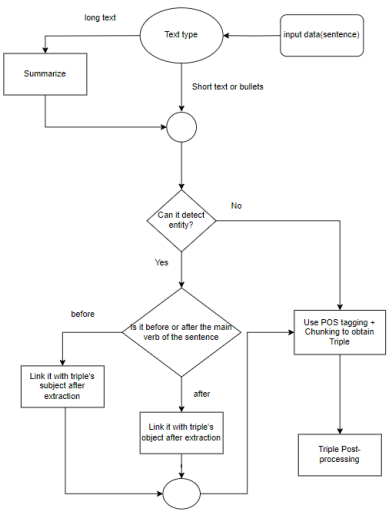

Figure 3. Triple construction pipeline

First, the input data, which in this case is a sentence, will be checked if it is a "long text" or a "short text/bullets". The text will go through a text summarization process before continuing if it is classified as a "long text".

Next, the output will go through an entity detection process to see if it does have an entity in it or not, if the entity is not detected, the input will go through a part of speech tagging and chunking process to obtain the triple in the form of (subj, verb, obj) of the sentence, which is considered to be (head, relation, tail) respectively. An example of this can be seen in Table 3.

TABLE III. AN EXAMPLE OF POS-TAGGING COMBINED WITH CHUNKING

	Values
Input	ผู้เข้าร่วมโครงการพึงพอใจต่อกำไรที่ได้รับ
Translation of input in English	The project's participants are satisfied with the profit obtained
POS-tagging and chunking	(ผู้เข้าร่วม, b-np),(โครงการ, i-np),(พึงพอใจ, b-vp),(ต่อ, i-vp),(กำไร, i-vp),(ที่, i-vp),(ได้รับ, i-vp)
Result	(ผู้เข้าร่วมโครงการ, พึงพอใจ, ต่อกำไรที่ได้รับ)
Translation of result in English	(The project's participants, are satisfied, with the profit obtained)

To explain Table III, after tagging the input with pos-tagging, the chunking will group the related word into a phrase, such as a Noun phrase (NP) or Verb phrase (VP), in IOB format, where the words in each group will not overlap with another group. Then the Noun phrase will be the "head" of the triple, The first word of the Verb phrase, which is usually a main verb, will be the "relation" in the triple, and the leftover part will be the "tail" of the triple. The output without name-entity will only have one triple as a result, as shown in Figure 4.

TABLE IV. AN EXAMPLE OF OUTPUT TRIPLE WITHOUT AN ENTITY

	Values
Input	อาจารย์สอนนักเรียนได้อย่างมีประสิทธิภาพ
Translation of input in English	The professors educate their students effectively
Result	(อาจารย์, สอน, นักเรียนได้อย่างมีประสิทธิภาพ)
Translation of result in English	(The professors, educate, their students effectively)

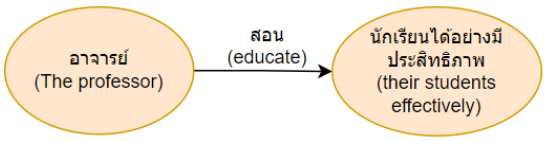

Figure 4. An example of output triple without entity as a graph

However, in the case that the entity is detected, The position of the entity in the sentence will be checked, if it is before the main verb, it will be linked to "subject" (head of the triple) or if the position of an entity is after the main verb, it will be linked to "object" (tail of the triple). The part of the triple that contains the entity will also be linked to the entity itself with the "has entity" relation. And the entity will be linked with the entity type by the relation "type". The output with the name entity can be seen in Figure 5.

TABLE V. AN EXAMPLE OF OUTPUT TRIPLE WITH ENTITY

	Values
Input	นิสิตของจุฬาลงกรณ์มหาวิทยาลัยมีชื่อเสียงเป็นที่ยอมรับในระดับสากล
Translation of input in English	Chulalongkorn University's students are famous and accepted internationally
Result	(นิสิตของจุฬาลงกรณ์มหาวิทยาลัย, มีชื่อเสียง, เป็นที่ยอมรับในระดับสากล), (นิสิตของจุฬาลงกรณ์มหาวิทยาลัย, has_entity, จุฬาลงกรณ์มหาวิทยาลัย), (จุฬาลงกรณ์มหาวิทยาลัย, type, Organization)
Translation of result in English	(Chulalongkorn University's students, are famous, and accepted internationally), (Chulalongkorn University's students, has_entity, Chulalongkorn University), (Chulalongkorn University, type, Organization)

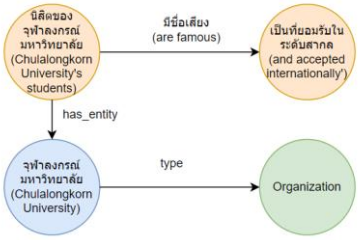

Figure 5. An example of output triple with entity as a graph

3) Triple post-processing

However, we still need to connect these triples to their respective project name to separate each project's information. This can be done by using a data field name of those triples as a relation to connecting the project name to the head and tail of each triple. So, each triple will be changed from

(head, relation, tail) to

(head, relation, tail)

(project name, [data field name]-start, head)

(project name, [data field name]-end, tail)

After this process is finished, we filtered only the unique and completed triple, (triple that is not null in every position) to be imported into Neo4j

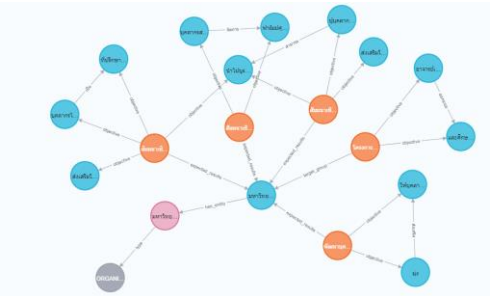

Figure 6. Sample of the final graph. The orange node represents the project's name, the blue node represents the project's details, the pink node represents the entity and the gray node represents the entity type.

IV. RESULTS

There will be two types of evaluation. The first one is to check the correctness of the constructed triple and the second

is to evaluate the constructed knowledge graph usability, the KG will then be imported into the Neo4j platform for visualization and querying.

1) Constructed triple's evaluation

According to [10] , One of the ways to evaluate the quality of the constructed KG is by randomly selecting a certain number of sentences and annotating it manually to find the "ground truth" results, and together with the results obtained from the pipeline, used them to calculate precision, recall, and F1.

For this work, the criteria for checking the triple whether it is correct or not is "if the main verb/relation of the triple from the pipeline match with the ground truth, it is considered to be correct". The reason that only the "relation" of the triple is evaluated instead of the whole triple is because the chunking process "divide" the sentence into three parts (The NP before the main verb, The main verb, and The NP after the main verb) with main verb being the separator of "head" and "tail" of the triple. So, if the main verb is correct, the whole triple is correct.

For triple's evaluation, a sample of 187 sentences were obtained from 30 projects and annotated manually to find the ground truth "main verb" of the sentence, with one sentence containing one main verb, resulting in 187 ground truth main verb. And those sentences pass through the pipeline resulting in 179 triples obtained, which means there are 179 main verbs. And among those, only 119 of them matched the ground truth. The evaluation process can be seen in Figure 7

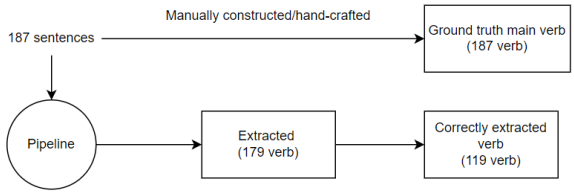

Figure 7. The pipeline for evaluation

By using the obtained information to calculate the precision, recall, and F1. The results can be seen in Table 6

TABLE VI. TRIPLE'S EVALUATION RESULT

	Values
Precision	0.664
Recall	0.636
F1	0.650

The result is about 65%, which is not very impressive in terms of output, but considering that the methodology applied was also not very complicated, as in using POS-tagging combined with chunking that operates on the principle of phase structure grammar for extraction, it's not unexpected. One of the goals of this research is to explore the performances of the existing tools, so this result can be considered a baseline for comparison with other more-complex methodologies. As for a way to improve the results, there are alternatives like using other tools/methods like dependency parsing, combining the use of many tools together, or even using a deep learning model that is customized for this task

2) Evaluation of query

In order to demonstrate the system's capability, two types of queries have been prepared. 1.) the basic queries, which can already be executed originally, and 2.) the advanced queries, which are queries that make use of the characteristics of the graph database to gain deeper insight into the data. Some of the advanced queries are:

Q1: Find the organizations that are involved in many projects

The purpose of this query is to inspect the distribution of the organizations that are focused on by the government. This query cannot be executed originally because the "organization" specified in the query refers to the product of NER in the extraction process, which originally doesn't exist as a separate field in the original JSON file. To achieve similar results in the document model, such as MongoDB, a lot of complex functions like unwind, group, or reduce are necessary.

```
1 match (a:`โครงการ`)-[]→(b)-[r:has_entity]
2 →(c)-[d:type]→(e)
3 where e.name = "ORGANIZATION"
4 return count(distinct (a)) as frequency
5 ,c.name as organization
6 order by count(distinct (a))  desc
```

frequency	organization
22	"มหาวิทยาลัยเทคโนโลยีราชมงคลธัญบุรี"
12	"มหาวิทยาลัยเทคโนโลยี"
6	"คณะเทคโนโลยีการเกษตร"

Figure 8. Q1 query and result

Q2: Find possible redundancy projects (similar expected results)

The purpose of this query is to find out the possible duplication of project entry, as well as to compare those projects with each other on some other dimension, such as budget; why 2 projects with the same results have a different amount of budget? or maybe on the masterplan direction; why projects with the same results are classified in different categories of masterplan?. This query can be used as a starting point for those inspections. Also, This query also makes use of path traversal, which is one of the graph characteristics to obtain the answer indirectly through the connected network of graphs instead of directly specifying the answer. The idea is "we want to obtain the results that are the same on some dimensions/field without knowing of or directly specifying it". This is difficult to achieve in the document model because we are required to specify the exact value in the field we are looking for in order to search or filter.

```
match(n)-[r:expected_results]→(k)-[p]→
(s)←[x:expected_results]-(n)
with n,k,p,s
match(j)-[r:expected_results]→(k)-[p]→
(s)←[x:expected_results]-(j)
where j.name <> n.name
return n,j,k,p,s
```

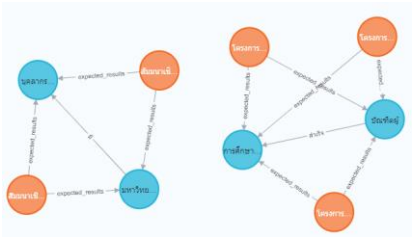

Figure 9. Q2 query and result

Q3: Find projects with the same outcome but have different budget

This query is the extension of Q2, by using outcome as a basis to compare the project budget, and also filter only the pair of projects that have different budget. The purpose of this query is to inspect on the potential bias on those two projects. Of course, there may also be other factors that result in the projects having different amount of budgets, but this can be used as a starting point for inspection.

```
match(n)-[r:outcome]→(k)-[p]→(s)←[x:outcome]-(n)
with n,k,p,s
match(j)-[r:outcome]→(k)-[p]→(s)←[x:outcome]-(j)
where j.name <> n.name
with distinct n,k,s
match (s)←[]-(k)←[q:outcome]-(n)-[r:budget_sum]→(x
with x,count(r) as times,k,s,n
where times = 1
return collect(id(n)) as project_code,
collect(x.name) as Budget_allocated
```

project_code	Budget_allocated
[275, 237]	["509209800", "32732400"]

Figure 10. Q3 query and result

V. Conclusion

This research has fulfilled two main purposes: 1.) to explore methodologies/tools for Thai language information extraction and 2.) to create a knowledge graph to perform queries that are hard to execute on the original dataset as well as find more insight into the data. For the information extraction part, this paper has mostly explored the PyThaiNLP library and how to use it for knowledge extraction combined with some of the techniques from the related work to obtain the triples. For graph queries, The demonstrated queries are just examples of what kind of query and knowledge can be retrieved when the data is in graph formats, such as path traversal or relationship count. More insight than this can be gained by importing a useful data field in the graph and by using a more complex query.

The improvement in this work that can be made are as follows; for the extraction part, as mentioned in the result section, more variations of methodologies should be applied and compared with the current approaches to clearly measure the difference in the capabilities of a tool and/or methods. As for the graph construction part, the addition of knowledge graph embedding into the construction process should be implemented in order to improve graph reasoning abilities as well as compare the performances with the current approaches.

References

[1] X. Zou, "A Survey on Application of Knowledge Graph," *Journal of Physics,* p. 2, 2020.

[2] S. Wen, J. Li, X. Zhu and M. Liu, "Analysis of financial fraud based on manager knowledge graph," *Procedia computer science,* 2022.

[3] A. Berquand and A. Riccardi, "From engineering models to knowledge graph : delivering new insights into models," in *9th International Systems & Concurrent Engineering for Space Applications Conference (SECESA 2020)*, 2020.

[4] J. L. Martinez-Rodriguez, I. Lopez-Arevalo and A. B. Rios-Alvarado, "OpenIE-based approach for Knowledge Graph construction from text," *Expert Systems With Applications,* pp. 339-355, 2018.

[5] M. Enkhsaikhan, "Understanding ore-forming conditions using machine reading of text," *Ore Geology Reviews,* p. 8, 2021.

[6] R. Arreerard, S. Mander and S. Piao, "Survey on Thai NLP Language Resources and Tools," in *European Language Resources Association*, Marseille,, 2022.

[7] W. Phatthiyaphaibun, K. Chaovavanich, C. Polpanumas, A. Suriyawongkul, L. Lowphansirikul and P. Chormai, "PyThaiNLP," 27 June 2016. [Online]. Available: https://github.com/PyThaiNLP/pythainlp.

[8] K. Pykes, "Part Of Speech Tagging for Beginners," 26 Nov 2020. [Online]. Available: https://towardsdatascience.com/part-of-speech-tagging-for-beginners-3a0754b2ebba#:~:text=Part%2Dof%2Dspeech%20(POS,the%20word%20and%20its%20context..

[9] C. Marshall, "What is named entity recognition (NER) and how can I use it?," 18 Dec 2019. [Online]. Available: https://medium.com/mysuperai/what-is-named-entity-recognition-ner-and-how-can-i-use-it-2b68cf6f545d.

[10] D. Papadopoulos, "A Methodology for Open Information Extraction and Representation from Large Scientific Corpora: The CORD-19 Data Exploration Use Case," *MDPI,* p. 14, 2020.