Research Paper Recommendation System (Second Stage Project Presentation)

Prepared by

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

Guided by: **Dr. Rupa G. Mehta**

Computer Engineering Department

Contents

- ➔ Problem Statement
- ➔ Theoretical Background
- ➔ Literature Survey
- ➔ Proposed Solutions and Workflow
- ➔ Implementation and Results
- ➔ Applications
- ➔ Summary
- ➔ Future Prospects

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.
- To be followed by a personalized look-up in the search space and ultimately a semantic comparison in the resultant search space.

Theoretical Background

- ➔ Similarity Measures
- ➔ **Evaluation Matrix**

Evaluation Matrix

Evaluating measures:

- Exact match evaluation
- Manual evaluation
- Partial match evaluation

Ranking Quality measures:

- Mean Reciprocal Rank (MRR)
- Mean Average Precision (MAP)
- Binary Preference Measure (Bpref)
- Average of Correctly Extracted Keyphrases (ACEK)

Literature Survey

- ➔ Research Paper Recommendation Approaches
- ➔ Research Paper Recommendation Systems and their Flow of Operation
- ➔ **Keyword Extraction Techniques**
- ➔ **Context Aware Approaches**
- ➔ **WordNet Similarity Measures**

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

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 Context Aware Approaches I

AUTHOR	HIGHLIGHTS
Adomavicious et. al. [1]	- Proposed addition of an extra dimension to store contextual information - Lead to increase in sparsity
Adomavicious et. al. [2]	- Divided CARS approaches into pre filtering, post filtering and contextual modeling
Baltrunas et. al. [3]	- Proposed item splitting by converting multi dimensional user-item matrix into a 2D matrix by creating fictitious items for contextual data - Later a bipartite graph is created between user and items and kNN was used for predictions - Problems are sparse matrix and use of in-memory kNN algorithm

Existing Context Aware Approaches II

AUTHOR	HIGHLIGHTS
Cheng et. al. [4]	Focussed on 2 factors: - Semantic Cohesiveness - Semantic Completeness
Magara et. al. [5]	- Utilised BisoNets for recommendation tasks - Nodes represent research papers - Keyword Extraction is done using TF-IDF - Jaccard similarity measure is used to determine existence of edges

WordNet Based Similarity Measures I

AUTHOR	FEATURES UTILISED	FORMULA
Wu and Palmer [6]	depth of synsets, depth of LCS	$Sim_{Wu}(c_1,c_2) = (2*H)/(N_1 + N_2 + 2*H)$
Li et. al. [7]	depth of LCS, shortest path between two words	$Sim_{Li}(c_1,c_2) = (e^{-\alpha L} * (e^{\beta H} + e^{-\beta H}))/ (e^{\beta H} + e^{-\beta H})$
Liu et. al. [8]	common features of two concepts	$Sim_{Liu-1}(c_1,c_2) = (\alpha * d)/(\alpha * d + \beta * l)$
Resnik et. al. [9]	distance measure using information content	$Sim_{Res}(c_1,c_2) = IC(LCS(c_1,c_2))$

WordNet Based Similarity Measures II

AUTHOR	FEATURES UTILISED	FORMULA
Jiang et. al. [10]	information content	$dist_{jiang}(c_1,c_2) = IC(c_1) + IC(c_2) - 2*IC(LCS(c_1,c_2))$
Lin et. al. [11]	information content, LCS	$Sim_{Lin}(c_1,c_2) = (2*IC(LCS(c_1,c_2))) / (IC(c_1) + IC(c_2))$
Meng et. al. [12]	exponential measure of information content and LCS	$Sim_{Meng}(c_1,c_2) = e^{Sim\_Lin(c1,c2)} - 1$
Jaccard [13]	removed defect in results derived by Lin et.a la.	$Sim_{Jaccard}(c_1,c_2) = (IC(LCS(c_1,c_2))) / (IC(c_1) + IC(c_2) - IC(LCS(c_1,c_2))$

Proposed Solution and Workflow

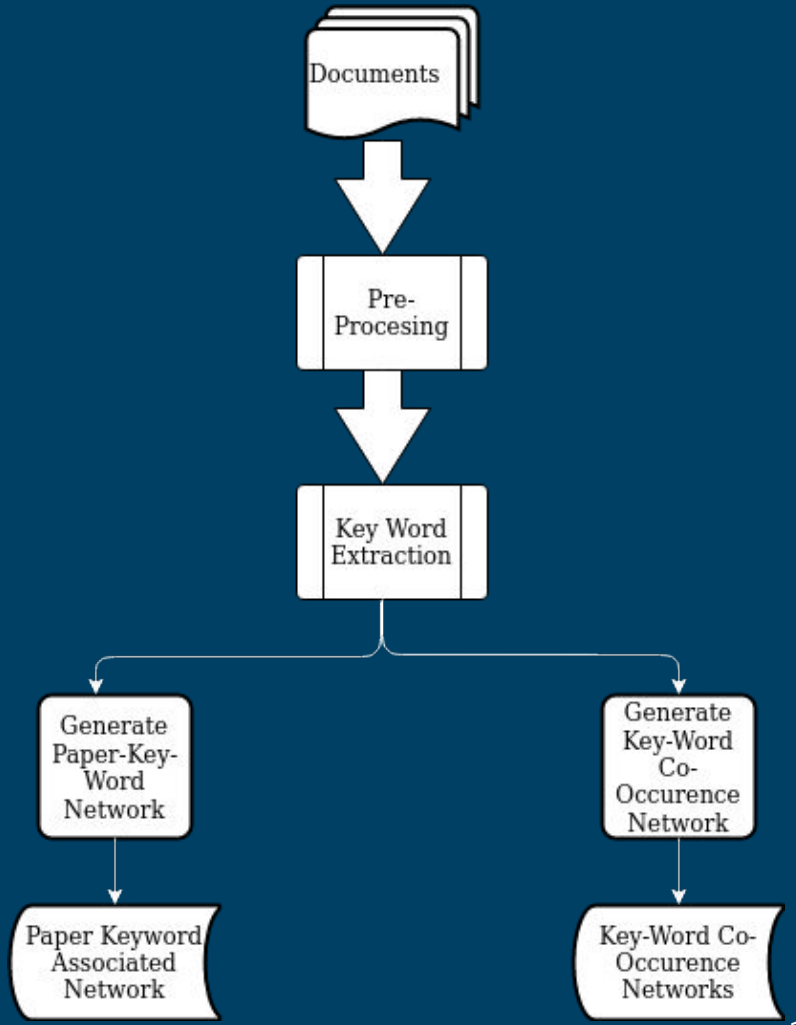
Makes use of a Graph based approach involving use of Keyword Co-Occurrence Network and Paper Keyword Associated Network.

Divided into two sections :

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

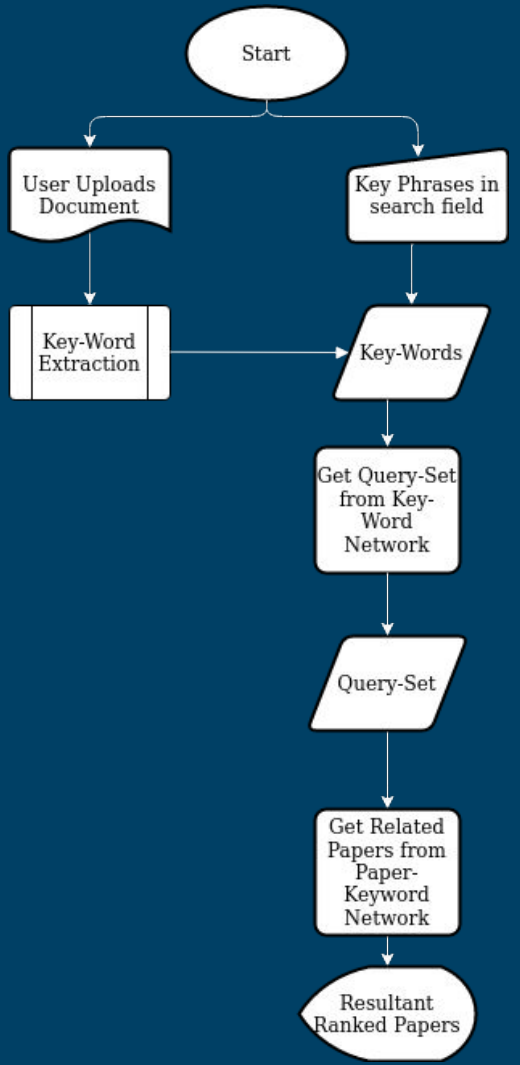
Model Preparation

(explains preprocessing and network creation stages)



Model Application

(explains how the user interacts with the system to get the recommendations)



Implementation and Results

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

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

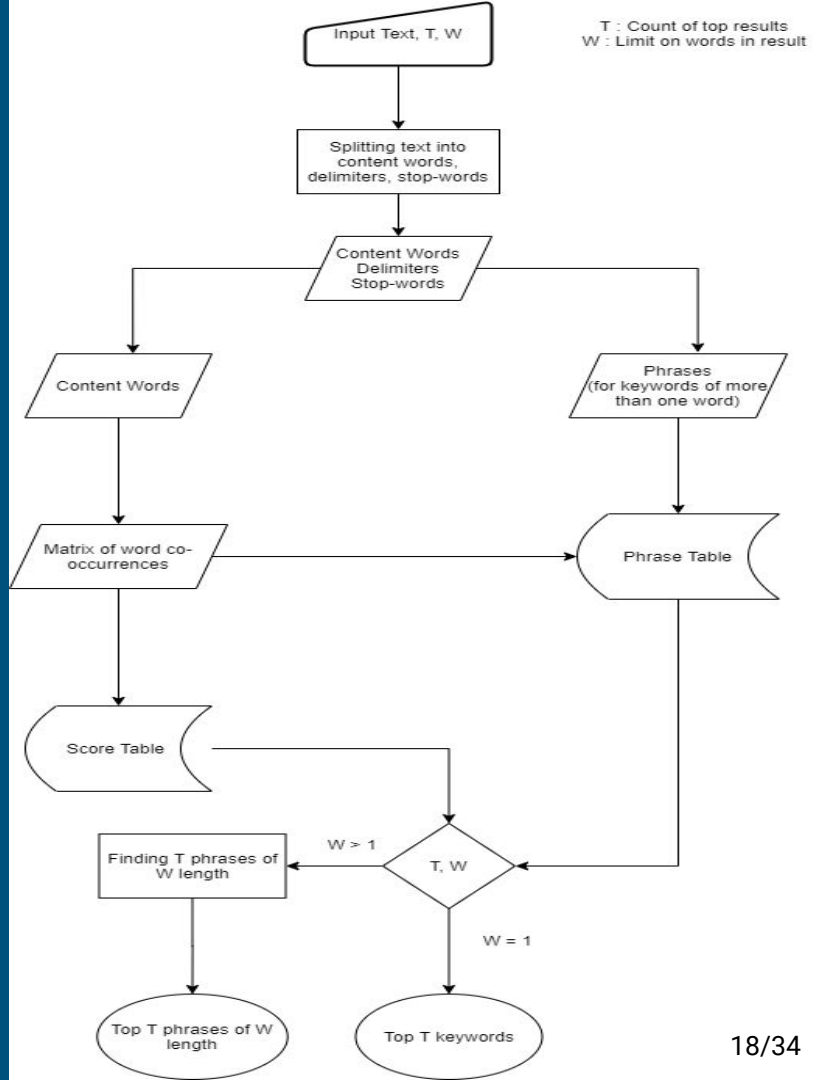
Keyword Extraction - RAKE

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.

Score = degree of word / frequency of word

5. 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



Keyword Extraction - 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 : YAKE

Keyword Extraction - 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 : keyBERT

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 have been implemented in Python and offers 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.

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.

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

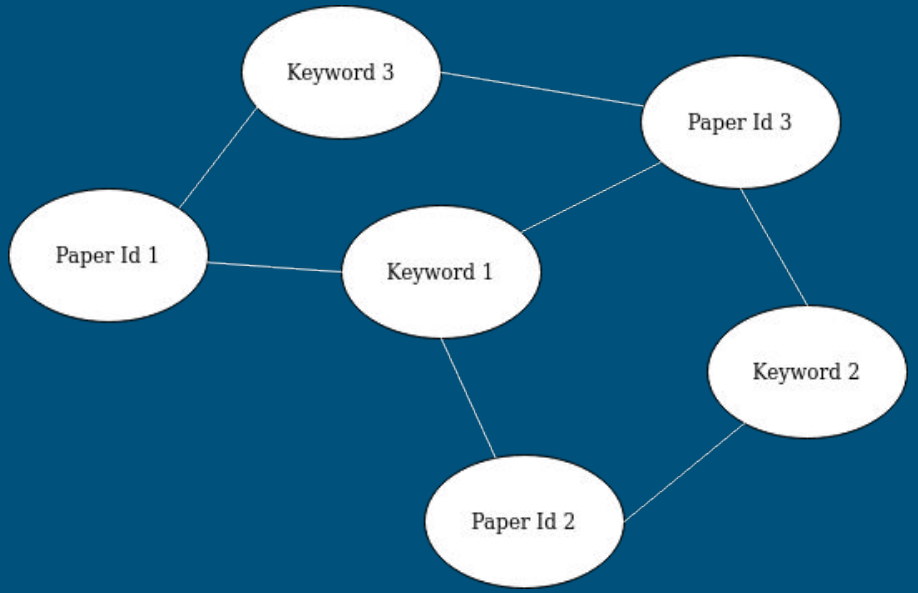
Papers List for K1: [P1, P2, P3, P4, P5, P6]
Papers List for K2: [P1, P2, P6, P7, P8]

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.



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.
   a. Path Similarity: It is output of 1 divided by the shortest distance between the given two words in wordnet taxonomy for an entity.
   b. Wu-Palmer Similarity : It calculates relatedness using depths of two synsets along with taking the depth of their Lowest Common Ancestor.

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

Code: https://github.com/Am-Coder/Document-Analysis/blob/working/wordnet.ipynb

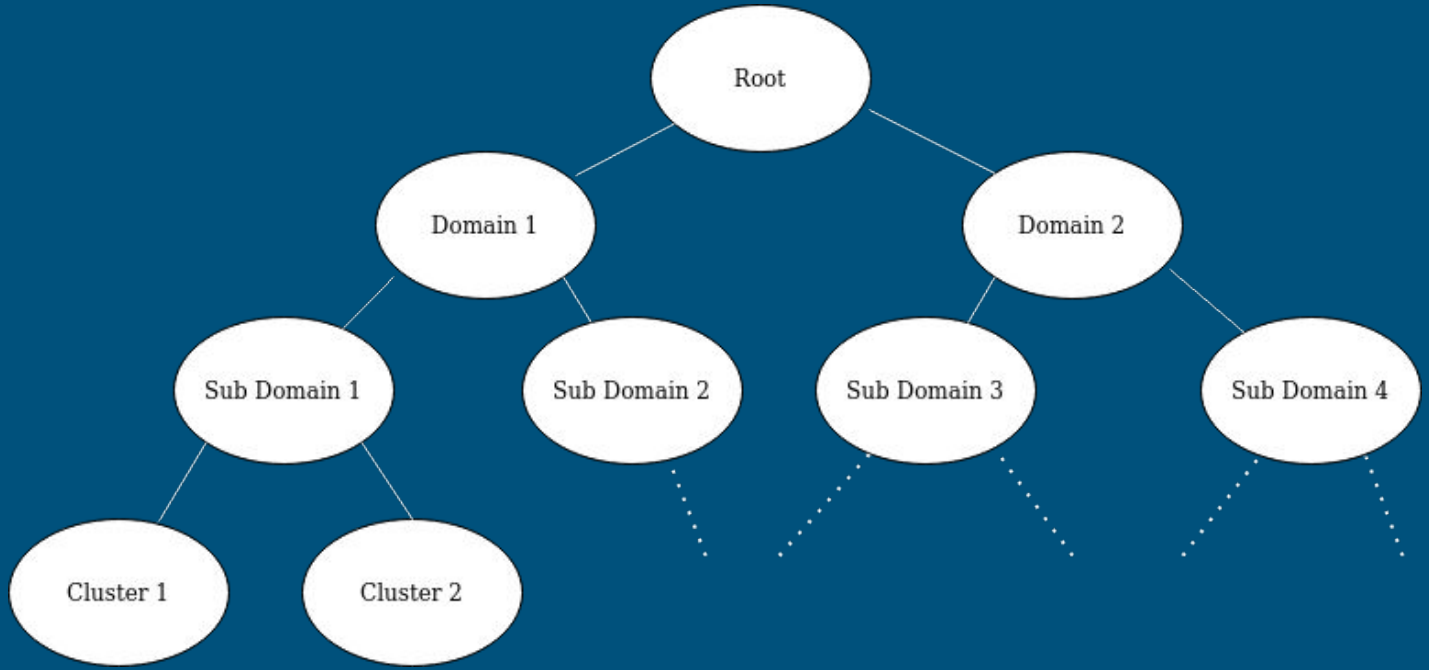
Wordnet Similarity Results

Phrase 1	Phrase 2	Similarity
machine learning	deep learning	0.085714285714285 72
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.
- The way forward is to make a custom ontology that removes this issue.
- Dataset to be used for this ontology: https://data.mendeley.com/datasets/9rw3vkcfy4/6

Custom Ontology Structure



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.

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 Prospects

- 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 I

[1] 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.

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

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

[4] 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.

[5] 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.

[6] 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

[7] 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.

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

[9] P. Resnik, “Using information content to evaluate semantic similarity in a taxonomy,” in IJCAI, 1995.

References II

[10] 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

[11] 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.

[12] 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.

[13] 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.

Thank You!!