**Towards partial fulfillment for Undergraduate Degree Level Programme
Bachelor of Technology in Computer Engineering**

A Project Preliminary Evaluation Report on:

Research Paper Recommendation System

Prepared by :

Admission No.	Student Name
U17CO099	Pankhi Khandelwal
U17CO101	Himanshu Choudhary
U17CO109	Shubhi Agarwal
U17CO112	Aman Mishra

Class   :   B.TECH. IV (Computer Engineering)   7th Semester

Year   :   2020-2021

Guided by :   Dr. Rupa G. Mehta

**DEPARTMENT OF COMPUTER ENGINEERING
SARDAR VALLABHBHAI NATIONAL INSTITUTE OF TECHNOLOGY,
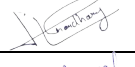SURAT - 395 007 (GUJARAT, INDIA)**

Student Declaration

This is to certify that the work described in this project report has been actually carried out and implemented by our project team consisting of

Sr.	Admission No.	Student Name
1	U17CO099	Pankhi Khandelwal
2	U17CO101	Himanshu Choudhary
3	U17CO109	Shubhi Agarwal
4	U17CO112	Aman Mishra

Neither the source code there in, nor the content of the project report have been copied or downloaded from any other source. We understand that our result grades would be revoked if later it is found to be so.

Signature of the Students:

Sr.	Student Name	Signature of the Student
1	Pankhi Khandelwal	
2	Himanshu Choudhary	
3	Shubhi Agarwal	
4	Aman Mishra	

Certificate

This is to certify that the project report entitled <u>Research Paper Recommendation</u>

<u>System</u> *is prepared and presented by*

Sr.	Admission No.	Student Name
1	U17CO099	Pankhi Khandelwal
2	U17CO101	Himanshu Choudhary
3	U17CO109	Shubhi Agarwal
4	U17CO112	Aman Mishra

Final Year of Computer Engineering and their work is satisfactory.

SIGNATURE :

Rupa G. Mehta

GUIDE   JURY   HEAD OF DEPARTMENT

Abstract

With the ever-increasing amount of data being accessible over the web, the need to make recommendation systems more and more accurate is pressing. The approaches involving the semantic comparison of documents tend to become infeasible when querying a very large amount of data. This problem is not just restricted to a few domains but is slowly and gradually becoming a part of almost all the domains involving look-up for bulk data. The same goes for the research community as well. Research paper recommendation aims to recommend new articles that match researchers' interests. It has become an attractive area of study since the number of scholarly papers increases exponentially. There are already approaches that make use of personalized suggestions based on user information but these approaches deal with the issues of lack of data for a newly registered user. This problem is referred to as cold-start. Cold-start problem occurs when trying to suggest a newly registered user regarding whom we don't have much data and hence the recommendations systems are not able to figure out what to suggest.

The aim of this project is to devise a graph-based network involving the use of Natural Language Processing based techniques for filtering to reduce the search space for documents. This will be followed by a personalized look-up in the search space and ultimately a semantic comparison in the resultant search space. The goal is to reduce the search space for semantic comparisons from as large as 1,00,000 documents to a few thousand documents. This will lead to really fast searches. Moreover, the addition of personalized search approaches will also contribute to increased accuracy in search results.

Keywords: Semantic Comparisons - Cold-Start - Graph - Search Space - Personalized Search - Keyword Networks - Natural Language Processing

Contents

List of Tables

List of Figures

ix

List of Acronyms

ACEK	Average of Correctly Extracted Keyphrases
AP	Average Precision
Bpref	Binary Preference Measure
CARS	Context Aware Recommendation System
CF	Collaborative Filtering
KCK	Keyword-Citation-Keyword
kNN	k Nearest Neighbour
LCS	Least Common Subsumer
MAP	Mean Average Precision
MRR	Mean Reciprocal Rank
NLTK	Natural Language Toolkit
RAKE	Rapid Automatic Keyword Extraction
TF-IDF	Term Frequency Inverse Document Frequency
YAKE	Yet Another Keyword Extractor

List of Symbols

δ Greek Symbol Delta

$\cap$ Intersection of Sets

$\cdot$ Dot Product of Vectors

Chapter 1

Introduction

The researchers around the globe find it an extensively time-consuming task to search for the related work and articles throughout the course of their research and dissertation from the digital repositories. Due to an upsurge in the number of publications (growth is exponential at a rate of 3.7% per annum) during the last few decades, the corpus for this search procedure has become manifold. The problem also gets intensified when the researchers possess very skimmed knowledge of operating these repositories. Performing these search operations manually is also a very time-consuming and tedious task making it almost infeasible to conduct.

In this project, we aim to build a graphical-network-model based on keywords, which co-relates the input publication to its n-neighbors on the basis of their semantic relationships and associations. The scope of this project provides a lead to the related research works in which only the quantitative correlation between the co-existing keywords is analyzed. Basically the proposed recommender is a software application that helps a user to search for the related and relevant documents automatically based on certain preferences described in the query thereby reducing the manual load.

1.1 Applications

A researcher in the current scenario has to manually search for the similar publications corresponding to his research to use them for preparing and studying the content related to his research or to perform the corresponding literature survey or to quote the citations. Therefore, a system that can automatically query an existing database to structure the co-related documents in a network and then look-up for the required one in the search space by reducing the search cost as compared to that of traditional means is of great significance and importance for practitioners.

Along with researchers, any educational and research institution can also customize this module in order to provide an efficient e-library system for the members of their organization.

1.2   Motivation

The structure of the research papers has made it significant to devise a system that checks for both the weighted and semantic similarity between documents before recommendation.

The bulk of publications present and its exponential growth rate have forced the existing search engines or recommendation systems to use enormous storage and search-space thereby increasing the search cost.

Moreover, inexperienced researchers find it quite challenging to deal with digital repositories. Sometimes, due to unfamiliarity of the search criteria incorporated in these search engines, land them in a situation where they get irrelevant data resulting in unnecessary delay in their progress.

Hence, a handy system that is easy to understand and exploit will receive an appreciation among the practitioners and researchers.

1.3   Objectives

The primary objective of this project is to explore innovative and efficient graphical network-based approaches to reduce the search space for semantic comparisons of documents while at the same time maintaining a high level of accuracy. A research paper almost all of the time follows a fixed format - a title then an abstract followed by the content of the paper and in the end a list of all the related citations. Networks based on co-occurrence of citations or keywords among papers are being explored. At the same time, efficient preprocessing techniques to extract the desired data for instance keywords from papers are also being looked into.

1.4   Contribution

To fulfill the project objectives hitherto discussed, in this project report, we have conducted an in-depth survey of pre-existing literature regarding the topics of document structuring and similarity analysis of legal documents. Furthermore, we have performed initial levels of preprocessing involving NLP based techniques like lemmatization, stemming, and created two different networks. At the same time, we have also tested one of the many approaches for keyword extraction from research papers that is to be used as a part of the preprocessing pipeline while creating the network.

1.5   Organization of project report

Chapter one includes a brief introduction of the project, its application in the real world, the motivation behind choosing this topic, its objectives, and our contribution towards this topic.

Chapter two describes the theoretical background involving a discussion on different similarity measures for documents. A brief overview about evaluation and performance measuring criteria is also given.

Chapter three includes the literature survey which gives an overview of the research work implemented in this domain and gives an overview of some of the concepts relevant to our project. A detailed emphasis has been laid on the existing methodologies and modules and the scope and technologies involved in developed systems have been explained in depth. Different content-based, citation-based, collaborative-filtering based, topic-based and keywords-based recommendation systems are discussed in-depth and conclusions are drawn.

Chapter four includes the proposed methodologies and its logical development.

Chapter five includes the implementation details and the tech stacks which we are utilizing to implement the project.

Chapter six presents our conclusions drawn and guides the reader to possible future works that can be built upon the proposed system.

Chapter 2

Theoretical Background

This chapter attempts to drive focus on the theoretical background required to understand the approach to analyzing the process of developing a research paper recommendation system. Here different similarity measures are discussed to create the base required to understand further discussion. A section of the chapter is utilized for explaining the basic criteria involved to determine the efficiency of a classifier or model. The discussion is concluded by mentioning different evaluation matrix parameters for deriving accuracy of keyword extraction techniques.

2.1 Similarity Measures

This section discusses the various approaches to estimate the similarity between documents.

2.1.1 Cosine Similarity

Cosine Similarity is the most common similarity measure used when documents are represented as term vectors. Here, the cosine of the angle between the term vectors in vector space corresponds to the correlation between these vectors [2]. The closer the cosine similarity value is to 1, the smaller the angle between the vectors, and hence, the greater the similarity.

Given two documents vectors, d1 and d2, over the term set T, the cosine similarity between them is calculated as:

$$CosineSimilarity(d1, d2) = \frac{d1 \cdot d2}{|d1| \cdot |d2|} \tag{2.1}$$

2.1.2 Bibliographic Coupling

Bibliographic Coupling is a link based similarity measure that uses citations to assert similarity between two documents. Two documents are said to be bibliographically coupled if they reference one or more common works in their bibliography. Thus, instead of the contents of the documents, only citations are compared to determine similarity [3].

Given two documents, D1 and D2, the set of out-citations of D1 and D2 are defined as OC1 and OC2, respectively. The bibliographic coupling between the two given documents is then

defined as the number of common out-citations, or:

$$B(D1, D2) = OC1 \cap OC2 \qquad (2.2)$$

The two documents are determined as similar if their bibliographic coupling is greater than or equal to the threshold value, δ .

2.1.3   Co-Citation

Co-citation is similar to Bibliographic Coupling because it uses citations to assert similarity, but differs in the sense that while bibliographic coupling links source document citations, co-citation links the number of times, the two given documents are cited together. Thus, if document D cites two documents, D1 and D2, together, D1 and D2 are said to be co-cited. The more co-cited the two documents are, the higher the likelihood that they are semantically similar [4].

Given two documents, D1 and D2, the set of in-citations of D1 and D2 are defined as IC1 and IC2, respectively. The co-citation between the two given documents is then defined as the number of common in-citations, or:

$$C(D1, D2) = IC1 \cap IC2 \qquad (2.3)$$

The two documents are determined as similar if their co-citation is greater than or equal to the threshold value, δ.

2.1.4   Keyword Co-Occurrence

Keyword Co-Occurrence is similar to Co-citation. The only difference is that it makes use of keywords rather than citations. Thus, if document D had two keywords K1 and K2 together, K1 and K2 are said to be co-occurred [5].

Given two documents, D1 and D2, the set of keywords of D1 and D2 are defined as K1 and K2, respectively. The co-occurrence of keywords between the two given documents is then defined as the number of common keywords, or:

$$C(D1, D2) = K1 \cap K2 \qquad (2.4)$$

The two documents are determined as similar if their co-occurrence of keyword is greater than or equal to the threshold value, δ.

2.2   Performance Measures

Confusion Matrix (refer to **Figure 2.1** is utilized to evaluate the performance of a classifier. It is a two-dimensional matrix viz., Actual and Predicted. It consists of four parameters that are used to calculate different measures: True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN) [6].

Figure 2.1: Confusion Matrix [1]

Following performance measures are studied based on Confusion Matrix:

- Classification Accuracy: It gives a measure about the correct predictions made by the classifier out of all the predictions and is given by the formula:

$$ClassificationAccuracy = \frac{TP+TN}{TP+FP+FN+TN} \qquad (2.5)$$

- Recall/Sensitivity: It is described as a probability that the test tuple sets to positive in the unhealthy population. It is represented by:

$$Sensitivity = \frac{TP}{TP+FN} \qquad (2.6)$$

- Specificity: It is described as a probability that the test tuple sets to negative in the healthy population and calculated as follows:

$$Specificity = \frac{TN}{TN+FP} \qquad (2.7)$$

- Precision: It gives the ratio of correct positive predictions and is found as:

$$Precision = \frac{TP}{TP+FP} \qquad (2.8)$$

- F Score: It is a measure of the test's accuracy. It considers both precision and recall of the system. It can be calculated as shown below:

$$F_{\beta} = \frac{(1+\beta^2)*precision*recall}{(\beta^2*precision)+recall} \qquad (2.9)$$

$\beta < 1$: Precision oriented evaluation

$\beta > 1$: Recall oriented evaluation

$\beta = 1$: Balance between Precision and Recall

F_1 score particularly is defined as a harmonic mean of precision and recall. It computes to be the best when the system has comparable values of both. It gives a measure better than accuracy metrics and is given by putting $\beta = 1$:

$$F_1 = \frac{2 * precision * recall}{precision + recall} \tag{2.10}$$

2.3 Evaluation Matrix for Keyword Extraction

Performance measures as explained in previous section along with some ranking quality measures are utilised for evaluating the efficiency of keyword extraction methodologies. The detailed process will be discussed in the upcoming subsections.

2.3.1 Ranking Quality Measures

It is used to measure the ranked extracted phrases. Such measures take into account the relative order of the phrases extracted by the keyphrase extraction systems. Popular ranking evaluation measures in the keyphrases extraction task are:

- Mean Reciprocal Rank (MRR): MRR is a measure to evaluate models that return a ranked list of key-phrases to documents. MRR only cares about the single highest-ranked relevant item.

$$MRR = \frac{1}{|D|} \sum_{d \in D} \frac{1}{rank(d)} \tag{2.11}$$

where rank(d) is the rank of the first correct keyphrase with all extracted keyphrases, D is the document set for keyphrase extraction, and d is a specific document.

- Mean Average Precision (MAP): It takes the ordering of a particular returned list of keyphrases into account. The Average Precision (AP) of the list is defined as follows:

$$AP = \frac{\sum_{r=1}^{|L|} P(r) \cdot rel(r)}{|L_R|} \tag{2.12}$$

where |L| is the number of items in the list, $|L_R|$ = number of relevant items, P(r) = precision when the returned list is treated as containing only its first r items, and rel(r) equals to 1 if the rth item of the list is in the golden set and 0 otherwise. By averaging AP over a set of n document cases, we obtain the MAP:

$$MAP = \frac{1}{n} \sum_{i=1}^{n} AP_i \qquad (2.13)$$

where AP_i is the average precision of the extracted keyphrases list returned for a document.

- Binary Preference Measure (Bpref): It is a summation-based measure of how many relevant phrases are ranked before irrelevant ones and it is defined as follows:

$$Bpref = \frac{1}{R} \sum_{r \in R} 1 - \frac{|nrankedhigherthanr|}{M} \qquad (2.14)$$

- Average of Correctly Extracted Keyphrases (ACEK): It is the average number of the extracted keyphrases that also belong to the document's golden set of keyphrases. This was the first type of performance evaluation that is used in the keyphrase extraction task but it is not widely used anymore, as precision and recall offer a more complete view for a system's performance in terms of the set of the extracted keyphrases.

The above measures are usually calculated following one of the following directions:

1. Exact match evaluation where the number of correctly matched phrases with the golden ones are determined based on string matching. In most cases, stemming is a preprocessing step to determine the match of two keyphrases.

2. Manual evaluation where experts decide whether the returned keyphrases by a system are wrong or right. However, this type of evaluation requires the investment of time and money and is characterized by great subjectivity

3. Partial match evaluation, a looser evaluation process that is proposed by Rousseau and Vazirgiannis(2015), which calculates the Precision, Recall and F1-measure between the set of words found in all golden keyphrases and the set of words found in all extracted keyphrases. Again, stemming is a required preprocessing step. However, such type of evaluation cannot evaluate the syntactic correctness of the phrases or deal with more complex issues such as over-generation problems and overlapping keyphrases candidates.

Chapter 3

Literature Survey

The overall workflow for framing a concept of dealing with structured data like research publications follows the path from forming the graphical networks. Then, ranking the suggestions made by each, using some comparative algorithms or ensembling the results, can also serve the purpose. Several ways have been employed by various researchers to analyze how to form a system that efficiently recommends the publications. Prior works have been discussed here, along with their advantages and disadvantages to conclude these implemented methodologies. Further, some existing research paper recommendation search engines and their adopted approaches have been explained to get insight into such systems' practicality. After that, different methods deployed to measure the similarity between two natural language text entities and previous efforts made to analyze the similarity between documents have been described.

3.1 Research Paper Recommendation Approaches

Various researchers in the above-explained domain have already explored several techniques. Some prominent ones out of those are as described below.

3.1.1 Citation-Based

Gipp et. al. proposed a method that tries to derive meanings out of the latent relationships that co-exist between any publication and its citations. These associations help in devising a system regardless of the domains explained through the article and authors' specializations as well. It generally uses bibliographic coupling [7] or co-citation analysis [8] to find publications similar to the given input. Scholastic search engines like Google Scholar work on the conventional text mining approach and citation counts.

3.1.2 Content-Based

Lops et. al. designed a system that recommends an item to a user, based on the description of the item and the profile of the user's interests. It matches the user's interests and profile with the content object and then recommends new items. Since we have to examine all the items in a set to correlate it with the user's interests, it takes a vast amount of item set, which is a considerable disadvantage [9].

3.1.3 Collaborative Filtering-Based

As stated by Jung et. al. [10], Collaborative Filtering (CF) identifies the need of the user before making recommendations. Hence the search becomes more specific rather than generic as it used to be in content-based recommendation systems. Information sources get rated in the first step resulting in a model that remains saved in the recommendation system. Whenever a query has been raised, the ratings get analyzed as per the user's information needs, and the output is generated. Emphasis has been laid on the fact that information need ratings have been utilized in contrast to the classical collaborative filtering methods wherein users' ratings were considered.

3.1.4 Topic-Based

Pan et. al. conducted a study and devised that instead of using some labeled or pre-classified data, the cardinal topics of a publication can also be incorporated as a tool for the recommendation system. Chenguang Pan and Wenxin Li [11] suggest that topic modeling principles can be used to study the thematic similarity between the topics. Recommendation results will be deduced by comparing the similarity parameter.

3.1.5 Keywords-Based

Content-based recommendation systems typically build attribute vector representations of contents and user preferences and generate recommendations according to the degree of similarity between user interests and items, as explained in [12] by Cheng et. al. Basically, two types of networks are possible [13], [14], one is a Co-word network in which an undirected graph has been generated by analyzing the co-occurrence nature of the keywords, and the other one is a Keyword-Citation-Keyword (KCK) network, it is a directed network in which the direction specifies the citing link between two publications and then the co-occurrence keyword property gets utilized [13], [15], the more links point to keywords for a particular paper signifies the importance of keywords thus providing more information as compared to the Co-word network. It is worth mentioning that no link exists between the same keywords in a KCK. In [15], the KCK approach has been demonstrated through its application to nano-related Environmental, Health, and Safety (EHS) risk literature.

3.1.6 Meta-Data-Based

Bharadhwaj et. al. discussed that recommendation systems are susceptible to the problem of cold-start. The user cold-start problem refers to the task of recommending items to a new user, whose previous item preferences are not present in the system [16]. To resolve this issue, meta-data, like demographic information of [17], can be used. But the problem with this approach is that metadata may not always be available and may not always be right. For example, the sensitive demographic information of users might not always be known as being used in [17].

3.1.7 Conclusion of Prior Works

All the methods, as discussed above, have both issues and advantages. As mentioned in [13], the co-word analysis only focuses on the count of keywords rather than their semantic relationships, making it challenging to derive context based information. Topic-based approaches and summarization techniques involve huge data processing and storage limitations due to the size of the data under consideration ranging to the extent of whole research paper data in summarization approaches. Only the listed keywords do not provide the main ideas of the entire literature; hence semantics play an essential role in overall development. The author in [11] has emphasized the need for textual information of any literature work and proposed a topic analysis approach by considering thematic similarity. Homographs identification becomes insignificant in citation analysis [18]. It sometimes results in assigning a publication to the wrong author. Despite having comparatively successful results using collaborative-filtering, it encounters the first-rater problem, which makes it compulsory that the items involved must be rated by at least one of its neighbors as discussed in [19]. Moreover, CF also faces sparsity problem due to the unlikeliness of users in rating all the available items.

3.2 Research Paper Recommendation Systems and their Flow of Operation

There are various open-source search engines for providing quick access to scholarly articles. They use different Natural Language Processing, semantic analysis, and ranking algorithms to serve the purpose. Some of these are as discussed below.

3.2.1 JournalFinder

Powered by the Elsevier Fingerprint Engine, JournalFinder uses semantic search technology and field-of-research specific vocabularies to match your abstract to relevant Elsevier journals. The Elsevier Fingerprint Engine applies various Natural Language Processing techniques to mine the text you enter into the JournalFinder for mentions of key concepts spanning all the major scientific disciplines and creates a structured index of weighted terms that defines the text, known as a Fingerprint. JournalFinder then compares the Fingerprint of your abstract with the Fingerprints of all journal articles in Scopus and recommends up to 50 of the most relevant journals for you to consider [20].

3.2.2 Google Scholar

Google Scholar provides a simple way to search for scholarly literature broadly. From one place, users can search across many disciplines and sources: articles, theses, books, abstracts, and court opinions. Google Scholar aims to rank documents the way researchers do, weighing the full text of each document, where it was published, who it was written by, as well as how often and how recently it has been cited in other scholarly literature.

3.2.3 Springer

Springer uses semantic technology to help the user quickly choose the right journal for the paper user wants. Users can refine the results based on requirements for the Impact Factor or publishing model, including an option to match to journals that are fully open access or have open access options.

3.3 Keyword Extraction Techniques

It is cumbersome to deal with the massive volume of data or content available over any network. The same applies to the publications' content, thus raising a need to summarise it. Keywords are a far shorter summary, as stated in [21]. There are various methods available to extract keywords from available text documents. Some prominent ones are noted here.

3.3.1 Simple Statistical Approaches

These approaches don't require training data to extract the most important keywords in a text. Since they rely on stats, they may overlook relevant words or phrases mentioned once but are still considered appropriate. Different Statistical approaches:

3.3.1.1 Word Frequency

Word frequency consists of listing the words and phrases that most commonly appear. Word frequency approach considers documents as a mere bag of words leaving aside crucial aspects to the meaning, structure, grammar, and sequence of words.

3.3.1.2 TF-IDF

TF-IDF measures how important a word is to a document in a collection of documents. This calculates the number of times a word appears in a text and compares it with the inverse document frequency. Multiplying these two quantities provides the TF-IDF score of a word in a document. The higher the score is, the more relevant the word is to the document.

3.3.1.3 RAKE

Rapid Automatic Keyword Extraction (RAKE) is a well-known extraction method that uses a list of stop words and phrase delimiters to detect the most relevant words or phrases in a piece of text as mentioned by Rose et. al. [22].

3.3.1.4 YAKE

Campos et. al. investigated and concluded that Yet Another Keyword Extractor (YAKE) is a novel feature-based system for multilingual keyword extraction, which supports texts of different sizes, domains or languages [23, 24]. Unlike other approaches, YAKE does not rely on

dictionaries nor thesauri, neither is trained against any corpora [25]. Instead, it follows an unsupervised approach that builds upon features extracted from the text, making it thus applicable to documents written in different languages without the need for further knowledge. This can be beneficial for a large number of tasks and a plethora of situations where access to training corpora is either limited or restricted.

3.3.1.5 keyBERT

As mentioned by Grootendorst, BERT is a bi-directional transformer model that allows us to transform phrases and documents into vectors that capture their meaning. BERT is used for this purpose as it has shown great results for both similarity and paraphrasing tasks. [26, 27].

3.3.2 Linguistic Approaches

Linguistic Approaches use the linguistic properties of the words, sentences, and documents. Lexical, syntactic, semantic, and discourse analysis are some of the most commonly examined properties.

3.3.3 Machine Learning Approaches

Machine Learning approaches consider supervised or unsupervised learning from the examples, but related work on keyword extraction prefers the supervised approach. Supervised machine learning approaches induce a model that is trained on a set of keywords. They require manual annotations of the learning dataset, which is too tedious and inconsistent. Thus, supervised methods require training data and are often dependent on the domain. A system needs to re-learn and establish the model whenever a domain changes [28].

3.4 Context Aware Approaches

Being aware of the context under which the recommendation has to be done can drastically improve the results of a recommendation system. It adds to a recommendation system what we can say at the very least, a humane touch to the output. There is no fixed context-related information that can be used in different recommendation systems. Instead, it varies widely from domain to domain. For instance, context can cover information that may include conditions under which the user has suggested an item, the relationship between different items and their common consumers. In many cases, this may change periodically, and the system has to account for it.

The first work on Context Aware Recommendation System (CARS) was done by Adomavicious et. al. [29]. They proposed that we must have extra dimensions added to each item to store context-related information. But the problem with this is that it would lead to an increase in sparsity. Adomavicius et. al. [30] divided the different approaches to model CARS into three

categories viz., pre-filtering, post-filtering, and contextual modeling. In pre-filtering, contextual information is used before and in post-filtering, it is used after the output of a context-free recommendation system. In the case of contextual modeling, the contextual information becomes an integral part of the recommendation algorithm. Phuong et al. in 2019, suggested a graph-based approach. They converted the multidimensional user-item matrix into a two-dimensional matrix by creating fictitious items to represent the context information. This is called item splitting as said by Baltrunas et. al. [31]. Then a bipartite graph was created between the user and items following which the similarity between users or items was computed. Subsequently, a k Nearest Neighbour (kNN) algorithm was used to make the predictions. The problem still here is a sparse matrix and the use of an in-memory algorithm i.e. kNN to make the predictions.

Cheng and Kharlamov [32], gave two important points regarding good semantic relationships search over keyword networks. These include ensuring the semantic cohesiveness of an answer by jointly analysing its constituent entities and properties and the other one is ensuring semantic completeness of an answer by analysing its context to avoid unnatural answers. [33] made use of bisonets for research paper recommendation tasks. Nodes represented the research papers. TF-IDF was used for extracting keywords. They used a variation of Jaccard measure of similarity to determine the existence of edges between the nodes.

3.5 WordNet Based Similarity Measures

WordNet is a large lexical database of English. Nouns, verbs, adjectives, and adverbs are grouped into sets of cognitive synonyms (synsets), each expressing a distinct concept. Synsets are interlinked by means of conceptual-semantic and lexical relations [34]. It can be easily considered one of the largest collections of concepts consisting of different words interlinked with each other to form synsets and these synsets in turn being linked to each other. Most used relations that are prevalent in WordNet comprise of Hypernymy and Hyponymy, thus forming a considerable number of levels of Is-a relationship for a concept. A lot of research has been done to derive semantic similarities among different concepts using this. Some of them include:

1. Wu and Palmer [35] proposed the similarity measure that makes use of depth of the synsets along with the depth of their Least Common Subsumer (LCS). LCS is basically the lowest common ancestor of the two synsets. It is given by:

$$Sim_{Wu}(c_1, c_2) = \frac{2 * H}{N_1 + N_2 + 2 * H} \tag{3.1}$$

where,
N_1 = Number of edges from LCS to c_1
N_2 = Number of edges from LCS to c_2
H = Number of edges from root of taxonomy to LCS

2. Li et. al. [36] made use of the shortest path between two words and reduced this result in terms of H which is the depth of LCS and L which is the shortest path length between the two words. The other parameters are the weighting parameters to adjust the contributions of H and L to the result. This type of similarity can be calculated using:

$$Sim_{Li}(c_1, c_2) = \frac{e^{-\alpha L} * (e^{\beta H} - e^{-\beta H})}{e^{\beta H} + e^{-\beta H}} \tag{3.2}$$

3. Liu et. al. [37] made use of the common features between the two concepts. It presented the similarity as the ratio of common features to the total features of the two concepts under some weighted parameters that were used for smoothing. Liu gave the following two measures, the second one being the exponential version of first and shows much more distinction in output. It is as shown in:

$$Sim_{Liu-1}(c_1, c_2) = \frac{\alpha * d}{\alpha * d + \beta * l} \tag{3.3}$$

$$Sim_{Liu-2}(c_1, c_2) = \frac{e^{\alpha * d} - 1}{e^{\alpha * d} + e^{\beta * l} - 2} \tag{3.4}$$

where,

d = Depth of LCS

l = length of shortest path between c_1 and c_2

4. Resnik et. al. [38] introduced the use of information content that the LCS of two concepts holds to determine the semantic similarity.

$$Sim_{Res}(c_1, c_2) = IC(LCS(c_1, c_2)) \tag{3.5}$$

5. Jiang et. al. [39] derived the distance measure from the use of information content. Larger this distance, the less similar are the two concepts.

$$dist_{jiang}(c_1, c_2) = IC(c_1) + IC(c_2) - 2 * IC(LCS(c_1, c_2)) \tag{3.6}$$

6. Lin et. al. [40] made use of the information content of the two concepts along with their LCS to decide the semantic similarities.

$$Sim_{Lin}(c_1, c_2) = \frac{2 * IC(LCS(c_1, c_2))}{IC(c_1) + IC(c_2)} \tag{3.7}$$

7. Meng et. al. [41], made use of the results derived by Lin et. al. [40] as an exponent to compute the similarity between the two concepts.

$$Sim_{Meng}(c_1, c_2) = e^{Sim_{Lin}(c_1, c_2)} - 1 \tag{3.8}$$

15

8. Results deduced by Lin et. al. [40] were sometimes inaccurate due to the result being quite large then expected. Thus, Jaccard [42] came up with the equation given below to solve this defect.

$$Sim_{Jaccard}(c_1, c_2) = \frac{IC(LCS(c_1, c_2))}{IC(c_1) + IC(c_2) - IC(LCS(c_1, c_2))} \tag{3.9}$$

Chapter 4

Proposed Work

The proposed project aims to produce a deliverable that can be utilized among research institutions and individuals to serve their purpose of an extensive search of related publications. Moreover, the storage space and search-space complexity would also be dealt with by reducing the need for entire publication content for recommendation to some structured parts. At later stages, a ranking algorithm will be used to collaborate the results produced during intermediate stages. Context-based factor and semantic analysis are also considered for graphical-network creation of the available dataset. The whole project workflow will be as shown in **Figure 4.1** and **Figure 4.2**.
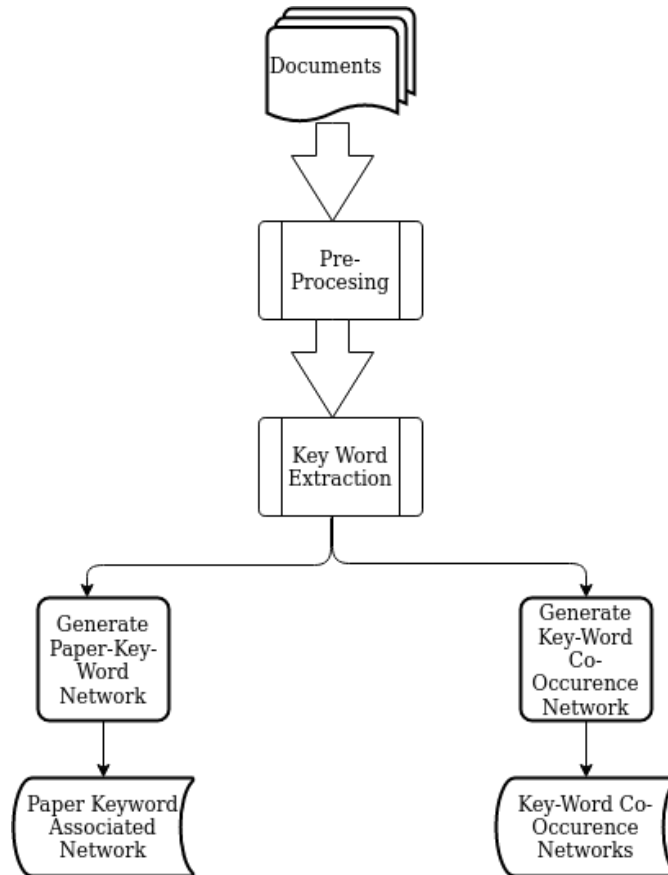
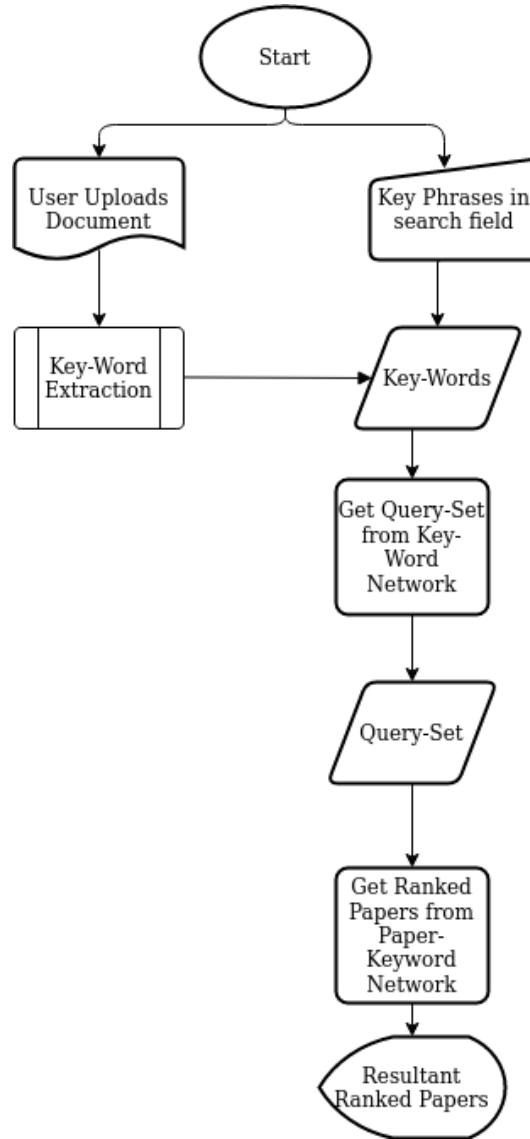Figure 4.1: Block Diagram depicting the preprocessing and network creation stages

Figure 4.2: Block Diagram depicting the user interaction with the system

4.1 How does the user interact with the system?

The proposed approach holds two ways by which the user can interact with our system:

4.1.1 By Uploading a Research Paper

Here the user can directly upload a research paper and, as an output, get all the research papers that are related to it. Internally, the key phrases will be extracted from the document, and the keyword-based network will be queried based on these keywords to get the desired results.

4.1.2 By Providing Key Phrase

Here the user can provide key phrases. Based on these key phrases the keyword-based network will be queried, and as an output user will get the related papers.

4.2 How does the system work?

As shown in **Figure 4.1** and **Figure 4.2**, the workflow of the proposed system can be explained in the following manner:

- From the user's query, a set of all possible keywords will be generated (either provided by the user directly or extracted from the provided content) and fed to the system during the initial stages, as explained in the previous section.

- Now, the model prepares a query set by getting relatable keywords obtained for the input set of keywords or keyphrases using context-information and semantic and thematic relationships.

- This query set is further utilized to get a set of related and interlinked documents by analyzing the matrix or map pairs of publications and keywords using both quantitative weights and semantic similarities. Note that the use of Paper Keyword associated network will be much more efficient then using a sparse matrix to represent the relationship between documents and keywords.

- From this collection of related papers, top n-ranked papers will be recommended to the user as an output.

Chapter 5

Simulation and Results

Under implementation, a bit of pre-processing using stemming, lemmatization, and tokenization has been carried out. Two types of graph-based networks (using keywords and paper id as nodes) are also generated as initial product development stages. Some keyword extraction techniques are also explored during the research and implementation tasks.

5.1 Pre-processing

Data-Preprocessing techniques in Data Mining are used to pre-process the raw data in a structured, well-defined, and clean format. Some of the pre-processing methods will be discussed in following subsections.

5.1.1 Stemming

Stemming is a technique used to extract the base form of the words by removing affixes from them. It is just like cutting down the branches of a tree to its stems. Python's *LancasterStemmer* class of Natural Language Toolkit (NLTK) module can be utilized for the same.

5.1.2 Lemmatisation

Lemmatization is similar to stemming. The process generates a 'lemma' as an output. Lemma is a root word rather than root stem, the output of stemming. After lemmatization, we will be getting a valid word that means the same thing. Python's *WordNetLemmatizer* class of NLTK module can be utilized for the same.

5.1.3 Tokenization

Tokenisation is splitting up a larger body of text into smaller lines, words, or even creating words for a non-English language. The various tokenization functions in-built into the NLTK module of Python itself can be utilized for the same.

5.2 Keyword Extraction Techniques

Keyword Extraction is an automated process of extracting words, group of words, or expressions from the given text. There are various approaches to extract n-gram keywords from the given text input. Some of them have been explored as mentioned in Chapter two. Three of the discussed algorithms have been implemented using Python frameworks and results have been drawn. Following subsections will be focused towards explaining the approaches along with their implementation.

5.2.1 RAKE

The first thing this method does is splitting the text into a list of words and remove stop words from that list. Then, the algorithm splits the text at phrase delimiters and stopwords to create candidate expressions. Once the text has been split, the algorithm creates a matrix of word co-occurrences. After that matrix is built, words are given a score and top T keywords are given as output.

5.2.2 YAKE

YAKE is an unsupervised keyword extraction method which depends on individual documents without relying on the existence of the corpus. In the first step of YAKE algorithm pre-processing 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.

5.2.3 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. Here it is assumed that the most similar candidates to the document are good keywords for representing the document. To calculate the similarity between candidates and the document, cosine similarity between vectors is used.

5.2.4 Results

Results obtained after implementing the RAKE, YAKE, and keyBERT algorithms on a common text input, *S* are as shown in **Table 5.1**.

S = "Supervised learning is the machine learning task of learning a function that maps an input to an output based on example input-output pairs.[1] It infers a function from labeled training data consisting of a set of training examples.[2] In supervised learning, each example is a pair consisting of an input object (typically a vector) and a desired output value (also called the supervisory signal). A supervised learning algorithm analyzes the training data and produces an inferred function, which can be used for mapping new examples. An optimal scenario will allow for the algorithm to correctly determine the class labels for unseen instances. This requires the learning algorithm to generalize from the training data to unseen situations in a 'reasonable' way (see inductive bias)."

Table 5.1: Results of Keyword Extraction Techniques

Algorithm	Framework	Extracted Keywords
RAKE	rake-nltk	'learning algorithm', 'training examples', 'learning machine', 'learning algorithm', 'machine learning'
YAKE	yake	'learning algorithm', 'learning machine', 'learning algorithm', 'machine learning', 'supervised learning'
keyBERT	KeyBERT	'learning algorithm, 'learning machine', 'machine learning', 'supervised learning', 'learning function'

5.3   Evaluation Matrix for Keyword Extraction Techniques

In the above section, three different approaches for extracting keywords have been discussed. These have been implemented in Python and a comparative manual study has been conducted to evaluate their performances. Results have been summarised in **Table 5.2**. Observation are done on abstracts of single dataset.

Table 5.2: Comparison of Keyword Extraction Techniques

Algorithm	Language	Language Support	MRR Score	MRR Rank	MAP Score	MAP Rank
RAKE	Python	Multi-Lingual	0.509	1	0.65	2
YAKE	Python	Multi-Lingual	0.456	2	0.652	1
keyBERT	Python	Multi-Lingual	0.32	3	0.53	3

5.4 Networks based on Keywords

In this section, different approaches for building keyword networks based on their co-occurrence property have been discussed. The given two approaches have been implemented using undirected graph data structure in Python.

5.4.1 Network 1 (Paper IDs as nodes)

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 (see **Figure 5.1**).
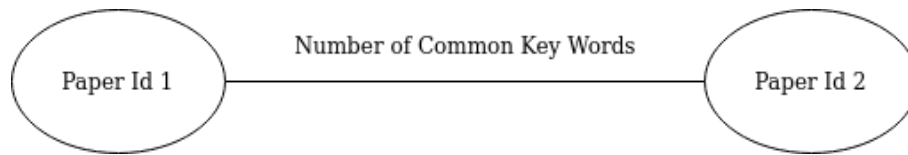
Figure 5.1: Paper-Keyword Network

5.4.2 Network 2 (Keywords as nodes)

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 (see **Figure 5.2**).
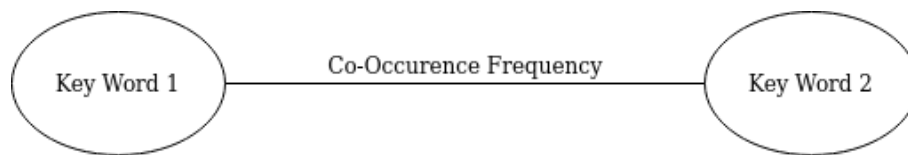
Figure 5.2: Keyword Co-Occurrence Network

5.4.3 Network 3 (Paper-Keyword Association Network)

This network consists of paper IDs and Keywords as nodes and there exist a link between a paper ID and a keyword if that keyword occurs in a particular paper. In this network, no two keyword or paper nodes are directly connected. The structure of the network is as shown in **Figure 5.3**
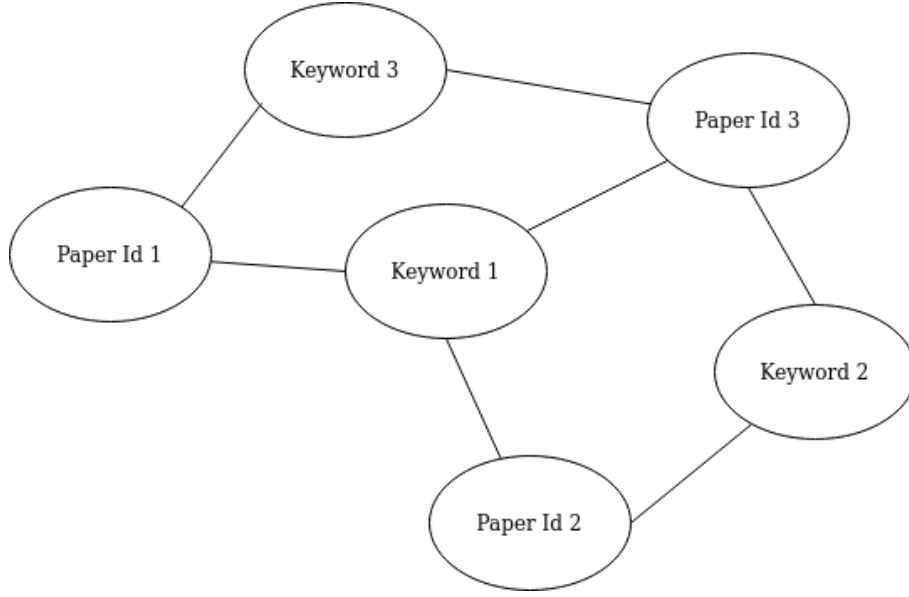
Figure 5.3: Paper-Keyword Association Network

5.5 Adding semantic weights to Keyword Co-Occurrence Network

In order to provide a touch of contextual similarity between the key-words so extracted while preparing the keyword network, we are making use of similarity measures that make use of wordNet corpus. There are some slight variations in the process. Following is the step-wise implementation of our approach:

1. Tokenize the two input phrases. Then, attach the part of speech to each of the tokens.

2. In each token set we remove all the tokens except Nouns, Verbs and Adjectives.

3. Then, make use of Lesk. Lesk returns a Synset with the highest number of overlapping words between the context sentence and different definitions of given word from each Synset. This is done for both of the token sets.

4. Then, compute the path similarity matrix and Wu Palmer similarity matrix for all possible pairs, with first word from set one and second word from set two.

5. Now let the sum of max value in each row be denoted by s1 and that the sum of max value in each column be denoted by s2. Let the length of set one of words be n1 and that of the second set be n2. The overall similarity score is given by:

$$Similarity = \frac{s1 + s2}{2 * n1 * n2} \tag{5.1}$$

6. The results derived for some test cases are as shown in **Table 5.3**. Here, the calculated similarity values are normalized on a scale of 0 to 1. Results are not very promising for the given corpus currently and optimization is under process by utilising a custom ontology.

24

Table 5.3: Results of Similarity Calculation Algorithm

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-scraper near by	0.35079365079365077
war dismantles	battle levelling	0.33333333333333333

Chapter 6

Conclusion and Future Work

As it has been witnessed that the system for an 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 that also takes semantic relationships into consideration during the model development and utilization phases. Another worth noting fact is the capability of the model to reduce the search-space.

As of now we are using WordNet corpus for deriving the semantic relations between the keywords. WordNet is rather a generalized corpus and does not account for scientifically related terms and domains. The future work will involve developing an ontology that would hold the domains, sub-domains and keywords in a tree like manner and then use this for doing the semantic measurements. We have already found a dataset for this purpose. This ontology can further be used to evaluate the results of keyword extraction techniques. Once the keyword extraction technique finalizes we can follow the steps described in the proposed work to make the predictions.

References

[1] "Understanding confusion metrics," https://towardsdatascience.com/understanding-confusion-matrix-a9ad42dcfd62, accessed: 2020-10-01.

[2] A. Huang, "Similarity measures for text document clustering," *Proceedings of the 6th New Zealand Computer Science Research Student Conference*, 01 2008.

[3] B. Jarneving, "Bibliographic coupling and its application to research-front and other core documents," *Journal of Informetrics*, vol. 1, no. 4, pp. 287 – 307, 2007. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1751157707000594

[4] H. Small, "Co-citation in the scientific literature: A new measure of the relationship between two documents," *Journal of the American Society for Information Science*, vol. 24, pp. 265 – 269, 07 1973.

[5] S. Radhakrishnan, S. Erbis, J. A. Isaacs, and S. Kamarthi, "Novel keyword co-occurrence network-based methods to foster systematic reviews of scientific literature," *PLOS ONE*, vol. 12, no. 3, pp. 1–16, 03 2017. [Online]. Available: https://doi.org/10.1371/journal.pone.0172778

[6] "Machine learning - performance metrics," https://www.tutorialspoint.com/machine_learning_with_python/machine_learning_algorithms_performance_metrics.htm, accessed: 2020-10-01.

[7] O. Hanif, Z. Donghua, W. Xuefeng, and M. S. Nawaz, "Refining the measurement of topic similarities through bibliographic coupling and lda," *IEEE Access*, vol. 7, pp. 179 997–180 011, 2019.

[8] B. Gipp and J. Beel, "Citation proximity analysis (cpa) : A new approach for identifying related work based on co-citation analysis," in *Proceedings of the 12th International Conference on Scientometrics and Informetrics, vol. 1*, B. Larsen, Ed. São Paulo: BIREME/PANO/WHO, 2009, pp. 571–575. [Online]. Available: http://www.sciplore.org/wp-content/papercite-data/pdf/gipp09a.pdf

[9] P. Lops, M. de Gemmis, and G. Semeraro, *Content-based Recommender Systems: State of the Art and Trends*, 01 2011, pp. 73–105.

[10] Seikyung Jung, Juntae Kim, and J. L. Herlocker, "Applying collaborative filtering for efficient document search," in *IEEE/WIC/ACM International Conference on Web Intelligence (WI'04)*, 2004, pp. 640–643.

[11] Chenguang Pan and Wenxin Li, "Research paper recommendation with topic analysis," in *2010 International Conference On Computer Design and Applications*, vol. 4, 2010, pp. V4–264–V4–268.

[12] D. De Nart and C. Tasso, "A personalized concept-driven recommender system for scientific libraries," *Procedia Computer Science*, vol. 38, pp. 84 – 91, 2014, 10th Italian Research Conference on Digital Libraries, IRCDL 2014.

[13] Q. Cheng, J. Wang, W. Lu, Y. Huang, and Y. Bu, "Keyword-citation-keyword network: a new perspective of discipline knowledge structure analysis," *Scientometrics*, vol. 124, no. 3, pp. 1923–1943, September 2020.

[14] H. Li, H. An, Y. Wang, J. Huang, and X. Gao, "Evolutionary features of academic articles co-keyword network and keywords co-occurrence network: Based on two-mode affiliation network," *Physica A: Statistical Mechanics and its Applications*, vol. 450, pp. 657 – 669, 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S037843711600025X

[15] S. Radhakrishnan, S. Erbis, J. A. Isaacs, and S. Kamarthi, "Correction: Novel keyword co-occurrence network-based methods to foster systematic reviews of scientific literature," *PLOS ONE*, vol. 12, no. 9, pp. 1–1, 09 2017. [Online]. Available: https://doi.org/10.1371/journal.pone.0185771

[16] H. Bharadhwaj, "Meta-learning for user cold-start recommendation," 04 2019.

[17] A. K. Pandey and D. S. Rajpoot, "Resolving cold start problem in recommendation system using demographic approach," in *2016 International Conference on Signal Processing and Communication (ICSC)*, Dec 2016, pp. 213–218.

[18] B. Gipp, J. Beel, and C. Hentschel, "Scienstein: A research paper recommender system," 01 2009.

[19] R. Torres, S. M. McNee, M. Abel, J. A. Konstan, and J. Riedl, "Enhancing digital libraries with techlens," in *Proceedings of the 2004 Joint ACM/IEEE Conference on Digital Libraries, 2004.*, 2004, pp. 228–236.

[20] "Elsevier fingerprint engine," https://www.elsevier.com/solutions/elsevier-fingerprint-engine, accessed: 2020-09-15.

[21] Y. HaCohen-Kerner, "Automatic extraction of keywords from abstracts," 09 2003, pp. 843–849.

[22] S. Rose, D. Engel, N. Cramer, and W. Cowley, *Automatic Keyword Extraction from Individual Documents*, 03 2010, pp. 1 – 20.

[23] R. Campos, V. Mangaravite, A. Pasquali, A. Jorge, C. Nunes, and A. Jatowt, "Yake! collection-independent automatic keyword extractor," 02 2018.

[24] C. et. al., "A text feature based automatic keyword extraction method for single documents," 02 2018.

[25] "Yet another keyword extractor (yake)," https://github.com/LIAAD/yake, accessed: 2020-10-25.

[26] "Bert explained: State of the art language model for nlp," https://towardsdatascience.com/bert-explained-state-of-the-art-language-model-for-nlp-f8b21a9b6270, accessed: 2020-10-27.

[27] "Keybert," https://github.com/MaartenGr/KeyBERT, accessed: 2020-10-28.

[28] S. Beliga, A. Meštrović, and S. Martincic-Ipsic, "An overview of graph-based keyword extraction methods and approaches," *Journal of Information and Organizational Sciences*, vol. 39, pp. 1–20, 07 2015.

[29] G. Adomavicius and A. Tuzhilin, "Multidimensional recommender systems: A data warehousing approach," in *Electronic Commerce*, L. Fiege, G. Mühl, and U. Wilhelm, Eds. Berlin, Heidelberg: pringer Berlin Heidelberg, 2001, pp. 180–192.

[30] G. Adomavicius, B. Mobasher, F. Ricci, and A. Tuzhilin, "Context-aware recommender systems," *AI Magazine*, vol. 32, pp. 67–80, 09 2011.

[31] L. Baltrunas and F. Ricci, "Context-based splitting of item ratings in collaborative filtering," 01 2009, pp. 245–248.

[32] G. Cheng and E. Kharlamov, "Towards a semantic keyword search over industrial knowledge graphs (extended abstract)," in *2017 IEEE International Conference on Big Data (Big Data)*, Dec 2017, pp. 1698–1700.

[33] M. B. Magara, S. O. Ojo, and T. Zuva, "Towards a serendipitous research paper recommender system using bisociative information networks (bisonets)," in *2018 International Conference on Advances in Big Data, Computing and Data Communication Systems (icABCD)*, Aug 2018, pp. 1–6.

[34] "Wordnet - a lexical database for english," https://wordnet.princeton.edu/, accessed: 2020-10-15.

[35] Z. Wu and M. Palmer, "Verbs semantics and lexical selection," in *Proceedings of the 32nd Annual Meeting on Association for Computational Linguistics*, ser. ACL '94. USA: Association for Computational Linguistics, 1994, p. 133–138. [Online]. Available: https://doi.org/10.3115/981732.981751

[36] Y. Li, Z. A. Bandar, and D. Mclean, "An approach for measuring semantic similarity between words using multiple information sources," *IEEE Transactions on Knowledge and Data Engineering*, vol. 15, no. 4, pp. 871–882, July 2003.

[37] X.-Y. Liu, Y.-M. Zhou, and R.-S. Zheng, "Measuring semantic similarity in wordnet," vol. 6, 09 2007, pp. 3431 – 3435.

[38] P. Resnik, "Using information content to evaluate semantic similarity in a taxonomy," in *IJCAI*, 1995.

[39] J. Jiang and D. Conrath, "Semantic similarity based on corpus statistics and lexical taxonomy," in *Proc. of the Int'l. Conf. on Research in Computational Linguistics*, 1997, pp. 19–33. [Online]. Available: http://www.cse.iitb.ac.in/~cs626-449/Papers/WordSimilarity/4.pdf

[40] D. Lin, "An information-theoretic definition of similarity," in *In Proceedings of the 15th International Conference on Machine Learning*. Morgan Kaufmann, 1998, pp. 296–304.

[41] L. Meng, J. Gu, and Z. Zhou, "A new model of information content based on concept's topology for measuring semantic similarity in wordnet 1," 2012.

[42] P. Jaccard, "Distribution de la flore alpine dans le bassin des dranses et dans quelques régions voisines." *Bulletin de la Societe Vaudoise des Sciences Naturelles*, vol. 37, pp. 241–72, 01 1901.

Acknowledgement