Performance Comparison between Neo4j-based and OWL-based Query Classification Process

Naw Thiri Wai Khin[*], Nyo Nyo Yee[*], Aung Aung Hein[**]

[*] Faculty of Information and Communication Technology, University of Technology (Yatanarpon Cyber City), Myanmar
[**] Department of Computer Technology, Defence Services Academy, Myanmar

DOI: 10.29322/IJSRP.9.04.2019.p8898
http://dx.doi.org/10.29322/IJSRP.9.04.2019.p8898

Abstract- Web query classification is emphasized by various search engines nowadays due to the increase in the size of the web as millions of web pages are added to it every day. Web query classification is to classify a user query Q_i into a list of n categories c_{i1}, c_{i2}, c_{in}. Search result pages can be grouped according to the categories predicted by query classification method. Providing query classification can help the information providers to understand users' needs based on the categories searched by the users. To build the domain corpus, most of the query classification system use ontology, Wikipedia category source, graph database etc. In this system, graph database and ontology are built as domain corpus for query classification process by using Neo4j and Web Ontology Language (OWL). Web Query Classification Algorithm (WQCA) with five steps is implemented as a web service by using XML web service technology. Proposed system classifies each domain term of user query into their relevant categories according to this WQCA algorithm by using different domain corpus. Finally, this system compares the performance between Neo4j-based and OWL-based WQCA to show the effectiveness of using graph database in the query classification process.

Index Terms- Web Query Classification, Web Service, Neo4j Graph Database, OWL

I. INTRODUCTION

Today, semantic logics are very important in query understanding to create successful web search engines. A user might not formalize the query when he seeks information although he knows what he wants. As a result, understanding the nature of the information that is needed behind the queries is important research problem. So, this system proposes the Query Classification Algorithm (QCA) for efficient Information Retrieval (IR) system.

Many other classification systems [1 - 4] have been proposed by using training dataset, user query logs, click URLs and user session data. The main advantage of using OWL or Neo4j graph database in query classification process is that the classification can be performed without requiring large data set. In this paper, Neo4j-based QCA is proposed and compared with OWL-based QCA.

The Neo4j-based QCA uses the graph database to classify the intended category of user query. Graph database is built by using Neo4j 3.1.3 which is an open source graph database supported by Neo Technology [9]. CYPHER query language is used to retrieve the matched terms and related category of domain words from the graph database. The node structure of graph database is shown in figure 3.

The OWL-based QCA uses the ontology dataset to classify the intended category of user query. Ontology file is constructed by using Protégé v 3.5 [10] and published as a dataset on the Apache Jena Fuseki server. Terms from ontology are extracted by SPARQL language using dotNetRDF which is a complete .NET library for parsing, managing, querying and writing RDF [11]. The class structure of ontology is as same as the node structure of graph database as shown in figure 5.

In this proposed system, QCA is implemented as a web service by using XML web service technology. To show the better performance of the Neo4j-based QCA, this system compares it with the OWL-based QCA.

The rest of the paper is organized as follows: related work is described in section 2. The proposed system is described in section 3. Web query classification algorithm is shown in section 4. Neo4j graph database and OWL construction are described in section 5 and 6. In section 7, implementation of proposed system is presented. Finally, evaluation of performance and conclusion are given in section 8 and 9.

II. RELATED WORK

In 2012, S. M. Fathalla and Y. F. Hassan [1] presented hybrid method for user query reformation and classification depending on fuzzy semantic-based approach and K-Nearest Neighbour (KNN) classifier. The overall processes of the system are query pre-processing, fuzzy membership calculation, query classification and reformation. Classification is performed using KNN classifier not just by keyword-based semantic but using a sentence-level semantics. After classification, user's query is reformulated to be submitted to a search engine which gives better results than submitting the original query to the search engine. Experiments show significant enhancement on search results over traditional keyword-based search engines' results.

In 2015, A. Katariya [2] presented ontology based web query classification. Query classification is one technique in which query should classify to the number of predefined categories. Query classification use ontology as a model to classify the input search queries. Ontology stores a set of concepts and semantic rules to classify user queries.

In 2014, M. M. Thannaing and A. N. Hlaing [3] proposed query classification algorithm for automatic topical classification of web queries based on domain specific ontology. In their proposed system, ontology with 12 classes is constructed as a controlled vocabulary for query classification process. To implement the proposed ontology-based query classification for information retrieval system, J2EE is used. According to their experimental results, the proposed system can outperform than traditional keyword search system. The results of their experiments showed that the accuracy of informational retrieval system is improved by using the ontology-based query classification process.

In 2006, W. Yue, Z.Chen and X. Lu [4] proposed a novel information retrieval algorithm based on query expansion and classification. The algorithm is induced by the observation that very short queries with the traditional information retrieval methods often have low precision, although they can get high recall. Their approach attempted to catch more relevant documents by query expansion and text classification. The results of the experiments showed that the proposed algorithm is more precise and efficient than the traditional query expansion methods.

III. PROPOSED SYSTEM

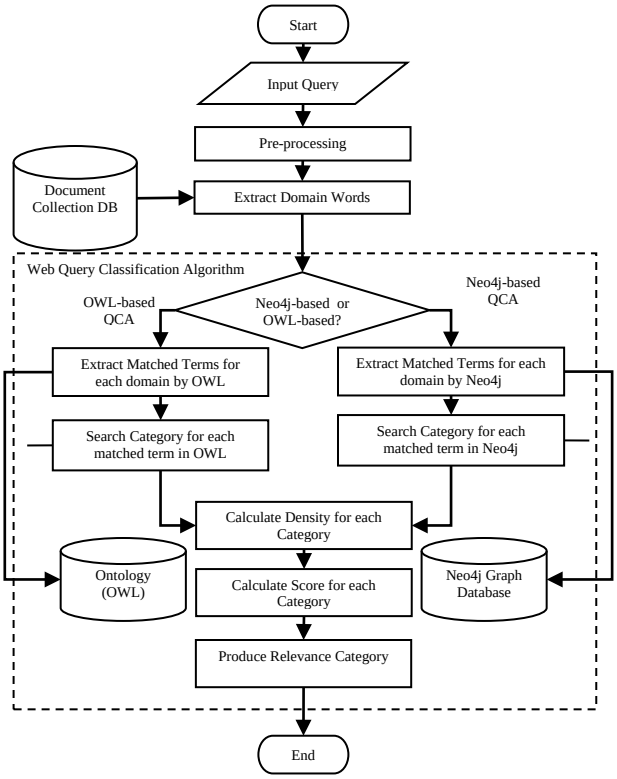

Figure 1: System Design

This system compares the performance between Neo4j-based and OWL-based query classification process. In this system, the user can choose the desired query classification process to classify the user query. In the OWL-based query classification, the user query is classified to intended category by using the domain ontology. But, the Neo4j-based classification process classifies the user query by using the Neo4j Graph Database to get the intended category for the user query. Both query classification process uses the same Web

Query Classification Algorithm (WQCA) based on different category data source: Neo4j graph database or ontology. System design is shown in figure 1.

IV. WEB QUERY CLASSIFICATION ALGORITHM

Web query classification is significant to search engines for the purpose of efficient retrieval of appropriate results in response to user queries. User queries are short in nature, contain noise and are ambiguous in terms of user intent. Web query classification is to classify a user query Q_i into a list of n categories $c_{i1}, c_{i2}, \ldots, c_{in}$ [5].

Web query classification includes three step processes. The first process is domain term extraction that is a categorization or classification task in which terms are categorized into a set of predefined domains [6]. The second process is learning step where a classification model is constructed. The third process is classification step where the model is used to predict class label for given data. If a certain category in an intermediate taxonomy is given, web query classification is directly mapped to a target category if and only if the following condition is satisfied: one or more terms in each node along the path in the target category appear along the path corresponding to matched intermediate category [2].

In the web query classification algorithm, the input is the domain terms of user query and the output is the relevance category that has the highest score. The web query classification algorithm (WQCA) is shown in figure 2.

ALGORITHM : Web Query Classification Algorithm
INPUT : Domain Terms
OUTPUT : User intended Categories
BEGIN

Step 1: Extract matched term for each domain from Neo4j or OWL.
Step 2: Search category for each matched term from Neo4j or OWL.
Step 3: Perform density calculation for each category [Density (C)]:

$$Density(C) = \left[(-P_{(i,j)} Log_2 P_{(i,j)}) \times w_i\right] + \left[-\sum_{j=1}^{m} (P_{(i,j)} Log_2 P_{(i,j)}) \times w_i\right]$$

Step 4: Perform score calculation for each category [Score(C)]:

$$Score(C) = \sum_{c=1}^{n} Category[Density(C)]$$

Step 5: Produce Relevance Category that has highest score.
END

Figure 2: Web Query Classification Algorithm (WQCA)

There are five steps in this algorithm. In the first step, this system extracts matched terms for each domain term using Neo4j graph database or ontology by specific query language. In the second step, related categories for each matched term are extracted from the Neo4j graph or ontology. In the third and fourth step, this system performs the density calculation for each category and performs the score calculation for each category. And finally, this system produces the Relevance Category that has the highest score.

V. NEO4J GRAPH DATABASE

Neo4j is a high performance NoSQL graph database which provides object oriented, flexible network structure. It is based on a Property graph data model which comprises of nodes and relationship along with their properties. It is reliable, ACID compliant, highly available and scalable. It offers REST interface and Java API quiet convenient to use. It can also be embedded into jar files. It uses CYPHER as its query language. Some of the fortune 500 companies that use Neo4j are Adobe, Accenture, Cisco, Lufthansa, Telenor and Mozilla [7].

A Neo4j database uses graph structures with nodes, edges and properties to represent and store information as shown in slide. Neo4j can be used in website link structures, hierarchical structures of categories and social networking. To classify the user query, concept terms in the area of computer science and technology are predefined in 22 sub categories using Neo4j database. These categories are grouped in four: Software, Hardware, Application and Information Science. The root category is Computer Science and Technology as our case study. The node structure of category graph is shown in figure 3. Each node in this graph is “Category” and the relationship between them is “INCLUDE”.

CYPHER query language is a latest query language that has been recently added to the Neo4j. Examples of CYPHER query to retrieve all matched terms listed by given domain words is shown in following figure 4.

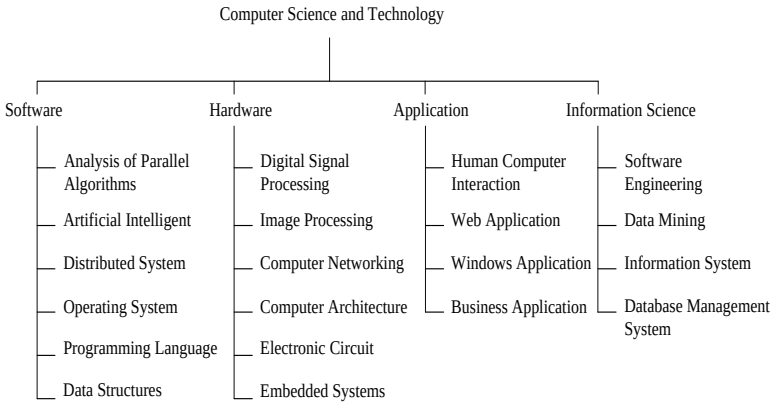

Figure 3: Category Structure in Neo4j

```
WITH ["artificial","intelligence","artificial intelligence"] AS domainTerms
UNWIND domainTerms AS domain
MATCH (c:Category)
WHERE LOWER(c.name) =~ ("(?i).*(?:[\\W-[+]]|^)" + domain + ".*")
RETURN domain as DomainTerm, collect(c.name) AS MatchTerms
```

Figure 4: Extraction of Matched Terms from Neo4j by CYPHER query

VI. ONTOLOGY CONSTRUCTION

Ontology renders shared vocabulary and taxonomy which models a domain with the definition of objects and/or concepts and their properties and relations [3]. In this system, ontology is constructed for query classification process using Protégé v 3.5. In construction of ontology model the target categories of domain area are created as classes. All relationship between classes are type of rdfs:subClassOf. There are 22 subclasses are created as the categories as same as the node structure of Neo4j graph database. These categories are grouped in four classes: Application, Hardware, Software and Information Science. Computer Science and Technology is the root class of ontology. The overview class structure of OWL is shown in figure 5.

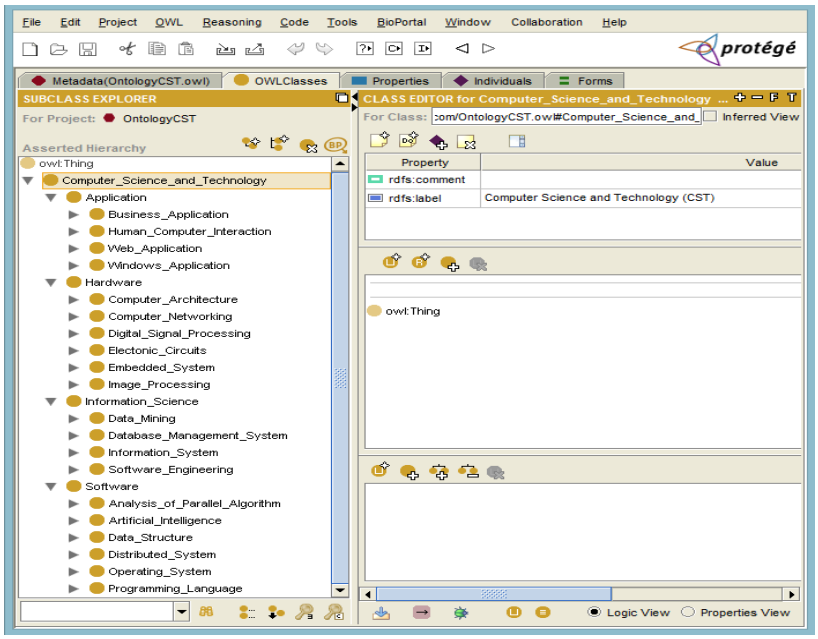

Figure 5: Class Structure in OWL

These categories consist of several subcategories or subclasses. For example, Web Application has subcategories namely Web Design, Website, Web Browsers, JavaScript (JS), Cascading Style Sheet (CSS), Hyper Text Markup Language (HTML) etc. Ontology is applied not only in the process of query classification to get the concepts of each domain word but also to match target category. Terms from ontology are extracted by SPARQL query language using dotNetRDF library as shown in figure 6.

```
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
SELECT ?className ?label
WHERE
{
    ?className rdfs:label ?label
    FILTER REGEX( ?label,'\\bartificial|\\bintelligence,'i')
}
```

Figure 6: Extraction of Matched Terms from OWL by SPARQL query

Although SPARQL supports the REGEX expression with multiple parameters, the results of query cannot be grouped by specific parameter, in this case the matched terms for artificial and intelligence domain words are returned by query as a common result. Therefore query must be executed one time for each domain word to know which terms are related to which domain.

VII. IMPLEMENTATION OF PROPOSED SYSTEM

Proposed system is implemented based on Service-Oriented Architecture (SOA) by using the XML based web service technology and ASP.NET. The logical architecture of proposed system is shown in figure 7.

Architecture of proposed system consists of one ontology dataset, two databases and three programming components. The functional module of proposed system is separately implemented as two web services by using C# programming language, because they relate to different database, in this case MySQL relational database and Neo4j graph database. Preprocess web service consists of functions for text preprocessing, extraction of domain terms and retrieving of information. Functions for QCA are implemented as a query classification web service. The functions of WQCAService are listed as shown in following figure 8. The functions for getting matched terms, getting category and calculating density value are the main functions of QCA web service. The user interface is designed and implemented as a web application in ASP.NET platform for testing these functions.

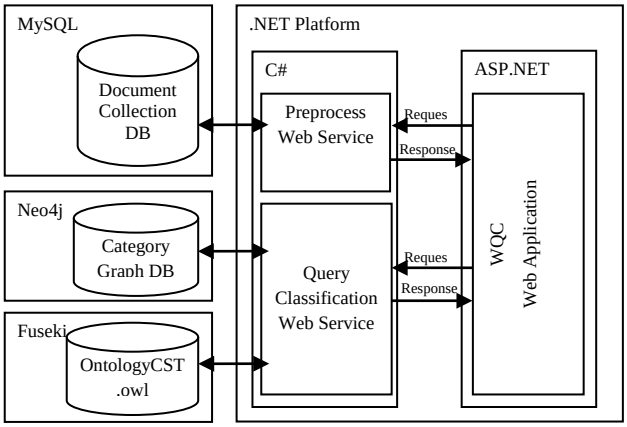

Figure 7: Proposed System Architecture

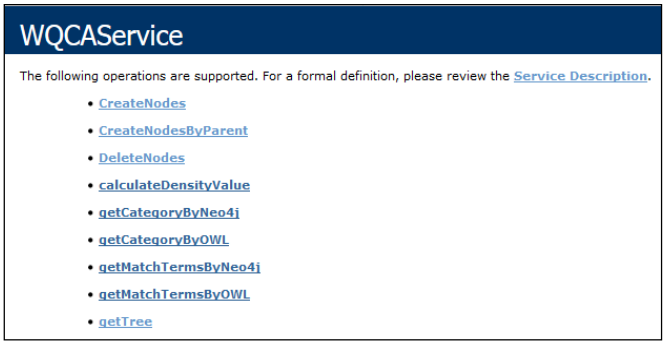

Figure 8: Function List of WQCA Web Service

VIII. EVALUATION OF CLASSIFICATION PERFORMANCE

To evaluate the performance of query classification process based on different category data source, ReadyAPI v2.6.0 tool is used. ReadyAPI [8] contains several powerful instruments for complex and overall testing of API and Web Services such as LoadUI which supports to simulate a massive load on web server to see how it works under the given conditions. Various testing strategies can be used to test different aspects of server. With LoadUI, tests can be run in parallel, as well as distribute tests among several test computers. The load test results of query classification by using LoadUI are shown in figure 9.

Neo4j-based and OWL-based QCA Load tests have been performed by invoking the calculateDensityValue function of WQCAService with 20 simultaneous virtual users (VUs) within 5 minutes. As a result, we can see that the average response time for Neo4j-based QCA is lower than the OWL-based QCA and the variation of response times is not smooth in OWL-based QCA. The wave forms of response times in OWL-based and Neo4j-based QCA load tests are shown in figure 10 and 11 respectively.

LoadTest	Project	Min (ms)	Max (ms)	Avg (ms)	Number of VUs	Duration (s)	Failures	Status	Target Executions	Date
QCA Load Test by OWL	WQCA-WebService	387	2717	1362	20	300	0	Pass	2493	2/9/19 5:59:15 PM
QCA Load Test by Neo4j	WQCA-WebService	85	351	139	20	300	0	Pass	5140	2/9/19 6:05:18 PM

Figure 9: Load Test Result for QCA Execution Time

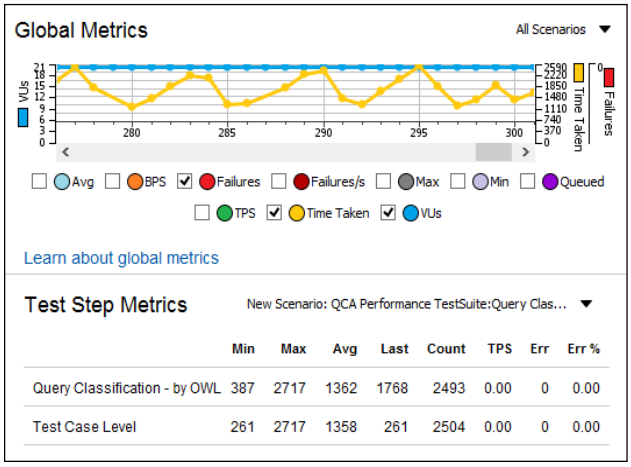

Figure 10: Load Test Result for OWL-based QCA

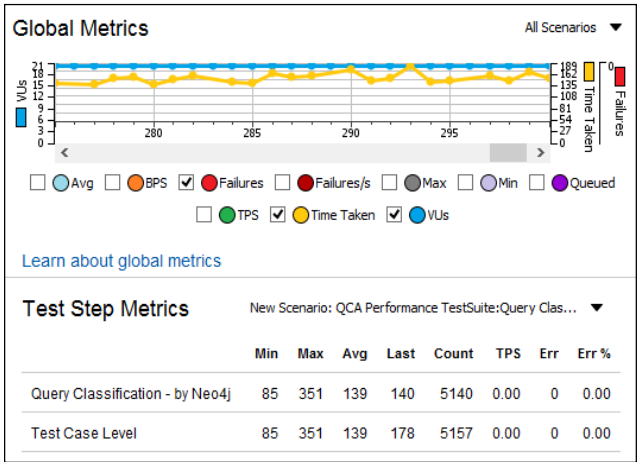

Figure 11: Load Test Result for Neo4j-based QCA

To evaluate the execution time of query classification process, total 220 queries (10 queries for each category) are tested from the ASP.NET web application. The execution times of Neo4j-based and OWL-based query classification are compared with bar chart group by intended categories of tested queries in figure 12.

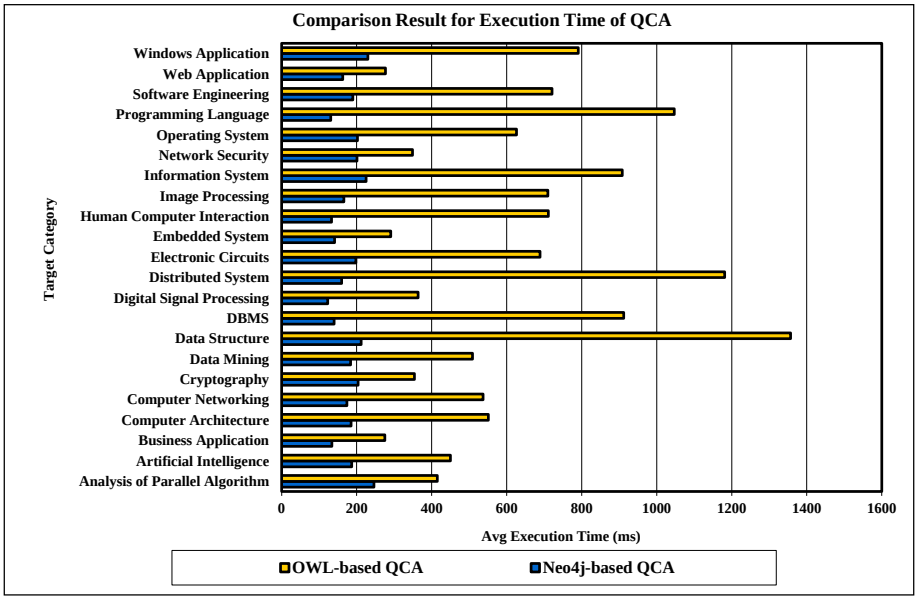

Figure 12: Comparison Result for QCA Execution Time

IX. CONCLUSION

According to the experimental results, Neo4j-based query classification process is more stable than the OWL-based classification process. The load test results performed by ReadyAPI shows that Neo4j database server can provide the stabilization of query classification process. Moreover, Cypher query language used in Neo4j can provide the WQCA to perform the querying process in time-effective manner. The performance comparison between Neo4j-based and OWL-based QCA shows the effectiveness of using graph database in the query classification process. So, in this paper Neo4j graph database is proposed for query classification process.

REFERENCES

[1] S. M. Fathalla and Y. F.14 Hassan, "A Hybrid Method for User Query Reformation and Classification", IEEE, pp. 132-138, 2012.

[2] A. Katariya, "Ontology-Based Web Query Classification", International Journal of Engineering Research and General Science, pp. 806-813, Volume 3, Issue 3, May-June, 2015.

[3] M. M. Thannaing and A. N. Hlaing, "Improving Information Retrieval Based on Query Classification Algorithm", Machine Learning and Applications: An International Journal (MLAIJ), pp. 21-31, vol.1, no.1, September, 2014.

[4] W. Yue, Z.Chen and X. Lu, "Using Query Expansion and Classification for Information Retrieval", Proceedings of the First International Conference on Semantics, Knowledge, and Grid (SKG), IEEE, 2006.

[5] D. Shen, J. Sun, Q. Yang, Z. Chen, "Building bridges for Web query classification". Proceedings of the 29th annual international ACMSIGIR conference on research and development in information retrieval, Seattle, WA, USA (2006), pp. 131-138.

[6] S. M. Kim and T. Baldwin, "An Unsupervised Approach to Domain-Specific Term Extraction", University of Melbourne, Australia, 2011.

[7] A. Nayak, A. Poriya and D. Poojary, "Type of NoSQL Databases and its Comparison with Relational Databases", International Journal of Applied Information Systems (IJAIS), Foundation of Computer Science FCS, USA, Volume 5, March 2013.

[8] ReadyAPI Documentation, URL: https://support. smartbear.com/readyapi/docs/index.html

[9] The Neo4j Graph Platform, URL: https://neo4j.com/ product/

[10] Protégé, A free open-source ontology editor and framework for building intelligent systems, URL: https://protege.stanford.edu/

[11] doNetRDF Documentation, URL: https://github.com/ dotnetrdf/dotnetrdf/wiki

AUTHORS

First Author – Naw Thiri Wai Khin, PhD Candidate, University of Technology (YCC), ntrwk87@gmail.com
Second Author – Nyo Nyo Yee, PhD, University of Technology (YCC), nny1ster@gmail.com.
Corresponding Author – Aung Aung Hein, PhD, Defence Services Academy, moezat198@gmail.com