KGC19: KNOWLEDGE GRAPH OF COVID-19 SCHOLARLY ARTICLES FOR ENHANCED INFORMATION RETRIEVAL & RECOMMENDER SYSTEM.

M Tech Dissertation

Submitted in

partial fulfillment of the requirement for the degree of

MASTER OF TECHNOLOGY

In

Computer Engineering

By

Warisahmed Nashrullah Bunglawala

200303201002

Under the supervision of

Dr. Jaimeel shah

Prof. Darshana Parmar

April 2022

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

PARUL INSTITUTE OF ENGINEERING & TECHNOLOGY

FACULTY OF ENGINEERING & TECHNOLOGY

PARUL UNIVERSITY

P.O. Limda – 391 760, GUJARAT, INDIA.

CERTIFICATE

This is to certify that the work contained in this dissertation thesis entitled "**KGC19: KNOWLEDGE GRAPH OF COVID-19 SCHOLARLY ARTICLES FOR ENHANCED INFORMATION RETRIEVAL & RECOMMENDER SYSTEM".** submitted by **Warisahmed Nashrullah Bunglawala**, **200303201002.** studying at **Parul Institute of Engineering &Technology** for the Phase - II Dissertation (M. Tech- COMPUTER ENGINEERING) is absolutely based on his own work carried out under our supervision and that this work/thesis has not been submitted elsewhere for any degree/diploma. To our satisfaction, this work is approved for the Phase - II Dissertation (M. Tech- COMPUTER ENGINEERING) External exam. The extent of plagiarism does not exceed the permissible limit laid down by the University.

Date:

Faculty Supervisors: **Signature:**

Dr. Prof. Jaimeel Shah

Prof. Darshana Parmar

Dr. Amit Barve	**Prof. Pratik Patel**	**Dr. Chintan Thacker**	**Dr. Vipul Vekariya**
HOD, CSE, PIET.	Department PG Coordinator	Department PG Coordinator	Principal, PIET.

THESIS APPROVAL CERTIFICATE

This is to certify that research work embodied in this dissertation thesis entitled "**KGC19: KNOWLEDGE GRAPH OF COVID-19 SCHOLARLY ARTICLES FOR ENHANCED INFORMATION RETRIEVAL & RECOMMENDER SYSTEM"** carried out by **Mr. Warisahmed Nashrullah Bunglawala, 200303201002** at **Parul Institute of Engineering & Technology** is approved for the degree of M. Tech with specialization of **Computer Engineering** by Parul University.

Date:

Place:

External Examiners" Sign and Name:

1)

2)

PAPER PUBLICATION CERTIFICATE

This is to certify that research work embodied in this dissertation thesis entitled **"KGC19: KNOWLEDGE GRAPH OF COVID-19 SCHOLARLY ARTICLES FOR ENHANCED INFORMATION RETRIEVAL & RECOMMENDER SYSTEM"** carried out by Mr. **Warisahmed Nashrullah Bunglawala, 200303201002** at **Parul Institute of Engineering &Technology** for partial fulfillment of M.Tech degree to be awarded by Parul University, has accepted article entitled "**KGC19: KNOWLEDGE GRAPH OF COVID-19 SCHOLARLY ARTICLES FOR ENHANCED INFORMATION RETRIEVAL & RECOMMENDER SYSTEM"** for publication by the **International Conference on Expert Clouds and Applications (ICOECA 2022)**, Paper ID:

ISSN: 2367-3370. Page Number: 230-243. at **Bengaluru, INDIA** during **3-4, Feb 2022.**

Date: 3-4 February 2022

Place: Bengaluru, INDIA. (Online Mode)

Warisahmed Bunglawala.	Dr. Prof. Jaimeel Shah.	Prof. Darshana Parmar.
Student Name & signature	**Signature and Name of Faculty Supervisor**	

ACKNOWLEDGEMENT

Behind any major work undertaken by an individual there lies the contribution of the people who helped him to cross all the hurdles to achieve his goal. It gives me the immense pleasure to express my sense of sincere gratitude towards my respected supervisor Dr. Jaimeel Shah and co-supervisor Prof. Darshana Parmar for their persistent, outstanding, invaluable co-operation and guidance. It is my achievement to be guided under them. They are a constant source of encouragement and momentum that any intricacy becomes simple. I gained a lot of invaluable guidance and prompt suggestions from them during entire project work. I will be indebted of them forever and I take pride to work under them. I also express my deep sense of regards and thanks to Prof. Pratik Patel (Associate Professor) and Faculty Representative (FR). And also, to Dr. Amit Barve, (Associate Professor) and Head of CSE Engineering Department. I feel very privileged to have had their precious advices, guidance and leadership. Adding to it this process was taken further for experimental purpose by testing it on super computer in order to test how accurate will the output be generated if done so. This was done on Super Computer "PARAMSHAVAK" which was received from GUJCOST, Government of Gujarat with the specifications 96gb ram 16 TB rom, processor "Intel® Xeon® Gold 6145" with 16 GB NVIDIA QUADRO RTX 5000 Graphic card. I am truly grateful to the environment and resources that are provided through or by university. Last but not the least, my humble thanks to the Almighty God.

Warisahmed Bunglawala

200303201002

PARUL UNIVERSITY, FACULTY OF ENGINEERING & TECHNOLOGY

Computer Science & Engineering Department

M.Tech Computer Engineering

Proposal for Dissertation Title

Title: KGC19: KNOWLEDGE GRAPH OF COVID-19 SCHOLARLY ARTICLES FOR ENHANCED INFORMATION RETRIEVAL & RECOMMENDER SYSTEM".

Submitted By:

Name of the student: BUNGLAWALA WARISAHMED NASHRULLAH.

Enrollment Number: 200303201002

Supervisor:

1) **Dr. Prof. Jaimeel Shah**
2) **Prof. Darshana Parmar**

Abstract:

Despite several efforts to make RS more efficient and personalized, it still faces issues with traditional systems along with data's inability of interconnectedness. And, because it is intended to be read only by humans, it cannot be processed or interpreted by a computer. Ontology facilitates knowledge sharing, reuse, communication, collaboration, and the construction of knowledge-rich and intensive systems. Adding semantically empowered techniques to recommender systems can significantly improve the overall quality of recommendations. There has been a lot of interest in creating recommendations using knowledge graphs as a side information source. Through this, we not only overcome the issues of traditional RS but also provide a flexible structure that naturally allows the integration of multiple entities all together. It is also helpful in explaining the recommended items. So, we proposed our very own work as a KGC19: knowledge graph of Covid-19 Scholarly Articles. We mentioned different use cases of our knowledge graph, which is majorly focused on information retrieval and recommender systems using a SPARQL and embedding-based approach. The proposed system has the potential to add significant value to the fields of semantic web and knowledge base systems.

Warisahmed Bunglawala.	Dr. Prof. Jaimeel Shah.	Prof. Darshana Parmar.
Student Name & signature	**Signature and Name of Faculty Supervisor**	

TABLE OF CONTENT

LIST OF ABBREVIATION

Symbol Name	Abbreviation
KG	Knowledge Graph
RS	Recommendation System
RDF	Resource Description Framework
W3C	World Wide Web Consortium
FOAF	Friend of a Friend (Ontology)
OWL	Web Ontology Language
KGE	Knowledge Graph Embedding
SPARQL	Simple Protocol and RDF Query Language
Cord-19	The Covid-19 Open Research Dataset
CBOW	Continuous Bag of Words
NLP	Natural Language Processing
MR	Mean Rank
MRR	Mean Reciprocal Rank
GCN	Graph Convolutional Networks
DKN	Deep Knowledge Network
ORKG	Open Research Knowledge Graph
LOD	Linked Open Data
CSV	Comma Separate Value
JSON	JavaScript Object Notation
JSONLD	JSON for Linked Data
KGC19	Knowledge Graph of COVID-19 Scholarly Articles
KGC19:	<http://www.semanticweb.org/warisahmed/ontologies/KGC19#>

LIST OF FIGURES AND TABLES

CHAPTER 1
INTRODUCTION

The advancement in digital technology has exploded the data tremendously. Social site such as Twitter, Instagram and Facebook are significant source of data generation. Moreover, question answering sites like Stack overflow, Quora and Reddit are also contributing in it. Furthermore, research and publications are also increasing in day by day [3]. with this advancement of the data, it brings advantages and disadvantages both. Advantage can be described as we are having a lot of data available within seconds. Disadvantage can be stated as loss of data or we can say abundance of data has increased, and due to that we are unable to get most relevant and required information.

To overcome this disadvantage recommendation system has been developed and still in research trend. A recommendation system is a type of information filtering system which attempts to predict a user's "rating" or "preference" for an item. Task of recommendation system can be defined in two parts (I) estimating a value of prediction for item. (ii) Recommending items to users [3]. There are various types of approaches available to achieve this task and most common or popular approaches are Content based Recommendation system, Collaborative Recommendation system. And by combining those techniques Hybrid Recommendation system is also developed [3][5][36]. However, Recommendation systems are in need of continuous modification due to exponential increment in data and knowledge.

In recent years, introducing a knowledge graph as a side information in recommendation system has attracted a lot of researchers and organizations [8]. First Knowledge graph is introduced by goggle in May 2012 [11]. There are lot of definition on knowledge graph is available according to its usage. However, none of them has become standard definition. as a term "Knowledge Graph" have different meaning according to usage. For simple understanding Knowledge graph can be define as a heterogeneous graph, where node represent entities and edge represent relationship between those entities [16][17]. Advantage of knowledge graph is its flexible basic graph structure and it provides the model of how everything is connected.

Talking about the latest pandemic COVID-19 has claimed so many lives worldwide. It also boosted the demand for tools that allow academics to search large scientific corpora for specific information, visualise data relationships, and uncover related data. Due to the

requirement for information retrieval connected to scholarly literature on Covid-19, several specialized search engines have been built. “Sketch Engine COVID-19”, “Sinequa COVID-19 Intelligent Search”, “Microsoft's CORD19 Search”, and “Amazon's CORD19” are examples of search engines. However, this search engines return thousands of search result that overlooked the inherent relationships like citation and subject topics [6]. Also, they do not provide the tool to visualize relationships, which can be beneficial for knowledge discovery. So, we need the system that can be specifically used for knowledge discovery and information retrieval. Also, we need to use every unique data that we can gather, and in situation of Covid-19 pandemic Scientific data can be very useful.

So let us understand some of the term like “Recommender System”, “Knowledge Graph”, “Ontology”, “Knowledge Graph Embedding” (KGE) and “SPARQL” in more details which is related to this particular work that we have carried out.

1.1 Recommender System

A recommender system is one that is designed to make recommendations to the user depending on a variety of parameters. These systems forecast the most likely product that users will buy and that they will be interested in [7]. Recommender system can be used where we want to recommend something to users, it can be article, E-commerce products, Movies, anything based on the type of application or services. The main component of recommender systems is its algorithm., which can be categories in many categories. From this category mainly mentioned is “Content Based Recommender System”, “Collaborative Recommender System”, “Knowledge Based Recommender System” and “Hybrid Recommender system” [7][12]. All of the categories that are mainly uses by the organization explained future in this document.

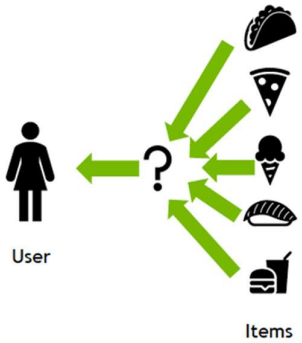


Figure 1: Recommendation System [13]

1.1.1 Collaborative Recommender System

In Collaborative Recommendation, A new item that is consumed by similar users is suggested by the recommendation system to a user [3]. Its aggregated item ratings or recommendations identify commonalities among users based on their ratings and provide new recommendations based on the cross comparisons. Collaborative filtering is founded on the idea that people who agreed in the past will agree again in the future, and that they will like objects that are similar to those they liked previously [36]. Advantage of this system is no domain knowledge is required and Serendipity. Disadvantage is this system can suffer from cold start problem. Also, it is hard to include side features in this model for example in movie recommendation, side features might include country or age.

1.1.2 Content based Recommender System

In Content Based Recommendation, the recommendation system recommends a new item based on their contents and attributes. In simple words Some of the user-related features could be explicitly provided by the user [36]. For example, let's say a user uses a play store and selects entertainment menu. While other features can be implicitly based on the previously installed apps. So, in content-based recommendation, model should recommend the items relevant to this particular user [33]. Note that recommendation is based on particular user no other user information is used. And to do this task a similarity metric can be applied such as dot product [33]. Advantage of this system can be defined as it does not uses other users' interest and recommendation are specific to particular user. It can capture very specific interest of user so that it can recommend a very personalized item. While Disadvantage can be stated as since the feature representation of the items are hand-engineered to some extent, this technique requires a lot of domain knowledge. Also, model has limited ability to expand on the base of users existing interest.

1.1.3 Knowledge based Recommender System

The Knowledge Based RS makes recommendations based on inferences about a user's needs and preferences [34]. It is based on functional knowledge: they understand how a specific item fits a specific user demand and can thus reason about the connection between that need as well as possible recommendation. Knowledge-based recommender systems can be very useful to combine with other types of recommender systems. They can be used to solve the cold start problem in the short term, then switched to "collaborative filtering" or "content-based systems" after enough ratings have been collected [32]. A potential knowledge

acquisition bottleneck, driven by the necessity to define recommended knowledge explicitly, is a related disadvantage. [19].

1.1.4 Hybrid Recommender System

Hybrid Recommender System can't be defined in particular definition but it can be described as a combination of any of the two or more system that suits for an application. This system is mostly adopted by companies or organization as it combines the advantages of two systems and eliminates the disadvantage which exist if we use only one system. There are no standards available for hybrid recommendation system but according to article on [12] they have mentioned three ways out of several way from which we can implement hybrid system. Those three ways are Weighted hybrid recommender, Switching hybrid recommender and Mixed hybrid recommender. Apart from this three there are several research papers that mentioned their own hybrid model. One of the papers included in literature survey has mentioned their own technique for data modelling and computation using graph structure [5].

1.2 Knowledge Graph

Some have tried to give the definition of knowledge graph but none of them has become a standard definition. As a term "Knowledge Graph" can have different views. So instead of the definition, characteristics of knowledge graph can be presented as: 1) It primarily describes real-world things and their interrelationships in the form of a graph. 2) In a schema, defines the classes and characteristics of entities. 3) Allows for the possible interconnection of arbitrary things. 4) Covers a wide range of topics [10].

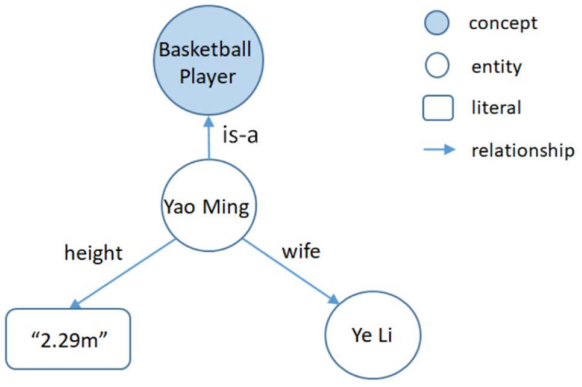


Figure 2: Knowledge Graph [10]

As shown in the figure entity is a thing present in a real world while concept is something define as a collection of individuals which have a same characteristic. Literal can be defined as a nothing but a specific value or strings of some relations. And the edge between entity or concepts can be define as a relation. For example, Yao Ming is individual entity and Basketball is a concept as so many players out there play basketball such as Kobe Bryant and Stephan Curry. While yao ming height can be define as a ”2.29m”, so this specific value can be said literal and on other hand yao ming have a wife ye li so wife is a relation between those two entities.

Important point to note here is that Knowledge can be of two types in KG, one is “schematic Knowledge” and other one is called “Factual Knowledge” [11]. Schematic knowledge consist of triple about concepts and properties, for example (Asian Country, subClassOf(), Country). While factual knowledge consists of statements about instances, for example triples given in the above graph is all factual knowledge. In most of the KG factual knowledge is in large size and small amount of knowledge is schematic knowledge. Knowledge graph have their logical foundation based on the ontology languages such as “Resource Description Framework” (RDF) and the “Ontology Web Language” (OWL), which are recommended creation from W3C. RDF can be used to describe the Factual Knowledge and OWL can be used to convey comprehensive and complicated entity knowledge, properties and relations. Owl can represent schematic and factual both types of knowledge.

1.3 Ontology

Ontologies are the foundation of a knowledge graph's formal semantics. They can be thought of as the graph's data schema. It ensures that data and its meanings are understood by everyone [39]. So, ontology can be understood as it explains what is in the knowledge graph without actually seeing or observing knowledge graph. Ontologies are frameworks for representing shareable and reusable knowledge across domains. They are the base for modelling high-quality, linked, and consistent data because of their capacity to describe relationships and high interconnectivity. The Web Ontology Language (OWL) is a family of knowledge representation languages for authoring ontologies. So, considering the ontology as a main logical foundation knowledge graph can be created in one of the two ways one is bottom-up approach and second one is top-down approach [4].

1.3.1 Bottom-Up Approach

We extract knowledge instances from "Linked Open Data" (LOD) or other knowledge resources using a bottom-up technique. The top-level ontologism is built using knowledge instances to create the entire KGs following knowledge fusing the completed instances [4]. Bottom-up approach of KG is and iterative update process, which includes knowledge acquisition, knowledge fusion, knowledge storage and retrieval. for better understanding let us look at the following architecture of bottom-up approach.

As shown in the figure-3 primary source of knowledge acquisition includes structured data, unstructured data and semi structured data. Knowledge extraction consist of attribute, relations and entity extraction. After that knowledge fusion can be define as an iterative process in which we construct the ontology and constantly evaluate it for the better quality. For knowledge graph storage and retrieval NoSQL databases are more popular. And further retrieval and visualization can also be done.

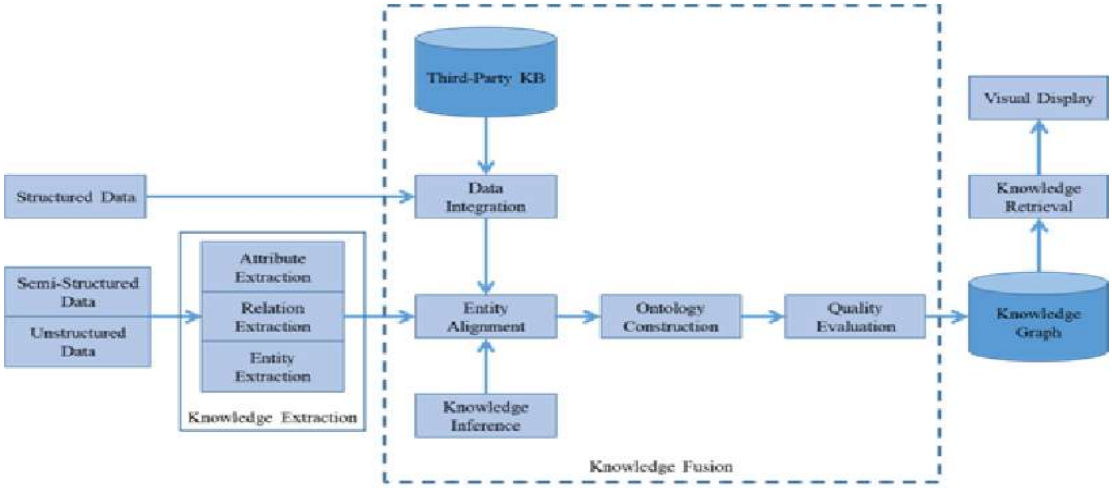


Figure 3: Architecture of Knowledge graph: Bottom-up Approach [4]

In knowledge extraction we can use all types of data whether it is from structured, semi structured or unstructured data source. And extracted knowledge is usually presented in machine readable formats such as RDF and JsonLD. While we can extract knowledge from any sources such a website or any record and datasets available, now a days most of the instances extracted from DBpedia or Yago and Wikipedia. For semi-structured and

unstructured data sources we need entity extraction, relation extraction and attribute extraction. Entity extraction is nothing but discovering the entity from wide variety of the data and try to classify them in the predefined categories such as person, place or location. After entity extraction relations among those entities are analyzed to conceptual extract relations. The attribute extraction is to define the intentional semantics of the entity and it is important for defining the concepts of entity more clearly. There are many tools available for knowledge extraction depending on the needs and functions, some of them are Stanford NER, OpenNLP, AIDA, Open Calais and Wikimeta.

In knowledge fusion the goal is to realize entity alignment and ontology construction, and this is an iterative process. Entity alignment is one step in this process and it is also known as entity matching. Purpose of entity matching is to judge whether or not different entities points to the same object of the real world. Point to note here is that entity alignment usually relies on the external sources such as manually developed corpus or Wikipedia links. After that ontology construction and evaluation step is there. We design the ontology as well as other works such as taxonomy, hierarchical structure, metadata, and other data sources. To ensure the KG's quality, general ontologies such as FOAF and general metadata from schema.org are required. In terms of KG storage, it is often saved in a NoSQL database. There are two basic storage types: RDF-based storage and graph database storage. The advantage of RDF based is that the efficiency of query and merge-join of triple patterns is good. However better query results request huge cost of storage space. Some popular RDF based data storage is 4store, RDF Store, TripleT and so on, most of the native storage system provides SPARQL or similar like query languages. On other hand graph-based storage have advantage that they themselves provide the perfect graph query languages and support a variety of graph mining algorithms. However, they do have disadvantages such as slow update of knowledge, high maintenance cost and in-consistence of distributed knowledge. The typical graph database Neo4j is popular as it is open source and provides native graph storage. While selecting a storage for KG the primary requirement of the large-scale knowledge graphs the following points are important, the underlying storage should be scalable, data segmentation can be done as required, cache and index are use timely and it should handle the large volume of knowledge graph effectively [4].

1.3.2 Top-Down Approach

Top-down approach is mostly the same for KG creation but only differs in terms of creating ontologies. While bottom-up approach for ontology starts with instances and leaves of the hierarchy, with subsequent group of more general concepts. A top-down approach focuses on the general concepts first and then follows on the leaves [20]. For example, let's say we want to create an ontology for food then top-down approach will follow basic concepts first like veg and non-veg while bottom-up approach will follow from the leaves like cabbage, tomato then go for general concepts like veg or non-veg. However, there is a hybrid concept is also developed in which we can combine these two concepts [20]. for better understanding let us consider the diagram [Figure 4] given below mentioned by Xiaogang Ma in his paper [21]. As discussed in the top-down approach we have to focus on the upper-level concepts first and then we have to go further towards instances. First step is Domain and as mentioned in diagram in this step we will have to focus on the subject areas and needs of it. After that we will have to move towards conceptual model so that we can collect and define major entities and categories. For example, mentioned above let us say we want to create a food related ontology then we first have to identify the areas and after that we can define major entities such as veg or non-veg and then we will be able to identify the interrelation and further concepts. After that author mentioned the next step as logical model and physical model, we which will add logical representation and assertions. Implementation and update will be the same as mentioned in bottom-up approach, in this step we will actually create the knowledge graph and as it can be iterative process we will constantly update according to the need and availability.

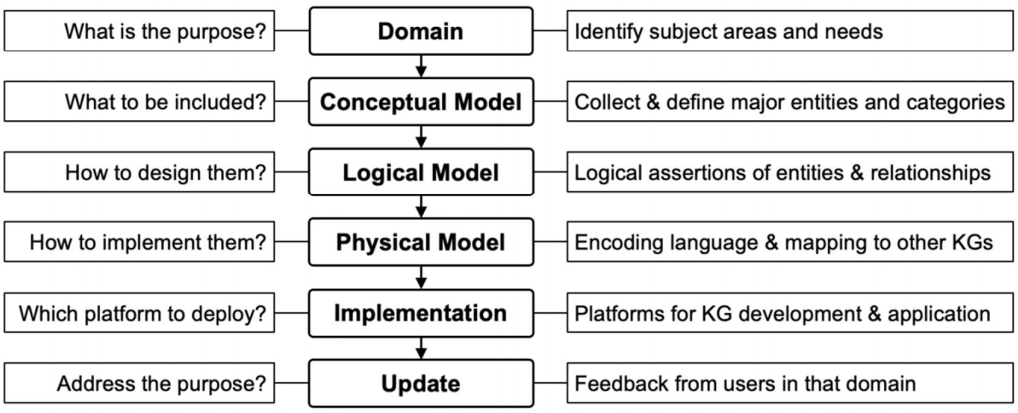


Figure 4: Architecture of Knowledge graph: top-down Approach [21]

For further understanding of Top-down approach, in paper [22] authors mentioned top-down approach as first the pattern layer of the knowledge graph is defined through the construction of domain ontology. After those entities, attributes and relationships between entities are identified. And at last, the knowledge graph is created. In paper, authors also mentioned the three basic steps to follow for ontology that are, 1) Define the class and its hierarchical structure. 2) Define the attributes of the class. 3) Define the relation between the classes. Although the bottom-up approach is able to process a large number of datasets and quickly build a big KG, a remaining challenge is the precise logical representation and assertions for the entities and relationships in the resulting KG. Very often, they still need to be specified by the domain experts and knowledge engineers, where existing KGs can be reused [21]. None of these two methods is better than each other it's depends on the view of the developer. If the developer has a better systematic top-down view of the domain then it may be easier for them to follow the top-down approach. However, the combine approach is easier for most of the ontology developers, since it uses the concept "In the middle" which tends to be more descriptive concept in the domain ontology.

1.4 Knowledge Graph Embeddings (KGE)

Knowledge graph embeddings are low-dimensional representations of the items and relations in a knowledge graph (KGEs). They give a generalizable context for inferring relationships throughout the entire KG. The embeddings of knowledge graphs are constructed in such a way that they fulfill certain characteristics, such as adhering to a specific KGE model. These KGE models define multiple score functions in the low-dimensional embedding space that assess the distance between two items relative to their relation type. These score functions are used to train the KGE models so that entities linked by relations are close together, while entities not linked are far apart.

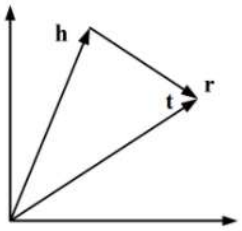
Popular Traditional model of KGE include TransE, TransR, ComplEx and RotatE. These models use as input a vector representation of entity and predicate embeddings in a triple. Embedding combined using scoring function to generate a score. On other hand we can also have convolutional model which include ConvKB and ConvE. These are convolutional model and they convert the embeddings to an image like representation and performs convolutions on them. So, we can think of it as 2 or 3 channel images where each channel represents S, P and O features. For the larger dataset such as in our case traditional models are best fitted to use so we used them in our work.

1.4.1 TransE

TransE is a representative translational distance model that depicts entities and relations as vectors in the same semantic space of dimension Rd, where d is the dimension of the target space with decreased dimension. In the source space, a fact is expressed as a triplet (h,r,t), where h is for head, r is relation, and t for tail. The relationship is read as a translation vector, resulting in a short distance between the embedded items connected by relation r.Let use (h,r,t) represent the triple, and the key idea of TransE is [40]:

$$t \approx h + r$$

Its optimization goal is to maximize the distance between positive and negative sampling data margin loss. The loss function is as follows:



$$L(h,r,t) = max(0, d_{pos} - d_{neg} + margin)$$

$$d = ||h + r - t||$$

Figure 5: Idea of TransE & Loss Function [40]

1.5 SPARQL

SPARQL is widely used query language for knowledge retrieval and almost all the large-scale knowledge graph system provides SPARQL query endpoint. SPARQL provides output in 'JSON', 'JSON-LD', 'XML', 'RDF/XML', 'RDF/N3', 'CSV' etc [4]. and almost all the outputs are in machine readable form. With machine readable form we require visualization tools also, visualization using browser are most common as some of the formats of query results are in text. Popular tools are graphdb, IsaViz, RDF Gravity, DBpedia Mobil and Open Link Data Explorer. Important key point is knowledge retrieval are mostly logic rules as ontology is based on description logic.

For more information regarding SPARQL and its functions please refer the documentation available here: SPARQL Query Language for RDF (w3.org)

CHAPTER 2

AIM AND OBJECTIVE OF THE STUDY

The motivation behind this study is to give a jump start and detailed information to researchers, who wants to pursue their research in this particular field. That's why study not only include basic information regarding system but also gives an overview on basic steps to follow for building the system. And study also include information available on datasets and tools that can be used. Also, this type of system can not only be used for ecommerce. But it can be majorly beneficial to the situation like Covid-19 where we want related data available within seconds for data discovery or in case of emergency information retrieval.

So major aim behind this study is to build an information retrieval and recommender system for situation like Covid-19 which is enriched with knowledge. And following are the research objective of this study.

- Building our own Knowledge graph KGC19 based on Cord-19 dataset to overcome traditional system & help semantic web.
- Faster information retrieval than traditional systems.
- Using KGE techniques to enhance recommendation from KG.
- Linked Data & Semantic web contribution.
- To increase Hits@K & MRR (Mean Reciprocal Rank) on our own KG.

CHAPTER 3
LITERATURE REVIEW

3.1 Literature Review

1) "Cone-KG: A Semantic Knowledge Graph with News Content and Social Context for Studying Covid-19 News Articles on Social Media", Feras Al-Obeidat, Oluwasegun Adedugbe, Anoud Bani Hani, Elhadj Benkhelifa, Munir Majdalawieh, IEEE 12-2020. **[1]**

In this paper authors used a 35GB of data from three raw dataset like aylien covid, IEEE dataport, covid-19 public media dataset then they performed a data extraction, cleaning and processing for doing so they first created a parser to extract the data and then they convert that to RDF format using Rdfizer tool. using SPARQL they can easily integrate and query over that data. after that they created a knowledge graph based on the data using fandet ontology and RDFizer tool. at last, they used graphDB to explore semantic graph dataset and then they performed the analysis on that so hidden insight can be find.

Outcome: From this paper we the main idea behind the concept of our work. Just like cone-KG focused on news article we have focused on scholarly article and information retrieval. Also, we learned about the how some data needs to be cleaned and what is the approach we should follow while building this kind of system. important of the use of SPARQL, GraphDB can also be understand.

2) "Open Research Knowledge Graph: Next Generation Infrastructure for Semantic Scholarly Knowledge", Mohamad Yaser Jaradeh, Allard Oelen, Et al. Research Gate 06-01-2020. **[2]**

In this paper authors mentioned their work or we can say project as ORKG and the concept behind this is to make scholarly article machine readable and easily manageable for that they have used labelled property graph, triple store and relational database storage technology. Authors Provided the features like literature comparison, contribution similarity and linking scholarly knowledge to another KG. Also, in evaluation time needed to perform state of the art comparison with 2 to 8 research contribution using the baseline and ORKG is very promising as ORKG reached 0.0204 second.

Outcome: Some of the concept regarding the enhancement in the scholarly article retrieval system can be learned from this paper. As authors introduces features like auto literature comparison and contribution similarity, all that can be very beneficial over the traditional system as they do not follow semantic and linked data approach unlike KG.

3) "Architecture of Knowledge Graph Construction Techniques", Zhanfang Zhao, Sung-Kook Han, In-Mi So, International Journal of Pure and Applied Mathematics 04-02-2018. **[4]**

Paper has mainly described the bottom-up approach for knowledge graph creation. In that they first shown the architecture in which different layers like knowledge extraction, Knowledge fusion, storage of knowledge graph and retrieval of KG has been described in details with their methods and tools available for it.

Outcome: We understood the difference between bottom-up and top-down approach for KG creation using ontology. Paper mainly focused on bottom-up approach so we learned that in details.

4) "An Integrated Hybrid Recommendation Model Using Graph Database", Angira Amit Patel, Dr. Jyotindra N. Dharwa, IEEE ICTBIG 06-04-2017. **[5]**

graph database is more efficient and expressive so they used a property graph [5]. In that they represent a multi-layer graph model and constructed a knowledge graph and returned the various top end N recommendation. As a result, they proposed a five-layer model, in which layer 1 is for users and information, layer 2 is for needs, layer 3 is for features and relevant information, and layer 4 is for all nodes linked to various item specifications and their related details. All nodes relating to various items and their associated details were included in layer 5. The creation of layers 2, 3, and 4 can be accomplished using preoccupied knowledge. In the process a system model is defined as a combination of different recommendation techniques hence can be called hybrid model RS, so that more efficient top-N recommendation can be done.

Outcome: From this paper we learned the use of heterogeneous graph usage in the Recommender system. Paper uses association of features for discovering relatedness and then the top N rec is derived from the similar specification from that feature. In our KGC19 work we are working with the association doc related to the original paper from which we can further find the information retrieval. Also, this paper claimed form the research that graph model always outperform the relational database and it can produce results in seconds.

5) "COVID-19 Knowledge Graph: Accelerating Information Retrieval and Discovery for Scientific Literature", Colby Wise, Vassilis N. Et al. Research Gate 07-12-2020. **[3]**

In this paper they constructed a CKG using latent schema and then enriched with biomedical entity. In methodology first step is to construct the KG and its curation for that they used

cord-19 dataset and defined different entity types, after that they used CKG to retrieve the information using different query, and as a third step they used CKG for recommendation in that they used a combination of semantic embeddings and KG embeddings (TransE) and then authors performed some analysis to verify and define their work.

Outcome: Form this paper we find two main useful things, one is Cord-19 dataset and the embedding approach like TransE which is used by our system right now. Paper has also mentioned semantic embedding and convolutional networks which can be used for the future work in our KGC19.

6) "Recommendation System Based on Heterogeneous Feature: A Survey", Hui Wang, Zichun LE, Xuan Gong, IEEE ACCESS OPEN ACCESS 15-09-2020. **[9]**

The authors of this work used a hierarchical design based on heterogeneous input attributes to learn text, behaviour, graph structure, and spatio-temporal properties from huge data using recommendation algorithms. They described the classification model design of RS, which is made up of three layers: feature input, feature learning, and output. They also talked about the evaluation metric, open-source implementation, relative merits, experimental comparison, and the path recommendation systems should pursue in the future.

Outcome: As mentioned, paper was focused on heterogenous feature for RS from which we learned the graph-based RS. In that we have learned the unsupervised techniques such as deep walk and node2vec etc. throughout we learner the enhancement from those algo and then leaned the jRDF2Vec which we have used in our KGC19 model as a tryout. Also, we learned evaluation parameters from which we have used MRR in our evaluation.

7) "Survey on Knowledge Graph-Based Recommender Systems", Jiangzhou Liu, Li Duan, IEEE: 5th IAEAC-2021 05-04-2021. **[11]**

In this paper authors have presented the basic knowledge of recommendation system and knowledge graph and after that they mentioned the key methods that used in the recommendation system with KG which includes path-based method, embedding based method and hybrid method further more they mentioned the user interest model, after that they have provided some basic future directions which include combination with graph neural network, enhanced representation of KG, KG completion and corrections.

Outcome: Along with path based and hybrid method one of the important things we learned from this paper was that KGE can be divided in two classes. Item based KG and User KG. here in our approach, we have currently focused on the item KG so we have focused on the

item KGE. Paper mention using neural network which can be used for the enhancement in our work too.

8) "A Survey on Knowledge Graph-Based Recommender Systems", Qingyu Guo, Fuzhen Zhuang, Chuan Qin, Hengshu Zhu, IEEE 07-10-2020. **[12]**

In this paper a deep survey has been done by authors on the recommendation system based on knowledge graph and they have categorized recommendation methods based on knowledge graph into three categories: embedding based, connection-based methods and propagation-based methods. they have provided fine grained information and advantages and disadvantages of methods mentioned into these three categories. after that they have also mentioned database related information and categorized them into different segments like movies, news, books, music etc.

Outcome: Generalized categories differentiation is learned. From that we learned some of the important algorithm used and details such as TransE, DistMult and node2vec. All that we have tried in our KGC19. Also, some of the important future direction that we learned such as dynamic RS and cross domain RS.

9) "A Comprehensive Survey of Knowledge Graph-Based Recommender Systems: Technologies, Development, and Contributions.", Chicaiza, Janneth, and Priscila Valdiviezo-Diaz. Information 12.6 (2021): 232. **[34]**

one of the detailed papers [6] mentioned KG based RS filtering approach into categories like ontology based, linked open database, embedding based and path based. in result they classified into two categories such as KG and semantic web and second is KG and AI methods. in first categories top 6 approaches mentioned some of them are KG and linked data, KGE and Ontologies, and in second category they compared different filtering approach and mentioned that hybrid system are most common. Future direction mentioned by the authors are interpretability of RS, explainable recommendation and KG based dynamic RS.

Outcome: Categorization of RS using KG is learned and where we get to understand them in clear detailed way. Paper mentioned that KG and linked data approach followed by KGE approach are most common approach in knowledge graph and semantic technology.

10) "Construction of Scenic Spot Knowledge Graph Based on Ontology.", Zeng, Wanghong, Hongxing Liu, and Yuqing Feng. IEEE, 2019. **[35]**

Authors created a scenic spot knowledge graph based on ontology, paper define the concept

of ontology on why and how should we use ontology so that the purpose can be served is greatly explained in this paper. Also, they present architecture that includes steps like data gathering and ontology building, entity alignment and knowledge graph storage tool. They used neo4j for storage and mentioned that it is one of the great databases that stores structured data in the form of network. For the evaluation purpose they also describe precision and recall metrices. There model outperform the string similarity method.

Outcome: Paper mentioned the steps to work with ontology which was very beneficial while working on KGC19.owl for our research work. Also, paper mentioned the results that ontology-based approach outperforms the string similarity method.

11) "Knowledge Graph-based Recommendation Systems: The State-of-the-art and Some Future Directions.", Sajisha, P. S., Anoop VS, and K. A. Ansal. International Journal- MLNCE. 2019 **[36]**

along with review some great future directions are mentioned such as bringing in more side information into knowledge graph so that power can be enhanced, also connecting social networks to know how social influence affect the recommendation, explainable recommendation and GCN are also in trend.

Outcome: We learned the benefit that KG based RS have over traditional RS. And also, we have learned about some of the model that we can apply in the future on KGC19 such as KGAT, DKN.

12) "RDF2Vec: RDF Graph Embeddings for Data Mining.", Petar Ristoski, Heiko Paulheim. Springer. International semantic web conference. 2016. **[37]**

This paper is introducing the original RDF2Vec approach from which other approaches are introduced. In RDF2Vec the very first step mentioned by the authors is to extract the substructure of the graph and generate the walks. After that it uses traditional word2vec algorithm to generate the embeddings this way it is used on the RDF graph but still depend on word2vec algorithm. Also, it provides all architecture from word2vec algo such as we can use it in CBOW and skip gram both ways. For the evaluation they have test this approach on small as well as large dataset and also presented the 2-dimensional PCA projection of cities vs countries graph. They claimed that their entity representation outperforming standard feature generation approaches. Also, for the future direction they aimed to build content-based recommender system.

Outcome: From this paper we learned original RDF2Vec approach and how its works. Also, from the observation we have learned that original paper did not used data properties and literals so considering the original work we also tried it for object properties only of the KGC19 graph. And as future direction suggested to use this approach further for the content base RS, we also tried for our KGC19 Content base RS. Unfortunately, we currently did not get satisfactory results but we aim to get it in the future.

13) "Knowledge Graph Embeddings with node2vec for Item Recommendation", Enrico Palumbo, Giuseppe Rizzo, Rapha¨el Troncy , Elena Baralis , Michele Osella and Enrico Ferro. Springer, European Semantic Web Conference 2018. **[38]**

Paper focuses on Node2vec algorithm for KGE. In the RDF2Vec paper and its documentation we found that it follows the similar structure like Node2vec and deep walk algorithm. They have focused on using cosine similarity. For the evaluation they have used standard metrices such as MRR, P@5, P@10, MAP and NDCG. From the result we observed that the best performing model of the node2vec on a given dataset was achieving 0.441 MRR. And 0.224, 0.196 of the P@5 and P@10 respectively.

Outcome: From this paper we learned the similar approach of RDF2Vec which is Node2vec. And can observed that on the given dataset the best MRR was 0.441 on the MovieLens dataset of 1M entity which can be said a good result considering the entity size.

3.2 Literature Review Summary Table

Title	Source	Methodology / Tools & Dataset	Advantages	Limitations	Future Work
"Cone-KG: A Semantic Knowledge Graph with News Content and Social Context for Studying Covid-19 News Articles on social media" [1]	IEEE SNAMS 2020	RDFizer tool, Fandet Ontology, GraphDB.	Schema-less Representation	Focused on the use of a semantic knowledge graph to model, structure and store, centrally and semantically no system is provided for further use.	Enhancement of KG by making it publicly available and Integration of Diff KG on Covid-19
"Open Research Knowledge Graph: Next Generation Infrastructure for Semantic Scholarly Knowledge" [2]	Research Gate 2020	Labled Property Graph (LPG), Triple Store.	Time Saving concept, Helpful for Scientific literature.	No detailed explanation, and dataset are not mentioned.	Integration of Diff KG on Covid-19 which are available
"Architecture of Knowledge Graph Construction Techniques" [3]	International Journal of Pure and Applied Mathematic	Linked Open Data, KG.	Explain Bottom-up approach in details.	mainly describe the bottom-up approach architecture.	understanding of top-down approach required.
"An Integrated Hybrid Recommendation Model Using Graph Database" [4]	IEEE ICTBIG 2017	Property Graph, Neo4j, Synthesis Dataset.	Hybrid Recommender System using property graph model.	Property graphs kind of limits the common structure and makes it hard to integrate KG.	knowledge acquisition, representation and self-learning like neural network.
"COVID-19 Knowledge	Research Gate 2020	Property Graph, Amazon	Medical domain information	only for scientific	Enhancement of CKG

Graph: Accelerating Information Retrieval and Discovery for Scientific Literature" [5]		Neptune, SciBERT; KGE; RGCN, Cord-19 Dataset.	added which is helpful in enriching knowledge graph.	literature which is also the benefit of their paper but we can always try to add some features.	information retrieval capabilities such as: expanding biomedical entity extraction.
"Recommendation System Based on Heterogeneous Feature: A Survey" [6]	IEEE ACCESS OPEN ACCESS 2020	softmax, Pearson correlation coefficient, Cosine Similarit, adjusted cosine similarity, BinarySimilarity.	Based on heterogenous features different recommendation techniques introduces with algorithm choice	It includes only general structure, there is no specific techniques or methods has been pointed for the best or efficient implementation.	Combination of sentiment analysis and interpretability of RS etc.
"Survey on Knowledge Graph-Based Recommender Systems" [7]	IEEE: 5th IAEAC 2021 2021	NGCF, DKN; SR-GNN,	Detailed information on recommender system using KG.	Lacks in details of how Knowledge Graph can be created.	Dynamic, Cross domain, Knowledge enhanced language representation.
"A Survey on Knowledge Graph-Based Recommender Systems" [8]	IEEE 2020	Emb based, Connection based, Propagation based.	Three major categories of KG based RS are explained in details	-	Research direction like dynamic & cross domain RS
"A Comprehensive Survey of Knowledge Graph-Based Recommender Systems: Technologies, Development, and Contributions." [34]	MDPI open access 2021	Knowledge based: Ontology based, linked open data based, KGE based, Path based RS using KG	They have categorized the section based on the technology used, application and development	-	Interpretability of RS, explainable RS, dynamic RS are the Future Direction

"Construction of Scenic Spot Knowledge Graph Based on Ontology" [35]	IEEE DCABES 2019	Web crawler, Ontology construction, Entity alignment	Out perform string similarity method, construction of scenic spot KG.	Property graph is used, limitation of it can affect the use case in our work.	More information retrieval and discovery can be done.
"Knowledge Graph-based Recommendation Systems: The State-of-the-art and Some Future Directions." [36]	International Journal-MLNCE 2019	Rule based learning, Knowledge aware graph neural network, KGE.	Full survey on RS with KG based on different categories mentioned in methodology.	No analysis is given only information is mentioned.	Using more side information & social network for RS engine
"RDF2Vec: RDF Graph Embeddings for Data Mining" [37]	AAAI-19	Weisfeiler Lehman Subtree RDF Graph Kernels & graph walks and word2vec	Embedding approach for RDF graph based on traditional word2vec model	Not giving satisfactory results on KGC19 yet	More versions of rdf2vec using different parameters for KGC19.
"Knowledge Graph Embeddings with node2vec for Item Recommendation" [38]	Springer Semantic Web Conference 2018	Movie Lens dataset, Node2vec Algorithm.	Node2vec outperforms item KNN, SVD and Random.	Comparison with different Embedding family can be done. MRR can be increased.	Comparison with different embeddings family.

Table 1: Summary of Literature

CHAPTER 4

Methodology: Materials and Methods

This section includes a workflow diagram, a basic generalised diagram presenting the overall steps to consider while building a system, and the actual proposed methodology followed to complete our own work. Now that we have a basic understanding of all of the types, we can build both the systems individually. As stated earlier, the need for the recommender system is on the trend because of the data explosion in this era. With an abundance of data, we need a proper system that we can use for better preference. We have discussed how the knowledge graph is beneficial for RS. Let us understand the basic workflow diagram.

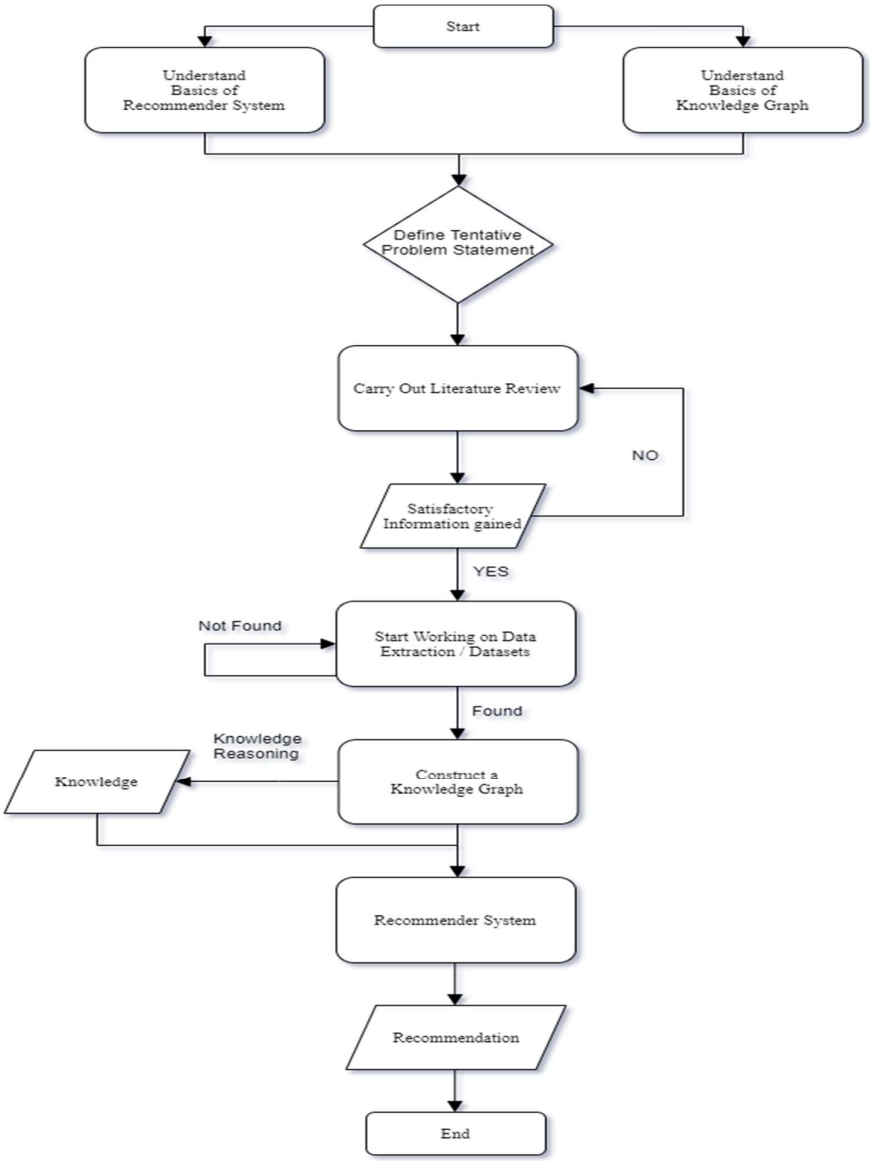


Figure 6: Process flow Diagram.

4.1 Generalized Approach For a recommender system using knowledge graph

The basic understanding of the recommender system with a knowledge graph diagram is mentioned below [Figure 7]. It can be used to build any KG-based RS or information retrieval system. As mentioned in the figure, the first and foremost step is to collect the data or to extract the data. After that, data cleansing and processing can be done for better evaluation. Now we have the data, we can create a knowledge graph. After creating a KG, it can be used for recommendation, and that will be our next step. Retrieval and visualisation are not necessary, yet they can be beneficial steps for better understanding and enhancement of KG for RS. The last step is the maintenance of KG and RS with it. For better understanding, see the methodology followed in our own work.

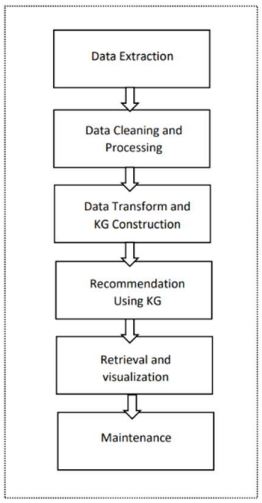


Figure 7: Generalized diagram for RS using KG.

4.1.1 Data Extraction

Data extraction is the most basic yet crucial part of any system, as knowing your data can explicitly increase the efficiency and performance. However, we know that there is a lot of diversity in data in terms of its types. For example, data can be structured, semi-structured, and unstructured. And there are lots of types available in those three categories, such as JSON, JSONLD, CSV, text, pictures, audio, graphs, and many more. And to extract that diverse data, there are lots of different methodologies and tools available. So, we need web scrapping or any other method to extract related data only. Moreover, we can also use any existing databases, such as cord-19 in our case. And we might integrate different sources for our use. In the cone-kg paper, the authors mentioned that they have implemented an efficient and scalable parser for the data types i.e., JSON, JSONLD, CSV, and TXT. And for the JSON files specifically, they have implemented a JSON streaming parser [1]. After that, they

stored the data in a MySQL database table and then exported it in CSV format.

4.1.2 Data Cleaning and Processing

Before performing operations such as creating KG or data transformation, we must clean the data, as data can be observed as noisy and maybe with missing values sometimes. Also, some objects or entities that should be represented as RDF resources may require operations like removing white space or special characters so that generated resources in RDF can be valid URIs. Also, dealing with the different datasets might have different date formats or different representations of the author/person names; all of that needs to be in the same format, such as the date should only be in dd-mm-yy or space in the names should be replaced by a dash so that the generated new name can be a valid URI [1]. With data, missing values are also common, so we can also replace the missing values with unknown ids. So, before creating a KG, this necessary cleansing and processing of data can be an important part of the recommended system with KG.

4.1.3 Data transformation and KG construction

The data can be divided into smaller chunks for ease of import and storage.This also facilitates ease of fitting into memory. After that, as mentioned earlier, we can either create or use an existing ontology for further processing. The fandet ontology (available at https://github.com/rif-zu/fandet-ontology) was used by the authors of the Cone-KG paper [1]. can also create our own ontology in order to create a knowledge graph for our system and we can verify it according to the ontology by running queries using a query language such as SPARQL.

4.1.4 Recommendation using KG

According to the survey done by the paper [34], the KG-based recommendation approach can be classified into a total of four categories. All four categories are mentioned in the diagram below.

In ontology-based systems, ontologies are used to model knowledge about users and their context, knowledge about items, and knowledge about the domain. Furthermore, an ontology's structure and semantics make it easier to create rules that generate recommendations based on explicitly described constraints or rules.

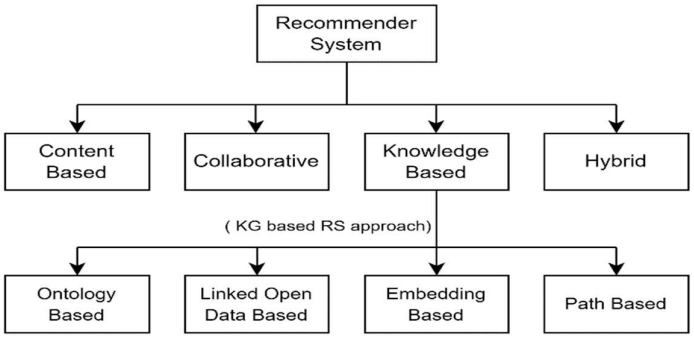


Figure 8: General Classification Diagram of RS using KG

Making semantic information query able from LOD data sets can improve the system's information, allowing for the discovery of related attributes among recommended items. The main benefit of doing so is that it solves the data sparsity problem. However, because the recommendation process is reliant on external data, the integrity of that data may have an impact on the recommendation outcomes.

The KG is converted using KGE algorithms, and the learned entity and relationship embeddings are then used by a recommendation system to provide a set of outcomes. In a continuous vector space, an embedded KG can represent entities and relationships while maintaining network information. The purpose of using KGE methods is to make it easier to process a KG while keeping its structure.

A natural and obvious technique to employ KGs in recommendation systems is path-based recommendation. In order to obtain additional information for suggestions, the algorithms seek to explore alternative patterns of connections between nodes in a KG. This method is based on hand-crafted designed meta-paths, which are difficult to optimize in practice and are impossible to create in some circumstances where entities and their interactions are not contained within a single domain.

4.1.5 Retrieval and Visualization

After creating a KG and Building a recommender system, querying over it, retrieving useful information and validating it or using it accordingly is also an important step. While we apply KG into the RS, we might use additional information and we may want to consider combining them together. So, to know our data and for the better understanding we may want to perform different analytic by executing various SPARQL queries. For example, we can find which particular author published article about COVID-19 through multiple publishers. Also, we can query over various article and find out if specific keywords are included such as Corona virus. And as SPARQL query results in machine readable format

mostly we might require visualization tools or analytical tools to perform some operation efficiently. So, retrieval and visualization can be considered as an important step in this flow diagram.

4.1.6 Maintenance

As we know that data is increasing day by day and especially in trending topic, for example in case of COVID-19 there is a high chance of increasing new data and knowledge discovery every day. So, maintaining and updating our system or in particular our knowledge graph can become a necessary step. Although in context of KG, maintenance techniques are not very common but We will consider the following categories of changes to the knowledge graph: changing world, changing requirements, changing sources, changes affecting previous inferences, and changes requiring redesign [23]. So, for better or worse this step needs to be taken into an account as we might need it at some point.

4.2 Proposed Methodology

Figure 9: Proposed Diagram

4.2.1 Dataset (Cord-19) [24]

In response to the COVID-19 pandemic, the White House and a coalition of leading research groups have prepared the COVID-19 Open Research Dataset (CORD-19). CORD-19 is a resource of over 500,000 scholarly articles, including over 200,000 with full text, about COVID-19, SARS-CoV-2, and related coronaviruses. This publicly available dataset is made available to the world's researchers so that they can use recent developments in NLP and other AI approaches to generate new insights in aid of the ongoing battle against the contagious diseases. Because of the tremendous acceleration of new coronavirus literature, which makes it very difficult for the medical research community to stay up, there is an increasing urgency for these approaches.

That is why the concept of KG was born, as well as the idea of information retrieval and recommendation based on it. We have used latest release of the dataset contain around 67GB of data. Version published on 2022-01-31 and it is a current latest version at a time of writing this thesis. Dataset can be found at Kaggle website under the name COVID-19 Open Research Dataset Challenge (CORD-19) [24].

4.2.2 Preprocessing

The need of preprocessing is now become a necessary step in ML, NLP and in almost all field. While creating KG as stated earlier in this document, making data a valid URI is required such cleaning and preprocessing of the data.

First of all, big metadata file was split into chunks of file for the ease of use and fitting into memory while processing. And in our case invalid Cord_UID from metadata file is also removed before passing it into KG because, our KG is using Cord_UID and Sha id of the paper as a key element to create an entity. And Cord_uid is considered as key element because cleaning and processing of 60 GB of data would have been a bottle neck for this research. But as an iterative process we can still update our KG according to our need so in future we can create a newer version too.

Also, in the metadata file source of the paper, sha ids associated with same PMC article and names of all authors were in the same column respectively. But in order to create the KG the preprocessing is used to separate them so that we can use easily using python script to map those data. and the last step was to extract needed data from the full text pdf files available in dataset and preprocess in order that no NAN value is place in KG and every data is cleaned and converted to str format for ease of use. All of this is done using python scripts written on our own.

4.2.3 Data Mapping and KG Construction

After the cleaning data the process of transforming those data into KG was initiated. RDFLib library of python is used which is one the easy to use and efficient library out there to transform the data. There are almost all RDF formats supported by RDFLib. It provides easy way to describe URIRef, Literals and many more for example:

```
# Parse in an RDF/owl file on the internet or available at locally
g = Graph()
g.parse("Link to the file on internet/local machine")
# Process the data and make a valid URI URIRef
Var = URIRef('The data you want to passed as entity')
# Process the data as literals
Var = Literal('1', datatype=XSD.integer)
# To add triple into Graph using add method (example for foaf ontology)
g.add((donna, RDF.type, FOAF.Person))
# Bind the FOAF namespace to a prefix for more readable output
g.bind("foaf", FOAF)
# Print out the entire Graph in the RDF Turtle format
print(g.serialize(format="turtle"))
```

For better understanding you can read Documenation and example from official document of RDFLib available HERE [41].

4.2.4 Information Retrieval using GraphDB & SPARQL

We have used GraphDB a popular triple store provided by Ontotext to store and query over the data. GraphDB provide loading of the RDF data in many ways from which we have used preload methodology. Preload can be used for huge dataset and it provides Initial offline import with no inference and plugins and Ultra-fast speed without speed degradation. It also provides support for explore section which include graph overview, class hierarchy, class relationship and visual graph option. And not to mention support for SPARQL query language.

SPARQL is widely used query language and it provides easy syntax to retrieve data from KG. for example, loading all the triple from the graph with max limit of 1000 would look like this:

```
select * where {
        ?s ?p ?o .
} limit 1000
```

More results can be found in the result section.

4.2.5 Recommendation using KGE (TransE)

We have used many traditional RDF2Vec methods which follows word2vec embedding techniques on the RDF graph such as wang2vec and JRdf2vec using different parameters. But none of them touched the expected result on our KG KGC19 as compared to semantic embedding traditional models such as TransE, ComplEx, DistMult and HolE. However, wang2vec and JRdf2vec provides one of the best RDF2Vec model implementation and can be used on different KG if suited, so it is worth mentioning them it's just didn't work on KGC19 due to its structure.

Using semantic Traditional models such as TransE we have achieved a good result as it mapped vector correspondence to the relationship that entities have. For that we have used opensource AmpliGraph Library which is one of the finest available to implement and play with RDF embeddings. It provides so many different APIs to perform different task with scalable formats. It also provides evaluation APIs and Knowledge discovery APIs. We can generate .pkl file for the model using our rdf graph which contains vector representation of our graph. AmpliGraph provides CPU and GPU processing whichever we find suitable.

For more information documentation of the Ampligraph can be found HERE [42]. And for the better understanding, example of KGE by ECAI 2020 can be found HERE [43]. It is a very detailed and good tutorial to understand KGE and the use of AmpliGraph.

This section provided the overview on the methodology we followed to achieve the task for our system. Experimental results and detailed Discussion can be found in the next section.

CHAPTER 5

OBSERVATION, RESULTS AND DISCUSSION

Throughout the study the challenges that we have faced is mentioned below with the possible solutions and results of our approach that we have followed.

5.1 Ontology creation using protégé (KGC19.owl)

As stated in the previous sections we now understand that ontology is the foundation of the knowledge graph so our first step was to build the ontology. But in order to build the ontology, as we have discussed earlier, we have two kind of approaches that are bottom up and top down. And there is new approach which is called hybrid approach and that what we have followed here due the fact that we were having a dataset cord-19 and on that we tried to make our knowledge graph. But instead of following bottom-up approach directly we found that staying in the middle is better.

Protégé software is one of the leading software with many plugins available to use as ontology building tools. Such as VOWL and OntoGraph for visualizing ontology.

Base URI: http://www.semanticweb.org/warisahmed/ontologies/KGC19

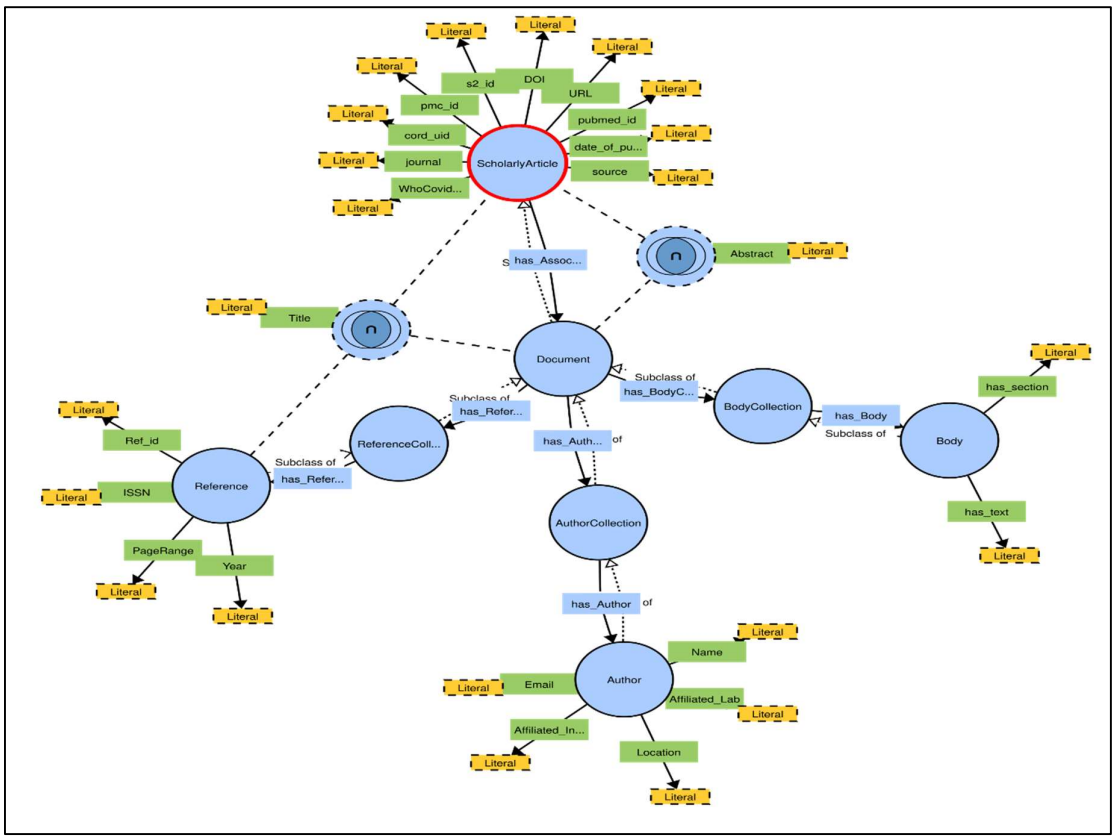


Figure 10: KGC19 ontology

5.2 Creating RDF file and loading data into triple store.

Now for the populating KGC19.owl with instances or we can say with data we first tried to do it using protégé software plugin named Cellfie, in that we can write rules to extract the data from excel workbook in the form of axioms but it failed due the fact that we were dealing with the large dataset. So, while working with the limited sized data it is great tool which can be used within the protégé itself.

During this try-out we also have found the bug in that plugin and reported it into the github page and they have releases new update with the fix.
Find more details at: https://github.com/protegeproject/cellfie-plugin/issues/159

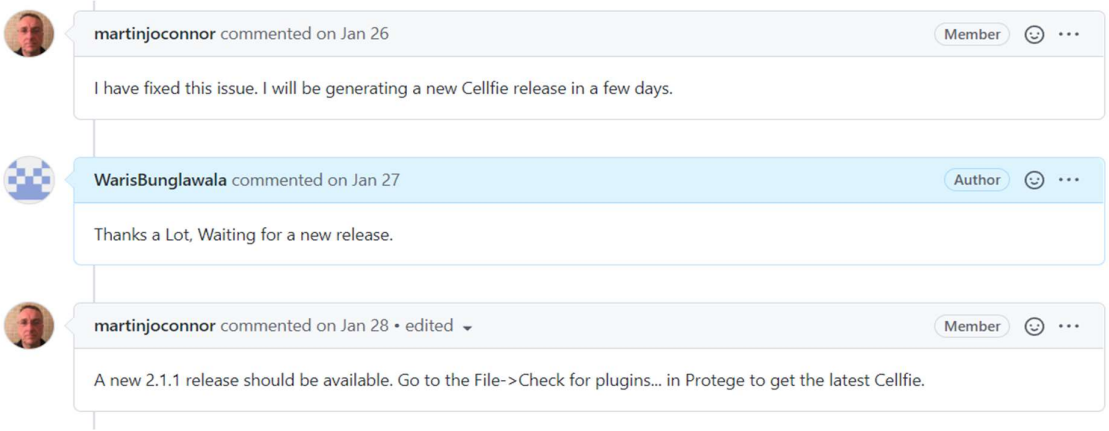


Figure 11: GitHub Bug report for cellfie-plugin

After Cellfie plugin we have moved to the python scripts and used RDFLib to extract the data from the dataset and to add it into the RDF file base on ontology owl file. As explained earlier RDFlib is a great library to work with rdf data. So, we have generated rdf file into ttl (turtle) format. Turtle is human readable type format which allows defining prefix at the top of the file and it can group related data into blocks so the URI for same is not repeated all the times. In our work we are using prefix as KGC19 which indicates full URI as:

<http://www.semanticweb.org/warisahmed/ontologies/KGC19#>

Creation of RDF file was divided into several parts for the ease of fitting into the memory we have created one rdf file for metadata and three more rdf file for the full text papers.
GraphDB was used as triple store and in order to load the big files into GraphDB we used preload method, which will load data into store without inferences and without speed degradation. The command that is used is:

$ <graphdb-dist>/bin/preload -i <repo-name> <RDF data file(s)>

More details on loading data into GraphDB can be found here: [Loading Data — GraphDB Free 9.10.0 documentation (ontotext.com)](#) [44]

The ontology creation and loading data process is finished here and now let us see some results of information retrieval that we can achieve.

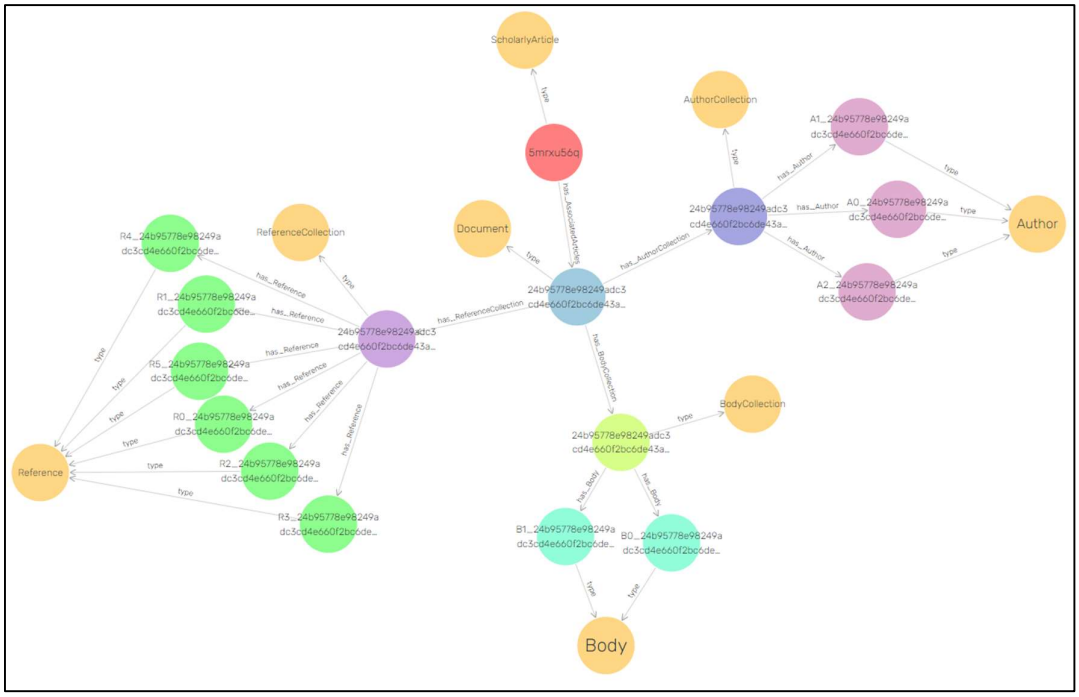


Figure 12: KGC19 Sub Graph of "5mrxu56q" Obj Properties only

5.3 Information retrieval using SPARQL

Reported result of distinct count of Total triple, Subject, Predicate and Object is mentioned in the table here:

Triple	Subject	Predicate	Object
112456898	10223616	37	2621440

Table 2: KGC19 Distinct Subject, Predicate & Object count

a) Total triple count of KGC19

Total Triple Count from GraphDB = 11,24,56,898

Query:

```
Prefix KGC19:<http://www.semanticweb.org/warisahmed/ontologies/KGC19#>
select (count(*) as ?triple) where {?s ?p ?o}
```

Results:

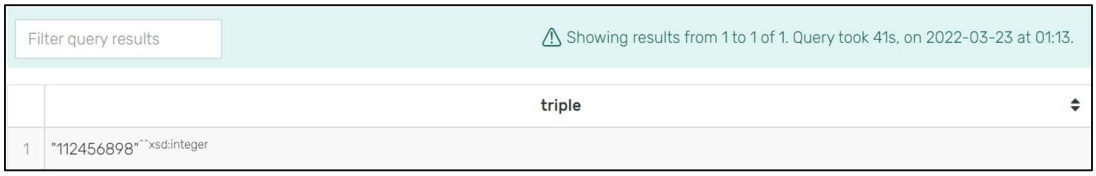


Figure 13: Output - Total Triple Count

b) Retrieving all the distinct article title from KGC19

Query:

```
Prefix KGC19: <http://www.semanticweb.org/warisahmed/ontologies/KGC19#>
select distinct ?title where {
   ?ScholarlyArticle KGC19:Title ?title
}
```

Results:

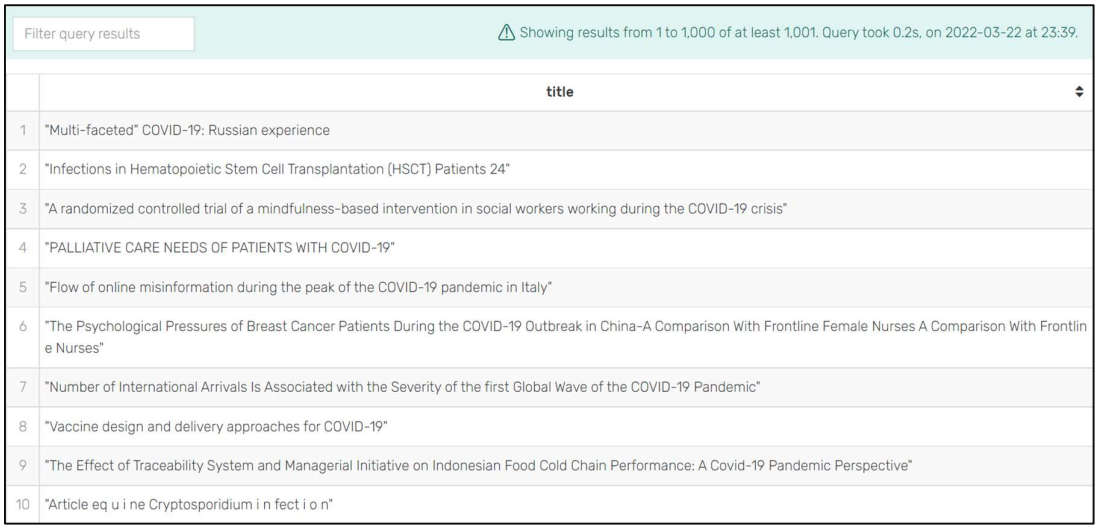


Figure 14: Output – All distinct Article from KGC19

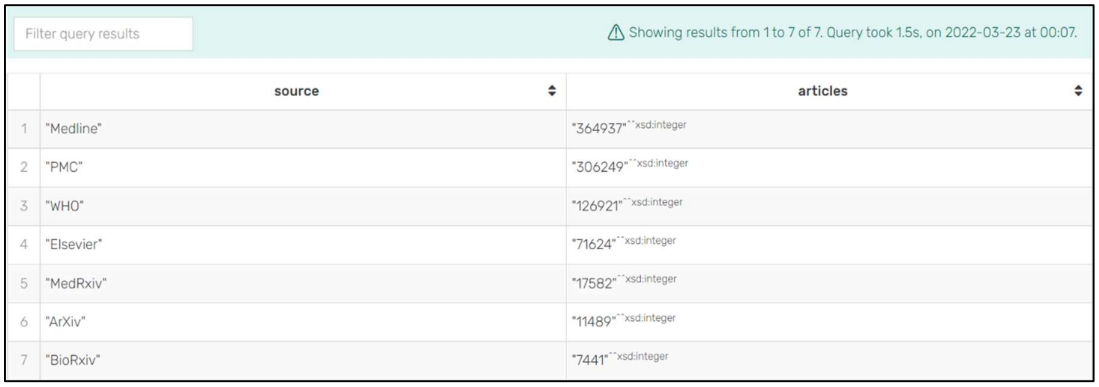
c) List all the source KG have, with highest articles published count first

Medline is the Highest with count = 364937 articles

Query:

```
Prefix KGC19: <http://www.semanticweb.org/warisahmed/ontologies/KGC19#>
select ?source (count(distinct ?URL) as ?articles) where{
   ?ScholarlyArticle KGC19:URL ?URL.
   ?ScholarlyArticle KGC19:source ?source.
}
group by ?source order by DESC(?articles)
```

Results:



Filter query results

Showing results from 1 to 7 of 7. Query took 1.5s, on 2022-03-23 at 00:07.

	source	articles
1	"Medline"	"364937"^^xsd:integer
2	"PMC"	"306249"^^xsd:integer
3	"WHO"	"126921"^^xsd:integer
4	"Elsevier"	"71624"^^xsd:integer
5	"MedRxiv"	"17582"^^xsd:integer
6	"ArXiv"	"11489"^^xsd:integer
7	"BioRxiv"	"7441"^^xsd:integer

Figure 15: Output – Source wise Article Count

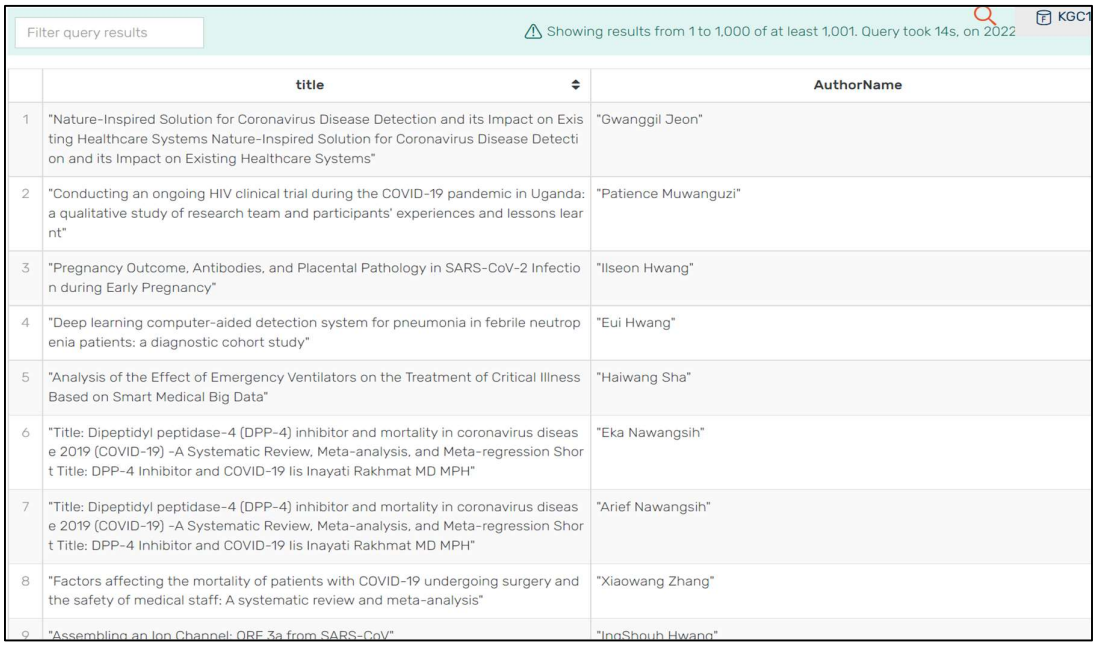
d) Listing all of the article containing author name as “wang”

More than 1000 articles have author name containing wang

Query:

```
Prefix KGC19: <http://www.semanticweb.org/warisahmed/ontologies/KGC19#>
select ?title ?AuthorName where {
   ?ScholarlyArticle KGC19:Title ?title.
   ?ScholarlyArticle KGC19:has_AuthorCollection ?AuthorCollection.
   ?AuthorCollection KGC19:has_Author ?Author.
   ?Author KGC19:Name ?AuthorName.
   filter regex(?AuthorName, 'wang')
}
```

Results:



Filter query results

Showing results from 1 to 1,000 of at least 1,001. Query took 14s, on 2022

KGC1

	title	AuthorName
1	"Nature-Inspired Solution for Coronavirus Disease Detection and its Impact on Existing Healthcare Systems Nature-Inspired Solution for Coronavirus Disease Detection and its Impact on Existing Healthcare Systems"	"Gwanggil Jeon"
2	"Conducting an ongoing HIV clinical trial during the COVID-19 pandemic in Uganda: a qualitative study of research team and participants' experiences and lessons learnt"	"Patience Muwanguzi"
3	"Pregnancy Outcome, Antibodies, and Placental Pathology in SARS-CoV-2 Infection during Early Pregnancy"	"Ilseon Hwang"
4	"Deep learning computer-aided detection system for pneumonia in febrile neutropenia patients: a diagnostic cohort study"	"Eui Hwang"
5	"Analysis of the Effect of Emergency Ventilators on the Treatment of Critical Illness Based on Smart Medical Big Data"	"Haiwang Sha"
6	"Title: Dipeptidyl peptidase-4 (DPP-4) inhibitor and mortality in coronavirus disease 2019 (COVID-19) -A Systematic Review, Meta-analysis, and Meta-regression Short Title: DPP-4 Inhibitor and COVID-19 Iis Inayati Rakhmat MD MPH"	"Eka Nawangsih"
7	"Title: Dipeptidyl peptidase-4 (DPP-4) inhibitor and mortality in coronavirus disease 2019 (COVID-19) -A Systematic Review, Meta-analysis, and Meta-regression Short Title: DPP-4 Inhibitor and COVID-19 Iis Inayati Rakhmat MD MPH"	"Arief Nawangsih"
8	"Factors affecting the mortality of patients with COVID-19 undergoing surgery and the safety of medical staff: A systematic review and meta-analysis"	"Xiaowang Zhang"
9	"Assembling an Ion Channel: ORF 3a from SARS-CoV"	"IngShouh Hwang"

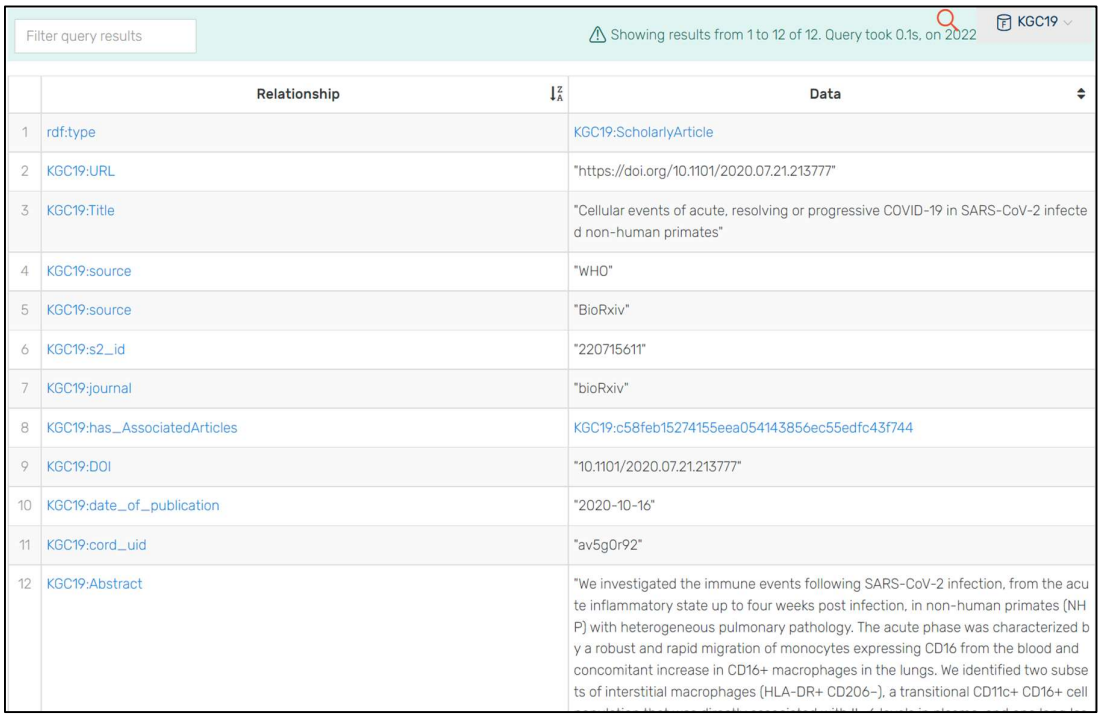
Figure 16: Output – Author Name which contains “Wang”

e) List All the triple containing cord_uid “av5g0r92” as Subject

Query:

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
Prefix KGC19: <http://www.semanticweb.org/warisahmed/ontologies/KGC19#>
select * {
   KGC19:av5g0r92 ?Relationship ?Data.
}
```

Results:



	Relationship	Data
1	rdf:type	KGC19:ScholarlyArticle
2	KGC19:URL	"https://doi.org/10.1101/2020.07.21.213777"
3	KGC19:Title	"Cellular events of acute, resolving or progressive COVID-19 in SARS-CoV-2 infected non-human primates"
4	KGC19:source	"WHO"
5	KGC19:source	"BioRxiv"
6	KGC19:s2_id	"220715611"
7	KGC19:journal	"bioRxiv"
8	KGC19:has_AssociatedArticles	KGC19:c58feb15274155eea054143856ec55edfc43f744
9	KGC19:DOI	"10.1101/2020.07.21.213777"
10	KGC19:date_of_publication	"2020-10-16"
11	KGC19:cord_uid	"av5g0r92"
12	KGC19:Abstract	"We investigated the immune events following SARS-CoV-2 infection, from the acute inflammatory state up to four weeks post infection, in non-human primates (NHP) with heterogeneous pulmonary pathology. The acute phase was characterized by a robust and rapid migration of monocytes expressing CD16 from the blood and concomitant increase in CD16+ macrophages in the lungs. We identified two subsets of interstitial macrophages (HLA-DR+ CD206−), a transitional CD11c+ CD16+ cell

Figure 17: Output – All the triple which have subject as “av5g0r92”

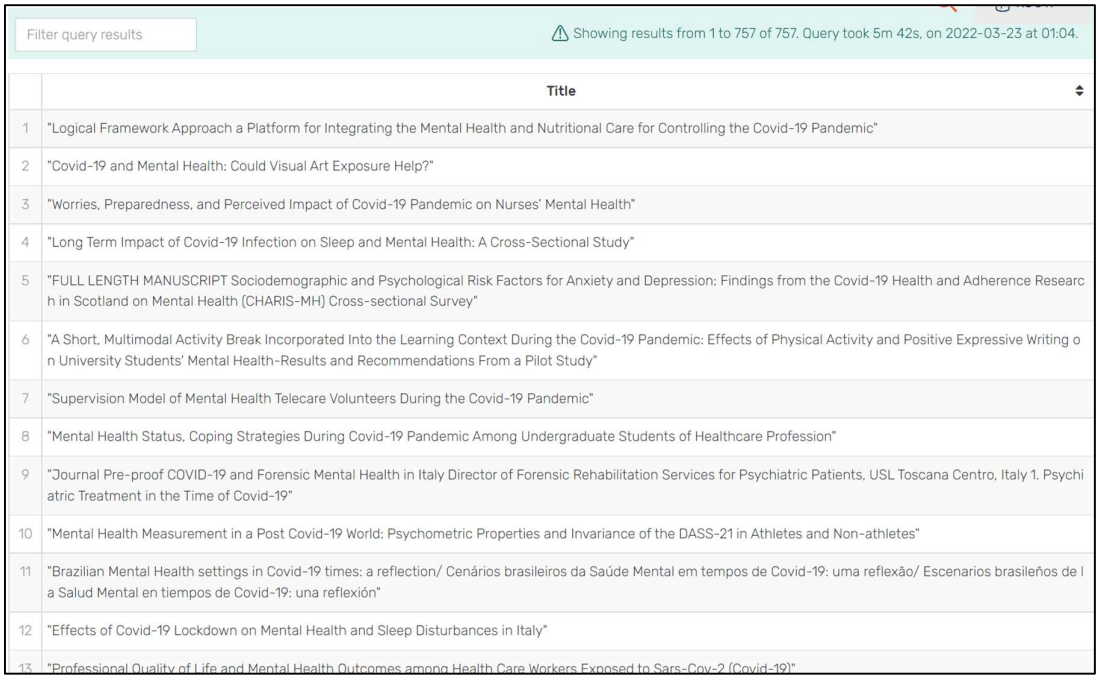
f) List all the Title Containing “Covid-19” + “Mental Health” word in it.

Found 757 Articles from KGC19

Query:

```
PREFIX rank: <http://www.ontotext.com/owlim/RDFRank#>
PREFIX KGC19: <http://www.semanticweb.org/warisahmed/ontologies/KGC19#>
Select ?Title
where {
   ?subject KGC19:Title ?Title
   FILTER (Contains(?Title, 'Covid-19') && Contains(?Title, 'Mental Health'))
}
```

Result:



Filter query results

⚠ Showing results from 1 to 757 of 757. Query took 5m 42s, on 2022-03-23 at 01:04.

	Title
1	"Logical Framework Approach a Platform for Integrating the Mental Health and Nutritional Care for Controlling the Covid-19 Pandemic"
2	"Covid-19 and Mental Health: Could Visual Art Exposure Help?"
3	"Worries, Preparedness, and Perceived Impact of Covid-19 Pandemic on Nurses' Mental Health"
4	"Long Term Impact of Covid-19 Infection on Sleep and Mental Health: A Cross-Sectional Study"
5	"FULL LENGTH MANUSCRIPT Sociodemographic and Psychological Risk Factors for Anxiety and Depression: Findings from the Covid-19 Health and Adherence Researc h in Scotland on Mental Health (CHARIS-MH) Cross-sectional Survey"
6	"A Short, Multimodal Activity Break Incorporated Into the Learning Context During the Covid-19 Pandemic: Effects of Physical Activity and Positive Expressive Writing o n University Students' Mental Health-Results and Recommendations From a Pilot Study"
7	"Supervision Model of Mental Health Telecare Volunteers During the Covid-19 Pandemic"
8	"Mental Health Status, Coping Strategies During Covid-19 Pandemic Among Undergraduate Students of Healthcare Profession"
9	"Journal Pre-proof COVID-19 and Forensic Mental Health in Italy Director of Forensic Rehabilitation Services for Psychiatric Patients, USL Toscana Centro, Italy 1. Psychi atric Treatment in the Time of Covid-19"
10	"Mental Health Measurement in a Post Covid-19 World: Psychometric Properties and Invariance of the DASS-21 in Athletes and Non-athletes"
11	"Brazilian Mental Health settings in Covid-19 times: a reflection/ Cenários brasileiros da Saúde Mental em tempos de Covid-19: uma reflexão/ Escenarios brasileños de l a Salud Mental en tiempos de Covid-19: una reflexión"
12	"Effects of Covid-19 Lockdown on Mental Health and Sleep Disturbances in Italy"
13	"Professional Quality of Life and Mental Health Outcomes among Health Care Workers Exposed to Sars-Cov-2 (Covid-19)"

Figure 18: Output – Title containing "Covid-19" + "Mental Health"

5.4 Embeddings for Recommendation

One of the major works apart from information retrieval that we focused on is recommendation using KG. now as we know that recommendation can be achieved using different embedding techniques so similarly, we tried two types of embedding models' family to achieve the task.

One is based on traditional word2vec algo, another is traditional KGE models.

- Rdf2vec models which follows traditional word2vec embedding as base
  - JRdf2vec [45]
  - Wang2vec [46]
- KGE models which is a Traditional knowledge graph embedding approach
  - TransE
  - ComplEx
  - HolE
  - DistMult

In the first rdf2vec models, we changed different parameters and trained over 10 models to achieve the desires results but due to the KGC19 structure those models failed to achieve the desired results as they are based on word2vec traditional approach. However, JRdf2vec is one of the best rdf2vec model out there which uses java implementation. So, depending on

the graph you are working on it is worth trying rdf2vec model if it is suited for your KG. wang2vec model can be used to achieve ordered rdf2vec results. From trained models, we have observed that the best resulting model from these were trained using parameters mentioned below.

Wang2vec: Entity training, Training algorithm – CBOW, Epochs – 200, Window size – 3, Dimension – 50, Negatives – 3

but still its performance wasn't good enough for our desired outcomes so we won't be including it here. Also, we have found one enhancement issue in jrdf2vec we which have reported to gihub page and recently the newer version of it is released by author with the fix.

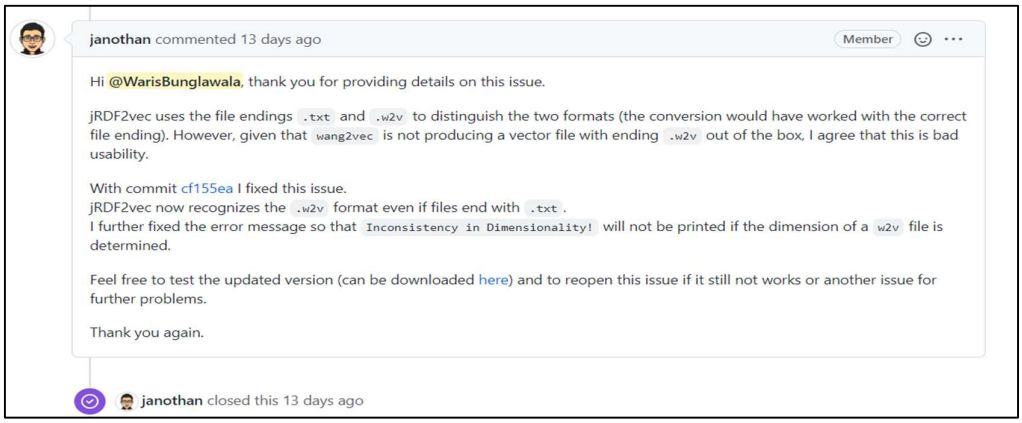


Figure 19: GitHub issue report for JRDF2Vec

Now, Considering Traditional KGE we have trained models using all of the mentioned algorithms. From which we found the best performing model which is TransE. Ampligraph provides many APIs including best model selection so that one can find the best model for their knowledge graph with best suited parameters. But that can consume so much time and resources so we tried only for the parameters range that we thought was best suited or meaningful for training. For now, the TransE model is the only one that is generating desired outcomes but for the future enhancement we will try to increase the performance of other models.

Parameters that are used to trained TransE model:

```
# Embedding size, Num of Epochs, number of corruptions to generate during training
K=150, Epochs=20, eta=2,
# Loss type and it's hyperparameters
loss='pairwise', loss_params={'margin': 1}
# Initializer type and it's hyperparameters
```

```
initializer='xavier', initializer_params={'uniform': False}
# regularizer along with its hyperparameters
regularizer='LP', regularizer_params= {'lambda': 0.001, 'p': 3}
# Optimizer to use along with its hyperparameters
optimizer='sgd', optimizer_params={'lr': 0.01}
seed=0, verbose=True
```

In KGE models we have focused on the evaluation parameters / matrices called:

- **Per triple metrices:**

  Score & Rank: it is a metrics that is computed for each test set triple:

  - **Score:** This is the value assigned to a triple, by the model, by applying the scoring function.
  - **Rank:** For a triple, this metric is computed by generating corruptions and then scoring them and computing the rank(position) of the triple score against the corruptions.

- **Aggregate metrices:**

  Once we have the ranks for all the test set triples, we can compute the following aggregate metrics: MR, MRR, Hits@N. These metrics indicate the overall quality of the model on a test set. These metrics come from Information Retrieval domain and are always computed on a set of True Statements.

  - **MR:** is the average of all the ranks of the triples. The value ranges from 1 (ideal case when all ranks equal to 1) to number of corruptions (where all the ranks are last) [43].

$$MR = \frac{1}{|Q|} \sum_{i=1}^{|Q|} rank_{(s,p,o)_i}$$

  - **MRR:** is the average of the reciprocal ranks of all the triples. The value ranges from 0 to 1; higher the value better is the model [43].

$$MRR = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \frac{1}{rank_{(s,p,o)_i}}$$

  - **Hits@N:** is the percentage of computed ranks that are greater than (in terms of ranking) or equal to a rank of n. The value ranges from 0 to 1; higher the value better is the model [43].

$$Hits@N = \frac{1}{|Q|} \sum_{i=1}^{|Q|} 1 \text{ if } rank_{(s,p,o)_i} \leq N$$

For KGE we have used small portion of the knowledge graph for the ease of use and best

fitting, total triples count by the ampligraph from the dataset and the size of np array splits that we used for train, valid and test sets are mentioned below: (3 indicates array is of 3 dimensions because we are having a form of triple- subject, predicate, object)

Total triples set	Size of train set	Size of valid set	Size of test set
(1412877, 3)	(1337877, 3)	(25000, 3)	(50000, 3)

Table 3: Split of KGC19 Dataset for Train, Test & Valid set

Using score, we can interpret two tasks:

- We can create a list of Hypothesis that we want to test, score them and then choose the top N hypothesis as True statements.
- As described earlier in the theory section, unlike classification task, we are doing a learning to rank task. In order to interpret the score, we can generate the corruptions and compare the triple score against the scores of corruptions to see how well does the model rank the test triple against them.

We can create our own set of Hypothesis, score it and then sort it and from that we can choose top N triples to be true triples. For the hypothesis we have used the set of associated articles set mentioned below:

```
# Hypothesis triple (last 4 are the wrong sha that means from another cord_uid and others are from largest sha set (116))
list_of_AssociatedArticles = [
                    'KGC19:83af5e246f8886d630fbc7e692ebc32d60c7ba43',
                    'KGC19:6c915a41fb646b41b1d30bc07049223a33a7a7ee',
                    'KGC19:4c9c90c73e40435b0e5d80f9a8d7df3c5cd670dc',
                    'KGC19:210e021ba2050d395326dc70f6f5505f24b03e39',
                    'KGC19:70a4622adc00dac981d1cbc407da93d6e5591bad',
                    'KGC19:b5792d92409c6daee37a0eaecb6eafe90dd7e7c0',
                    'KGC19:b21316debb3cfac6820f0aa2b04028a3b63bc587',
                    'KGC19:efeaa7412bc8c74a4a4ca3365e6a1d4d19c3c409',
                    'KGC19:4d177174a18d68179fcffdb54ea1b2f3b19abd50', # starting of wrong 4
                    'KGC19:4ad56700823b5e8696b7b1f5aac6f16d9c1fadc8',
                    'KGC19:b52be1d2a2c8e089f1a1d9bfd84f257f53fd1c04',
                    'KGC19:ff7807b699f1469608f627c6bafead27b815fb6c', # end of wrong 4
]
```

All of the hypothesis is checked with respect to the

- Subject:**'KGC19:hzzm3fcv'**
- Predicate:**'KGC19:has_AssociatedArticles'**

Output generated by the TransE model in a sorted list is mentioned below with additional column added by us to mentioned the wrong triple placed by model to indicate the position

that they are at. As seen in the table from top 5 only one triple is not a true triple other four are well placed.

Associated Article Node	Score	True Triple?
'KGC19:6c915a41fb646b41b1d30bc07049223a33a7a7ee'	'-196.79282'	Yes
'KGC19: 4d177174a18d68179fcffdb54ea1b2f3b19abd50'	'-197.75183'	No
'KGC19: 210e021ba2050d395326dc70f6f5505f24b03e39'	'-199.55902'	Yes
'KGC19:70a4622adc00dac981d1cbc407da93d6e5591bad'	'-199.79323'	Yes
'KGC19:b5792d92409c6daee37a0eaecb6eafe90dd7e7c0'	'-200.11891'	Yes
'KGC19: b52be1d2a2c8e089f1a1d9bfd84f257f53fd1c04'	'-200.30153'	No
'KGC19: 83af5e246f8886d630fbc7e692ebc32d60c7ba43'	'-200.4366'	Yes
'KGC19: 4ad56700823b5e8696b7b1f5aac6f16d9c1fadc8'	'-201.81355'	No
'KGC19: ff7807b699f1469608f627c6bafead27b815fb6c'	'-201.99171'	No
'KGC19: 4c9c90c73e40435b0e5d80f9a8d7df3c5cd670dc'	'-202.51756'	Yes
'KGC19: b21316debb3cfac6820f0aa2b04028a3b63bc587'	'-202.8655'	Yes
'KGC19: efeaa7412bc8c74a4a4ca3365e6a1d4d19c3c409'	'-203.18156'	Yes

Table 4: Output – sorted Hypothesis for true triple validation

for the evaluation we have used triple set of: S & O mentioned in the table 5 while Predicate is used as: **'KGC19:has_AssociatedArticles'**

'KGC19:hzzm3fcv'	'KGC19: 83af5e246f8886d630fbc7e692ebc32d60c7ba43'
'KGC19:hzzm3fcv'	'KGC19:6c915a41fb646b41b1d30bc07049223a33a7a7ee'
'KGC19:hzzm3fcv'	'KGC19: 4c9c90c73e40435b0e5d80f9a8d7df3c5cd670dc'
'KGC19:anwoyhdd'	'KGC19:4d177174a18d68179fcffdb54ea1b2f3b19abd50'
'KGC19:anwoyhdd'	'KGC19:4ad56700823b5e8696b7b1f5aac6f16d9c1fadc8'
'KGC19:anwoyhdd'	'KGC19:ff7807b699f1469608f627c6bafead27b815fb6c'
'KGC19:gbmjuhgv'	'KGC19:d08c4a6506c1a9b8fe7ace82d4c14f9636e1c33a'
'KGC19:gbmjuhgv'	'KGC19:0f7bb2b30b0eba1a065a6dfc88dbbd99053ff1ba'
'KGC19:gbmjuhgv'	'KGC19:81b416703ca6e099dec54d55e5fa56e532f6aa9e'

Table 5: Input – corruption set of Subject & object (Test_triple2)

Now as mentioned ComplEx, HolE and DistMult is giving very poor performance right now

on our KG. So, we are currently focusing on TransE as the base model for the task of recommendation. Besides TransE is the very first, simple and traditional model out there.
For the evaluation we can corrupt the entire triple or we can corrupt the specific side if the KG is big enough. Because for the practical performance it is under stable and efficient task. Evaluation metrices for the TransE model with respect to the corruption side id mentioned below. Where corrupt_side parameters indicate the value for evaluate_performance it can take on the following values:

- s for subject corruption only
- for object corruption only
- s+o for subject and object corruption. Returns a single rank.
- s,o for subject and object corruption separately (default). Returns 2 ranks. This is equivalent to calling evaluate_performance twice with s and o.

as discussed above the MRR and Hits@N values are considered good if they are closer to the 1, however the factor of corruption size is also needs to be consider. So, in our case we seen that size is too big. And the model performance in a practical scenario should be considered for the first 10% of the corruption. So, we have evaluated Hits@10 and Hits@20 to check how good our model performs with respect to the first 10% and 20% of the corruption.

a) TransE Evaluation:

Distinct entities used for corruption is counted: 1098440

Corruption side	**MR**	**MRR**	**Hits@10**	**Hits@20**
O	168156.777	6.45086e-05	0.444	0.555
S	**62807.88**	**5.32668e-04**	**0.666**	**1.0**
S, O	115482.33	2.98588e-04	0.555	0.777
S + O	230963.66	4.16786e-05	0.222	0.555

Table 6: Output – TransE Evaluation

We can clearly observe that TransE model on KGC19 is performing very well overall. Yet the best performance can be considered for KGC19 is corrupting subject side instead of object side. Hits@10 is more than half and Hits@20 is giving 1.0 as output that means first 20% of the corruption contains all the true triple and that can be considered as a good result.

Now next we have evaluated the performance of the model based on querying top N triples. For that we have evaluated model performance 3 times, two is using entity list for subject and object respectively and one is without the entity list.

b) Top N = 8, Entity list = list_of_AssociatedArticles

```
triples, scores = query_topn(model, top_n=8,
head='KGC19:hzzm3fcv', relation='KGC19:has_AssociatedArticles', tail=None,
ents_to_consider=list_of_AssociatedArticles,
rels_to_consider=None )
```

Output:

```
Score: -196.79281616210938
['KGC19:hzzm3fcv'
 'KGC19:has_AssociatedArticles'
 'KGC19:6c915a41fb646b41b1d30bc07049223a33a7a7ee']
Score: -197.7518310546875
['KGC19:hzzm3fcv'
 'KGC19:has_AssociatedArticles'
 'KGC19:4d177174a18d68179fcffdb54ea1b2f3b19abd50']
Score: -199.55902099609375
['KGC19:hzzm3fcv'
 'KGC19:has_AssociatedArticles'
 'KGC19:210e021ba2050d395326dc70f6f5505f24b03e39']
Score: -199.79322814941406
['KGC19:hzzm3fcv'
 'KGC19:has_AssociatedArticles'
 'KGC19:70a4622adc00dac981d1cbc407da93d6e5591bad']
Score: -200.11891174316406
['KGC19:hzzm3fcv'
 'KGC19:has_AssociatedArticles'
 'KGC19:b5792d92409c6daee37a0eaecb6eafe90dd7e7c0']
Score: -200.30152893066406
['KGC19:hzzm3fcv'
 'KGC19:has_AssociatedArticles'
 'KGC19:b52be1d2a2c8e089f1a1d9bfd84f257f53fd1c04']
Score: -200.4365997314453
['KGC19:hzzm3fcv'
 'KGC19:has_AssociatedArticles'
 'KGC19:83af5e246f8886d630fbc7e692ebc32d60c7ba43']
Score: -201.8135528564453
['KGC19:hzzm3fcv'
 'KGC19:has_AssociatedArticles'
 'KGC19:4ad56700823b5e8696b7b1f5aac6f16d9c1fadc8']
```

c) Top N = 15, Entity list = List_of_Corduid

Where:

```
list_of_Corduid = [
      'KGC19:anwoyhdd',
      'KGC19:005xh6cg',
      'KGC19:hzzm3fcv',
```

```
    'KGC19:gbmjuhgv'
]
```

```
triples, scores = query_topn(model, top_n=8,
head=None, relation='KGC19:has_AssociatedArticles',
tail='KGC19:4d177174a18d68179fcffdb54ea1b2f3b19abd50',
ents_to_consider=list_of_Corduid,
rels_to_consider=None)
```

Output:

```
Score: -197.7518310546875
['KGC19:hzzm3fcv'
 'KGC19:has_AssociatedArticles'
 'KGC19:4d177174a18d68179fcffdb54ea1b2f3b19abd50']
Score: -200.0795135498047
['KGC19:anwoyhdd'
 'KGC19:has_AssociatedArticles'
 'KGC19:4d177174a18d68179fcffdb54ea1b2f3b19abd50']
Score: -201.60638427734375
['KGC19:005xh6cg'
 'KGC19:has_AssociatedArticles'
 'KGC19:4d177174a18d68179fcffdb54ea1b2f3b19abd50']
Score: -201.6295623779297
['KGC19:gbmjuhgv'
 'KGC19:has_AssociatedArticles'
 'KGC19:4d177174a18d68179fcffdb54ea1b2f3b19abd50']
```

d) Top N = 15, Entity list = None (All distinct entites)

```
triples, scores = query_topn(model, top_n=15,
head='KGC19:hzzm3fcv', relation='KGC19:has_AssociatedArticles', tail=None,
ents_to_consider=None,
rels_to_consider=None)
```

Output:

```
Score: -194.5412139892578
['KGC19:hzzm3fcv'
 'KGC19:has_AssociatedArticles'
 'KGC19:6a169438a0da03e267a038ed7f90daed3edd7cce']
Score: -194.5775909423828
['KGC19:hzzm3fcv'
 'KGC19:has_AssociatedArticles'
 'KGC19:9cc79d84407d895d042eab8eb9dd338609f7b570']
Score: -194.68699645996094
['KGC19:hzzm3fcv'
 'KGC19:has_AssociatedArticles'
 'KGC19:4832e43e341b59b2121495082577852202dcd5e0']
Score: -194.78042602539062
['KGC19:hzzm3fcv'
 'KGC19:has_AssociatedArticles'
 'KGC19:f8ca30dc26327dd0e1eb6f23e7aef0838fa886bc']
Score: -194.7874755859375
```

```
['KGC19:hzzm3fcv'
 'KGC19:has_AssociatedArticles'
 'KGC19:24dcc80f0b4073329cbe437e3da4b7b32004544e']
Score: -194.85696411132812
['KGC19:hzzm3fcv'
 'KGC19:has_AssociatedArticles'
 'KGC19:69c469c998861af6b1cf644691790d87c6b79048']
Score: -194.87342834472656
['KGC19:hzzm3fcv'
 'KGC19:has_AssociatedArticles'
 'KGC19:ee9ecf938304f7f068074ccc6271845eb6128c76']
Score: -194.9174346923828
['KGC19:hzzm3fcv'
 'KGC19:has_AssociatedArticles'
 'KGC19:fec80a5e43e3465ea29efc84f69e898869a7f60e']
Score: -194.9423828125
['KGC19:hzzm3fcv'
 'KGC19:has_AssociatedArticles'
 'KGC19:bb65464b902f4b80dcccd76ebd70c37ee9f5a594']
Score: -195.0416717529297
['KGC19:hzzm3fcv'
 'KGC19:has_AssociatedArticles'
 'KGC19:bad721bd5425837dac395a2ba1af0570dce83f82']
Score: -195.10092163085938
['KGC19:hzzm3fcv'
 'KGC19:has_AssociatedArticles'
 'KGC19:220593adee4eec8b5aec178d78f796b028f1de49']
Score: -195.14892578125
['KGC19:hzzm3fcv'
 'KGC19:has_AssociatedArticles'
 'KGC19:26a10ba5e6f4a2569cff458fd4c42e0b26e48059']
Score: -195.15126037597656
['KGC19:hzzm3fcv'
 'KGC19:has_AssociatedArticles'
 'KGC19:b141e40390df4a96a7ebcf6aff4be3bf39a56e17']
Score: -195.1827850341797
['KGC19:hzzm3fcv'
 'KGC19:has_AssociatedArticles'
 'KGC19:5b9d744c6d6c51df22a573cc2a2a4f2977e8d414']
Score: -195.19322204589844
['KGC19:hzzm3fcv'
 'KGC19:has_AssociatedArticles'
 'KGC19:047bdedc6d500d1602d174e5c46ed37bbeb4df8c']
```

e) Clustering:

Apart from this retrieval and querying top n recommendation there is much more we can do with knowledge graph. Just to show you the glimpse of it, Ampligraph also provide the API to find the clustering from the provided knowledge graph, we can also find nearest neighbors, we can discover new facts from existing KG, we can also find duplicates from KG and many more. Point to note here is that all of that are not limited to the Ampligraph library but instead we are recommending it due to its ease of use and efficiency. For example, we have a set of entities here for which we are going to cluster them using ampligraph API called find_Cluster it can support various types of algo we used KMeans.

```
all_entities = [
'KGC19:hzzm3fcv',
'KGC19:83af5e246f8886d630fbc7e692ebc32d60c7ba43',
'KGC19:6c915a41fb646b41b1d30bc07049223a33a7a7ee',
'KGC19:4c9c90c73e40435b0e5d80f9a8d7df3c5cd670dc',
'KGC19:anwoyhdd',
'KGC19:4d177174a18d68179fcffdb54ea1b2f3b19abd50',
'KGC19:4ad56700823b5e8696b7b1f5aac6f16d9c1fadc8',
'KGC19:ff7807b699f1469608f627c6bafead27b815fb6c',
'KGC19:gbmjuhgv',
'KGC19:d08c4a6506c1a9b8fe7ace82d4c14f9636e1c33a',
'KGC19:0f7bb2b30b0eba1a065a6dfc88dbbd99053ff1ba',
'KGC19:81b416703ca6e099dec54d55e5fa56e532f6aa9e',
'KGC19:snldtqsn',
'KGC19:5692a8a106385686787b73c718b9d9991b96299e',
'KGC19:w8sjyclm',
'KGC19:94545e8428017783b5cc94aa23464874689aef51',
'KGC19:488bpx3z',
'KGC19:fc6479a39fefb7814acfcfc8d277659c43ca6e02',
]
```

Output: from the output we can see the well-defined two cluster using KMeans

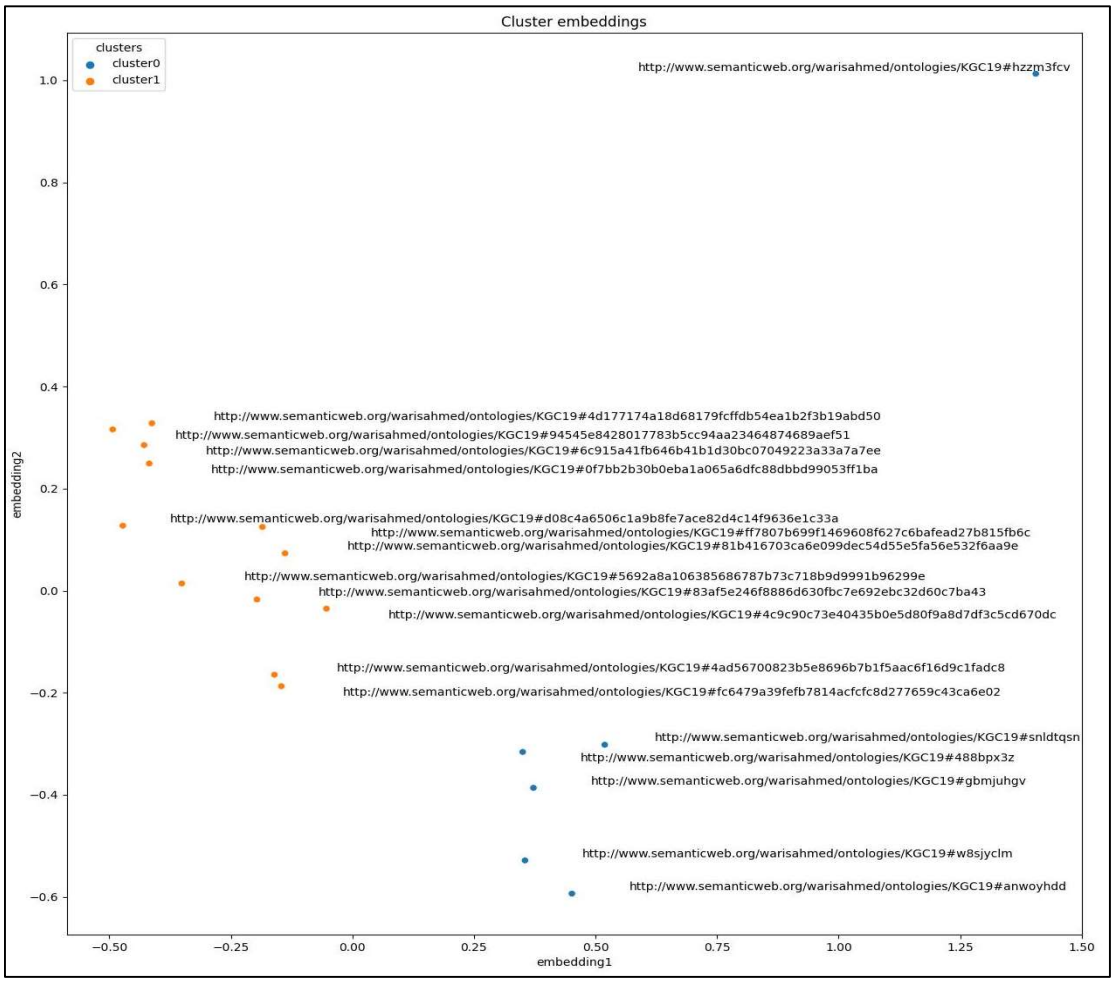


Figure 20: Output – Clustering using KMeans

5.5 Available Database and Datasets

With this discussion we would like to also mentioned some of the useful tools and dataset that we found. As a data storage lots of database available to store knowledge graph and graph data. And most of the NoSQL databases are used to store the KG. Some are listed below [14][15][18].

Database Name	Link	Database Model
Neo4j	https://neo4j.com/	Graph
GraphDB	https://graphdb.ontotext.com/	Multi Model
Cosmos DB: Azure	Introduction to Azure Cosmos DB	Multi Model
OrientDB	https://orientdb.org/	Multi Model
ArangoDB	https://www.arangodb.com/	Multi Model
Janus Graph	https://janusgraph.org/	Graph
Virtuoso	https://virtuoso.openlinksw.com/	Multi Model
Amazon Neptune	https://aws.amazon.com/neptune/	Multi Model
Stardog	https://www.stardog.com/	Multi Model
Dgraph	https://dgraph.io/	Graph

Table 7: List of Database

Also, there are many general as well as domain specific datasets available and we can use any according to our need. Some of the popular general datasets are mentioned below along with few covid-19 datasets [14][15].

Dataset Name	Description
Kaggle: Cord-19 [24]	"COVID-19 Open Research Dataset (CORD-19). it is a resource of over 500,000 scholarly articles, including over 200,000 with full text, about COVID-19, SARS-CoV-2, and related coronaviruses."
CORONAVIRUS (COVID-19) TWEETS DATASET [25]	"Dataset includes CSV files that contain IDs and sentiment scores of the tweets related to the COVID-19 pandemic. The real-time Twitter feed is monitored."

AYLIEN: COVID-19 [26]	"Corona virus news datasets"
WordNet [27]	"A free large lexical database of English from Princeton University."
YAGO [28]	"A huge semantic knowledge base, derived from Wikipedia, WordNet and GeoNames."
DBPedia [29]	"DBpedia is a crowd-sourced community effort to extract structured content from the information created in various Wikimedia projects."
Wikidata [30]	"It is a free, multilingual, dataset collecting structured data to provide support for Wikipedia, Wikimedia Commons."
Google KG [31]	"Google's Knowledge Graph has millions of entries that describe real-world entities."

Table 8: List of Datasets

CHAPTER 6
CONCLUSION AND FUTURE WORK

In the situation of the COVID-19 pandemic, we need a faster information retrieval system for knowledge discovery. Also, with common structure the proposed system uses KG for RS and information retrieval. A basic approach is mentioned to achieve this task. We have also proposed our own work as KGC19 (Knowledge Graph of Covid-19 Scholarly Articles), which is a KG based on the cord-19 dataset. Using KGC19, we have shown some of the information retrieval results using SPARQL queries, the GraphDB database, and the Ampligraph library in Python. Through the literature, we observed that KGE is a popular and easy-to-use technique for recommendation. That is why we have trained different models using the rdf2vec approach and traditional KGE techniques for our new KGC19 graph. As a result, we determined that TrasnE is the best model for KGC19 right now. We have shown evaluation metrics such as MR, MRR, and Hits@N. We observed that the resulting Hits@20 was able to achieve exactly 1 outcome, which indicates that the trained model is good enough for now considering the large size of KGC19. So, the proposed KGC19 as a new work can benefit the semantic web and be used to improve traditional RS and information retrieval systems, particularly in the area of COVID-19 analysis and knowledge discovery.

Now that we have our KGC19 graph and we have achieved some of the desired results, the future work is aimed towards enhancing KGC19 for its usage. One of them is to increase the MRR and Hits@N values. For that, enhancement using different KGE models can be done. RS can be achieved using inference algorithms too, such as DKN and Ripple Network, as mentioned in the literature. So, we can try to find the best suited algorithm or model for the RS. Basically, a newer approach and graph are generated now, so we can focus on the enhancement as a future work.

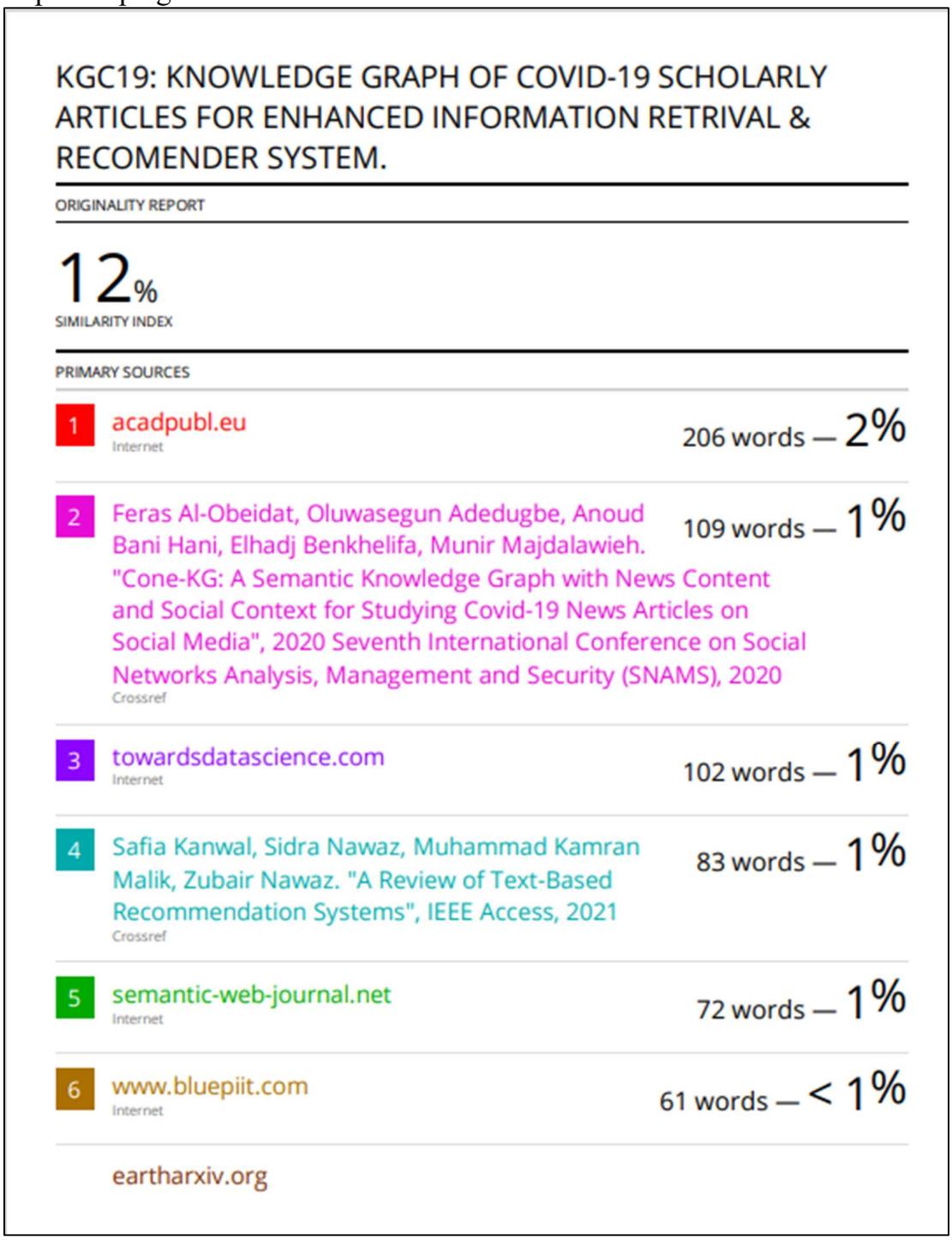
REFERENCES

1. Al-Obeidat, Feras, et al. "Cone-KG: A Semantic Knowledge Graph with News Content and Social Context for Studying Covid-19 News Articles on Social Media." *2020 Seventh International Conference on Social Networks Analysis, Management and Security (SNAMS)*. IEEE, 2020.
2. Jaradeh, Mohamad Yaser, et al. "Open research knowledge graph: next generation infrastructure for semantic scholarly knowledge." *Proceedings of the 10th International Conference on Knowledge Capture*. 2019.
3. Wise, Colby, et al. "COVID-19 knowledge graph: accelerating information retrieval and discovery for scientific literature." *arXiv preprint arXiv:2007.12731* (2020).
4. Zhao, Zhanfang, Sung-Kook Han, and In-Mi So. "Architecture of knowledge graph construction techniques." *International Journal of Pure and Applied Mathematics* 118.19 (2018): 1869-1883.
5. Patel, Angira Amit, and Jyotindra N. Dharwa. "An integrated hybrid recommendation model using graph database." *2016 International Conference on ICT in Business Industry & Government (ICTBIG)*. IEEE, 2016.
6. Kanwal, Safia, et al. "A Review of Text-Based Recommendation Systems." IEEE Access 9 (2021): 31638-31661.
7. Wang, Yuequn, et al. "An Enhanced Multi-Modal Recommendation Based on Alternate Training with Knowledge Graph Representation." IEEE Access 8 (2020): 213012-213026.
8. Wang, Hongwei, et al. "multi-task feature learning for knowledge graph enhanced recommendation." The World Wide Web Conference. 2019.
9. Wang, Hui, Zichun Le, and Xuan Gong. "Recommendation System Based on Heterogeneous Feature: A Survey." IEEE Access 8 (2020): 170779-170793.
10. Li, Weizhuo, Guilin Qi, and Qiu Ji. "Hybrid reasoning in knowledge graphs: Combing symbolic reasoning and statistical reasoning." Semantic Web 11.1 (2020): 53-62.
11. Liu Jiangzhou, and Li Duan. "A Survey on Knowledge Graph-Based Recommender Systems." 2021 IEEE 5th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC). Vol. 5. IEEE, 2021.
12. Guo, Qingyu, et al. "A survey on knowledge graph-based recommender systems." IEEE Transactions on Knowledge and Data Engineering (2020).
13. Nvidia Developer Blog, "How to Build a Winning Recommendation System, Part 1",https://developer.nvidia.com/blog/how-to-build-a-winning-recommendation-system-part-1/
14. Ji, Shaoxiong, et al. "A survey on knowledge graphs: Representation, acquisition, and applications." IEEE Transactions on Neural Networks and Learning Systems (2021).
15. GitHub, "totogo/awsome-knowledge-graph", Awesome Knowledge Graph – github
16. "Knowledge Graph", https://en.wikipedia.org/wiki/Knowledge_graph
17. Ontotext, "What is Knowledge Graph", https://www.ontotext.com/knowledgehub/fundamentals/what-is-a-knowledge-graph/
18. C#corner, "Most Popular Graph Database", https://www.csharpcorner.com/article/most-popular-graph-databases/
19. Wikipedia, "Knowledge base recommender system", https://en.wikipedia.org/wiki/Knowledge-based_recommender_system
20. Noy, Natalya F., and Deborah L. McGuinness. "Ontology development 101: A guide to creating your first ontology." (2001).
21. Ma, Xiaogang. "Knowledge graph construction and application in geosciences: A review." (2021).

22. Tan, Jiyuan, et al. "Research on the Construction of a Knowledge Graph and Knowledge Reasoning Model in the Field of Urban Traffic." *Sustainability* 13.6 (2021): 3191.
23. Stanford University, "What is Knowledge Graph", https://web.stanford.edu/class/cs520/2020/notes/What_is_a_Knowledge_Graph.html
24. Kaggle, "COVID-19 Open Research Dataset Challenge (CORD-19)", https://www.kaggle.com/allen-institute-for-ai/CORD-19-research-challenge
25. IEEE Data Port, "CORONAVIRUS (COVID-19) TWEETS DATASET", https://ieee-dataport.org/open-access/coronavirus-covid-19-tweets-dataset
26. Aylien, "Free Coronavirus News Dataset – Updated", https://aylien.com/blog/free-coronavirus-news-dataset
27. Princeton University, WordNet - A Lexical Database for English, "What is WordNet?", https://wordnet.princeton.edu/
28. Yago, "YAGO: A High-Quality Knowledge Base", https://yago-knowledge.org/
29. DBPedia, "Global and Unified Access to Knowledge Graphs", https://www.dbpedia.org/
30. Google Search Central, "Google Knowledge Graph Search API", https://developers.google.com/knowledge-graph
31. Bluepi, "Classifying Different types of Recommender Systems ", https://www.bluepiit.com/blog/classifying-recommender-systems/
32. Jackson wu, "Knowledge Based Recommender system: An Overview", https://medium.com/@jwu2/knowledge-based-recommender-systems-an-overview-536b63721dba
33. Google, "Recommender System", Introduction | Recommendation Systems | Google Developers
34. Chicaiza, Janneth, and Priscila Valdiviezo-Diaz. "A comprehensive survey of knowledge graph-based recommender systems: Technologies, development, and contributions." Information 12.6 (2021): 232.
35. Zeng, Wanghong, Hongxing Liu, and Yuqing Feng. "Construction of Scenic Spot Knowledge Graph Based on Ontology." 2019 18th International Symposium on Distributed Computing and Applications for Business Engineering and Science (DCABES). IEEE, 2019.
36. Sajisha, P. S., Anoop VS, and K. A. Ansal. "Knowledge Graph-based Recommendation Systems: The State-of-the-art and Some Future Directions."
37. Ristoski, Petar, and Heiko Paulheim. "Rdf2vec: Rdf graph embeddings for data mining." International Semantic Web Conference. Springer, Cham, 2016.
38. Palumbo, Enrico, et al. "Knowledge graph embeddings with node2vec for item recommendation." European semantic web conference. Springer, Cham, 2018.
39. Ontotext, "what is ontologies", https://www.ontotext.com/ontologies
40. Towards Data Science, "Summary of Translate Model for Knowledge Graph Embedding", https://towardsdatascience.com/summary-of-translate-model-for-knowledge-graph-embedding-29042be64273
41. Getting started with RDFLib — rdflib 6.1.1 documentation
42. AmpliGraph — AmpliGraph 1.4.0 documentation
43. ECAI 2020 KGE Tutorial - Hands on Session - Colaboratory (google.com)
44. Loading Data — GraphDB Free 9.10.0 documentation (ontotext.com)
45. GitHub- dwslab/jRDF2Vec: A high-performance Java Implementation of RDF2Vec, https://github.com/dwslab/jRDF2Vec
46. GitHub - wlin12/wang2vec: Extension of the original word2vec using different architectures, https://github.com/wlin12/wang2vec

Plagiarism Report from University:

Reported plagiarism is 12%



KGC19: KNOWLEDGE GRAPH OF COVID-19 SCHOLARLY ARTICLES FOR ENHANCED INFORMATION RETRIVAL & RECOMENDER SYSTEM.

ORIGINALITY REPORT

12%
SIMILARITY INDEX

PRIMARY SOURCES

1	acadpubl.eu Internet	206 words — 2%
2	Feras Al-Obeidat, Oluwasegun Adedugbe, Anoud Bani Hani, Elhadj Benkhelifa, Munir Majdalawieh. "Cone-KG: A Semantic Knowledge Graph with News Content and Social Context for Studying Covid-19 News Articles on Social Media", 2020 Seventh International Conference on Social Networks Analysis, Management and Security (SNAMS), 2020 Crossref	109 words — 1%
3	towardsdatascience.com Internet	102 words — 1%
4	Safia Kanwal, Sidra Nawaz, Muhammad Kamran Malik, Zubair Nawaz. "A Review of Text-Based Recommendation Systems", IEEE Access, 2021 Crossref	83 words — 1%
5	semantic-web-journal.net Internet	72 words — 1%
6	www.bluepiit.com Internet	61 words — < 1%
	eartharxiv.org	

Master of Technology

(Dissertation Review Card)

Name of Student: Bungawala Warisahmed N.

Enrollment No.: 2 0 0 3 0 3 2 0 1 0 0 2

Student's Mail ID:- ~~warris~~ 200303201002@paruluniversity.ac.i

Student's Contact No.: 7096113200

College Name: PIET

College Code: 0 3

Branch Code: 0 5 Branch Name: Computer Science Engineering

Broad area of Title: Semantic Web

Title of Dissertation: Recommender System based on knowledge graph for covid-19 scientific literature and social media

Supervisor's Details	Co-Supervisor's Details OR External Supervisor's Details
Name: Dr. Jaimeel Shah	Name: Prof. Darshana Parmar
Institute: PIET	Institute/Organization: PIET
Institute Code: 03	Institute Code (in case of co-supervisor): 03
Mail ID: jaimeel.shah@paruluniversity.ac.in	Mail ID: darshna.parmar2809@paruluniversity.ac.in
Mobile No. 96385 04566	Mobile No. 95868 83925

M.Tech. Dissertation Guidelines, FET, Parul University Page 23 of 28

Comments for Phase-I Dissertation
(Internal) (Subject Code) (Semester 3)

Date: 28/08/21

Sr. No.	Comments given by internal review panel	Modification done based on Comments (To be filled by Supervisor)
1.	In deapth parameters required	Done
2.	How to train the KG	Done
3.	How to extract knowledge from KG.	Done
4.	performance parameters need to be define	Done
		Name of Supervisor:
		Sign of Supervisor:

Particulars	Internal Review Panel		
	Expert 1	Expert 2	Expert 3
Name:	Dr. Daxa Vekariya	Nitin Pandya.	
Institute:	PIET-CSE	PIET-CSE	
Institute Code:	03	03	
Mobile No.:	9909969540	9825761233	
Sign:	28/8/2021	28/8/21.	

Comments for Phase-I Dissertation
(External P) (Semester 3)

Date: 14-10-21

Enrollment No. of Student: 200303201002

Title : Recommender System based on Knowledge Graph for covid-19 scientific literature and social media

1. Problem Definition (title) is appropriate or not (Yes/ No) yes

2. Clarity of objectives. (Yes/ No) yes

M.Tech. Dissertation Guidelines, FET, Parul University

Comments for Phase-I Dissertation (Subject Code)
(External) (Semester 3)

Date: 24-10-21

Enrollment No. of Student: 2003032010 02

Sr. No.	Comments given by External Examiners	Modification done based on Comments (To be filled by Supervisor)
-	**By Dr. Dhruti Sharma**	
1)	Overall Presentation is good.	
2)	Only a suggestion is to include performance metrics	Done
	(Time, Space Complexity) for the propose approach	
-	**By Dr. Bela Shrimali**	
1)	Identify the comparison parameters to compare the	Done
	Method with existing method	
2)	Overall presentation is good	
		Name of Supervisor: Jaimal Shah
		Sign of Supervisor:

- *Approved* ☐
- *Approved with suggested recommended changes* ☑
- *Not Approved* ☐

Please tick on any one

Details of External Examiners:

Particulars	Expert 1	Expert 2
Name:	Dr. Dhruti Sharma	Dr. Bela Shrimali
Institute/University/Organization:	SCET SURAT	LDRP Gandhinagar
Mobile No.:	9925010162	9925222054
Sign:		

M.Tech. Dissertation Guidelines, FET, Parul University

Master of Technology

(Dissertation Review Card)

Name of Student: Banglawala Warisahmed Nasheullah

Enrollment No.: 200303202002

Student's Mail ID:- 200303202002@parul university.ac.in

Student's Contact No.: 7096113200

College Name: PIET

College Code: 03

Branch Code: 05 **Branch Name:** Computer Science Engineering

Broad area of Title: Knowledge based Systems

Title of Dissertation: Recommender system using Knowledge graph for Covid-19 Scientific Literature.

Supervisor's Details	
Name:	Dr. Jaimeet Shah
Institute:	PIET
Institute Code:	03
Mail ID:	jaimeet.shah@parul university.ac.in
Mobile No.	96385 04566

Co-Supervisor's Details OR External Supervisor's Details	
Name:	Prof. Darshana Parwar
Institute/Organization:	PIET
Institute Code (in case of co-supervisor):	03
Mail ID:	darshana.parwar2809@paruluniversity.ac.in
Mobile No.	95868 83925

Comments for Phase-II Dissertation (Subject Code)
(Internal) (Semester 4)

Date: 05-02-22

Enrollment No. of Student: 200303201002

Sr. No.	Comments given by internal review panel	Modification done based on Comments (To be filled by Supervisor)
1.	complete the Implementation	
		Name of Supervisor:
		Sign of Supervisor:

Particulars	Internal Review Panel		
	Expert 1	Expert 2	Expert 3
Name :	Dr. Daxa Vekariya		
Institute :	PIET-CSE		
Institute Code :	09		
Mobile No. :	9909969540		
Sign :			