USING GRAPH DATABASE FOR EFFECTIVE VISUALIZATION IN LEARNING BASIC BUDDHIST VOCABULARY*

Ohnmar Win[1]

Abstract

Graph databases have a long academic tradition. At the heart of any graph database lies an efficient representation of entities and relationships between them. All graph database models have, as their formal foundation, variations on the basic mathematical definition of a graph, for example, directed or undirected graphs, labeled or unlabeled edges and nodes, hypergraphs, and hypernodes. More recently, semantic relations have become a major theme of interest of Computational Linguistics. Semantic relations among words have captured the interest of various brands of philosophers, cognitive psychologists, linguists, early childhood and second language educators, computer scientists, literary theorists, cognitive neuroscientists, psychoanalysts - investigators from just about any field whose interests involve words, meaning or the mind. The Pāḷi Canon is the complete scripture collection of the Theravāda school. Buddhist monks and scholars studied the Pāḷi language mainly to gain access to the Buddhist Canon and many religious works were written using the Pāḷi language. The objective of this study is to support for new Buddhist vocabulary learner to alternative view by using graph database, Neo4j.

Keywords: graph database, semantic relations, Neo4j, Pāḷi, Buddhist Vocabulary

Introduction

Semantics is the study of the relationship between the linguistics forms and entities in the world, that is, how words literally connect to things (meaning). It is a major branch of linguistics devoted to the study of meaning in language. In many research fields such as linguistics, cognitive science, psychology, artificial intelligence, biomedicine and information retrieval, computing semantic similarity/relatedness between concepts or words is considered as an important issue. More recently, semantic relations have become a major theme of interest of Computational Linguistics, as they present a convenient and natural way to organize huge amounts of lexical data in ontologies, Word Nets and other machine-readable lexical resources.

[1]. Lecturer, Department of Computer Studies, Yadanabon University

*. Best Paper Award Winning Paper in Computer Studies (2017)

Semantic relations may reflect relations in language including relations between objects and their symbols. Semantic relations can refer to relations *between concepts in the mind* (called conceptual relations), or relations *between words* (lexical relations) or text segments. Different domains develop continuously new kinds of semantic relations. Some kind of semantic relationships that exist in words of natural language have always been a challenge in the Fields of Natural Language Processing (NLP) and Information Retrieval (IR). When a word level semantic relation requires exploration, there are many potential types of relations that can be considered: synonym, antonym, homonym, polysemy, hyponym, meronym, etc. Semantic relations are fixed manually in various linguistic resources, such as thesauri, ontologies, and synonym dictionaries.

The relationships between words can be summarized briefly as follows:

Synonym : The notion that more than one linguistic form can be said to have the same conceptual or propositional meaning.

e.g., Nibbāna and Mokkha

Antonym : The notion of semantic oppositeness.

e.g., Amitta and mitta

Hyponym : Refers to a relationship existing between specific and general lexical items: the meaning of the specific item is included in, and by, the meaning of the more general item.

e.g., sunakha is a hyponym of tiricchāna.

Meronym : Refers to a part-whole relation.

e.g., Rukkha and Phala

The limitations of traditional databases, in particular the relational model, to cover the requirements of current application domains, has lead the development of new technologies called Graph Databases, which are oriented to store graph-like data. Recently the area is gaining attention because in trendy projects where a database is needed (for example chemistry, biology, Web Mining and semantic Web), the importance of the information relies on the relations more or equal than on the entities. Moreover, the continued emergence and increase of massive and complex graph-like data makes a graph database a crucial requirement. This renascence is showed by the availability of several graph databases systems.

With the needs to manage large and sparse datasets, with many kinds of relationships between them, new kinds of Database have been developed to supply it with a performance and capability better than the traditional databases technologies and queries languages. Many of these new kinds of Databases using graph structures like the main engine to allow to user to insert, update, query, delete and apply analysis techniques based in graphs in the networks of graphs.

Graph Database is a database system where the associations between objects or entities are similarly as important as the objects themselves. In a graph database, data are represented by nodes, edges and properties. Nodes are represented as objects and edges manifest the relationship between nodes. There are several implementations of graphical database. Both nodes and edges can have properties that illustrate their particular characteristics.

Graph databases are especially suited for highly connected data. Today, general-purpose graph databases are a reality, allowing mainstream users to experience the benefits of connected data without having to invest in building their own graph infrastructure. Today, there are many graph databases such as Allegro Graph, DEX/Sparkee, Hypergraph DB, Infinite Graph, Neo4J, Orient DB, Info Grid, Vertex DB, Flock DB, Graph DB etc.

An Overview of Neo4j Graph Database

Neo4j is the world's leading graph database. Neo4j is a high performance graph store with all the features expected of a mature and robust database, like a friendly query language and ACID transactions.

Neo4j is a graph database, which means that it does not use tables and rows to represent data logically; instead, it uses nodes and relationships. Both nodes and relationships can have a number of properties. While relationships must have one direction and one type, nodes can have a number of labels. The programmer works with a flexible network structure of nodes and relationships rather than static tables. For many applications, Neo4j offers orders of magnitude performance benefits compared to relational databases.

Neo4j is based on a network oriented model where relations are first class objects.

The most popular variant of graph model is the property graph. Property graphs are attributed, labeled, directed multi-graphs. The property graph balances simplicity and expressiveness. Property graphs sacrifice some graph purity for pragmatism by grouping properties into nodes, thereby making them easier to work with. The main abstractions in a property graph are nodes, relationships and properties. Neo4j uses Cypher Query Languages for property graphs. A Property graph has the following characteristics:

- It contains nodes and relationships
- Nodes contain properties
- Relationships are named, directed and always have a start and end node
- Relationships can also contain properties

Most people find the property graph model intuitive and easy to understand.

Neo4j has many features. The main feature is that neo4j not depend heavily on index because it supplies a natural adjacency by the graph. It is easy to write queries about relationships with many types of deep.

The Cypher Query Language in Neo4j

Cypher is a declarative graph query language that allows for expressive and efficient querying and updating of the graph store. Cypher is designed to be a humane query language, suitable for both developers and operations professionals who want to make ad hoc queries on the database. Cypher is a database expressive and compact query language. It is primarily used in Neo4j, although it can also be used to programmatically describe graphs in a precise manner due to its close affinity to graphs. It is easy to learn and understand since it follows the way humans intuitively describe graphs using diagrams. Cypher is a relatively simple but still very powerful language. Very complicated database queries can easily be expressed through Cypher. Like most query languages, Cypher is composed of clauses. A reasonably simple query is made up of START, MATCH and RETURN clauses.

The some clauses of Cypher are:

- START – specifies one or more starting points – nodes or relationships – in a graph, which are obtained via index lookup (starting points are rarely accessed via IDs).
- MATCH – it makes use of the relationships
- RETURN – returns nodes and relationships that match the criteria
- WHERE – acts as a filter pattern for matching results
- CREATE or CREATE UNIQUE – creates (unique) nodes and relationships
- DELETE – removes nodes, relationships or properties
- SET – sets property values
- UNION – merges results from two or more queries
- WITH – chains subsequent query results and pipelines results

The Operations of Neo4j Graph Database

Neo4j has CRUD operations. They are Create, Read, Update, and Delete.

CREATE Operation

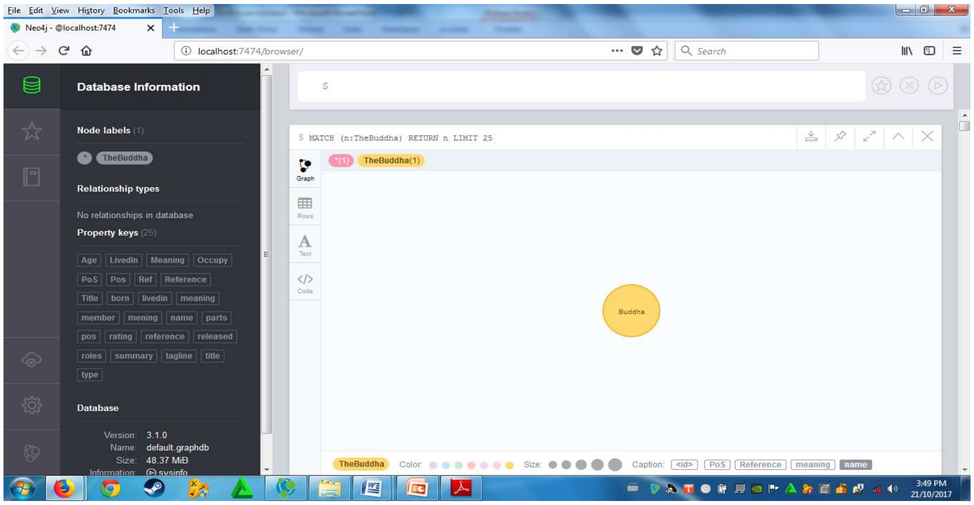
Create operation is used to create nodes and relationships.

e.g., Creating a node

```
create(a:TheBuddha{name: 'Buddha', meaning: 'Supreme Man',
PoS: 'Noun', Reference: 'Pāḷi Canon'})
```

Create clauses can create nodes and relationships.***( ) parenthesis*** is to indicate a node. In ***a:TheBuddha***, '***a***' is variable and***The Buddha***is label for the new node. ***{}*** bracket can be used to add properties to the node.

Result in console:



Database Information

Node labels (1)

* TheBuddha

Relationship types

No relationships in database

Property keys (25)

Age LivedIn Meaning Occupy PoS Pos Ref Reference Title born livedin meaning member mening name parts pos rating reference released roles summary tagline title type

Database

Version: 3.1.0
Name: default.graphdb
Size: 48.37 MiB
Information: sysinfo

```
$ MATCH (n:TheBuddha) RETURN n LIMIT 25
```

*(1) TheBuddha(1)

TheBuddha Color: Size: Caption: <id> PoS Reference meaning name

-Creating multiple nodes

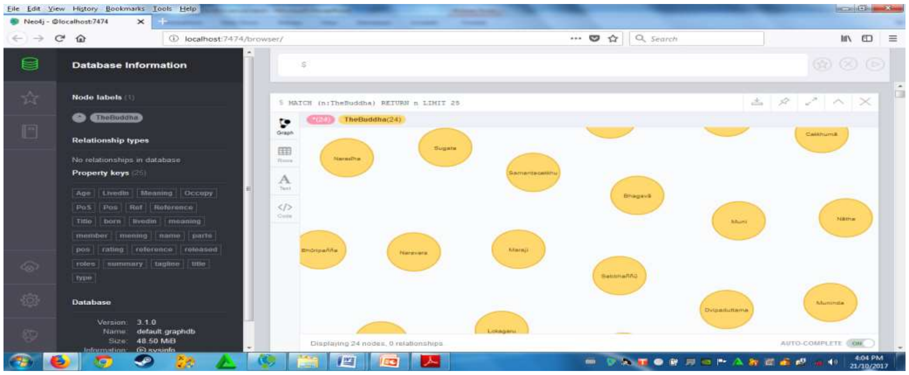
```
create (a:The Enlightened One {name: "Buddha", meaning: "Supreme
Man", PoS:"Noun", Reference: "PāḷiCanon"}), (b:The Enlightened
One{name:" Dasabala", meaning:"Ten powers of Buddha", PoS: "Noun",
Reference: "Pāḷi Canon"}), (c:The EnlightenedOne {name:"Satthā",
meaning:"A Supreme teacher", PoS:"Noun", Reference:
"PājikakaṇdaPāḷi"}), etc.,
```

In the following table, there are some of the epithets of the Buddha and its properties.

Table 1. The epithets of the Buddha and its properties.

Name	Meaning	PoS	Reference
Dasabala	Ten powers of Buddha	Noun	Pāḷi Canon
Satthā	A Supreme teacher	Noun	PārājikakaṇdaPāḷi
Sabbaññū	All-Knowing	Noun	Theragāthāand MajjhimapaṇṇāsaPāḷi
Dvipaduttama	The best of Men	Noun	Buddhavaṁsa
Munindа	The chief of monks	Noun	ApadānaPāḷi
Bhagavā	The Blessed One	Noun	PārājikakaṇdaAṭṭhakathā
Nātha	Protector	Noun	three piṭakas
Cakkhumā	Having eyes	Noun	Pāḷi Canon
Muni	Monk	Noun	Pāḷi Canon
Lokanātha	TheRefuge of the human beings	Noun	Suttapiṭaka
Anadhivara	There is no one who is superior to oneself	Noun	Buddhavṁsa
Mahesi	The Great Sage	Noun	Pāḷi Canon
Vināyaka	One who admonishes the living beings	Noun	Sutta and VinayaPiṭakas
Samantacakkhu	All-Seeing	Noun	Pāḷi Canon
Sugata	Meritorious act	Noun	Five Nikāyas
Bhūripañña	abundant knowledge as the earth	Noun	Majjhimapaṇṇāsa
Maraji	Supreme Man	Noun	Pāḷi Canon
Narasīha	The Noble Man	Noun	TheragāthāPāḷi
Naravara	The Noble Man	Noun	TheragāthāPāḷi
Dhammarājā	The King of righteousness	Noun	Theragāthā
Mahāmuni	The Great Sage	Noun	Suttaand VinayaPiṭakas
Devadeva	The God of gods	Noun	Theragāthā and ApadānaPāḷi
Lokagaru	The One who is a teacher deserving the special veneration of human beings	Noun	SakulātherīApadāna

Result with graph view for creation of multiple nodes;



```
$ MATCH (n:TheBuddha) RETURN n LIMIT 25
```

Displaying 24 nodes, 0 relationships

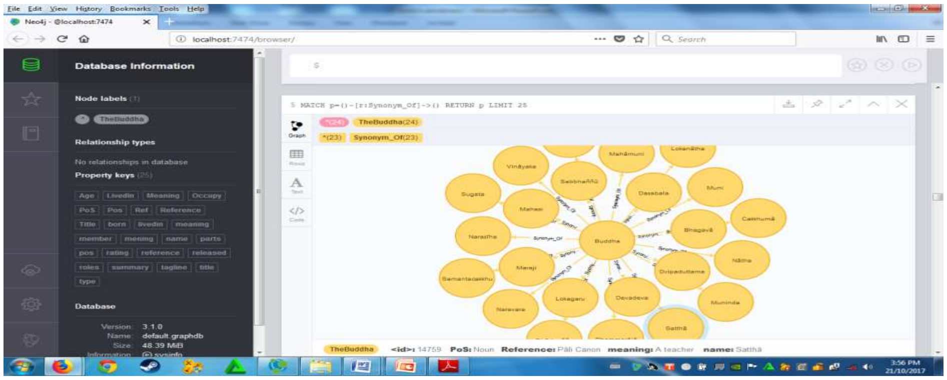
-Creation of relationships

These are coding for relationships which are created after the creation of nodes' codes under the following:

```
create
 (a)-[:synonym_of]->(b),(a)-[:synonym_of]->(c),(a)-[:synonym_of]->(d),
 (a)-[:synonym_of]->(e),(a)-[:synonym_of]->(f),(a)-[:synonym_of]->(g)
```

In Relationships, Cypher uses a pair of dashes (--) to represent undirected relationship. Directed relationships have an arrow head at one end (eg.,<-- , -->). Bracketed expression may include types of relationships, properties, and attributes.

Result with graph view for creation of relationships between nodes;



```
$ MATCH p=()-[r:Synonym_Of]->() RETURN p LIMIT 25
```

TheBuddha <id>: 14759 PoS: Noun Reference: Pāli Canon meaning: A teacher name: Satthā

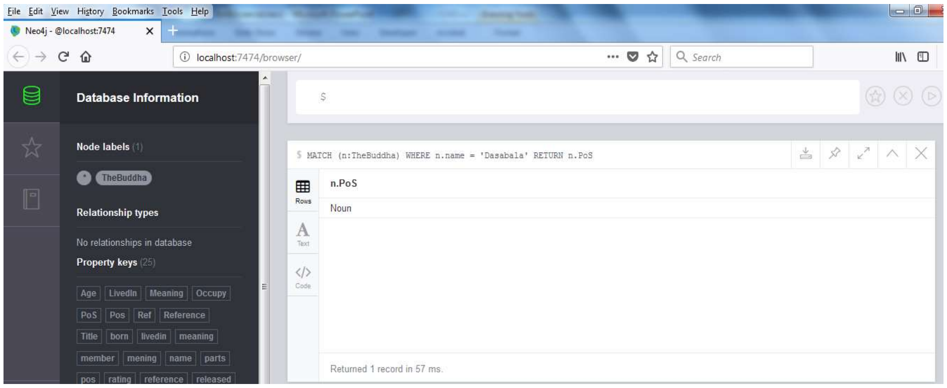
Read Operation

Reading a node named ‘Dasabala’ and return the PoS.

e.g.,

```
MATCH (n:TheEnlightenedOne) WHERE n.name =‘Dasabala' RETURN
n.Pos
```

Result in console:



```
$ MATCH (n:TheBuddha) WHERE n.name = 'Dasabala' RETURN n.PoS
```

```
n.PoS
Noun
```

Returned 1 record in 57 ms.

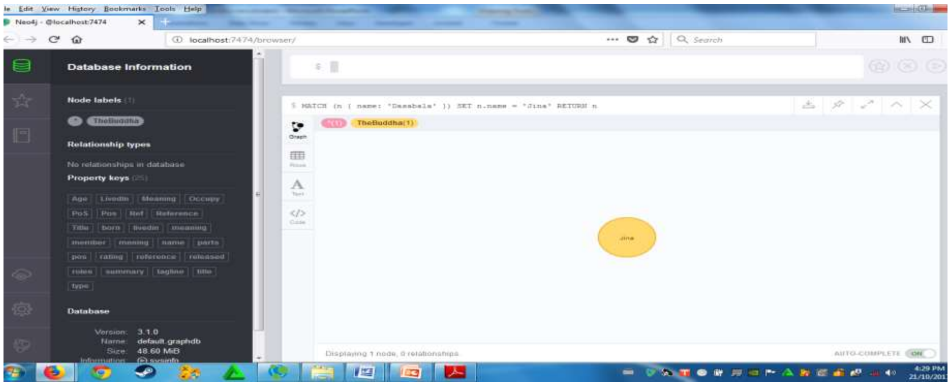
Update Operation

-Updating a node named ‘Dasabala’ with 'Jina':

e.g.,

```
MATCH (n {name: ‘Dasabala'}) SET n.title = ‘Jina' RETURN n
```

Result in console:



```
$ MATCH (n { name: 'Dasabala' }) SET n.name = 'Jina' RETURN n
```

Displaying 1 node, 0 relationships

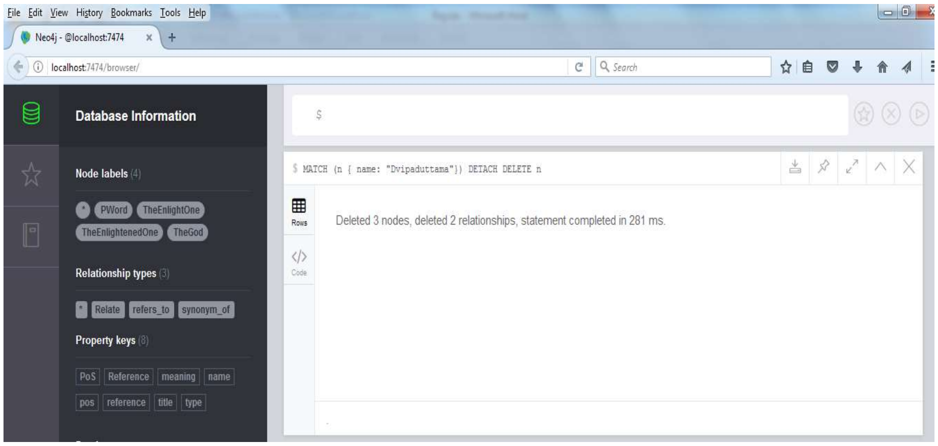
Delete Operation

- Deletinga node named with "Dvipaduttama"and all its relationships

e.g.,

```
MATCH (n { name: ‘Dvipaduttama' }) DETACH DELETE n
```

Result in console:

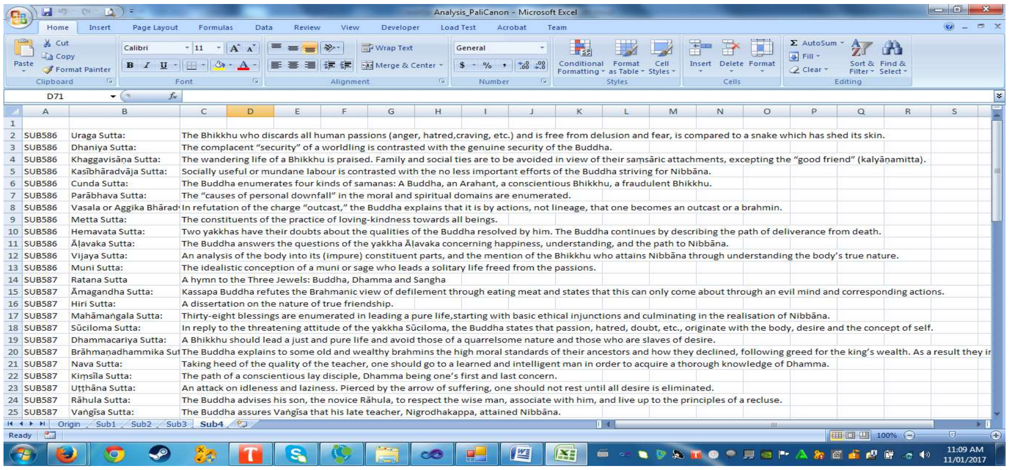


```
$ MATCH (n { name: "Dvipaduttama"}) DETACH DELETE n
```

```
Deleted 3 nodes, deleted 2 relationships, statement completed in 281 ms.
```

Advance Feature of Neo4j

Cypher query language can use LOAD CSV to import data from CSV (Comma Separated Value) fileto get the data into query. The data can be loaded from standard CSV with LOAD CSV function. Firstly, the Buddhist vocabulary from the Pāli Canon was created with CSV format. From this the raw CSV data turn into a graph database which shows the nodes and the relationships between them but keeps the other details such as the meanings, PoS, and references as properties within the database. The following figure contained some words of Tipitaka and stored with CSV file format.



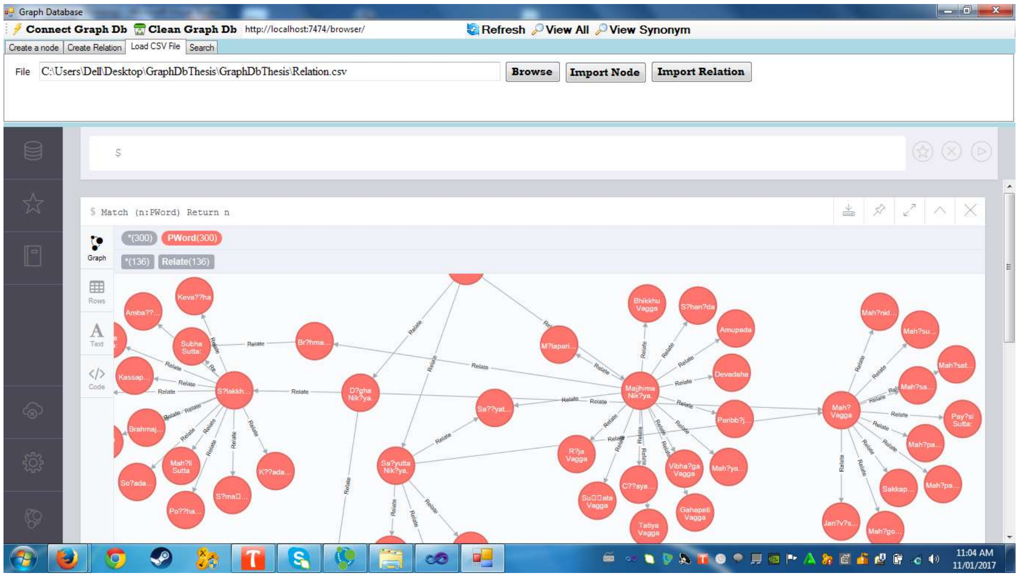
	A	B	C
1			
2	SUB586	Uraga Sutta:	The Bhikkhu who discards all human passions (anger, hatred,craving, etc.) and is free from delusion and fear, is compared to a snake which has shed its skin.
3	SUB586	Dhaniya Sutta:	The complacent "security" of a worldling is contrasted with the genuine security of the Buddha.
4	SUB586	Khaggavisāṇa Sutta:	The wandering life of a Bhikkhu is praised. Family and social ties are to be avoided in view of their saṃsāric attachments, excepting the "good friend" (kalyāṇamitta).
5	SUB586	Kasībhāradvāja Sutta:	Socially useful or mundane labour is contrasted with the no less important efforts of the Buddha striving for Nibbāna.
6	SUB586	Cunda Sutta:	The Buddha enumerates four kinds of samanas: A Buddha, an Arahant, a conscientious Bhikkhu, a fraudulent Bhikkhu.
7	SUB586	Parābhava Sutta:	The "causes of personal downfall" in the moral and spiritual domains are enumerated.
8	SUB586	Vasala or Aggika Bhārad	In refutation of the charge "outcast," the Buddha explains that it is by actions, not lineage, that one becomes an outcast or a brahmin.
9	SUB586	Metta Sutta:	The constituents of the practice of loving-kindness towards all beings.
10	SUB586	Hemavata Sutta:	Two yakkhas have their doubts about the qualities of the Buddha resolved by him. The Buddha continues by describing the path of deliverance from death.
11	SUB586	Āḷavaka Sutta:	The Buddha answers the questions of the yakkha Āḷavaka concerning happiness, understanding, and the path to Nibbāna.
12	SUB586	Vijaya Sutta:	An analysis of the body into its (impure) constituent parts, and the mention of the Bhikkhu who attains Nibbāna through understanding the body's true nature.
13	SUB586	Muni Sutta:	The idealistic conception of a muni or sage who leads a solitary life freed from the passions.
14	SUB587	Ratana Sutta	A hymn to the Three Jewels: Buddha, Dhamma and Sangha
15	SUB587	Āmagandha Sutta:	Kassapa Buddha refutes the Brahmanic view of defilement through eating meat and states that this can only come about through an evil mind and corresponding actions.
16	SUB587	Hiri Sutta:	A dissertation on the nature of true friendship.
17	SUB587	Mahāmaṅgala Sutta:	Thirty-eight blessings are enumerated in leading a pure life,starting with basic ethical injunctions and culminating in the realisation of Nibbāna.
18	SUB587	Sūciloma Sutta:	In reply to the threatening attitude of the yakkha Sūciloma, the Buddha states that passion, hatred, doubt, etc., originate with the body, desire and the concept of self.
19	SUB587	Dhammacariya Sutta:	A Bhikkhu should lead a just and pure life and avoid those of a quarrelsome nature and those who are slaves of desire.
20	SUB587	Brāhmaṇadhammika Sut	The Buddha explains to some old and wealthy brahmins the high moral standards of their ancestors and how they declined, following greed for the king's wealth. As a result they ir
21	SUB587	Nava Sutta:	Taking heed of the quality of the teacher, one should go to a learned and intelligent man in order to acquire a thorough knowledge of Dhamma.
22	SUB587	Kiṃsīla Sutta:	The path of a conscientious lay disciple, Dhamma being one's first and last concern.
23	SUB587	Uṭṭhāna Sutta:	An attack on idleness and laziness. Pierced by the arrow of suffering, one should not rest until all desire is eliminated.
24	SUB587	Rāhula Sutta:	The Buddha advises his son, the novice Rāhula, to respect the wise man, associate with him, and live up to the principles of a recluse.
25	SUB587	Vaṅgīsa Sutta:	The Buddha assures Vaṅgīsa that his late teacher, Nigrodhakappa, attained Nibbāna.

Loading the data

The LOAD CSV statement can be used to load the data in from a CSV file as the following:

```
LOAD CSV WITH HEADERS FROM
C:\Users\Dell\Desktop\GraphDbThesis\GraphDbThesis\Relation.csv As
line
```

Result in console:



File C:\Users\Dell\Desktop\GraphDbThesis\GraphDbThesis\Relation.csv

```
$ Match (n:PWord) Return n
```

Classification of Pāḷi Canon

The Pāḷi Canon is the complete scripture collection of the Theravāda school. As such, it is the only set of scriptures preserved in the language of its composition. It is called the *Tipiṭaka* or "Three Baskets" because it includes the *Vinaya Piṭaka* or "Basket of Discipline," the *Sutta Piṭaka* or "Basket of Discourses," and the *Abhidhamma Piṭaka* or "Basket of Higher Teachings".

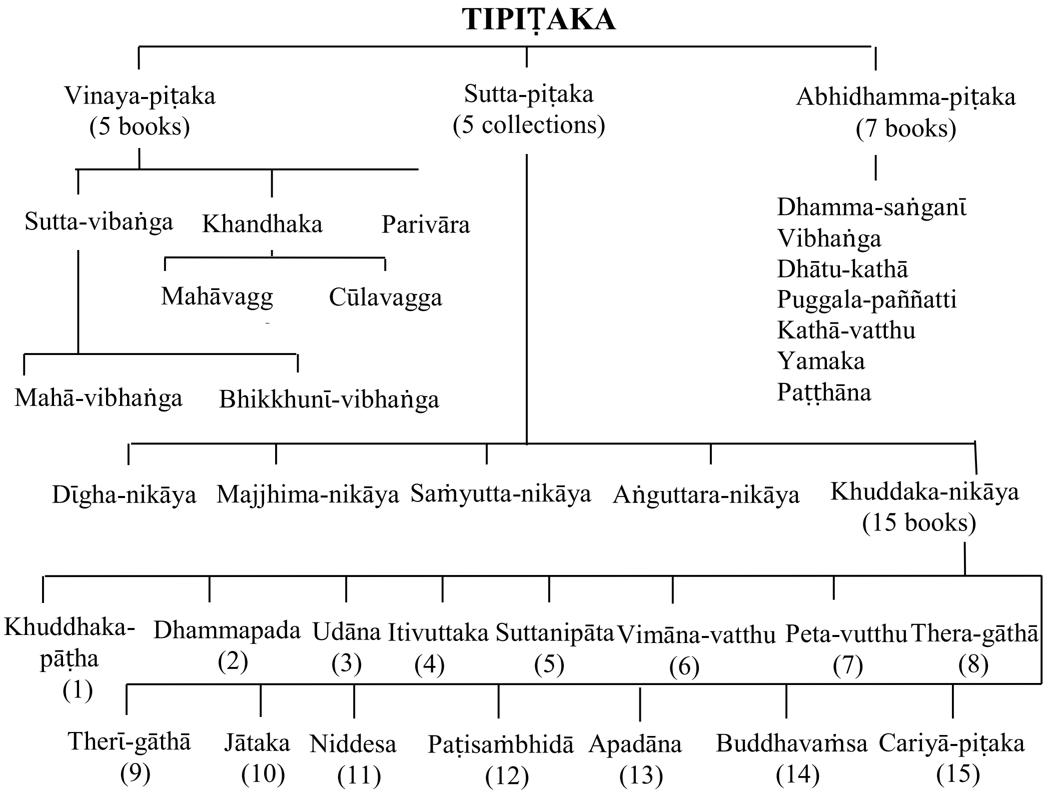


Figure 1. Classification of PāḷiCanon

Implementation for Semantic Relationship of Buddhist Vocabulary with Neo4j

Firstly, words are extracted from the Pāḷi Canon and various Buddhist literatures written by Pāḷi. The relationship implementation was focused on Pāḷi word definitions and semantic relationship in the dictionaries, where the meaning of a word is explained by other words in its gloss.The Pāḷi words and their activities are built in the spread sheet and stored with CSV file format.The graph database extracts the words that match a user-query and sets relationships between words by using Load CSV. The user can search the desired words via graphical user interface which provides to find the words with semantic meaning. The system will display the result all of the words and its relationships with graph view. The process flow of the words and the semantic relationship of Pāḷi words implemented by Neo4j graph databasewas provided in figure 2.

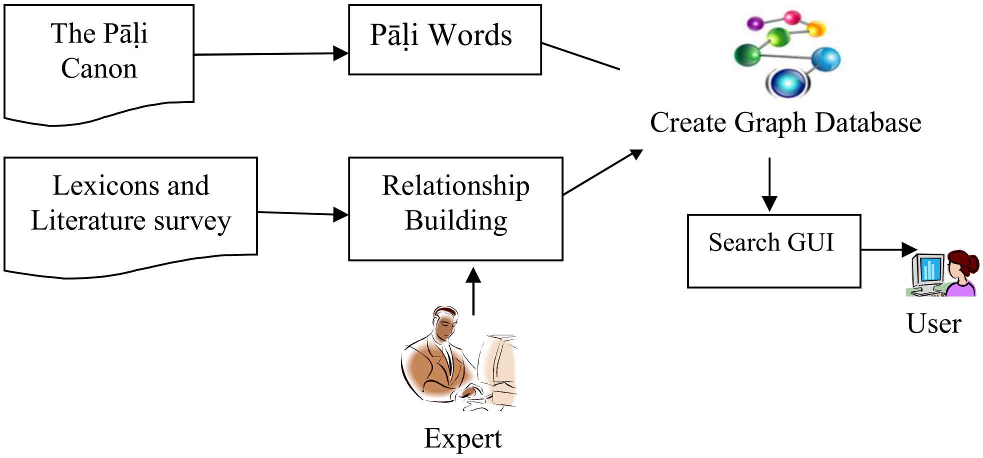
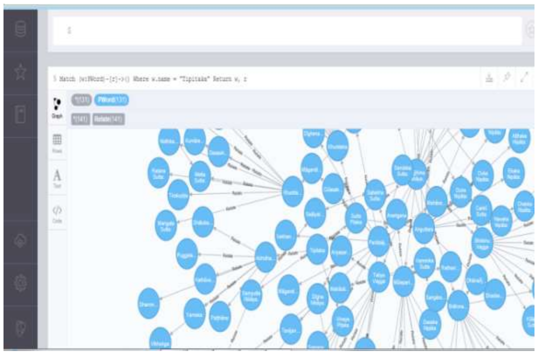
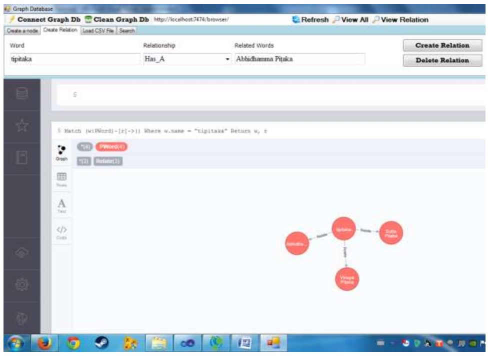


Figure 2. System architecture for Semantic Relationship between Pāḷi words



```
$ Match (w:PWord)-[r]->() Where w.name = "Tipitaka" Return w, r
```



http://localhost:7474/browser/

Word	Relationship	Related Words
tipitaka	Has_A	Abhidhamma Piṭaka

```
$ Match (w:PWord)-[r]->() Where w.name = "tipitaka" Return w, r
```

Figure (3) Relationship for Tipiṭaka

Figure (4) User Interface for Tipitaka Nodes and its Relationships

Conclusion and Future Work

In this work, a storage technique for Pāḷi Dictionary especially hyponym and meronym relationships was implemented based on graph database. Graph databases are a major pillar of the No SQL movement with lots of emerging products, such as Neo4j. Main contribute of this work is to support with Pāḷi words learner with understandable format. Yet, this is only the beginning. The automatic extraction of semantic relations of Pāḷi words form various resources will be future work. The evaluation and comparison with other graph databases and relational database were the future work. And also plan on migrating several researches done on relationship mining to work on graph database back-ends.

Acknowledgement

The author is deeply indebted to Supervisor Dr. Soe Mya Mya Aye (Professor, Head of Department, Department of Computer Studies, and University of Yangon) and Co Supervisor Dr. Khin Sandar Myint (Lecturer, Department of Computer Studies, University of Yangon) and Daw Wai Wai Myint (Lecturer and Head, Department of Computer Studies, Yadanabon University) for her encouragement. Thanks are due to Dr Thet Thet Hlaing (Lecturer, Department of Computer Studies, University of Yangon) for her helpful suggestions and valuable discussions.

References

1. Christopher S. G. Khoo and Jin-Cheon Na (2006).*Semantic Relations in Information Science.* Division of Information Studies, School of Communication &Information Nanyang Technological University.
2. http://www.neo4j.org/
3. Ian Robinson, Jim Webber, and Emil Eifrem (2013). *Graph Databases*. O'REILLY Media, Inc.,978-1-449-35626-2.
4. Justin J. Miller. Georgia Southern University, *Graph Database Applications and Concepts with Neo4j.*Proceedings of the Southern Association for Information Systems Conference, Atlanta, GA, USA March 23rd-24th, 2013.
5. Khin Khin Oo (2014). *The Epithets of Buddha*, Department of Oriental Studies, University of Mandalay.
6. Moggallāna, Ashin. (1982). *Abhidhānappadīpikāpaṭha*. Yangon: Department of affairs Press.
7. Murphy M. Lynne (2003). *Semantic relations and the lexicon: antonymy, synonymy, and other paradigms,* University of Sussex.
8. Renzo Angles, *A Comparison of Current Graph Database Models*, Department of Computer Science, Engineering Faculty, Universidad de Talca Camino Los Niches, Km. 1, Curic´o, Chile.
9. Rhys Davids, T. W. and Stede, W. (1979).*The Pali Text Society's Pali-English Dictionary.* London: The Pāḷi Text Society.
10. U .H. Myint, Dictionary of Pāḷi-derived Words, First Edition, Universities Press, Yangon, Myanmar, 1986.
11. Ven. Pannyavaro, *buddhist studies*:, Buddha Dharma Education Association Inc., Buddhanet.net