International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878 (Online), Volume-8 Issue-4, November 2019

Concept Relation Knowledge Visualization with CR Logic using Neo4j

Praveena Rachel Kamala S, Justus S

Abstract*: *Knowledge Visualization using graphical structure is a field in which the researchers have a strong influence on comprehensibility. Based on the information provided by the users, the knowledge is created. By mapping the logic generated based on the knowledge, we have created a new graphical structure using Cypher Query Language that shows the connectivity between each concept and their relationship. These knowledge visualization techniques are mainly suitable in finding the outcome of a situation and also in discovering new knowledge from the existing or by adding new information. An automated code is generated to build the graph structure for visualization in Neo4j. The reduction in content size and graphical structure helps the user to easily gain knowledge about the system.

Keywords*: *CR Logic, Cypher Query Language (CQL), Knowledge Visualization, Neo4j.

I. INTRODUCTION

The expansion of digital communication technologies has made a huge impact on storage and sharing of information. Associated through large networks, many users communicate and transfer information as documents and store in massive volume. The accessibility of information has exploded over the decades and this will increase as the years go by. The innovations in technology is fast developing in terms of complexity, getting quickly, precise, and less costly every year.

In any case, with expanding probability comes expanding complexity; the support by computerized and other advanced technologies of this complex and dynamic data space has not really made increasingly effective in the communication of knowledge [1]. The present innovation gives a framework where information can be documented and transmitted; however efficiency of current system for accessing, arranging and exploring data are lacking. Crosswise over personal and expert societies, individuals are dedicating more time and energy to extract and process loads of information so as to get meaningful information. Current information technology systems regularly present spontaneous and contextualized details which make it hard to confirm and place information to utilize. An improvement in procedures for sorting out and showing data is required to reduce the disappointment frequently felt by even experienced information searchers. An incredible challenge today isn't really to create new information [2], but to create improved methods for investing and making sense of what we have in hand.

Knowledge is a blend of framed experience, values, relevant data, and expert insight that makes the system to assess and consolidate new encounters and information [3]. It starts and is applied in the minds of experts. In institutions, it becomes not only as reports or repositories as well as in hierarchical schedules, procedures, practices, and standards.

The procedures of learning and comprehension include moving along the range from data to information to knowledge. To comprehend the procedures associated with creating knowledge from information and information from data, the initial step is to show attributes of every one of the three components and the structure they produce together. While there is no widespread model of this structure as points of view and definitions change depending upon the sector like engineering, structure, etymology, or reasoning, it is in any case a valuable exercise to draw from every one of these areas for clear interpretations.

The knowledge space is developing and complicated; inserted knowledge (as information) is increasingly disseminated for efficient interpretation and transferable digital or analog (print, communicate media) records, for all time and pervasively accessible on the web. The huge digital information system and the simplicity of generation and conveyance of digital records is a massive asset, however the confused (non-)organization makes getting to the exact information complex. Knowledge that is available in some untraceable records is useless; it picks up esteem just in its application and simplicity of transfer.

The display of complex data spaces by content heavy interfaces is not exactly ideal; information visualization is the alternate solution. Visualization is the reduction and interpretation of complex semantic and topologic interrelationships of the information space to an intelligent visual map [4]. These maps are digitally created from cartography and data graphics, and concentrate visual methodologies from visual art and graphic design. The advantage of such visualization is the capability to process data, show 2D or 3D images, update progressively, and react interactively to a user's activities, these maps can discover notable details and give multi-dimensional input to help orientation and navigate through information.

Manuscript published on November 30, 2019.
* Correspondence Author
Praveena Rachel Kamala S*, Research Scholar, School of Computing Science and Engineering, VIT University, Chennai, India.
Justus S, Associate Professor, School of Computing Science and Engineering, VIT University, Chennai, India.

Visualizations are information maps, and it's a kind of representation and reproduction of the semantic space.

Knowledge Visualization is a field of study that examines the intensity of visual structure to place knowledge. It targets supporting cognitive procedures in creating, representing, organizing retrieving sharing and utilizing knowledge [5].

Visualization has demonstrated to be a powerful strategy for supporting users in handling with complex knowledge and information rich situations. There are software for visualizing the information. These programming software changes continually. There is always upgradation in the version of software constantly [6]. Regardless of whether the earlier version works accurately, changes to the software may lead to potential error. The designer interprets the planned semantic changes of the program's behavior into syntactic changes of the program's source code and starts executing the changes.

In this paper the section 2 with basic background on knowledge visualization, section 3 describes the architecture and algorithm for our proposed Concept Relation Knowledge Visualization (CRKV) model, section 4 elaborates experiments and analysis the results done on 2 different domains using our CRKV model and finally section 5 gives the conclusion.

II. BACKGROUND

A. Knowledge Graph

It is a database that improves Google's web search tool results with semantic search data accumulated from a wide collection of sources. A knowledge graph [7] is a multi-relational graph made out of elements as nodes and relationships as edges with various sorts that depict facts in the environment. Out of the numerous highlights associated with the processing of information sources to make a knowledge graph, Natural Language Processing (NLP) takes a significant role. The search objective changes depending on the area in which it is utilized, so it would vary generously among web search and product list navigation in a web based business webpage, scientific revelation, and expert search noticeable in medication, law and research. Every domain varies as far as business objectives, the meaning of importance, meaning, ontologies, etc.

Relevant search is done using these four components: content, client, context, and business objective: Information extraction and NLP are vital for giving search items as per the client's content question; Client modeling and suggestion engines take into modifying results as per client inclinations and profiles; Context information like location, time, etc, further refine results depending upon the necessities of the client while answering the question; Business objectives drive the whole execution, as search exists to add to the achievement and benefit of the institution.

B. Graph Database

There are two main components that distinguish native graph technology: storage and processing. Graph storage refers to the fundamental structure of the database that contains graph data [8]. The framework specifically constructed for putting graph-like data, it is known as native graph storage. Graph databases with native graph storage are especially for graphs in each perspective, guaranteeing that data is kept productively by naming nodes and relationships with one another. Graph storage is classified as non-native when the storage originates from an outside source, for example, a relational, columnar, or other database [9]. These databases utilize different procedures to store information about nodes and relationships, which may wind up being set far separated. This non-native methodology can lead to hidden outcomes as their storage layer isn't sophisticated for graphs. Native graph handling is another key component of graph technology, alluding to how a graph database performs database operations, including both storage and queries. Index free adjacency graph is the key differentiator of native graph processing.

C. Neo4j

It is an open-source, NoSQL, native graph database that gives an ACID-compliant value-based backend for your applications [10]. Development started in 2003, yet it has been freely accessible since 2007. Neo4j is alluded to as a native graph database since it productively executes the property of graph model down to the storage level. This means that the data is put away precisely as denoted, and the database utilizes pointers to navigate and traverse across the graph. As opposed to graph processing or in-memory libraries, Neo4j likewise gives full database attributes, including ACID transaction consistence, cluster support, and runtime failover – making it appropriate to utilize graph for data in progress situations.

III. OUR CR KNOWLEDGE VISUALIZATION MODEL

The methodologies used in Concept Relation Knowledge Visualization (CRKV) Model are divided into three stages. The figure 1 gives the pictorial architecture of CR Knowledge Visualization Model. In the first stage, the text input is taken and converted into CR logic by segregating concepts and relation. In the second stage, the concepts are defined as per Cypher Query Language (CQL). In the final stage, the generated code returns all the concept nodes with their relationship.

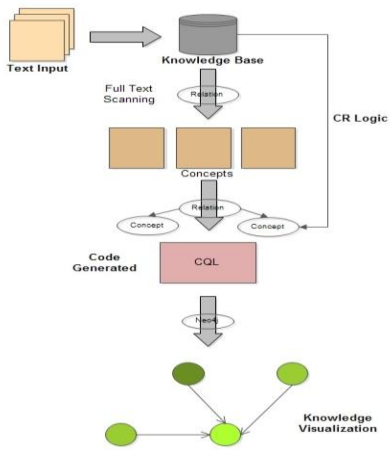

Fig. 1.CR Knowledge Visualization Architecture.

Retrieval Number: D9792118419/2019©BEIESP
DOI:10.35940/ijrte.D9792.118419
Journal Website: www.ijrte.org

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

Our Proposed CR Knowledge Visualization Model works as per the Algorithm given below. The input text is analyzed for recording the words and word count. Then the knowledgebase is checked for concept words and relation. Then the CR logic is generated using the concepts and relation in the given sentence referring the knowledgebase. From the logic created, the code is generated in CQL by defining the concepts with label name. Finally all the concept nodes are viewed in Neo4j.

Input: Text Input
Process: Convert into CR logic -> translate into CQl
Output: Knowledge Visualized in graphical structure

Procedure:

```
DETERMINE the length of the content (words) in the sentence
INITIALIZE words={w1,w2,w3….wn}
CHECK Knowledgebase for words as concept or relation
IF present in Concept_Table
   THEN already existing concept word
ELSE IF present in Relation_Table
   THEN relation word
ELSE new word
CHECK Relation_Library
IF present in library
   THEN relation word
ELSE new concept word
   ADD to Concept_Table

PASS words in sequence as in sentence (text input)
IF concept word
   THEN represent as (concept)-
ELSE IF ending concept word
   THEN represent as (concept)
ELSE relation word
   THEN represent as [:relation]->

DEFINE concepts      #CQL code generation
DETERMINE the frequency of occurrence for each concept word
CREATE a concept name with node property
ARRANGE the logic generated as patterns
RETURN all the concept words

VIEW the graphical structure in Neo4j platform

END
```

A. CR Logic

The text input is processed for consolidating the data as concept and relation. The sentence and order of occurrence is also recorded. These are done in the initial stage and the information is mapped. Thus the knowledge is represented in our CR Knowledge Representation Model [11].

The concepts are connected with each other through a relation. This is done with the help of CR Logic [12]. The pre-defined relation with respect to each scenario is employed for connecting every concept with each other.

B. Definition

Concept: Entities, attributes are denoted as concepts. The concept is depicted as ( ).

Relation: The occurrence of the concept they are linked to is termed as relation. The relation is shown as [ ].

CR Logic structure: Every concept is linked with another concept through a relation. The below given depicts CR logic representation.

(Concept1)-[:relation]->(Concept2)

C. Components

The parsed sentence contains both concepts and relation words. The CR representation algorithm recognizes and categorizes the word as either concept or relation. These inferences are recorded in the sentence_form table along the order of occurrence.

In the CR logic algorithm, each word is taken in order and identified as either concept or relation. The first word is normally recognized as concept. The other words are taken one by one, if it is a concept then it is represented as (concept_name)- and if it is predicted as relation then it will be shown as [:relation_name}->. Each relation should always begin and end with concept, otherwise it will be treated as an error. Concept can also be represented as a single component. Whereas such single concept and ending concept in the statement are represented as (concept).

D. Neo4j Code Generation using Cypher Query Language

Neo4j enables us to store and retrieve data from the graph database using Cypher Query Language (CQL) [13]. Syntax provided by CQL helps to visualize and logically match patterns of nodes and relationships in the graph.

In CQL each node is represented as (node). It can be represented as a variable or with a variable name as (p) or (person). To extend the relationships between nodes CQL uses an arrow → or ←. Undirected relationships are represented with no arrow --. The relationships are represented as [:relationship]. Node properties is given as (p:Person {name: “Peter”}).

Initially the nodes are created with their properties. Each node will be uniquely created using the CREATE function. Then the nodes along with their relationship are created for representation is the graph database.

E. Mapping

The building blocks for graph patterns are nodes and reships. Simple or complex patterns are expressed using these building blocks. The most powerful capacity of graphs are patterns. In CQL, these patterns are written as continuous or separate paths using commas. This query language enables us to find existing pattern or start a new pattern for feeding the information.

F. Interpretation

The nodes and relationships are logically placed, CQL has reserved a few words for performing specific actions like to search, update, or delete data in Neo4j.

MATCH works like SELECT in SQL, where from graph database search for existing nodes, relationship, label, property or pattern is done. It displays the node and relationship if it finds the particular node in the database.

RETURN in CQL, displays the exact value or result what is expected from the cypher query. These nodes can further be expanded for more description and exhibiting their relationships.

G. Knowledge Visualization

The graphical structure for knowledge visualization enables us to view the nodes of information and also help to traversal in order to understand and reason out better. This Neo4j graph database platform fulfils our requirement by logically defining the nodes and relationships from the input text and placing them in the correct order as per our CR knowledge representation model.

CQL makes it possible to create the nodes whether it is existing or new and link these nodes with relationship as defined in our CR logic model. Later in constructing the patterns associated with the nodes. RETURN * function is used for visualizing the complete graphical structure.

IV. EXPERIMENTAL RESULT

The implementation of our CRKV Model is done on these two different scenarios. The first is based on the chatpot in university, where the relationships specified are based on educational institution requirement. Whereas the second is based on the malware monitoring system, in this the relationships were identified and added based on attributes affecting the behavior of the file system.

A. CRKV implementation for ChatPot

Input:

VIT offers course on KE.
KE is Offered in the Department of SCSE.
BTech students are eligible to take KE.
Subject_code for KE is CS1088.

Output:

CR Logic:

```
(vit)-[:offers_course_on]->(ke),
( ke)-[:is_Offered_in_the]->(department)-[:of]->(scse),
(
btech)-[:_]->(students)-[:are]->(eligible)-[:to_take]->(ke),
( subjectcode)-[:for]->(ke)-[:is]->(cs1088)
```

CQL code generated:

CREATE(vit:Concept {Name:"vit"}),(ke:Concept {Name:"ke"}),(ke:Concept {Name:" ke"}),(department:Concept {Name:"department"}),(scse:Concept {Name:"scse"}),(btech:Concept {Name:" btech"}),(students:Concept {Name:"students"}),(eligible:Concept {Name:"eligible"}),(subjectcode:Concept {Name:" subjectcode"}),(cs1088:Concept {Name:"cs1088"}),(:Noun {Name:""})

CREATE(vit)-[:offers_course_on]->(ke),(ke)-[:is_Offered_in_the]->(department)-[:of]->(scse),(btech)-[:_]->(students)-[:are]->(eligible)-[:to_take]->(ke),(subjectcode)-[:for]->(ke)-[:is]->(cs1088)

RETURN *

Visualization using Neo4j:

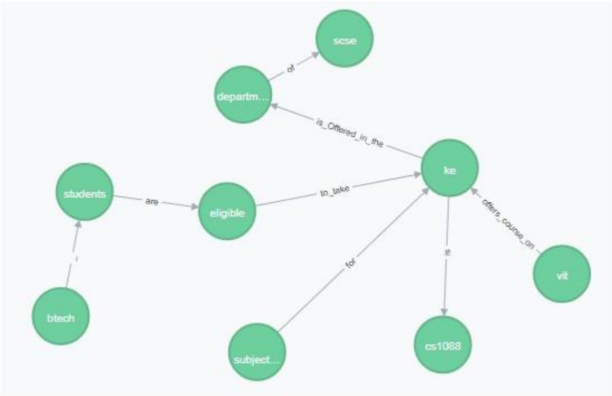

Fig. 2. Knowledge Visualization of Chatpot using CR Model in Neo4j.

Result Analysis:

The text representation (input) has been analyzed and classified as concept and relation. Further, the content was represented using CR logic to retain the meaning of the sentence. Later the content is visualized as nodes and relationship as in Fig. 2. This has reduced the content size and helps in easy capturing of knowledge.

The Table.1 and graph in Fig. 3 below shows the number of words in the input text, concepts and relation identified through our implementation.

Table 1. Performance of CR Knowledge Visualization Model in Chatpot

Input Text length (words)	26
concept nodes	9
relation	17

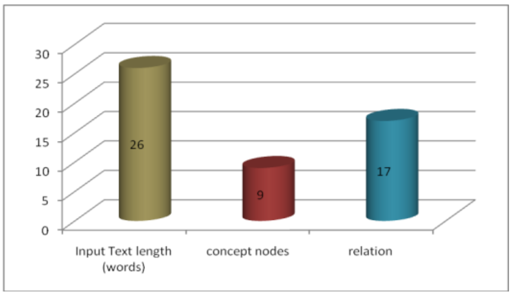

Fig. 3. Reduced content size in CRKV in Chatpot.

B. CRKV implementation for Malware

Input:

malwareName Adware debugSize 0.
Adware IatRVA 5976.
Adware ExportSize 0.
Adware ImageVersion 0.
Adware ResourceSize 68280.
Adware VirtualSize 69632.
Adware NumberOfSections 2.

Output:

CR Logic:

```
(malwarename)-[:_]->(adware)-[:debugSize]->(_0),
(adware)-[:IatRVA]->(_5976),
(adware)-[:ExportSize]->(_0),
(adware)-[:ImageVersion]->(_0),
(adware)-[:ResourceSize]->(_68280),
(adware)-[:VirtualSize]->(_69632),
(adware)-[:NumberOfSections]->(_2)
```

CQL code generated:

```
CREATE(malwarename:Concept
{Name:"malwarename"}),(adware:Concept
{Name:"adware"}),(_0:Concept
{Name:"_0"}),(_5976:Concept
{Name:"_5976"}),(_68280:Concept
{Name:"_68280"}),(_69632:Concept
{Name:"_69632"}),(_2:Concept {Name:"_2"})

CREATE
(malwarename)-[:_]->(adware)-[:debugSize]->(_0),
(adware)-[:IatRVA]->(_5976),
(adware)-[:ExportSize]->(_0),
(adware)-[:ImageVersion]->(_0),
(adware)-[:ResourceSize]->(_68280),
(adware)-[:VirtualSize]->(_69632),
(adware)-[:NumberOfSections]->(_2)

RETURN *
```

Visualization using Neo4j:

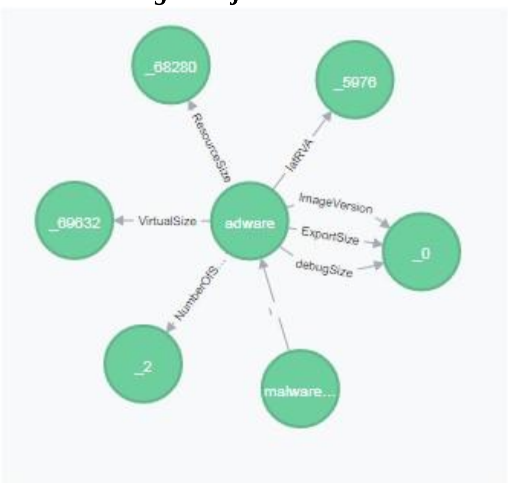

Fig. 4. Malware Parameter Visualization using CR Model in Neo4j.

Result Analysis:

The text representation (input) for this case study is analyzed and identifies the variable values as concept and attribute name as relation. Further, these parameters were represented using CR logic. Then these attribute values are visualized as nodes and names as relationship as in Fig. 4. This enables the analyst in monitoring the deflected files and helps in easy recovery from the attack.

The Table.2 and graph in Fig. 5 below shows the number of words in the input text given as file attributes, concepts as value of attribute and relation as name of attribute are identified through our implementation.

Table 2. Performance of CR Knowledge Visualization Model in Malware Detection

Input Text length (file attributes)	22
concept nodes (attribute values)	7
relation (attribute name)	15

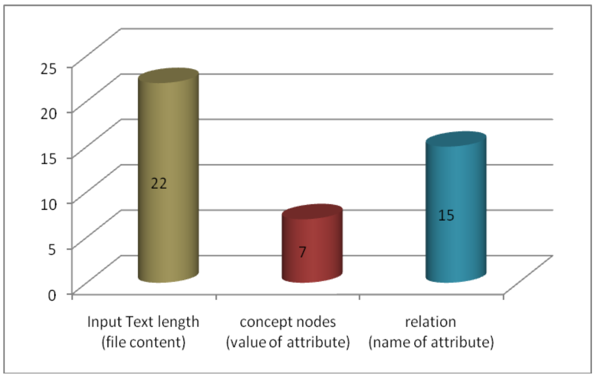

Fig. 5. Content size reduction using CRKV in Malware File Monitoring.

V. CONCLUSION

This Concept Relation Knowledge Visualization Model works well in identifying the concepts, generating logic and the code in CQL for execution. These recognized concept and relation are substituted as the node and relationship in Neo4j for visualizing the knowledge acquired.

Our CRKV Model creates different knowledge patterns using relations and concepts. This model reduces the content size making the knowledge generation process more accurate for the users to retrieve the exact answer.

Our CR Knowledge Visualization currently works using basic functionalities of CQL. In future, making use of more functions and properties, the system will be able to solve more complex logical entailment easily.

REFERENCES

1. Burkhard R. A., "Learning from architects: the difference between knowledge visualization and information visualization", Proceedings. Eighth International Conference on Information Visualisation, 2004.
2. Martin J. Eppler, "Knowledge Visualization: Towards a New Discipline and its Fields of Application", Paper #2/2004, Institute for Corporate Communication, 2004.
3. Martin J. Eppler and Remo A. Burkhard, "Visual representations in knowledge management: framework and cases", Journal of Knowledge Management, Vol. 11 NO. 4, pp. 112-122, 2007.
4. Burkhard R.A., "Towards a Framework and a Model for Knowledge Visualization: Synergies Between Information and Knowledge Visualization", Knowledge and Information Visualization. Lecture Notes in Computer Science, vol 3426. Springer, Berlin, Heidelberg, 2005.
5. S. Praveena Rachel Kamala and Dr. S. Justus, "Towards MORK: Model for Representing Knowledge", I. J. Modern Education and Computer Science. 2016; pp. 45-53.
6. Marchese F., "Tables and Early Information Visualization", Knowledge Visualization Currents. Springer, London, 2013.
7. "The Power of Graph-Based Search", white paper - The #1 Platform for Connected Data, neo4j.com, April 2015.
8. Mark Needham and Amy E. Hodler,"Graph Algorithms: Practical Examples in Apache Spark and Neo4j", O'Reilly Media Publisher.
9. Ian Robinson, Jim Webber, and Emil Eifrém, "Graph Databases " Edition 2, O'Reilly media Publisher.

10. https://neo4j.com/developer/get-started/
11. S. Praveena Rachel Kamala and Justus Selwyn, “Concept-Relation Constructs for Knowledge Representation”, International Journal of Control Theory and Applications, Volume 9, Number 52, 2016, pp. 463-473.
12. S. Praveena Rachel Kamala and Justus. S, "Concept Relation Logic for Visualizing File Behavior in a Knowledge Representation Model", International Journal of Engineering and Advanced Technology (IJEAT), ISSN: 2249-8958, Volume-8 Issue-5, June 2019, Page No.:1732-1740.
13. https://neo4j.com/developer/cypher-basics-i/

AUTHORS PROFILE

Praveena Rachel Kamala S is a full time Ph.D. Research Scholar at the School of Computer Science and Engineering, VIT University, Chennai, TN, India. Her areas of research include Knowledge Engineering, Knowledge Representation, Knowledge Storage and Retrieval. She has received her undergraduate and postgraduate degrees in Computer Science & Engineering from Madras University and Anna University, Chennai. She has worked as lecturer in several Engineering Educational Institutions in India.

Dr. S. Justus received his doctorate degree from Madurai Kamaraj University, Madurai, India. His research specializations include Object-relational data modeling, knowledge engineering and Big Data.
He has been into academic research and has published several of his research work results in International Journals and Conferences – including SwSTE in Israel and DASMA in Germany. He has also practitioner’s experience while working with Software development companies. He is a member of IEEE, ISTE, IAENG professional associations. He has served as research & project coordinator for pg studies at engineering institutes. Presently he is working as associate professor at VIT University, chennai.