KGC19: Knowledge Graph Of COVID-19 Scholarly Articles for Enhanced Information Retrieval & Recommender System.

Warisahmed Bunglawala[1*], Jaimeel Shah[2], and Darshna Parmar[3]

Research Scholar, Dept. Computer science Engineering, Parul University,
Vadodara, INDIA.
warisahmedb@gmail.com [1]
Dr, Dept. Computer science Engineering, Parul University,
Vadodara, INDIA.
jaimeel.shah@paruluniversity.ac.in [2]
Prof, Dept. Computer science Engineering, Parul University,
Vadodara, INDIA.
darshana.parmar2809@paruluniversity.ac.in [3]

***Corresponding author**

Warisahmed Bunglawala
Dept. Computer science Engineering, Parul University,
Vadodara, INDIA.
warisahmedb@gmail.com [1]

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

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 itself. 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. Soon KG and ontology will be published online for the open access under the name KGC19.

Keywords: Knowledge Graph, Recommender System, Knowledge Graph Embedding, Information Retrieval, Ontology.

1. Introduction

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].

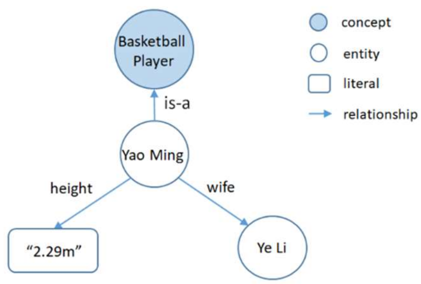


Figure 1: Idea of Knowledge Graph [10]

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.

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 further in this document.

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.
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].
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].

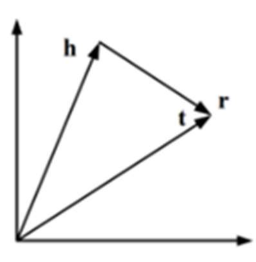
Knowledge graph embeddings are low-dimensional representations of the items and relations in a knowledge graph (KGEs). They give a generalizable context for inferring relationships throughout the entire KG. The embeddings of knowledge graphs are constructed in such a way that they fulfil certain characteristics, 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.

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 2: Idea of TransE & Loss Function [40]

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.

Major Objective of the research work includes:

- 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 on our own KG.

2. Material and methods

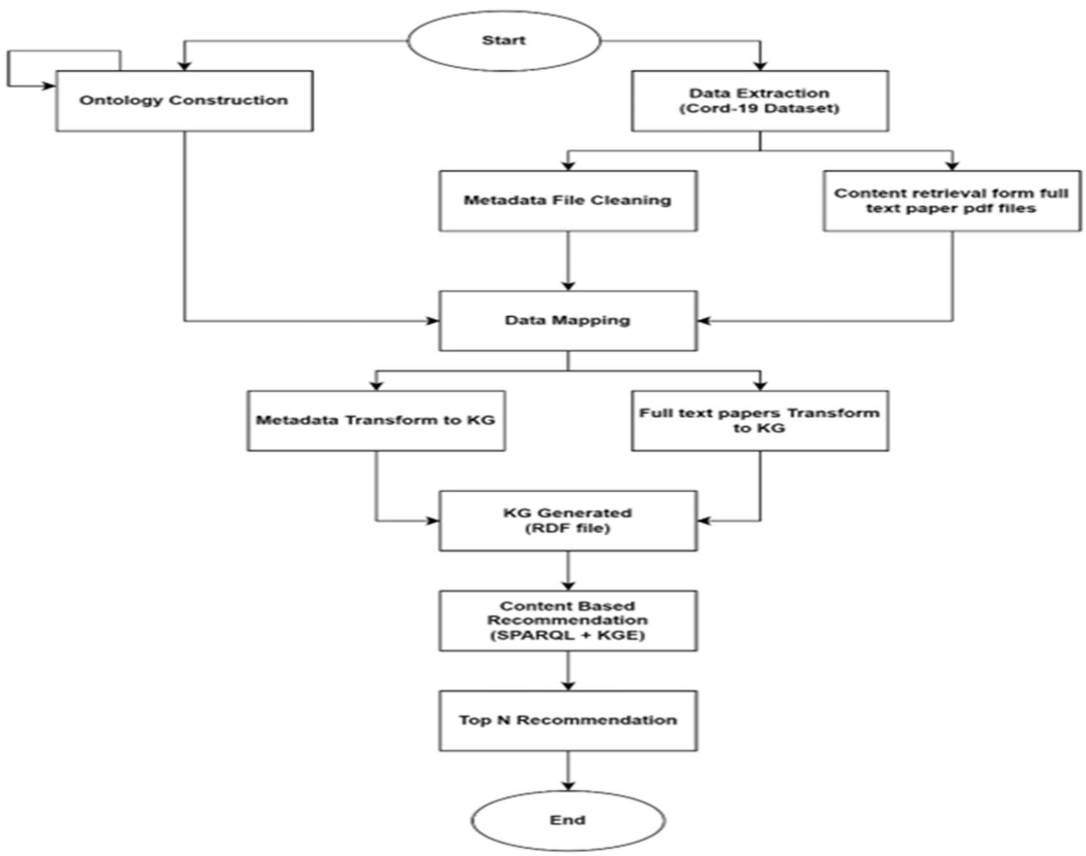


Figure 3: Proposed Diagram

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. 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 research. Dataset can be found at Kaggle website with name COVID-19 Open Research Dataset Challenge (CORD-19) [24].

2.2 Pre-processing

While creating KG 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.

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

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.

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 yet. 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.

3. Results

First of we created a KGC19 ontology which is based on cord19 dataset model and the created ontology is mentioned in the figure below. After that on this ontology the cord19 data is mapped and we created the KGC19 knowledge graph.

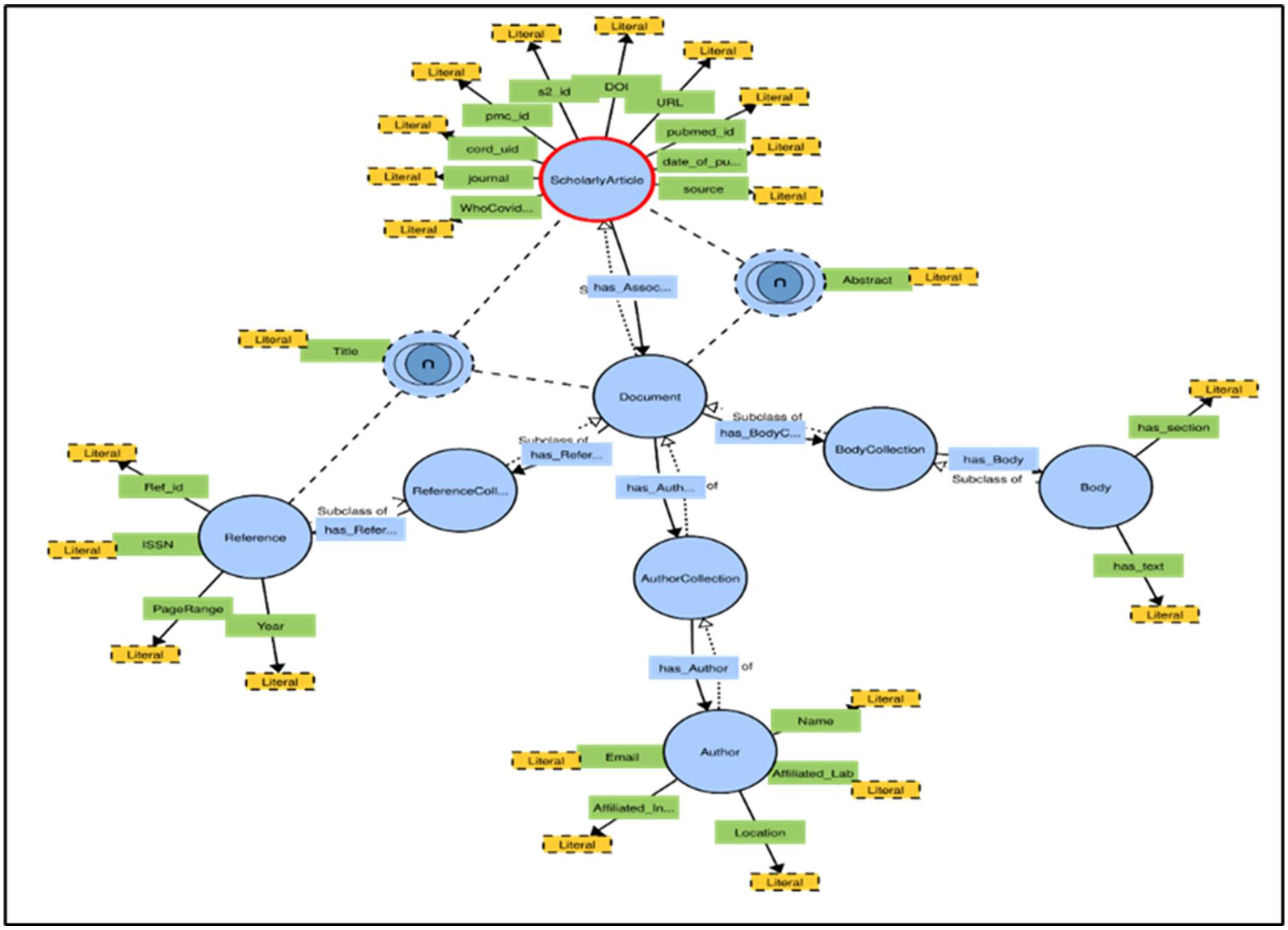


Figure 4: KGC19 ontology

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 1: KGC19 Distinct Subject, Predicate & Object count

Here we are presenting some of the SPARQL queries which we used to analyses and retrieve the results from the KGC19:

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

Medline is the Highest with count = 364937 articles

Query & Results:

```
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)
```

Source	Articles
Medline	364937
PMC	306249
WHO	126921
Elsevier	71624
MedRxiv	17582
ArXiv	11489
BioRxiv	7441

b) Listing all of the article containing author name as "wang"

More than 1000 articles have author name containing wang form the given data.

Query & Results:

```
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')
}
```

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 5: Output – Author Name which contains "Wang"

c) List all the Title Containing "Covid-19" + "Mental Health" word in it.

Found 757 Articles from KGC19

Query & Results:

```
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')) }
```

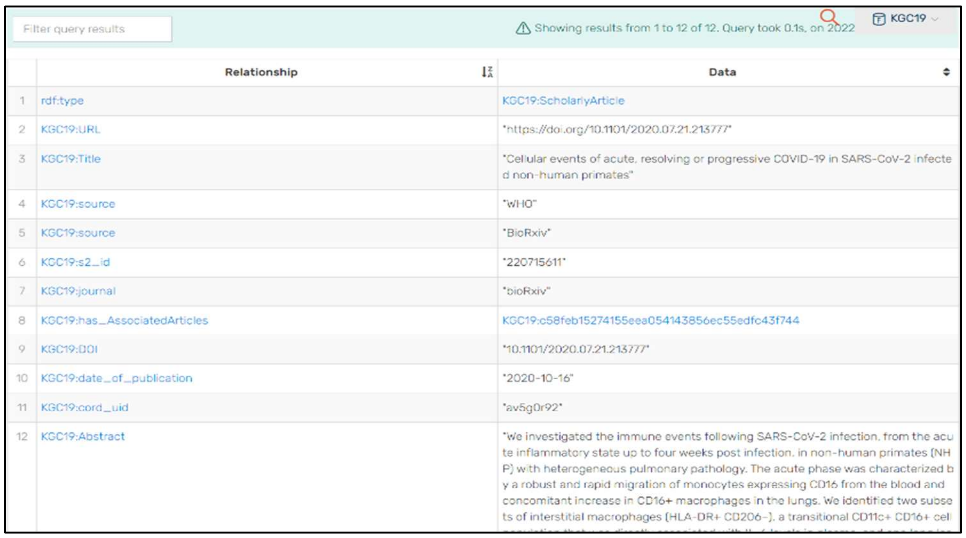
	Title
1	"Logical Framework Approach a Platform for Integrating the Mental Health and Nutritional Care for Controlling the Covid-19 Pandemic"
2	"Covid-19 and Mental Health: Could Visual Art Exposure Help?"
3	"Worries, Preparedness, and Perceived Impact of Covid-19 Pandemic on Nurses' Mental Health"
4	"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 Research 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 on 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. Psychiatric 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"

Figure 6: Output – Title containing "Covid-19" + "Mental Health"

d) List All the triple containing cord_uid "av5g0r92" as Subject

Query & Results:

```
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. }
```


Filter query results

Showing results from 1 to 12 of 12. Query took 0.1s, on 2022

KGC19

	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 7: Output – All the triple which have subject as "av5g0r92"

Now after that we have trained our KGC19 using different model such as rdf2vec and KGE models such as ComplEx and TransE but none of them were giving expected results except TransE. 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
```

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

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'**

a) TransE Evaluation:

Distinct entities used for corruption is counted: 1098440

Corruption side	**MR**	**MRR**	**Hits@10**	**Hits@20**
O	168156.77	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 4: Output – TransE Evaluation

We can also retrieve top N items using TransE model some of the results are mentioned below.

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']
```

KMeans Clustering: For the given list we have mapped KMeans Clustering and from the output we can see the well-defined two cluster using KMeans so here we can see that KGC19 is not only limited to recommendation but rather can be used for Information Retrieval and analysis purpose too.

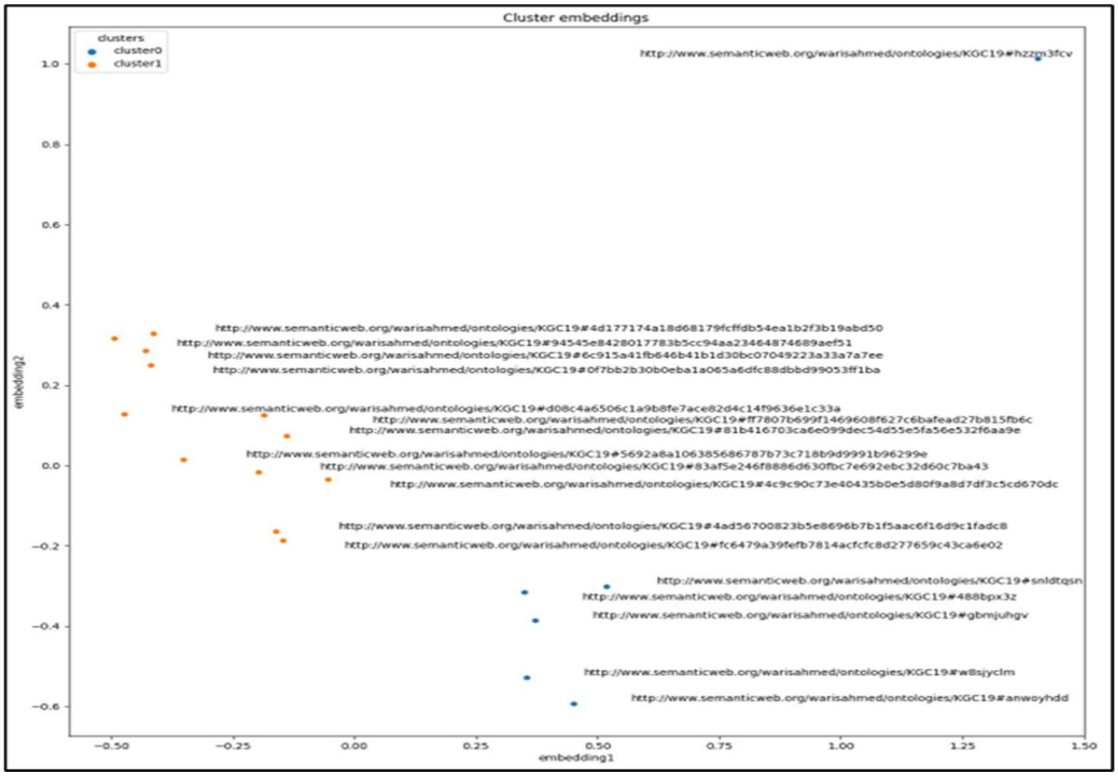


Figure 8: Output – Clustering using KMeans

4. Conclusions

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. We have 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.

5. Future Work

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