

A Knowledge Representation Model using Concept-Relation Graph

Praveena Rachel Kamala S., Justus S.



Abstract: Huge volume of relevant and irrelevant information is available from different sources for gaining knowledge about a system. If the required data is in a structured form, then the fact can be easily understood otherwise the process becomes laborious and results are vague after intense analysis. In this paper, we are proposing a framework for fetching knowledge from unstructured source of data. The algorithms proposed identifies the concepts and separates the concepts and relation words, enables to add new concepts and also modify the old concept word with new concept and locate the concept. This Concept-Relation Model enables the system to work according to the users' connivance and deliveries accurate knowledge. This model does not manipulate or interpret the information provided but only represent and share the desired knowledge.

Index Terms: Knowledge Representation, Logic, Visualization.

I. INTRODUCTION

With the advent of big data and IoT data, knowledge engineered intelligent and smart systems are in need for knowledge representation models that would answer the end users' queries. The existing knowledge representation models use one or combination of first-order-logic (FOL), petri-net, concept-maps, frames etc. These models are quite useful when the data size is small and static, but do not address a growing dataset with variety of data flowing in at different velocity [1]. The regular procedures for acquiring knowledge are focused more on capturing atomic concepts and knowledge units (KU) but the redundant KU, and the relationships among the KUs and concepts are insufficient. Redundancy brings to gather similar KU under various categories or domains. Searching and sharing of KU turns out to be more mind boggling and confounded because of these issues. Nonappearance of relationship between concepts winds up in conveying confused responses for the question posted by the user.

Another open issue is choosing a more suitable storage and retrieval of knowledge units. As far as representation, a precise structure is required for adapting information into a knowledge unit and codifies it into a functional model for intelligent systems. The challenge is how these KUs can be made persistent by storing in a more suitable storage structure and better way of retrieving them so as to answer the end users' query in a more effective way.

Manuscript published on 30 December 2019.

* Correspondence Author(s)

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.

© The Authors. Published by Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP). This is an [open access](#) article under the CC-BY-NC-ND license <http://creativecommons.org/licenses/by-nc-nd/4.0/>

The knowledge shared to the user from the acquired information is done by pre-training the system or predicting the outcome through experiences. In that situation if any misassumption or new scenario occurs then the knowledge cannot be generated.

To beat these disadvantages, a point by point study on all the available models and techniques was carried out earlier. As a result of the study, a new knowledge representation model is proposed, called a concept-relation graph. We have focused on an orderly approach to translate the data, and cautiously setting it in an efficient structure withstanding the rules and restrictions.

In this paper, the process involved in representing knowledge is discussed in Section 2. In Section 3, a knowledge representation model using description logic and conceptual graph is illustrated. In Section 4, the four key algorithms for content parsing, adding nodes, modifying nodes and traversing nodes are explained with complexity. In Section 5, conclusion and future work is given for the knowledge representation model designed.

II. PROCESSES IN REPRESENTING KNOWLEDGE

Our proposed models for representing knowledge utilize the logical concepts like predicate logics, and first order logic for the information to be deciphered. Description logic is connected to form rules and restrictions. The acquired concepts are mapped to a graphical structure to decide the relationship between every condition. This empowers the development of Concept Relation (CR) diagram to be simple.

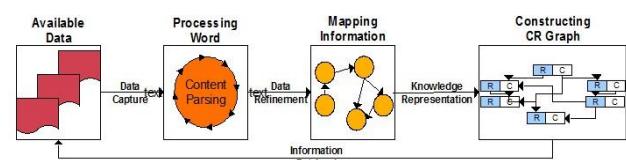


Fig. 1: A Model for Representing Knowledge

Fig. 1 clarifies the general working of the system. Initially the content is received from the available source, in this case, from a structured repository, and the content is processed. The identified concepts are then connected together by means of relations between them. These relations determine the semantics of the two concepts it connect. Our model supports structured and semi-structured form of information for processing and knowledge acquisition. After the information is fed into the structure, content parsing is done where each and every word in processed for identifying it as a concept or a relation that connects two concepts.



A Knowledge Representation Model using Concept-Relation Graph

Conflicts in validating the concepts and its relations are carried through logical entailment algorithms. After resolving the conflicts, the graph is constructed using the algorithms we have proposed in this paper.

Conceptual Graph (CG)[2] idea composed by John Sowa[3], defines the conceptual form of knowledge representation. The concepts are represented to as nodes in the diagram having logic based relationship. A CG based knowledge representation encourages a clear and organized approach for codifying the information. CG additionally allows a more precise move towards reasoning. The graphical plan of CG empowers the user to plainly screen the structure and operational technique of the system. Studying the benefits of CG, we have fused this graphical concept into our model for greater clarity, functional structure, logical entailment, and for user assistance.

III. REPRESENTING USING DL AND CG

In this Model, the knowledge represented is done by converting the statement into description logic using first order logic[4]. Rules are generated and suitable reasoning is done by applying conceptual graphs structure. Accordingly required output is obtained from the diagram.

Consider this as input statement: "If no man is reading the book, then is there someone who wrote the book reads it?"

Step 1:

The given statement is converted into logical statements using first order logics (FOL).

1. FOL for the given statement "If no man is reading the book, then is there someone who wrote the book reads it?"

$$\forall x [man(x) \wedge \neg Read(x, book)] \supset \exists x [Author(x, book) \wedge Read(x, book)]$$

2. No man reads the book.

$$\forall x [\neg man(x) \wedge Read(x, book)]$$

3. Adult is a group containing men and women. (if men are not reading then women are reading)

$$\forall x [adult(x) \supset man(x) \wedge woman(x)]$$

4. Woman reads the book.

$$\forall x [adult(x) \supset woman(x) \wedge Read(x, book)]$$

5. Assuming negation approach: The author does not read the book.

$$\forall x, \forall y [Author(x, y) \supset \neg Read(y, x)]$$

6. The author of the book is not Jane.

$$(y=Jane) \forall y [woman(y) \wedge y \neq Jane \supset Author(book, y)]$$

7. A woman, who is not Jane, does not read the book. (from 5)

$$\forall y [woman(y) \wedge y \neq Jane \supset \neg Read(y, book)]$$

8. Conclusion: Jane is the author.

Step 2:

General properties for different concepts can be given by

using restrictions and distinguish primitive definitions. The Primitive concepts are: Adult \sqsubseteq T, Woman \sqsubseteq Adult, Man \sqsubseteq Adult, book \sqsubseteq {1}

Defined Concepts:

1. [ALL :Man [FILLS :hasn't Read]] -----> From 2
of Step 1

2. [ALL :Adult[AND Man, Woman]] -----> From 3
of Step 1

3. [AND Woman [FILLS :has Read]] -----> From 4
of Step 1

4. [AND Author [FILLS :hasn't Read]] -----> From 5
of Step 1 (negation approach)

5. [ALL :Woman[EXISTS 1 :Woman (Jane -> [FILLS :can't_be Author])[FILLS :can_be Author]]] -----> From 6
of Step 1

6. [ALL :Woman[EXISTS 1 :Woman (Jane -> [FILLS :can't_be Author])[FILLS :hasn't Read]]) -----> From 7
of Step 1

7. [ALL :Woman [EXISTS 1 :Woman [ALL :except Jane]
[FILLS :can't_be Author]]] -----> From 5 and 6 of Step 2

8. [AND Woman [FILLS :Jane Author]] -----> From
7 of Step 2 (conclusion).

Step 3:

Using the theory of conceptual graphs, the information is mapped using the concept – relation graph. Consider the Fig. 2, which gives the conceptual structure for the above mentioned statement.

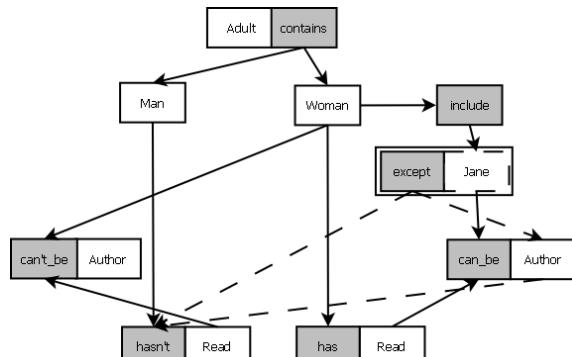


Fig. 2: Concept-Relation Graph for Author – Reader example

In this CR graph, the assumptions are indicated by using a double dash border node as in case of Jane. Since the assumed concept along with the relation signifies a negative approach, they are indicated by a double bordered node.

Step 4:

The concept along with their relationship is represented as a single node. Mapping between nodes enable the graph to reason the sequence of action.

From the above CR graph, the problem is defined as Adult contains both Men and Women. Since no man reads the book, only a woman can be the Author of the book. By introducing a constant Jane under the category Woman and by applying negation approach to the situation, the conclusion is "Jane is the author of the book".



IV. THE ALGORITHMS FOR CONSTRUCTING CONCEPT-RELATION GRAPH

A few algorithms are designed for working with the graphical structure. The distinctive operations done in a Concept-Relation (CR) diagram are including new nodes, modifying the current nodes and finding the presence of the desirable node. The design used for recovering the knowledge from the graphical structure resembles an attractive magnetic ball impact. At the point when a concept is searched and discovered, at that point all the related concepts (nodes) will be brought relying on their level of relationship. Therefore the retrieval of the stored concept is completed by following the path of their relation. Our Model includes different stages in creating significant extraction of knowledge. This is accomplished by handling the information through various levels. The working methodology and algorithms is discussed below.

A. Algorithm for Content Parsing

This technique for capturing data [5] assumes a crucial part in knowledge representation [6]. The meaning of the source ought to be same while obtaining the facts. There ought to be no loss of records. The area of storage should be simple bringing of data for further manipulation.

Notation:

Given a sentence 'S', containing data 'D' with concepts 'C' and relations 'R'.

Such that $S: D \rightarrow \{C \cup R\}$, where $C = \{C_1 \cup C_2 \cup \dots \cup C_n\}$ and $R = \{R_1 \cup R_2 \cup \dots \cup R_m\}$

Algorithm:

Content Parsing(word_document)

Input: Data: $\{D_1 \cup D_2 \cup D_3 \cup \dots \cup D_k\} = \{C_1 \cup R_1 \cup C_2 \cup R_2 \cup \dots \cup C_n \cup \dots \cup R_m\}$

Given: A set of predefined Relation, Concept Table (CT) and Relation Table (RT)

Output: Parsed content in Sentence Table (ST)

$\{C_1 \cup C_2 \cup \dots \cup C_n \cup R_1 \cup R_2 \cup \dots \cup R_m\}$

Procedure:

If Data!= NULL

Add Data in Table with s_id

Ch = COUNT(Data)

While (!=Ch)

For cmp 0 to Ch

If Dcmp \in Relation

Check Dcmp in RT

If Dcmp \in RT

 Add Dcmp in ST with r_id

Else

 Add Dcmp in ST with new r_id

Else

 Check Dcmp in CT

If Dcmp \in CT

 Add Dcmp in ST with c_id

Else

 Add Dcmp in ST with new c_id

Repeat till Ch

Else

 Print "Enter Valid Value"

In this algorithm Content_Parsing, the word record from where information must be taken is dealt with as the input. This record is opened; every word of the lines in the document are perused and stored in the short-term memory. These data are part word by word and recorded in the memory. A spreadsheet is opened [7] and the records are included by composing the split words onto diverse cells in column wise. At that point the recently made spreadsheet is updated for later use. These captured words are included in the sentence table for computational reason.

A unique number is created for the word present and a sequence number is additionally produced as the words must be organized in correct arrange. The information captured is investigated by checking the content report containing the list of relations defined. If the word is already defined in relation list then it is treated as relation otherwise it has to verify with the list of concepts enrolled in the concept table. If the concept is already present then the existing unique number is fetched and used. Adding new concepts to the list requires new unique number.

The Concept Table is created, if no concepts are as of now present. Updating of Concept Table is done only if a single or more concepts are available and just if the concept is absent prior. If the concept previously exists then the unique number is retrieved for completing the sentence table.

The outcome of this algorithm involves creation of sentence table containing the statement from the record given and updating the sentence table with the word, identification of the word as concept or relation, generation of unique number regarding concept and relation and sequence number demonstrating the order of arrangement for simpler computation.

s_id	Sentence
175	It is a very old maxim that "Health is Wealth".
176	A healthy person can work with efficiency to earn wealth.
177	The man who is not having a good health spends lot of money on medicines and doctors.
178	He is never cheerful.
179	Which in itself is a wealth?
180	But we have to take precautions for building up a good health.
181	Healthy body is always dependent on the healthy mind.
182	For maintaining good health we should always be away from the bad habits like smoking, drinking and drug addiction.
183	We should have morning walk regularly, light exercise in fresh air and a stroll after dinner.
184	We should not be jealous of others.
185	We should always take balanced and nutritious diet.
186	One should always develop the habits of cleanliness to keep healthy.

Fig. 3: Content Stored in Sentence Table with Sentence ID

A Knowledge Representation Model using Concept-Relation Graph

fs_id	s_id	c_id	r_id	word	order	type
1718	176	1426	o	earn	9	C
1719	176	1427	o	wealth	10	C
1720	177	o	459	The	1	R
1721	177	o	460	who	3	R
1722	177	o	461	is	4	R
1723	177	o	462	a	7	R
1724	177	o	463	of	12	R
1725	177	o	464	and	16	R
1726	177	1428	o	man	2	C
1727	177	1429	o	not	5	C
1728	177	1430	o	having	6	C
1729	177	1431	o	good	8	C
1730	177	1432	o	health	9	C
1731	177	1433	o	spends	10	C
1732	177	1434	o	lot	11	C
1733	177	1435	o	money	13	C
1734	177	1436	o	on	14	C
1735	177	1437	o	medicines	15	C
1736	177	1438	o	doctors	17	C
1737	178	o	465	is	2	R
1738	178	1439	o	He	1	C

Fig. 4: Parsed Content stored in Table

Complexity:

Assume that 'n' numbers of words are given as input; each word is taken one by one for categorized as either concept or relation and adding to the table. The time taken for processing this algorithm is 'n' unit time for sorting and appending.

The time complexity for Content Parsing Algorithm is $O(n)$.

B. Algorithm for Adding Nodes

To take care of the issues in knowledge representation a powerful structure for symbolizing the data is required. A graphical structure is utilized to streamline this circumstance; consequently Concept Relation (CR) diagram is conveyed.

Notation:

Given a sentence table ST containing concepts C and relations R with their unique ids c_id and r_id respectively. Such that $ST: \{C \cup R\}$, where $CT = \{C_1 \cup C_2 \cup \dots \cup C_n\}$ and $RT = \{R_1 \cup R_2 \cup \dots \cup R_m\}$ are separated and placed in a unique position in the Concept Relation (CR) graph.

Algorithm:

Adding_Content(word1,...,wordn)

Input: Content (word₁,..., word_n) to be included from ST

Output: Entire content added to the CR graph structure

/* Relation Table (RT) contains a set of relations */

/* Concept Table (CT) contains clauses, which are existing in the graph*/

Procedure:

DETERMINE the length of the content

INITIALIZE count=0

IF count < length of the content

CHECK IF the word is in RT

RETURN (relation)

count++

ELSE

CHECK IF the word is present in CT

RETURN (concept)

count++

ELSE

CHECK IF CT ≠ φ

FIND the word position to be added

ADD the word to CT

count++

ELSE

ADD the word to CT as position=1

```
count++
END
```

As a pre imperative, two tables are made. One table known as Relation Table (RT) is loaded with the relations. This RT is pre-characterized and changes can't be made. The other table is Concept Table (CT), which continues adding concepts to the table as new undefined words are added.[8]

At the point when the words captured are feed from the table, the quantity of words is calculated. Checking the count to be lesser than the quantity of words acquired for the table, the technique proceeds. Each and every word is taken independently and compared with the pre-characterized verb list in the RT. If the word present, then that word is marked as a relation and its position is returned. Otherwise, that word is verified with the clauses which are already added into the CT. If the CT contains no word, then add this word as the first word. Moreover find out the most suitable location for adding the word in CT. Increase the count by one and repeat the same routine for the next word till all the words from the spreadsheet are placed in the correct position.

s_id	c_id	concept	order
182	1488	bad	12
182	1489	habits	13
182	1470	like	14
182	1471	smoking	15
182	1472	Drinking	16
182	1473	drug	18
182	1474	addiction	19
183	1475	We	1
183	1476	should	2
183	1477	have	3
183	1478	morning	4
183	1479	walk	5
183	1480	regularly	6
183	1481	light	7
183	1482	exercise	8
183	1483	fresh	10
183	1484	air	11
183	1485	stroll	14
183	1486	after	15
183	1487	dinner	16

Fig. 5: Concept Table (CT) updated with Concepts

s_id	r_id	relation	order
177	464	and	16
178	465	is	2
179	467	in	3
179	468	is	5
179	469	a	6
180	470	But	1
180	471	to	4
180	472	for	7
180	473	a	10
181	474	is	3
181	475	on	6
181	476	the	7
182	477	For	1
182	478	the	11
182	479	and	17
183	480	in	9
183	481	and	12
183	482	a	13
184	483	be	4
184	484	of	6

Fig. 6: Relation Table (RT) with Relations



Complexity:

The parsed words are added to the respective tables if they are recently included (i.e.) Concept Table or Relation Table. The time taken for processing this algorithm is ‘n’ unit time for picking the words and adding the word to a particular table.

The time complexity for Adding Node Algorithm is O(n).

C. Algorithm for Modifying Nodes

The clauses in the graph can be changed or deleted. At the point such variation happens in appropriate substitution. At the point when a concept is modified, the relation existing between those nodes with alternate nodes should be carefully handled.

Notation:

In a Concept Table CT = {C₁UC₂U...C_n}, given an element C_i to be modified.

If C_i ∈ CT, then C_i = C_{i'}, Where C_i and C_{i'} are old and new modified values of C_i.

Algorithm:**Modify_Concept(word,whatchange)**

Input: Concept (C) to be modified as (C_{new})

Output: Modified CR graph.

Procedure:

```

IF CT ≠ φ
IF Concept (C) ⊑ CT
    THEN C = Cnew
    ALL the relation (r1..,rn) nodes associated with C are
    marked
    CHANGES are made as Cnew and (r1..,rn)
    ELSE
        RETURN Concept (C) not found in CT
RETURN CT is Empty
IF C ⊑ RT
    NO modification can be done
ELSE
    RETURN Concept (C) not found in RT

```

If any change must be made to an element then its accessibility in the CT is checked. If it is available then all the relation nodes rising up out of this concept node are marked. Fundamental changes like replacement or deletion can be performed to those marked nodes.

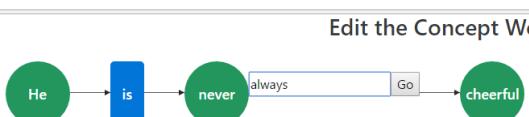


Fig. 7: Concept for Modification

If replacement of concept must be done, at that point the unique number must be changed. The newly replaced concept exist the concept list, then the corresponding unique number has to be updated in the concept table and sentence table, otherwise new unique number has to be generated.

For deletion of concept, the concept name along with unique number and sequence number has to be removed from the sentence table. If the concept is in the beginning of the

sentence formed, then the entire sentence is deleted from the sentence table and necessary updates like change of unique number, concept number are made in the concept table.

If the word is found in RT, then no modification can be done because this table is not created by appending data but by pre defining the verbs as relations.

Complexity:

For checking the presence of a specific concept or relation in knowledge base, all the elements from both the tables are checked. Suppose ‘n/2’ number of concepts is from Concept Table (CT) and ‘n/2’ number of relations is present in Relation Table (RT). Then the maximum number of comparison requested for identifying the word is ‘n/2 + n/2 = n’.

The time taken for figuring out the target node requires ‘n’ unit of time. Replacing the old word with a new word requires ‘1’ unit of time (i.e.) rewriting the old with new concept.

The time complexity for Modifying Node Algorithm is O(n).

D. Algorithm for Traversing Nodes

Reasoning [9] assumes a more prominent part in the process of knowledge representation [10]. Therefore finding the occurrence of a concept and getting their relation with other concept fetches the model [11].

Notation:

In a Concept Table CT = {C₁UC₂U...C_n}, given an element C_i to be modified.

If C_i ∈ CT, then C_i is displayed along with other neighboring nodes.

Algorithm:**Traverse_Content(word)**

Input: Concept (C) to be searched

Output: Displays the related information.

Procedure:

```

IF CT ≠ φ
IF Concept (C) ⊑ CT||RT
    THEN the related Nodes are fetched
    DISPLAY the concepts (C, C1, C2..,Cn)&& their
    relation(r1..,rn)
    ELSE
        RETURN Concept (C) not found
RETURN CT is Empty

```

This algorithm sees if the clause is available in CT or RT. Once the presence is confirmed then the clause alongside the other related neighboring concepts and their relationship are taken out and displayed.[12] Or else the presence of the searched entity is denied.



A Knowledge Representation Model using Concept-Relation Graph

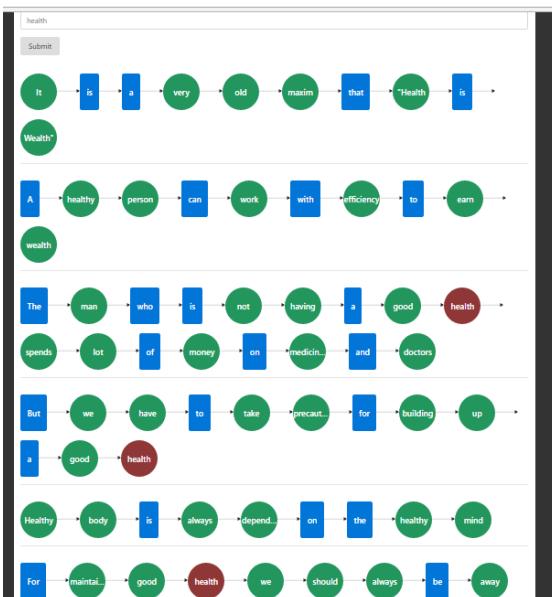


Fig. 8: Search for Concept

The extracted concepts if connected with more than one concept through a relation, at that point other concepts are also shown. Thus enabling the retrieval of every possible concept associated with the selected concept.

Complexity:

The time taken by this algorithm to determine the presence of a given word is checked in both Concept Table (CT) and Relation Table (RT), and to display if present is ' $n/2 + n/2 = n$ '.

The time complexity for Traversing Nodes Algorithm is $O(n)$.

V. CONCLUSION

In this paper, we have presented the working of our knowledge representation model to demonstrate with different algorithms for content parsing the record, including, changing and searching concepts. The objective of this paper is to bring the content from an source which is semi organized, to place the content in a systematic way and to extract significant content from the knowledge base using Concept_Relation (CR) Graph.

Brief description of a knowledge representation model using description logic and conceptual graph is given along with the various algorithms for constructing the concept relation graph. The data is taken and processed, to yield adequate knowledge. The main advantages in our representation model are no prior knowledge about the scenario is required and no data training [13] is done. As the data is fed into the system, it is processed and the graph is constructed. This can be easily adapted to any environment for generating knowledge.

A few more enhancements like logical entailment and reasoning a situation can be done as future work. This can eventually give conceptual information.

REFERENCES

- Frank van Harmelen, Vladimir Lifschitz and Bruce Porter, "Handbook of Knowledge Representation". 1st Edition: 2007; Volume 1.
- Madalina Croitoru et al, "Graphical norms via conceptual graphs". Knowledge-Based Systems. 2012; 29: pp. 31–43.
- John F Sowa, "Knowledge Representation: Logical, Philosophical and Computational Foundations". New York, PWS Publishing Co., 2000.
- 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.
- Chuntao Jiang et al, "Text classification using graph mining-based feature extraction". Knowledge-Based Systems. 2010; 23: pp. 302–308.
- Praveena Rachel Kamala S and Justus S, "Concept – Relation Constructs for Knowledge Representation". International Journal of Control Theory and Applications. 2016; 9: pp. 463 - 473.
- Tiago A. Almeida et al, "Text normalization and semantic indexing to enhance Instant Messaging and SMS spam filtering". Knowledge-Based Systems. 2016; 108: pp. 25–32.
- Julia Hoxha et al, "Automated learning of domain taxonomies from text using background knowledge". Journal of Biomedical Informatics. 2016; 63: pp. 295–306.
- Miao Fana et al, "Distributed representation learning for knowledge graphs with entity descriptions". Pattern Recognition Letters. 2016; 000: pp. 1–7.
- Praveena Rachel Kamala S and Justus S, "Concept Relation Logic for Visualizing File Behavior in a Knowledge Representation Model". International Journal of Engineering and Advanced Technology. 2019; 8: pp. 1732 -1740.
- Misael Mongiovì et al, "Merging open knowledge extracted from text with MERGILÓ". Knowledge-Based Systems. 2016; 108: pp. 155–167.
- Antonio Jimeno Yepes et al, "Knowledge based word-concept model estimation and refinement for biomedical text mining". Journal of Biomedical Informatics. 2015; 53: pp. 300–307.
- Alberto Tonon et al, "Contextualized ranking of entity types based on knowledge graphs". Web Semantics: Science.Services and Agents on the World Wide Web. 2016; 37–38: pp. 170–183.

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.