

Available online at <http://www.mecs-press.net/ijem>

## Development of Knowledge Graph for University Courses Management

Ismail Aliyu<sup>1\*</sup>, A. F. D. Kana<sup>2</sup> and Salisu Aliyu<sup>3</sup>

<sup>1, 2, 3</sup>*Department of Computer Science, Ahmadu Bello University Zaria, Nigeria*

Received: 10 January 2020; Accepted: 14 February 2020; Published: 08 April 2020

---

### Abstract

The task of Allocating courses to lecturers in many tertiary institutions is done manually by typing using word processor application. Motivated by the widespread application of knowledge graphs in different domains, we present automated approach based on knowledge graph to address the problem of manual course allocation to, a task usually carried out at the beginning of every semester or academic year by departments in tertiary institutions. The development of knowledge graph in a way that enables easy manipulation and automatic generation of course allocation schedule is the core contribution of this paper. Rather than storing the data in relational database tables, the system stores data in a knowledge graph which is in RDF/XML format and refer to it to support intelligent knowledge services. In addition to automatic generation of course allocation schedule, another important feature of the system proposed in this paper is its ability to enable easy implementation of tasks similar to Question Answering that are very important to education administrators, which the existing manual approach does not provide. Testing of the proposed system reveals its ability to perform effectively. Our approach of using Knowledge graph offers advantages such as flexibility and security.

**Index Terms:** Course allocation, Knowledge graph, Resource Description Framework (RDF)

© 2020 Published by MECS Publisher. Selection and/or peer review under responsibility of the Research Association of Modern Education and Computer Science

---

\* Corresponding author.  
E-mail address:

## 1. Introduction and Literature Review

Allocation of courses to lecturers is an important task that is carried out at the beginning of every academic year or semester. However, in most Nigerian tertiary institutions it is done manually by simply typing on Microsoft word application. A simple question of who taught CS121 in second semester of 2010 academic session?, for example, will amount to hours of searching for printout of course allocation schedule of that semester/year or several minutes of searching the computer for MS word file containing the information. In most cases, the word files stored on computer's hard drive get corrupted or lost. Another problem is lack of automated approach for centralize tracking of what is going on at various departments with regard to allocation and teaching of courses. Considering the facts that number of courses taught by a lecturer is an important parameter that determines the lecturer's Earned Academic Allowance (EAA), such centralize tracking systems can assist school management or directorate of academic planning of the university for verification of academic staff claim for payment of EAA. Unfortunately, such systems do not exist in many Nigerian universities.

In recent past, Knowledge Graph (KG) has been introduced as a model for knowledge representation in a structured way. Knowledge graph represents entities such as people, places, organizations etc and their relationships. Knowledge graphs have been incorporated in many applications in different domains to support various tasks. Previous research that applied knowledge graph in education domain such as [1] used it to support learning and teaching of mathematics. [2] constructed knowledge graph to support scientific resource retrieval for students. To the best of our knowledge no research work was found that leverages knowledge graph to support routine activities related to management and teaching of courses in tertiary institutions. Our aim in this paper is to address the problem manual course allocation using knowledge graph as a model of data storage. Specifically, we construct KG of courses and use the KG to support management of courses, automatic generation of course allocation schedule, and implementation of Question Answering tasks that are important to education administrator. The contributions of this paper are as follows:

- i. We apply knowledge graph to education domain to support management of courses and automatic generation of course allocation information in tertiary institutions.
- ii. Propose an approach that enable education administrator to easily have access to basic information on teaching of courses at various departments. Answers to simple questions (that are often difficult to answer) can be obtained easily. – Examples; How many courses did lecturerY taught in a particular session or semester? Which course is prerequisite to courseX? What are the recommended reading materials for course X?

The rest of this paper is organized as follows. Brief explanation on knowledge graph and some of its applications is the content of section 2. Review of the literature is presented in section 3. The method of building and querying the courses knowledge graph is described in section 4. We describe the case study in which the system was evaluated in section 5. Section 6 concludes the paper.

## 2. KNOWLEDGE GRAPH

Basically, Knowledge Graph is model for knowledge representation. It describes real world entities, objects, concepts etc and their relationships in a graph [3]. It is a structured knowledge base that represents knowledge as triple of the form  $(h,r,t)$ . Where  $h$  is the head entity,  $t$  is the tail entity, and  $r$  is the relation between  $h$  and  $t$ . Example of knowledge graph is shown in fig.1. which has courses and lecturers as entities. The triple  $(Prof\ S.B, taught\_2\_2018, CS311)$  is interpreted as Prof. S.B taught CS311 in the second semester of 2018 session. Similarly, the interpretation of the triple  $(CS121, prerequisite, CS211)$  is CS121 is the prerequisite of CS211.

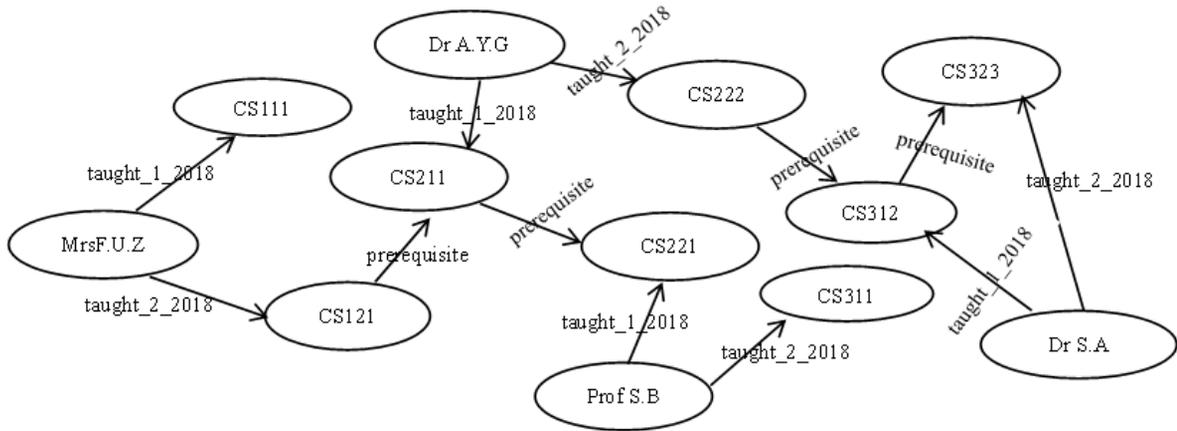


Fig.1. Courses knowledge graph

Knowledge graph can be seen as a network in which nodes are the entities and edges that link entities are the relations. Formally, Let  $E$  be the set of entities and  $R$  be the set of relations. Knowledge Graph,  $G$  is defined as collection of triple facts  $(e_h, r, e_t)$ . That is,  $G = \{(e_h, r, e_t) \mid e_h, e_t \in E, r \in R\}$ .

Knowledge graph was first introduced by Google as a semantic enhancement of their search functions, so that search is beyond “keyword” matching [4]. Since its introduction in 2012, knowledge graph has promoted the development of “semantic” network of entities, which the giant tech companies (Google, Facebook, etc) and many enterprises use to support development of intelligent applications in their domains [5]. Knowledge graphs powered many Artificial Intelligence applications such as search engines, Question Answering systems, Recommender system etc. They have also been applied to variety of domains like Education [1], Biomedical [6], Electric Power System [7], Automobile industry [8], Traditional Chinese Medicine (TCM) to support intelligent knowledge services (such as efficient retrieval, recommendation etc) [9] and many more. Examples of publically available large scale knowledge graphs are DBpedia, YAGO, NELL, Freebase, and Word net.

Knowledge graphs can be curated manually or automatically. They can be constructed by automatically extracting instances of entities, and relations from structured, semi-structured or unstructured resources like text corpus or web. State of the art Natural Language Processing (NLP) techniques for segmentation, tagging, Named Entity Recognition (NER) are employed in the process of entity extraction. Another way to construct knowledge graph is to use data of specific domain aiming at solving some specific questions. Mostly the data are stored in relational databases.

### 3. RELATED WORK

Our aim in this paper is to construct knowledge graph for use in education domain to support management of courses in tertiary institutions. [1] propose system called KnowEdu that automatically construct educational knowledge graph to support teaching and learning of mathematics. KnowEdu uses neural sequence labelling algorithm on pedagogical data (eg, curriculum) to extract concepts of subjects or courses and employs probabilistic association rule mining to identify relations. The nodes of their KG are instructional concepts of subjects. An instructional concept is a basic concept that learner is expected to fully understands eg “linear equation” in Mathematics, “photosynthesis” in Biology. [10] construct a knowledge graph utilizing education

data mined from web, and develop visualization analysis platform called EduVis which supports users to do associated analysis to reveal patterns and help education administrators take decisions based on the data in the graph. [2] designed scientific publication knowledge graph to support scientific resource retrieval and other services for students. The scientific publications knowledge graph seeks to integrate and link scientific publications with entities such as journal, researchers, their affiliations etc, in order to enhance retrieval efficiency and reduce the difficulty of exploring scientific publications, thereby improving students' and researchers' learning ability. The entities, some of which are metadata (such as, title, author, journal etc) were extracted from 3 scientific databases namely, Web of science, Engineering Village, and EBSCO. In educational domain, very few studies focus on construction of domain-specific knowledge graphs. However, some recent works investigated relation extraction between certain entities in educational domain. [11] create Prerequisite Structure Graph (PSG) using an unsupervised approach that utilizes text content and student activity log from heterogeneous sources. The nodes represent the universal concepts in an educational domain and the edges specify the pairwise ordering of concepts in effective teaching by instructors or for effective learning by students. [12] recover prerequisite relations from university course dependencies. [13] and [14] came up with a method called Concept Graph Learning (CGL) that automatically map online courses from different providers (universities or Massive Online Open Courses (MOOCs)) onto space of concepts, and predict latent prerequisite dependencies among both concepts and courses.

Like our work, the previous work reviewed above constructs knowledge graph and uses it to support some tasks in education domain. In contrast, our work constructs and use knowledge graph to simplify the task of course allocation and to support easy implementation Question Answering tasks that are important to education administrators. While some works [1] and [2] extract instance of entities from unstructured text, we use different approach instead. The instances of entities come from structured source such as relational database table, an approach used by [9]. Furthermore, we do not used NLP techniques for Name Entity Recognition (NER) and relation extraction.

## 4. METHOD

This section describes our method for constructing and querying the courses knowledge graph.

### 4.1 Data Schema

Knowledge graph is composed of entities of different types. The description of the types or classes of entities and constraints on their use is usually given in a schema or ontology [3]. Figure 2 shows the schema of the courses knowledge graph. The entities are of two types; person and thing. Therefore the courses knowledge graph proposed in this work do not include entities such as places (cities & countries), organizations etc that are usually included in conventional knowledge graphs.

The entities and relations used in constructing courses knowledge graph are shown in Tables 1 and 2 respectively.

TABLE 1. LIST OF ENTITIES OF COURSES KNOWLEDGE GRAPH

Entity	Type	Description
Lecturer (L)	person	Represents a lecturer that teaches a course
Course (C)	thing	Represent a course that lecturer teaches
courseBook (CB)	thing	Represent the text book recommended for a particular course
Bookauthor (BA)	person	The author's name of the recommended text book

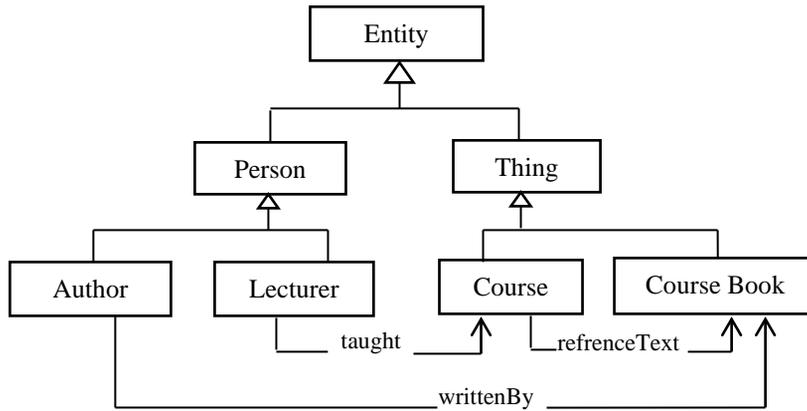


Fig.2. Data schema of the courses management model

TABLE 2. LIST OF RELATIONS OF COURSES KNOWLEDGE GRAPH

Relation	Description
Prerequisite	relation between <i>course</i> and another <i>course</i> entities
taught_in_semester_year	relation between <i>course</i> entity and <i>lecturer</i> entity
writtenBy	relation between <i>author</i> entity and <i>coursebook</i> entity
referenceText	relation between <i>course</i> entity and <i>coursebook</i> entity

#### 4.2 Building the Courses Knowledge Graph

Fig.3. illustrates the architecture of the system that builds courses knowledge graph. The two main modules are Entity extraction, and Relation extraction. In this work, instances of entities are extracted from database since list of courses and lecturers of a department are usually stored in database. Relationships between courses and lecturers are established in an adhoc manner during allocation of courses at the beginning of every semester or academic year. Because of this, we create a Graphical User Interface (GUI) for user to specify such relationships. When user selects lecturer's name, course code, semester and year, such action is interpreted as relation of the form *taught\_in\_semester\_year*. Consequently, a triple of the form (Course, *taught\_in\_semester\_year*, Lectuerer) is formed – eg (CS323, *taught\_in\_1\_2018*, Dr S.A), interpreted as *CS323 was taught in first semester of 2018 sessio by Dr S.A*. Through the GUI, user can equally specify recommended text book for a course, and prerequisite relationship between courses.

The information collected from user through the user interface is used to formulate the knowledge graph triples. The triples are finally transformed into RDF statements [15] and stored in xml file using Jena [16]. The xml schema in fig.4 shows course information in a tree format. This information are transformed into rdf statements. The snippet of the course knowledge graph automatically constructed in rdf format is shown in fig.5. RDF format is not the only way to represent knowledge graph. For example, FB15k-237 and NELL-955 knowledge graphs which are used as datasets to evaluate research works on knowledge graph completion [17]–[19] are plain text files and the information is not enclosed in any tag. However, in this work we settled for RDF format because it avail us the opportunity to easily encode relations of interest and other properties (attributes) of entities. Furthermore, manipulation (querying) of the graph is easier.

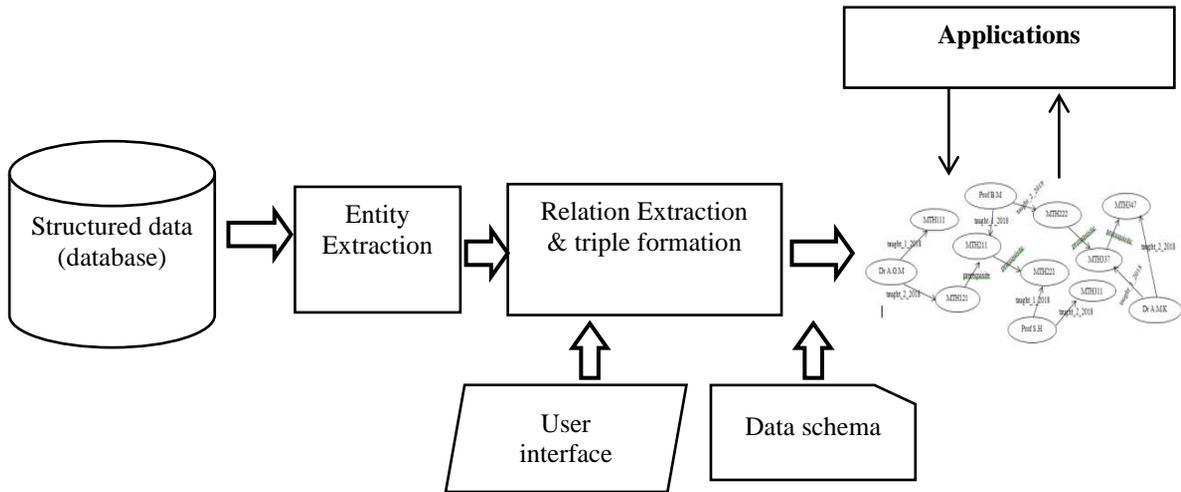


Fig.3. System architecture

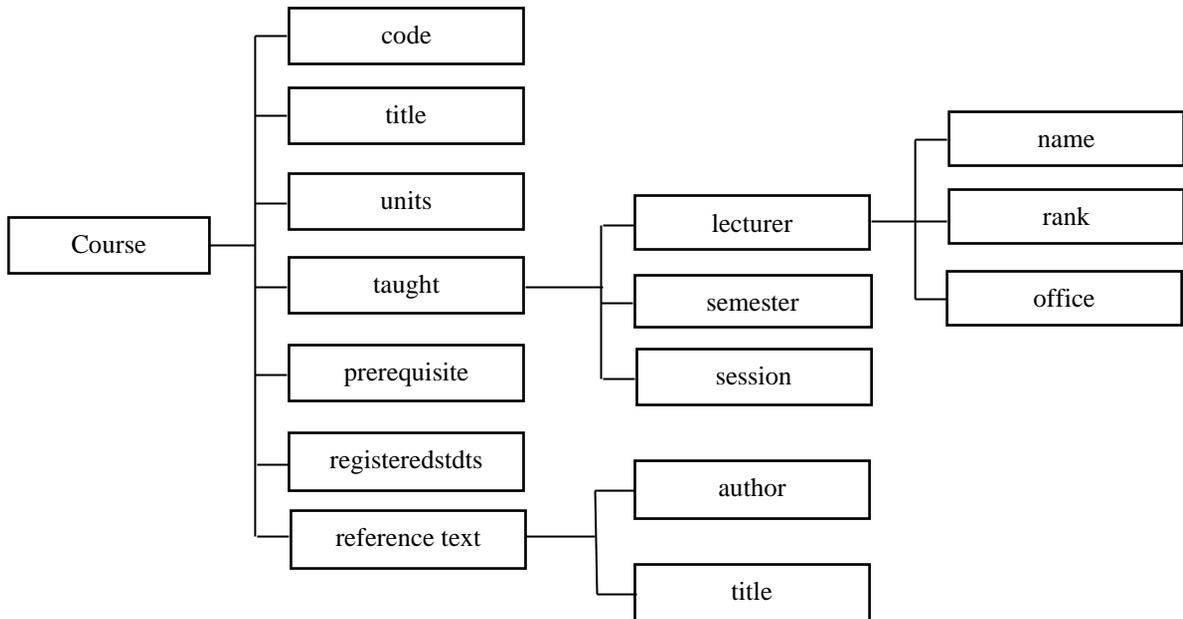


Fig.4. xml diagram (schema) showing the nodes in tree format

As stated earlier, the KG is in RDF format. Fig.4 shows one of the rdf triple from the KG and illustrates of how the knowledge graph triples can be deduced. For example, KG triple (*CS323*, *taught\_in\_2\_2018-2019*, *Dr S.A*) – Dr S.A taught CS323 in the second semester of 2018-2019 can be deduced. Similarly, triple (*CS312*, *prerequisite*, *CS323*) – CS312 is the prerequisite of CS323 can be deduced.

```

-----
<rdf:Description rdf:about="http://dms.atbu.edu.ng/courses#CS323">
  <j.0:prerequisite rdf:resource="http://dms.atbu.edu.ng/courses#CS312"/>
  <j.0:taught_in_2_2018-2019>
    <rdf:Description rdf:about="http://dms.atbu.edu.ng/staff#Dr S.A">
      <j.0:office>'BLK M005'</j.0:office>
      <j.0:rank>'Senior Lecturer'</j.0:rank>
      <j.0:phone>'0802715463'</j.0:phone>
    </rdf:Description>
  </j.0:taught_in_2_2018-2019>
  <j.0:registeredStudents>'72'</j.0:registeredStudents>
  <j.0:title>'HUMAN COMPUTER INTERACTION'</j.0:title>
  <j.0:units>'2'</j.0:units>
</rdf:Description>

```

Fig.5. snippet of the KG triples stored as rdf statements.

### 4.3 Querying the Courses Knowledge Graph

The essence of constructing the courses knowledge graph is to support further processing related to management of courses at departmental level. Once the graph is constructed, it can be queried to retrieve the desired information. Jena [16] provides functionality of querying RDF documents to retrieve subjects (entities in our case), predicates, and objects (mostly entities in our case). Note that in the snippet of RDF document in fig.5, the entity names are part of the **Uniform Resource Identifier (URI)**. However, the complete uri is not needed in the application. Only the last the characters after # symbol (i.e, entity name) are needed. So we have to split the **URI** and extract the entity name. Fortunately, Java provides a lot of functionalities for string manipulation.

## 5. CASE STUDY

To test the system, data (courses and lecturers) of Department of Mathematical Sciences of ATBU, Bauchi was used. The practice at the department is that at the beginning of every semester, every academic staff given course allocation schedule which indicate the course(s) a staff is expected to teach. In line with that practice the software first allow user to build the courses graph. Building the graph involves assigning courses to lecturers, specifying prerequisite relationship between courses, and specifying recommended text books for courses. The most important part of building the courses knowledge graph is assigning of courses to lecturers. When the graph is constructed, user can: (1) Print course allocation schedule for a particular semester which will be distributed to staff and place on notice boards. (2) Obtain answers to some questions related to a course. – eg, *Who taught courseX in first semester 2014? How many courses did lecturerX taught in first semester 2015?*

## 6. DISCUSSION AND CONCLUSION

The system functions well in delivering the two functional requirements mentioned above. Contrary to the current system which is ‘auto-manual’, the software proposed in this paper can maintain course allocation information of the department for several years. Although the software is meant to be used by one department, its scope can be extended in the future to cover other departments of the university. Such enhanced system should be configured in way to accommodate many different users representing departments, and be managed

centrally by an administrator. Such enhanced software would enable the university management particularly directorate of academic planning to easily have access to basic information on teaching to support decision making.

In this paper, automated approach for handling the task of course allocation and management is proposed. The system uses knowledge graph as model for data storage. Rather than storing the entire data in tables of relational database, the system stores data in a knowledge graph which is in RDF/XML format and queries it to support intelligent knowledge services. Incorporating knowledge graph as model for data storage into the system enable the software to easily accomplish tasks similar to Question Answering. Furthermore, it offers advantages such as flexibility – no sql queries, no joining of sql queries, security – no fear of SQL Injection, or hacking of database.

## ACKNOWLEDGEMENTS

We are grateful to anonymous reviewers for their vital comments and suggestions. We thank Rabiou Madaki of the Department of Mathematical Sciences ATBU Bauchi for helping us obtained the data used in testing the system. The first author is sponsored by Nigerian Tertiary Education Trust Fund (TetFund).

## References

- [1] P. Chen, Y. Lu, V. W. Zheng, X. Chen, and B. Yang, “KnowEdu : A System to Construct Knowledge Graph for Education,” *IEEE*, vol. 6, pp. 31553–31563, 2018.
- [2] Y. Chi, Y. Qin, R. Song, and H. Xu, “Knowledge Graph in Smart Education : A Case Study of Entrepreneurship Scientific Publication Management,” *MDPI/sustainability*, 2018.
- [3] H. Paulheim, “Knowledge Graph Refinement : A Survey of Approaches and Evaluation Methods,” *Semant. Web*, vol. 1, 2016.
- [4] L. Ehrlinger and W. Wöß, “Towards a Definition of Knowledge Graphs,” in *12th international conference on semantic systems SEMANTiCS*, 2016, vol. 1695, pp. 1–4.
- [5] N. Noy, Y. Gao, A. Naratanan, A. Patterson, and J. Taylor, “Industry-scale Knowledge Graphs Lessons and Challenges,” *QUEUE*, *ACM*, pp. 1–28, 2019.
- [6] Q. Cong, Z. Feng, F. Li, L. Zhang, G. Rao, and C. Tao, “Constructing Biomedical Knowledge Graph Based on SemMedDB and Linked Open Data,” in *2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, 2018, pp. 1628–1631.
- [7] Y. Wang, K. Zhang, Q. Dai, C. Peng, and K. Zhao, “Construction and Application of Knowledge Graph in Full-service Unified Data Center of Electric Power System.,” in *IMMAEE*, 2018.
- [8] M. Zhao et al., “Construction of an Industrial Knowledge Graph for Unstructured Chinese Text Learning,” *MDPI/applied Sci.*, pp. 1–22, 2019.
- [9] T. Yu et al., “Knowledge graph for TCM health preservation : Design , construction , and applications,” *Artif. Intell. Med. Elsevier*, vol. 77, pp. 48–52, 2017.
- [10] K. Sun, Y. Liu, Z. Guo, and C. Wang, “Visualization for Knowledge Graph Based on Educational Data,” *Int. J. Softw. Informatics*, vol. 10, no. 3, pp. 1–13, 2016.
- [11] D. S. Chaplot, Y. Yang, J. Carbonell, and K. R. Koedinger, “Data-driven Automated Induction of Prerequisite Structure Graphs,” in *9th Interntional Conference on Educational Data Mining*, 2016, pp. 318–323.
- [12] C. Liang, J. Ye, Z. Wu, B. Pursel, and C. L. Giles, “Recovering Concept Prerequisite Relations from University Course Dependencies,” in *7th Symposium on Educational Advances in Artificial Intelligence (EAAI)*, 2017, pp. 4786–4791.

- [13] Y. Yang, H. Liu, J. Carbonell, and W. Ma, "Concept Graph Learning from Educational Data," in WSDM, 2015, pp. 1–1.
- [14] H. Liu, W. Ma, Y. Yang, and J. Carbonell, "Learning Concept Graphs from Online Educational Data," *J. Artif. Intell. Res.*, vol. 55, pp. 1059–1090, 2016.
- [15] S. Decker, P. Mitra, and S. Melnik, "Framework for the Semantic Web: An RDF Tutorial," *IEEE Internet Comput.*, pp. 68–73, 2000.
- [16] Apache Jena, "Apache Jena," 2019. [Online]. Available: <https://jena.apache.org>. [Accessed: 22-Nov-2019].
- [17] W. Chen, W. Xiong, X. Yan, and W. Y. Wang, "Variational Knowledge Graph Reasoning," in NAACL-HLT, 2018, pp. 1823–1832.
- [18] Y. Lin, Z. Liu, M. Sun, Y. Liu, and X. Zhu, "Learning Entity and Relation Embeddings for Knowledge Graph Completion," in 29th AAAI Conference on Artificial Intelligence, 2015, pp. 2181–2187.
- [19] W. Xiong, T. Hoang, and W. Y. Wang, "DeepPath: A Reinforcement Learning Method for Knowledge Graph Reasoning," in Conference on Empirical Methods in Natural Language Processing, 2017, pp. 564–573.

**Author's Profile**

**Ismail Aliyu** is a Doctorate student at the Department of Computer Science, A.B.U Zaria. He obtained B.Tech degree in Computer Science from Abubakar Tafawa Balewa University Bauchi, Nigeria, and MSc in Advanced Software Engineering from University of Sheffield, UK in 2013. His research interests include Information Extraction, a sub-area of NLP, Machine Learning, Semantic Web, and Knowledge Representation.



**Armand F. Donfack Kana** received the B.Sc. degree in Computer Science from University of Ilorin, Nigeria, M.Sc and Ph.D. degrees in Computer Science from University of Ibadan, Nigeria. He is currently a Senior Lecturer in Computer Science at the Department of Computer Science, Ahmadu Bello University, Zaria, Nigeria. His current research interests include Knowledge representation and reasoning, Formal Ontologies and Soft computing.



**Dr Salisu's** research area is in Knowledge Representation and Reasoning, Artificial Intelligence. He obtained his PhD in Computer Science from A.B.U Zaria in December 2016, and he is presently a lecturer in the same institution. He has published many papers both locally and internationally.

**How to cite this paper:** Ismail Aliyu, A. F. D. Kana, Salisu Aliyu. " Development of Knowledge Graph for University Courses Management ", International Journal of Education and Management Engineering(IJEME), Vol.10, No.2, pp.1-10, 2020.DOI: 10.5815/ijeme.2020.02.01