Research Paper Recommendation System (Final Stage Project Presentation)

Presented by: **Group 15**
Pankhi Khandelwal (U17CO099)
Himanshu Choudhary (U17CO101)
Shubhi Agarwal (U17CO109)
Aman Mishra (U17CO112)

Guided by: Dr. Rupa G. Mehta
Computer Engineering Department

May 12, 2021

Contents

Problem Statement

- Availability of infinite amount of data over web whose growth rate is exponential.
- Semantic comparison becomes almost infeasible.
- To build a graphical-network-model based on keywords, which correlates the input publication to its n-neighbors on the basis of their semantic relationships and associations.
- To use Natural Language Processing based techniques for filtering to reduce the search space for documents.

Workflow

- Research Paper Recommendation Approaches
- Research Paper Recommendation Systems and their Flow of Operation
- Keyword Extraction Techniques
- Context Aware Approaches
- WordNet Similarity Measures
- **Existing Research Work on Network-Based Approaches**

Various Existing Keyword Extraction Techniques

- Simple Statistical Approaches
  - Word Frequency
  - Term Frequency Inverse Document Frequency (TF-IDF)
  - Rapid Automatic Keyword Extraction (RAKE)
  - Yet Another Keyword Extractor (YAKE)
  - keyBERT
- Linguistic Approaches
- Machine Learning Approaches

Features of Context Aware Approaches

- Need to understand the context under which recommendation has to be done.
- Semantic and contextual information add humane touch to the output.
- No fixed syntax about the use of context-related information.
- For instance, context can cover
  - information that may include conditions under which the user has suggested an item
  - the relationship between different items
  - common consumers
- Such type of information is susceptible to change over a duration of time.

Existing Research Work on Network-based Approaches I

Table 1: Various features explored in Context-Aware Domain

AUTHOR	HIGHLIGHT
Zhao et. al. [1]	• The entire system was divided into two sections • One for researcher level analysis that involves analysis of social relations, users research interest • The other one is Document Level analysis that involves use of users publications to make suggestions.
Gao et. al. [2]	• Made use of layers of networks , with each network covering a different semantic relation • 4 layers suggested (Author, Paper, Keywords, Topics)

Existing Research Work on Network-based Approaches II

Turowski et.al. [3]	• Made use of mindmaps • User model formed based on mindmaps • Then similarities between model and papers is analysed
M. Eto [4]	• Co-citation based approach • the closeness between two cited documents is measured by three grades such that they appear (1) within a sentence, (2) within a paragraph and (3) across two paragraphs
Bhattacharya et. al. [5]	• Analysis of user keyword similarity in online social networks • Semantically related keywords lie within the same tree

Workflow

Makes use of a Graph based approach involving use of Keyword Co-Occurrence Network and Paper Keyword Associated Network.

Divided into two sections:

1. Model Preparation: Covers the entire network creation
2. Model Application: How user will interact with our system

Pre-processing and network creation stages

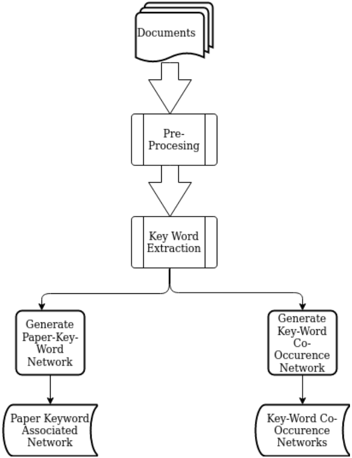


Figure 1: Block Diagram depicting the preprocessing and network creation stages

User interaction with the system

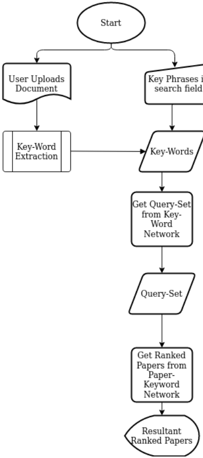


Figure 2: Block Diagram depicting the user interaction with the system

Simulation: Overview

1. Pre-processing
   - Stemming
   - Lemmatization
   - Tokenization
2. Keyword Extraction: RAKE ,YAKE, keyBERT
3. Evaluation Matrix for Keyword Extraction
4. Network 1 (Paper IDs as nodes): Paper Keyword Network
5. Network 2 (Keywords as nodes): Keyword Co-occurrence Network
6. Network 3 (Keywords as well as Paper IDs as nodes): Paper Keyword Association Network
7. Contextual Similarity Calculation to add weights to the network
8. Prediction Pipeline

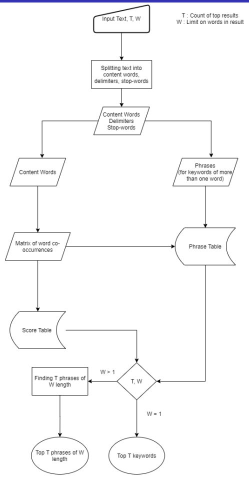
Preprocessing Techniques

- **Stemming:** It is the process of extracting the base from the given words by removing affixes from them, this extracted part is known as stem. It reduces the given words to common stem. For example, good will be reduced to "good" and better will be reduced to "bett" in stemmer.
- **Lemmatization:** It is the process of extracting the valid base (root word not root stem) from the given words by removing affixes from them, this extracted part is known as lemma. It reduces the given words to common lemma. For example, good and better both will be reduced to "good" in Lemmatization.
  Github Link: `https://github.com/Am-Coder/Document-Analysis/blob/master/preprocessing/preprocessing.py`
- **Tokenization:** It is a process of splitting up a larger body of text into smaller lines, words, or even creating words for a non-English language.
  Github Link: `https://github.com/Am-Coder/Document-Analysis/blob/master/preprocessing/tokenisation.py`

RAKE

1. Splitting the text into content, delimiters and stop words.
2. Then the algorithm splits the text at phrase delimiters and stopwords to create candidate expression.
3. Now after extraction, the algorithm creates a matrix of word co-occurrences.
4. After matrix is built, words are given a score and score table is generated.
5. Score = degree of word / frequency of word
6. Returns the result as top T keywords of W words from score table.

Github Link: `https://github.com/Am-Coder/Document-Analysis/blob/master/keywordExtraction.py`



YAKE

- In the first step of YAKE algorithm preprocessing of text is done and candidate terms are identified.
- In the next step feature extraction is performed on individual terms.
- In the third step, term scores are computed and combined to show the importance of each term.
- The fourth step generates and computes the candidate keyword score using n-gram generation.
- At last ,the fifth step compares likely similar keywords through the application of a deduplication distance similarity measure.

Github Link: `https://github.com/Am-Coder/Document-Analysis/blob/master/keywordExtraction3.py`

keyBERT

- Firstly, it creates a list of candidate keywords or keyphrases from a document.
- Next the document as well as the candidate keywords/keyphrases converted to numerical data.
- Finally, the candidates that are most similar to the document are extracted.
- To calculate the similarity between candidates and the document, cosine similarity between vectors is used.

Github Link: `https://github.com/Am-Coder/Document-Analysis/blob/master/keywordExtraction2.py`

Comparison

Table 2: Comparison of Keyword Extraction Techniques

Algorithm	MRR Score	MRR Rank	MAP Score	MAP Rank
RAKE	0.509	1	0.650	2
YAKE	0.456	2	0.652	1
keyBERT	0.320	3	0.530	3

Note: All algorithms offer a multilingual support.

Network 1 (Paper IDs as nodes): Paper Keyword Network

- A list of publication IDs for all the publications present in the dataset is generated in this system.
- Each element of the list acts as a node for an undirected graph.
- The weight of an edge is calculated by calculating the total number of common keywords between the two research papers depicted by the two nodes corresponding to that edge.

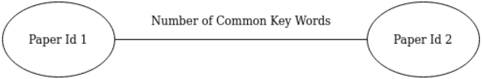


Figure 4: Paper-Keyword Network

For example, suppose there are two scholarly articles with paper ids 1 and 2, list of keywords for both are as shown below:

Keywords for paper 1: [K1, K2, K3, K4, K5, K6]
Keywords for paper 2: [K1, K2, K4, K6, K7, K8]

So, the number of common keywords in both the articles is 4 (viz., K1, K2, K4 and K6), hence the weight of the connecting edge would be 4.

Github Link:
`https://github.com/Am-Coder/Document-Analysis/blob/master/Paper-KeyWord-Network/graphGeneratorFromKeywords.py`

Network 2 (Keywords as nodes): Keyword Co-occurrence Network

- A list of common keywords extracted from all the publications present in the dataset is generated in this system.
- Each element of the list acts as a node for an undirected graph.
- The weight of an edge is calculated by calculating the co-occurrence frequency between the two nodes corresponding to that edge.

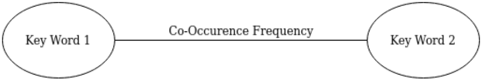


Figure 5: Keyword Co-occurrence Network

For example, suppose there are two keywords viz., K1 and K2, list of papers in which they are occurring ar as shown below:

Papers List for K1: [P1, P2, P3, P4, P5, P6]
Papers List for K2: [P1, P2, P6, P7, P8]

So, the number of articles having both K1 and K2 is 3 (viz., P1, P2 and P6), hence the weight of the connecting edge would be 3.

Github Link:
`https://github.com/Am-Coder/Document-Analysis/blob/master/Paper-KeyWord-Network/graphWithKeywordsAsNode.py`

Network 3 : Paper-Keyword Association Network

- Holds relationship between keywords and the papers in which they are present.
- No two keywords connected to each other.
- No two papers connected to each other.
- Query set used to generate prediction using this network.

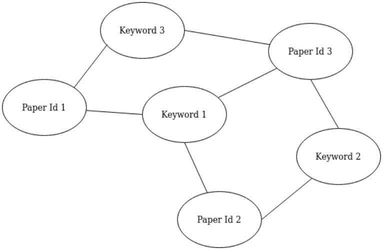


Figure 6: Paper-Keyword Association Network

Wordnet for Semantic Analysis

1. A large lexical database of English. Nouns, verbs, adjectives and adverbs are grouped into sets of cognitive synonyms (synsets).
2. We are using Path Similarity and Wu Palmer similarity to derive the semantic similarity between two keywords.
   - Path Similarity: It is output of 1 divided by the shortest distance between the given two words in wordnet taxonomy for an entity.
   - Wu-Palmer Similarity: It calculates relatedness using depths of two synsets along with taking the depth of their Lowest Common Ancestor.

$$Similarity = \frac{s1 + s2}{2 * n1 * n2} \tag{1}$$

Github Link: `https://github.com/Am-Coder/Document-Analysis/blob/working/wordnet.ipynb`

Wordnet Similarity Results

Table 3: Wordnet Similarity Results

Phrase 1	Phrase 2	Similarity
machine learning	deep learning	0.08571428571428572
computer science	computer engineering	0.11080586080586081
green grass	pasture	0.13675213675213674
a building by road side	sky-scrapper near by	0.35079365079365077
War dismantles	battle levelling	0.33333333333333333

Results: Not so promising?

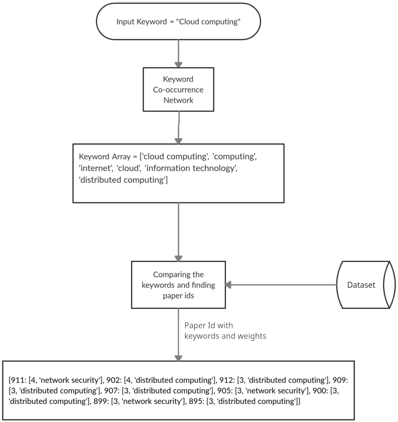
- A solid reason for this is that Wordnet is a general purpose lexical database and does not takes into account similarity based on scientific domains.
- A probable solution to this problem is the use of a custom ontology.

Workflow

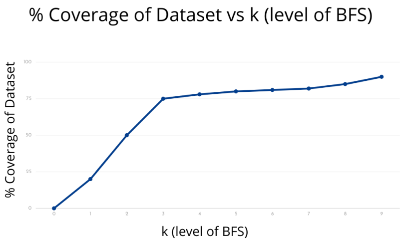
Consider the example shown in the flow diagram:

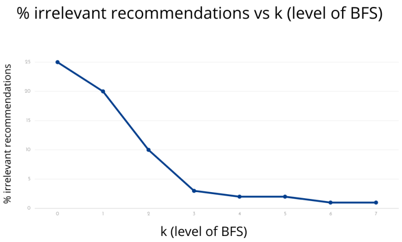
- First, the input keyword 'Cloud Computing' is used to create a query set from the keyword co-occurrence network. Three parameters have been considered to get the related keywords, viz., keyword co-occurrence frequency between two words, wordnet similarity measure and level of the BFS. By doing so, the required query set is generated.
- Using this query set, we then consult the keyword-document matrix and find the list of papers having keywords similar to that present in the query set. Currently, the exact match strategy is being used.
- In the output, we get the paper IDs along with the domain of the paper and the number of words that were common between the paper and the query set.
- Currently, we are using the number of common keywords for ranking the documents predicted by the system before making a final recommendation.

For experimental purpose, we have taken k=3 because in general scenario directly connected or nearest neighbours in a graph have more similarity index in the context in which they are connected. As we go deeper in the Breadth First Search, the nodes that are encountered, found to be less correlated with respect to the root.



How to choose k?





Approaches to generate the query set I

- Three parameters have been taken into consideration while deriving and evaluating different formulae to generate the query set. These are keyword co-occurrence frequency between two words, wordnet similarity measure and level of the BFS.
- Following symbols will represent different parameters involved in the further discussion:

$$k = \text{level of BFS}$$
$$W_{sim} = \text{Wordnet Similarity Measure}$$
$$E_{i,j} = \text{Keyword co-occurrence frequency of word i and word j}$$
$$C_{sim_j} = \text{Cumulative Similarity for word j}$$
$$T = \text{Minimum threshold } C_{sim_j} \text{ value}$$
$$l = \text{number of nodes at k-1 level}$$

- After doing a thorough study about the dependence of these on the similarity parameter between any two keywords, we have concluded that:

$$C_{sim_j} \propto \frac{(E_{i,j})(W_{sim})}{k} \tag{2}$$

- **Note:** For each category, a threshold has been set to remove the garbage output from the generated query set such that $C_{sim_j} >= T$.
- **Simple BFS Approach:** In this approach a breadth first search upto a level k over keyword co-occurrence network has been done to generate the query set. All the keywords lying in this scope are inclusively taken in the output.

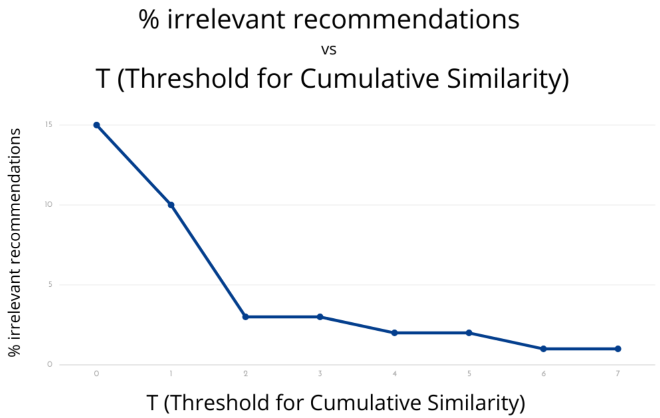
Approaches to generate the query set II

- **Linear Formulation Approach:** Here, after considering the direct proportion relationship between nature of keywords and their co-occurrence and contextual similarity parameters, a linear formula has been derived to get the cumulative similarity value between two keywords. The formula devised is as follows:
  For j_{th} node at level k,
  $$C_{sim_j} = \sum^{l} E_{i,j} + 10 * W_{sim} \tag{3}$$
  Here, $T = 4$
- **Basic Non-Linear Formulation Approach:** The output generated by the Linear Formulation Approach has been divided by the exponent value of the current level k of BFS in order to take into account the inverse relation characteristic occurring due to their distance in the network. The formula devised is as follows:
  For j_{th} node at level k,
  $$C_{sim_j} = \frac{\sum^{l} E_{i,j} + 10 * W_{sim}}{e^k} \tag{4}$$
  Here, $T = 2$

Approaches to generate the query set III

- **Exponential Formulation Approach:** This acts as a variant to the Basic Non-Linear Formulation Approach. Here the wordnet similarity value has been exponentiated to witness the effect caused due to this parameter as it comes to be a smaller value in the range of 0 to 0.5. The formula devised is as follows:
  For j_{th} node at level k,
  $$C_{sim_j} = \frac{\sum^{I} E_{i,j} + e^{W_{sim}}}{e^{k}} \tag{5}$$
  Here, $T = 2$
- **Enhanced Exponent Formulation Approach:** It is an enhanced version of Exponent Formulation Approach. The wordnet similarity component has been scaled up here by adding a multiplication factor of 10 to it. The formula is as follows:
  For j_{th} node at level k,
  $$C_{sim_j} = \frac{\sum^{I} E_{i,j} + e^{10 * W_{sim}}}{e^{k}} \tag{6}$$
  Here, $T = 4$

How to choose T?



Results to prove the relevance of Keyword Extraction techniques

- Results derived from user extracted keywords are much more consistent as compared to author labelled keywords as shown in the tables below.
- As for the example explained above,, the author labelled keywords, gives a prediction for a wide range of domains while the results from data extracted with RAKE and YAKE algorithms are more related to the queried domain.
- Moreover till now, only the top five keywords after extraction have been considered while preparing the network in the second case due to the computational capacity issues of the system. Once, the network becomes denser, then the related predictions are supposed to be more accurate.
- It also justifies the fact quoted by Zhao et. al. [1]. According to which for the article predictions, only the author-labeled keywords are used to represent the content of the given paper. But every research work contains a limited number of keywords that are insufficient to represent the whole content.

Github Link: `https://github.com/Am-Coder/Document-Analysis/tree/master/Pipeline`

Results derived from Author Labelled Keywords I

Table 4: Results derived from Author Labelled Keywords

Paper ID	Domain	Abstract
791	**relational databases**	Given research is related to databases used in big data.
914	network security	Cloud Computing is a new distributed computing paradigm. Use of autonomic computing in cloud computing especially in ERP has been explored in this research.
902	distributed computing	This paper discusses the concept of cloud computing. It also addresses some of the related issues, and available cloud computing implementation.
583	**relational databases**	The proposed model here deals with storage of health data in no sql databases which was found to be more effective than Relational Databases for handling such type of data. Implementation has been done in a cloud environment.

Results derived from Author Labelled Keywords II

919	distributed computing	In this paper, the author presented a successful implementation of a scalable low-level load balancer, implemented on the network layer.
912	distributed computing	This paper proposes a security framework to secure VMs Images in a virtualization layer in the cloud environment.
786	distributed computing	Based on Big Data security using HDFS.
785	**relational databases**	Based on Structured data (relational data) in the domain of Big Data.
525	distributed computing	This paper presents a sliding window-based dynamic load balancing algorithm, which specially aims at balancing the load among the heterogeneous nodes during the Hadoop job processing.

Results derived from Author Labelled Keywords III

401	network security	Developed a model combining cloud computing and ML related to Hadoop security.
326	image processing	In this paper, the first endeavor towards privacy-preserving image denoising from external cloud databases has been initiated.
320	**data structures**	Issue of allocating memory dynamically for VMs has been dealt with.
318	distributed computing	A paradigm for the computation of k-mer-based alignment-free methods for Apache Hadoop has been discussed.

Results derived from User-extracted Keywords I

Table 5: Results derived from User-extracted Keywords

Paper ID	Domain	Abstract
911	network security	Discussed **security issues in cloud computing.**
902	distributed computing	This paper **discusses the concept of cloud computing**. It also addresses some of the related issues, and available cloud computing implementation.
912	distributed computing	This paper proposes a security framework to secure VMs in a **virtualization layer in the cloud environment.**
909	distributed computing	Discusses **Mobile Cloud Computing** Security frameworks found in the literature related to Cloud Computing and its environment.
907	distributed computing	It explores heuristic task scheduling with artificial bee colony algorithms for **VMs in heterogeneous cloud computing.**

Results derived from User-extracted Keywords II

905	network security	Security solution for **Intrusion detection in cloud computing** has been explored.
900	distributed computing	Related to **scalability in Cloud Computing.**
899	network security	Algorithms for low overhead, edos attack, etc. on **cloud computing** have been proposed.
897	distributed computing	**Cloud Computing involved with autonomic computing** is the main focus here.
895	distributed computing	This research work focuses on the **security threats and Risk Assessments for cloud computing**, attack mitigation frameworks, and the risk-based dynamic access control for cloud computing.

Results derived after implementation from Prediction Pipeline

- This section contains the output generated by the prediction pipeline by utilising all the five methodologies.
- Further their results have been compared to conclude which one is the most efficient method for the given system.
- We have taken two test cases:

Test Case 1: Input Query Keywords = ['cloud computing', 'distributed computing', 'network']

Test Case 2: Input Query Keywords = ['machine learning', 'deep learning']

Results obtained from different methodologies (1)

Test Case 1: Input Query Keywords = ['cloud computing', 'distributed computing', 'network']

Table 5: Results for Test Case 1

Methodology	Domains of the recommended papers
Simple BFS Approach	'network security', 'distributed computing', 'relational databases', 'computer programming', 'parallel computing', 'algorithm design', **'computer vision'**, 'data structures', 'structured storage'
Linear Formulation Approach	'network security', 'distributed computing', 'parallel computing', **'relational databases'**, 'operating systems'
Basic Non-Linear Formulation Approach	'network security', 'distributed computing'
Exponential Formulation Approach	'network security', 'distributed computing'
Enhanced Exponential Formulation Approach	'network security', 'distributed computing'

Results obtained from different methodologies (2)

Test Case 2: Input Query Keywords = ['machine learning', 'deep learning']

Table 5: Results for Test Case 2

Methodology	Domains of the recommended papers
Simple BFS Approach	'network security', 'distributed computing', 'image processing', 'algorithm design', 'machine learning', 'computer vision', **'cryptography'**, 'software engineering', 'data structures', 'operating systems', **'bioinformatics'**, 'computer graphics', 'parallel computing', 'computer programming', 'relational databases', 'structured storage'
Linear Formulation Approach	'computer vision', 'machine learning', 'parallel computing', 'data structures', 'distributed computing', 'computer programming', 'relational databases', 'image processing', 'operating systems', 'software engineering', **'network security'**
Basic Non-Linear Formulation Approach	'computer vision', 'machine learning', 'distributed computing', 'image processing', 'data structures'
Exponential Formulation Approach	'computer vision', 'machine learning', 'distributed computing', 'image processing', 'data structures'
Enhanced Exponential Formulation Approach	'computer vision', 'machine learning', 'distributed computing', 'image processing', 'data structures', 'software engineering', 'operating systems'

Summarizing the results of five approaches

- With the small sample of population, the results suggest that improving simple BFS approach by adding some more parameters like keyword co-occurrence frequency between two words, wordnet similarity measure and level of the Breadth First Search was a good decision to improve the recommendation.
- However, all the approaches are not efficient for the proposed system. Basic Non-Linear Formulation Approach and Exponential Formulation Approach have come out to be the best ones followed by Enhanced Exponential Formulation Approach.

$$\textit{Non-Linear Formulation} = \textit{Exponential Formulation} > \textit{Enhanced Exponential Formulation}$$

UI and API

The UI consists of two simple forms , one takes the input of semi-colon separated keywords and the other form takes an abstract as input. On submitting the required form, the request is made to the backend service which then sends the required results.

3 end points have been exposed from our Django based backend system:

- **" app/ "** -> This endpoint will redirect you to the home-page of the application
- **" app/recommend-keywords/ "** -> To this endpoint we send a list of semi-colon separated keywords in request body and get the respective recommendations based on it.
- **" app/recommend-abstract/ "** -> To this endpoint we send the abstract in request body and get the respective recommendations based on it.

Demo Video: Research Paper Recommendation System

Summary

As it has been witnessed that the system for easy recommendation of scholarly articles is of great significance today due to the various quoted reasons, this project work would henceforth provide a probable solution to this demand. The main focus here is to develop a keyword-based recommendation system which also takes semantic relationships in consideration during the model development and utilization phases. Another worth noting fact is the capability of the model to reduce the search-space.

Future Works

1. As of now, in the prediction pipeline word to word comparisons have been made to get the related research articles. However, in the future semantic relations have to be studied for this comparison and a more generalized formula has to be devised in this regard.
2. Due to the system's processing limitations (12 GB RAM, Intel(R) Xeon(R) CPU @ 2.20GHz Processor) the keyword co-occurrence network that has been generated involves only the top five keywords corresponding to each article. Moreover, only a few hundred data points of a particular domain have been considered but further enhancement involves the inclusion of more data points as well as other domains to make the network more dense and connected.
3. Currently, we are using the number of common keywords between the query set and paper for ranking the documents predicted by the system before making a final recommendation. We may look into other ranking algorithms in future.
4. Basic Graph data structure has been used to save the network. However, more advanced DS like industry standard graph data structures can be used.

References

[1] P. Zhao, J. Ma, Z. Hua, and S. Fang, "Academic social network-based recommendation approach for knowledge sharing," *SIGMIS Database*, vol. 49, no. 4, p. 78–91, Nov. 2018. [Online]. Available: https://doi.org/10.1145/3290768.3290775

[2] F. Meng, D. Gao, W. Li, X. Sun, and Y. Hou, "A unified graph model for personalized query-oriented reference paper recommendation," 10 2013, pp. 1509–1512.

[3] J. Beel, "Towards effective research-paper recommender systems and user modeling based on mind maps," 2017.

[4] M. Eto, "Extended co-citation search: Graph-based document retrieval on a co-citation network containing citation context information," *Information Processing Management*, vol. 56, no. 6, p. 102046, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0306457318303637

[5] P. Bhattacharyya, A. Garg, and S. F. Wu, "Analysis of user keyword similarity in online social networks," *Social Network Analysis and Mining*, vol. 1, no. 3, pp. 143–158, Jul 2011. [Online]. Available: https://doi.org/10.1007/s13278-010-0006-4

Thank You!!