KLE Society's
KLE Technological University, Hubballi.

A Minor Project Report

on

Wiki-Tree – A Concept Relation Tree

submitted in partial fulfillment of the requirement for the degree of

Bachelor of Engineering

in

Computer Science and Engineering

Submitted by

Kishor R Rao	01FE18BCS095
Utkarsh A Koppikar	01FE18BCS243
Jinesh Nagda	01FE18BCS083
Maltesh Kulkarni	01FE18BCS110

Under the guidance of
Prof. Prakash Hegade

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

Hubballi – 580 031

2020 -21

KLE Society's
KLE Technological University, Hubballi.

2020 - 2021

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that Minor Project titled Wiki-Tree – A Concept Relation Tree is a bonafide work carried out by the student team comprising of team Mr. Kishor R Rao – 01FE18BCS095, Mr. Utkarsh A Koppikar – 01FE18BCS243, Mr. Jinesh Nagda – 01FE18BCS083, Mr. Maltesh Kulkarni – 01FE18BCS110 for partial fulfillment of completion of sixth semester B.E. in Computer Science and Engineering during the academic year 2020-21.

Guide	Head, SoCSE
Prof. Prakash Hegade	Dr. Meena S. M

Viva -Voce:

Name of the Examiners	Signature with date
1.	
2.	

REF: 2021/WL-04

Knit Arena
Software Research and Services Private Limited,
Hubballi - 31.
Registration Number: 113756
www.knitarena.com

Project Completion Letter

This letter is to certify that the project titled WIKI-TREE – A CONCEPT RELATION TREE from Knit Arena was successfully completed by the student team **Mr. Kishor Rao, Mr. Utkarsh Koppikar, Mr. Maltesh Kulkarni,** and **Mr. Jinesh Nagda** from KLE Technological University as a part of their VI Semester Minor Project. The project work was carried out under the guidance of **Mr. Prakash Hegade**. The team performance was graded excellent and has positively contributed towards the industry segment growth.

Regards,

Vishwanath T
Director,
Knit Arena.
28 June 2021

ACKNOWLEDGEMENT

We would like to thank our faculty and management for their professional guidance towards the completion of the project work. We take this opportunity to thank Dr. Ashok Shettar, Vice-Chancellor, Dr. N.H Ayachit, Registrar, and Dr. P.G Tewari, Dean Academics, KLE Technological University, Hubballi, for their vision and support.

We also take this opportunity to thank Dr. Meena S. M, Professor and Head, SoCSE for having provided us direction and facilitated for enhancement of skills and academic growth.

We thank our guide Prof.Prakash Hegade, SoCSE for the constant guidance during interaction and reviews.

We would also like to acknowledge the constant help and support provided to us by Knit Arena Software Research and Services Pvt. Ltd.

We extend our acknowledgement to the reviewers for critical suggestions and inputs. We also thank Project Co-ordinator Dr. Sujatha C. and faculty in-charges for their support during the course of completion.

We express gratitude to our beloved parents for constant encouragement and support.

Kishor R Rao - 01FE18BCS095
Utkarsh A Koppikar - 01FE18BCS243
Jinesh Nagda - 01FE18BCS083
Maltesh Kulkarni - 01FE18BCS110

ABSTRACT

Wikipedia is a wide-ranging database of human knowledge continually updated by a large community of contributors. The lack of defined distinctions, structures devoid of direction, and categorization within the system puts a reader down a rabbit hole of information. A reader is often confused with too much or too little information, hyperlinks, and related articles. In response to such scenarios, we propose a model that, for user input, provides a path that includes both generic and specific articles relevant to it. The articles crawled are represented as graphs based on the strength of the relationship between them measured via the shared links. These are then used as input to Minimum Spanning Tree algorithms to generate tree structure for the same. Besides, every node in the tree is annotated, enabling the user to test the credibility of information for their study. The results from the domains analyzed exhibit the relationships that exist within and across. It also provides the path taken to reach that output. A prospective future extension can be to subject the entire database to the system that might correlate with various other domains.

Keywords : *Annotations, category, graph, hierarchy, operational efficiency, tree*

CONTENTS

LIST OF FIGURES

Chapter 1

INTRODUCTION

The world is governed by hierarchies. Class structure and stratification have been at the center of evolution, leading to the organization and formation of categories. The concept of narrowing the scope with each subsequent step makes space for everyone, and everything is of exceptional importance in comprehending any data. Biological taxonomy, food chain, postal address system, and most of the computer networks are hierarchies. This exploration of orders and mapping them to Wikipedia- The Free Encyclopedia is presented in the project.

Technology and data have grown in tandem, if not as a consequence. This drastic change in the availability, similar to a butterfly effect, only generates more data due to flexibility in conducting extensive research reinforced by technological resources. This is another reason to have access to credible information supported by proper representation.

The present era is termed the age of information, as every action carried out by humans contributes to data generation. The need for defining and coining new terms for each of the actions taken has only contributed to the plight. The granularity of this accumulated data for every concept poses the complex task of documenting and categorizing it. With a larger amount of data, the need of the hour is an organized and lightweight version of the data, which summarises the available data with precise annotations.

The challenge of translating Wikipedia articles to various other languages is a challenge that various researchers have considered. However, categorizing and organizing this digital encyclopedia is still a significant challenge. We take up this convoluted task and attempt to resolve it.

1.1 Motivation

The massive size of Wikipedia proves very difficult to work with. Every concept and topic from Vedic mathematical principles to modern science, from traditional music and symphonies to KPOP bands and rock bands of today, from Socrates philosophy to today's mental health tendencies based off on the philosophical questions, from theories and hypotheses to real-world facts about people and phenomenon, everything is available on an online encyclopedia called the Wikipedia. The massive number of contributors that help Wikipedia in becoming what it is by writing articles and providing data to its database, which then becomes publicly available for everyone to read, is also food for thought for various researchers who analyze the consumption patterns of this data. Hence, researchers have come up with various techniques and methods to make it simple, understand its working, analyze the relationship of the various articles present in its database, and the nature of their relationship forms a network. The goal is to dismantle the said network and study it in isolation.

This same thirst for deriving insights from Wikipedia, now called a corpus for knowledge extraction, has led to properties within it like semantic relatedness, structural disambiguation of the database, and projects related to the same, to name a few. The same reason motivates us to solve this problem and create something that helps other users use Wikipedia data efficiently.

1.2 Literature Survey

Wikipedia has seen exponential growth and is now considered a very reliable source, making it vital for intelligent systems. It covers many concepts of various fields such as Arts, Geography, History, Science, Sports, and Games. It has become a frontier for researchers to derive and build on this data for various technical and non-technical purposes. Gone are the days of data where encyclopedias were a burden and unintelligible for machines. Wikipedia has now become a database that stores all human knowledge. Thus, it promises that Wikipedia can be an excellent source of information for various data mining studies.

Acquiring knowledge from the categories and category network of Wikipedia is a challenge on its own. This challenge can be countered by categorizing at the same time acquiring hidden knowledge of Wikipedia [1]. For the most part, Wikipedia categories are comprehensive and vague even after their complex names are overlooked.

These names closely follow the rules of human classification and thus organize the instances by encoding the data about class attributes, taxonomic, and other semantic relations. These names are then decoded to refer back to the network, and relations are induced between the concepts in Wikipedia represented conveniently through pages or relevant categories. This structure propagates the relation detected between the elements of a category name to a variety of concept links and supported the hypothesis of Wikipedia being a rich source of valuable knowledge.

While building the network of Wikipedia, one major hurdle faced is the problem of Dis-ambiguity [2]. Distinguishing articles that have similar names but refer to distinct properties and topics is a significant challenge. A large-scale system is a solution to the challenge. Systems have been designed which recognize and disambiguate the named entities based on semantics and information extracted from an extensive collection of encyclopedic and Web search results. A detailed description is stated regarding the disambiguation paradigm used and the process of extracting information from Wikipedia. Optimizing the agreement that signifies the relation between the context of a document and the contextual information extracted, the model also delves deep into the tags associated with the entities taken into consideration. The implemented system exhibits high accuracy on all the fronts proposed initially.

Subtree Mining models have patented that efficiently extract relations in Wikipedia articles [3]. The model makes data machine-processable and builds an algorithm to find relations between two or more data concepts. Here, the focus is more on extracting relations among English articles of Wikipedia that can eventually be used in intelligent systems and satisfy users' information needs. The proposed method primarily fixates the appearance of the said entities using heuristic rules supported by their encyclopedic style. Consequently, it does not use Named Entity Recognizer (NER) or the Coreference Resolution tool, both prominent sources of errors concerning extraction. SVM is employed to classify the relationships among entity pairs with the extracted features from the Web and the subtrees mines from the syntactic structure of the text. The method proposed makes use of characteristics of Wikipedia to allocate entities and classify them. The algorithm extracts a core tree that reflects a relationship between a particular entity pair accurately and further identifies critical features concerning the relationship from the core tree. Demonstrating the approach's effectiveness by evaluating manually annotated data from actual Wikipedia articles is also done.

Semantic relatedness is just a numerical strength of relation but does not have an explicit relation type. To extract inferable semantic relations with explicit relation types, we need to analyze the link structure and texts in Wikipedia. Enhanced optimization techniques for Wikipedia mining [4] have been researched upon.

These techniques are instrumental in extracting semantic relations from mined Wikipedia data. This detailed evaluation further proves that this approach of link structure mining improves accuracy and the scalability of the extraction of semantic relations.

PFIBF (Path Frequency - Inversed Backward link Frequency)[5] is an efficient link mining method. PFIBF and the extension method, "forward / backward link weighting (FB weighting)," effectively construct a large-scale association Thesaurus. These are extensions of analyzing association link extraction and knowledge acquisition for Thesaurus building. There is proof of the effectiveness of our proposed methods compared with other conventional methods such as co-occurrence analysis and TF-IDF.

1.3 Problem Statement

"To design and develop a knowledge system for Wikipedia data, which can categorize articles under different domains while creating a network of related articles. At the same time, briefly summarising an article with precise annotations."

1.4 Applications

Categorizing each of the articles into their root domain, the system can be used as follows:

- Ontology: to understand properties and relationships
- Multi-hop Connections: Walk-through from entry level to core domain of a concept
- Recommendation Systems: To suggest related articles
- NLP: Word sense disambiguation based on domain knowledge
- Application: Article annotations very similar to InShorts
- Application: Dynamic concept tree generator useful for research or light reading
- Application:Measure of Strength detector for given input of articles

1.5 Objectives and Scope

1.5.1 Objectives

- To represent Wikipedia data as concept tree
- To establish relationships between data concepts
- To build annotations for the analysed data
- To represent data for operational efficiency

1.5.2 Scope

There are various applications currently being drawn out, such as building a thesaurus for Wikipedia. Given the data size, there are many parameters to be considered and worked upon to enhance its marketable value. Building an annotated tree on the relationship between two or more articles adds to such a solution and can be implemented incrementally with the mentioned approach. Most of the APIs and interfaces to interact with the Wikipedia data are currently outdated; the focus can also be upon creating an informative and easy-to-use GUI that enables users to explore the database efficiently and effortlessly. Monetizing the said utility software then again depends on the licenses. The functions and methods used here are more generic and do not constrain themselves to the specifics of the data under consideration. They thus can be used upon any data and are not limited to only Wikipedia, opening many avenues for the idea to grow and adapt, eventually becoming a resourceful and helpful utility for the public.

Chapter 2

REQUIREMENT ANALYSIS

Requirement Analysis, also known as Requirement Engineering, is a critical process that includes understanding and articulating the users' expectations regarding the application being prepared for them. Also, know as the process of gathering the requirements in conventional software engineering terms, the output of this is often termed as Software Requirements Specification. These include the Functional Requirements, the operations that software performs on interaction with the user, Non-Functional Requirements, the operations and maintenance happening behind the scenes invisible to the user. Requirement Analysis takes into consideration everything and every opinion from the various stakeholders, users, owners, and investors. It is also a document to track the various versions of the software and how the requirements have changed over time.

2.1 Functional Requirements

User:

1. The user shall provide input serving as a root node

System:

1.1 The system should accept the input given by the user

1.2 The system should validate the input

1.3 The system should give appropriate warning message if the input is invalid

1.4 The system should generate the concept tree with the input as the root node

User:

2. The user shall give 2 inputs in case of finding a relationship between them

System:

2.1 The system should accept the two inputs

2.2 The system should validate the inputs

2.3 The system should give appropriate warning message if the input is invalid

2.4 The system should find the relationship between the two inputs

User:

3. The user shall be given with output according to their input

System:

3.1 The system should generate the output based on the input type

3.2 The system should display the output

2.2 Non Functional Requirements

1. Availability of the system = 0.99
   Mean Time to Failure (MTTF) : 100 hrs
   Mean Time to Repair (MTTR) : 1 hr
   Mean Time Between Failure (MTBF) : 100+1 = 101 hrs
   Availability = MTTF / MTBF = 100 / 101

2.3 Hardware Requirements

1. The minimum speed of the processor should be 1.9 gigahertz (GHz) x86- or x64-bit dual core processor but the recommended processor should be 3.3 gigahertz (GHz) or faster 64bit dual core processor.

2. The minimum Memory should be 2-GB RAM but the memory recommended is 4GB RAM or more.

3. Either Ethernet connection (LAN) OR a wireless adapter (Wi-Fi) is mandatory to operate to download the Wikipedia data .

2.4 Software Requirements

1. Recommended Operating Systems
    - Windows: 7 or newer
    - MAC: OS X v10.7 or higher
    - Linux: Ubuntu
2. Recommended Browsers
    - Firefox
    - Google Chrome
    - Microsoft Edge
    - Opera

Chapter 3

SYSTEM DESIGN

System Design is one of, if not the most critical phase of a Software Engineering project. Here, the designer has to divide the project into modules, where each module will answer one or more objectives. These modules must be specified systematically in the System Architecture of the project. Each module's interfaces, inputs, and outputs must be designed and planned even before even the implementation has begun. This will give the programmer a clear goal to achieve and provide the project manager with a blueprint to complete the project.

3.1 Architecture Design

Figure 3.2 depicts a general flow of the system and how the data traverses from one part to another, with each step getting closer to the desired output. Wikipedia data is first from one of its "Special Pages" called "What Links Here." It is then stored in a python dictionary. Similarly, the data from infoboxes is also acquired, and a textual summary is stored. The dictionary that contains the list of links becomes the input for the model that generates graphs and trees based on the relationship between the articles. This module tells us the existence of a relationship between two articles and can also indicate a sense of the strength of the mentioned relationship. With the help of this, a path is drawn out for any related root article. The textual summary and the data from the infoboxes become input to the system that generates annotations for the articles. These annotations give a brief and concise summary of the article to comprehend for the user's benefit.

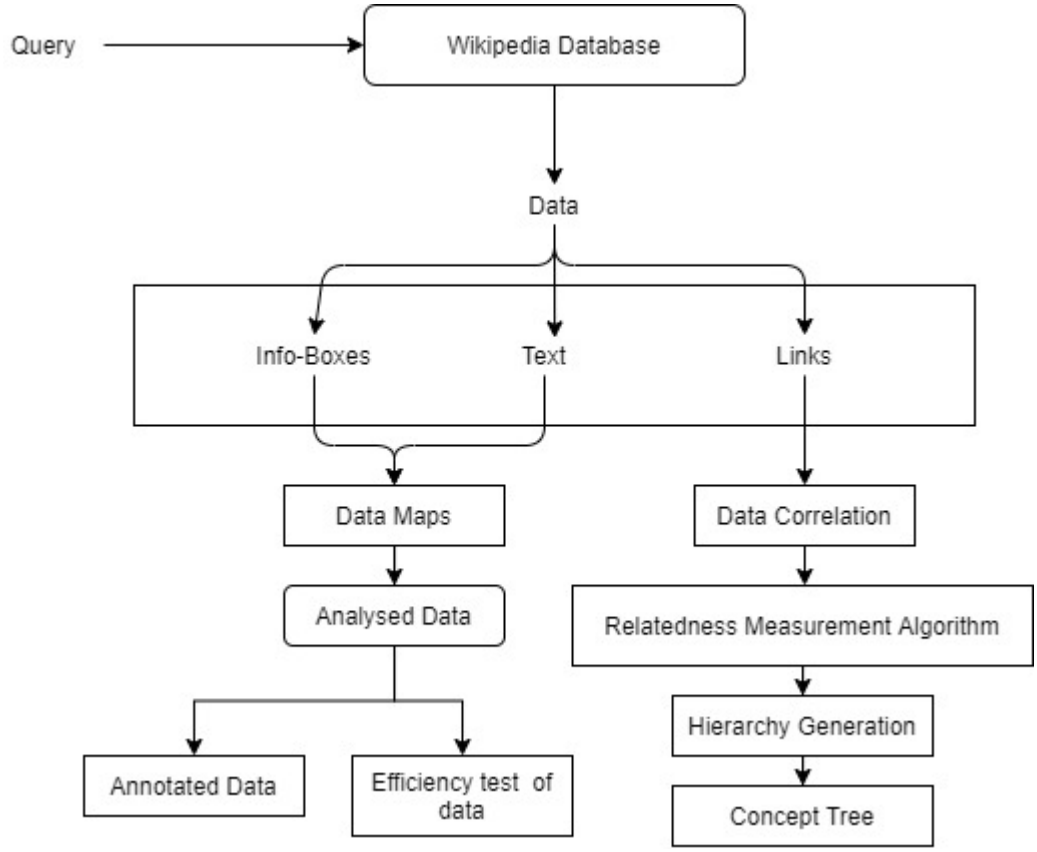


Figure 3.1: Proposed System

3.2 Data Dictionary

Acquiring data posed many challenges. Wikipedia releases their entire database and archives for the public to use in any way they see fit. There were no suitable sources and methods to retrieve this data from their encoded formats. There are various Python libraries, pybots, and crawlers used to get access to this information. Most of these, now being outdated, had little to no contribution to the project.

As a solution to the previously mentioned challenges, customized functions were written with the help of a python library called Wikipedia and the famous web-scraping library, BeautifulSoup. With the helper functions of these and required modifications blended in, it was possible to get the data in exactly the format required for the system to process them.

The data was segregated into three parts, all in the form of a python dictionary or JavaScript Object Notation saved in JSON files. The first kind has a list of links along with the name of the article. The second type has the article's name as the key, and the value is another python dictionary of info-box data of each article.

The final fragment of the input contains the textual summary of the articles. The following figures are the screenshots of the structure of our data set.

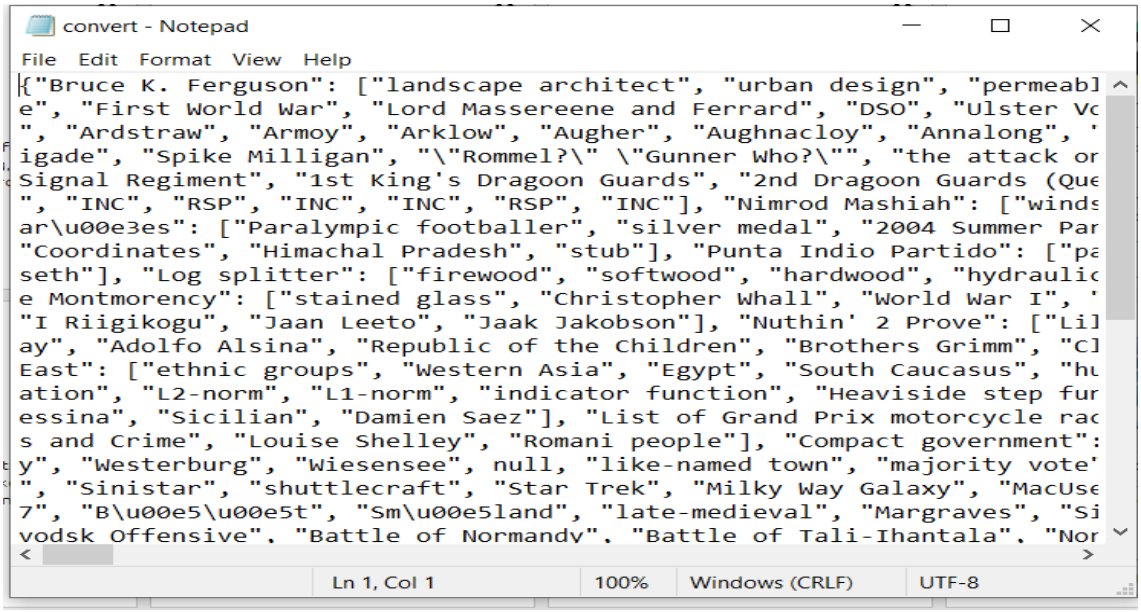


Figure 3.2: Links directing to the current article

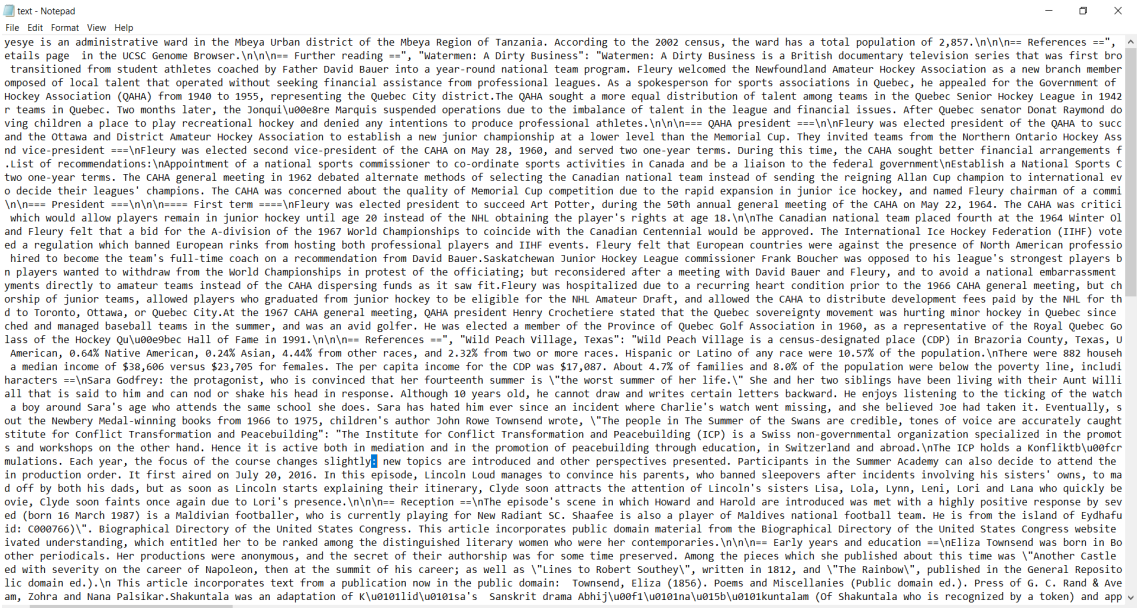


Figure 3.3: Textual Summary of the current article

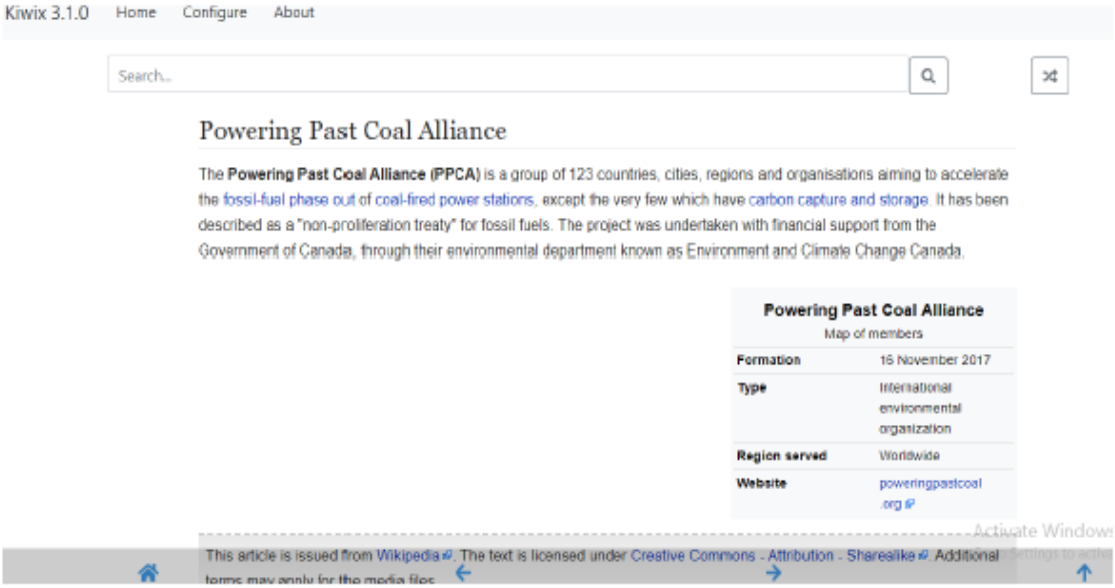


Figure 3.4: Infoboxes from Kiwix

3.3 Detailed Designs

Two prominent elements of the model are the graph and tree generator and the annotation generator. Figures 3.5 and 3.6 show the flowchart of these modules.

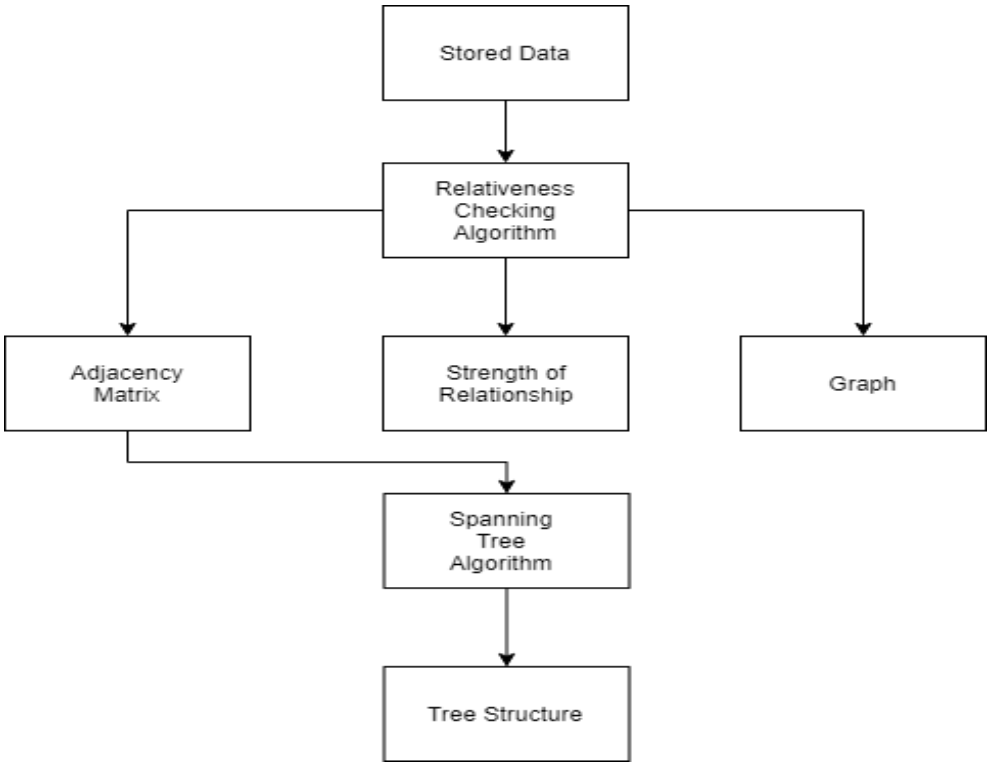


Figure 3.5: Model for Graph and Tree generation

In order to bring this amount of information down to just a few lines while maintaining the purity of the information our module "Annotating Wikipedia articles" has been built.The major challenge in annotating articles on Wikipedia is the versatility of the articles on the site. The topics are of wide range and a generic method of annotating an article might work perfectly for one topic while it would look improper for others. This keeping all this into consideration, a model has been proposed as shown in the Figure 3.7.

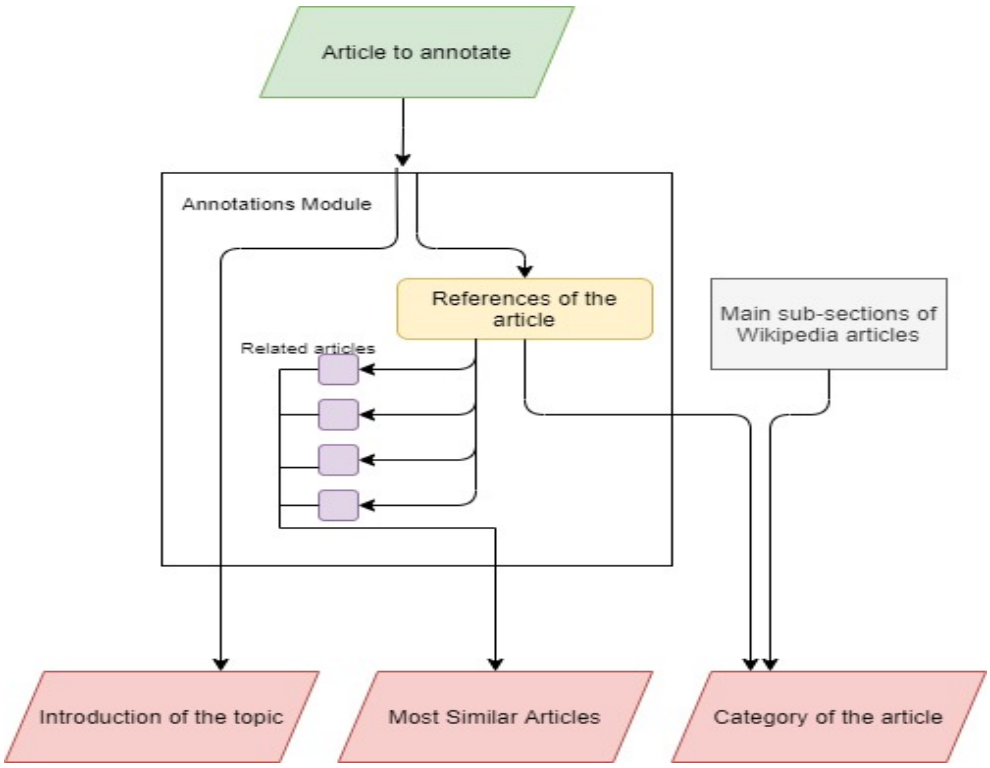


Figure 3.6: Model for annotation

Figures 3.7 and 3.8 portray the algorithms created for graph and tree representation of the articles. Pseudo-code of how a link between any two articles is established with the help of graph concepts and the road-map of how multiple articles are connected to each other and analyses all the connections and gives out the most optimal path to lead to a particular article.

```
articles ← Wikipedia.article()
while article! = articles.end() do
  article1, article2 ← articles.adjacent()
  links1 ← article1.links()
  links2 ← article2.links()
  while links1, link2 ≤ link.end() do
    if match(link1, link2) ≥ threshold then
      article1 ← article2
    end while
  end while=0
Display Graph
```

Figure 3.7: Graph Representation

```
articles ← Wikipedia.article()
while article! = articles.end() do
  links[max] ← root
  links[max − 1 ← childNode[]
  while i ≤ childNode[] do
    if match(root, childNode) ≤ threshold then
      maxSimilarity ← childNode[i]
    end while
  end while=0
Display tree
```

Figure 3.8: Tree Representation

Chapter 4

IMPLEMENTATION

Since Wikipedia is a database with millions of articles under its hood and new ones being added every day, it is a humongous task, and rather impossible with the resources available, to take in the entire data-set into the consideration. Hence, for ease of understanding and to establish a flow of comprehension from one module to other, five major clusters/domains of data that might produce interesting results are considered and explored in greater depth. The five domains are namely, Philosophy, Maths, Science, Logic and, Biography. Each being an independent topic and field by itself, the chosen domains also have a clear overlap between them and it is this commonality that is desired to unearth insightful details. The said objective is achieved via five modules.

The first module traverses from the root article that is given as an input to the system. It traverses the links present in the input article and produces a list that can be reached. This list of articles is then scraped for their data in the second module using the customized functions to retrieve the links and textual summary. These links become our foundation to build an adjacency matrix that helps us in drawing a graph. This graph gives a visual representation of the data taking the articles as nodes and edges representing the existence of a relationship between them. This adjacency matrix is further subjected to a minimum spanning tree algorithm to generate another matrix that is free of cycles and consisting of all the nodes considered. This matrix is drawn to generate the tree structure of the data-set. To each of the articles/nodes present in the graph, a three-line annotation is generated focusing on the nearest domain that the article belongs to, a one-line summary/definition of the article, and the list of strongly related articles the article currently under study. A detailed description of each of the modules is given in the ensuing pages.

4.1 Discovery through traversal

This is the part where all the related articles relevant to the user chosen domain are found by traversing from the root node. This is done using a Python library called Wikipedia. It can sometimes be different, for example, Science doesn't have a page in the exact same name as it has a disambiguation in its name and in such cases we go with the nearest article, in this case, Philosophy of Science. We gather 50 to 100 articles from the root node selecting a random link in the current page and traversing to the next. A sample image of this is shown below in figure 4.1.

The following is an example of the list we have obtained for the Philosophy domain ['Greek', 'existence', 'reason', 'knowledge', 'values', 'mind', 'language', 'Pythagoras', 'Philosophical methods', 'questioning', 'critical discussion', 'rational argument', 'philosopher', 'Ancient Greek', 'Aristotle', 'natural philosophy', 'astronomy', 'medicine', 'physics', 'Newton', 'Mathematical Principles of Natural Philosophy', 'research universities', 'professionalize', 'psychology', 'sociology', 'linguistics', 'economics', 'metaphysics', 'existence', 'reality', 'epistemology', 'knowledge', 'belief', 'ethics', 'moral value', 'logic', 'rules of inference', 'conclusions', 'true', 'premises', 'philosophy of science', 'political philosophy', 'aesthetics', 'philosophy of language', 'philosophy of mind', 'knowledge', 'Lives and Opinions of the Eminent Philosophers', 'Diogenes Laërtius', 'Pyrrhonist', 'Sextus Empiricus', 'Academic Skeptic', 'Cicero', 'astronomy', 'chemistry', 'biology', 'cosmology', 'social sciences', 'value theory', 'aesthetics', 'political philosophy', 'mathematics', 'philosophy of science', 'Newton', 'Mathematical Principles of Natural Philosophy', 'natural philosophy', 'astronomy', 'medicine', 'physics']

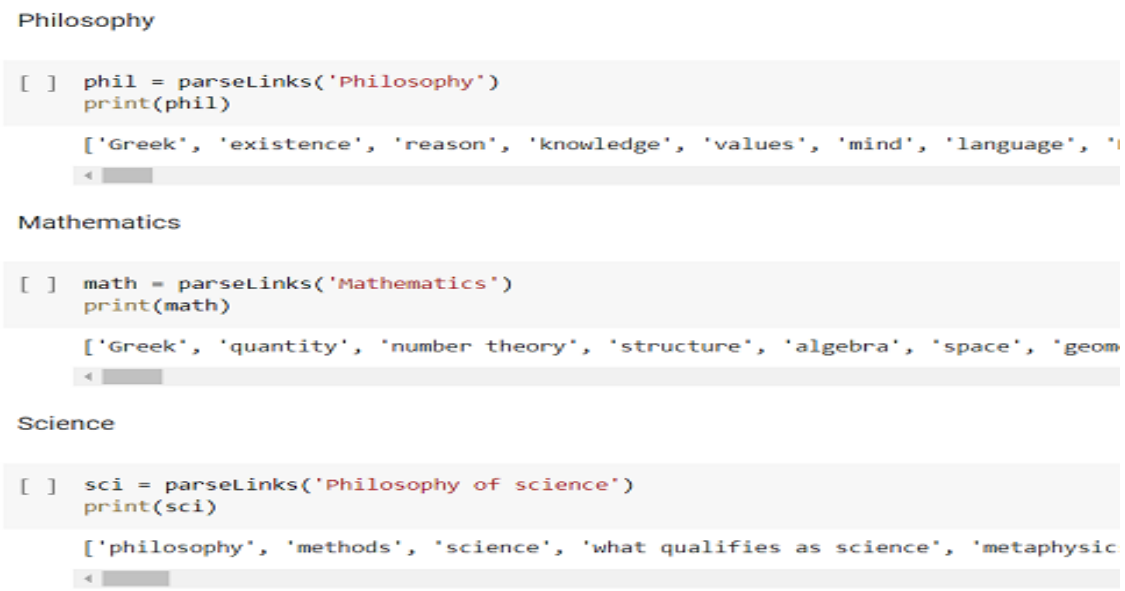


Figure 4.1: Article discovery from root node

4.2 Data Acquisition and Storage

In this module, the discovered articles in the previous module are scraped using the 'BeautifulSoup API'. the "Special Page" of each article that is, "What Links here" is retrieved with the name-space component as article. The number of links are limited to 50 for ease of processing but each link can have some sub links associated with it. These are stored as a python dictionary with the article name and the links directing it to the specified, key being the name of the article and value being a list of links. A sample snapshot of the dictionary for the cluster of Philosophy is shown below in figure 4.2.

```
In [7]: philosophy = {}
        for item in list1:
            url = item
            data  = requests.get(url).text
            soup = BeautifulSoup(data,"html5lib")
            name = soup.find('title')
            name = name.string
            start= '"'
            end = '"'
            title= (name.split(start))[1].split(end)[0]
            articlelinks = []
            paras = soup.find_all('ul',{"id": "mw-whatlinkshere-list"})
            for para in paras:
                links = para.find_all('a')
                for link in links:
                    data = re.findall('href="/wiki/', str(link))
                    if len(data)!=0:
                        x = re.findall("\[*\]", str(link.string))
                        if len(x)==0:
                            articlelinks.append(link.string)
            philosophy[title]=articlelinks
        philosophy

Out[7]: {'Greek': ['Ancient Greek',
          'Greece (disambiguation)',
          'The Greek',
          'Hellenic',
          'Greco (surname)',
          'Outline of Greece',
          'Grković',
          'Greeks (disambiguation)',
          'Macedonian Greeks (disambiguation)'],
         'Existence': ['Aristotle',
          'Altruism',
          'Arthur Schopenhauer',
          'Aikido',
```

Figure 4.2: Link Extraction

4.3 Graph

This module focuses on creating a graph, taking each of the 20 articles as nodes and creating an edge between them if there are 5 or more links common between two articles. For each of the 5 clusters, a 20 cross 20 matrix is created indicating the adjacency matrix of the articles. A graph is plotted based on this adjacency matrix and is shown in Figure 4.3. Even though the weights are not shown in the graph, the number of links common between any two articles act as weights and are used in the next module.

```
0   Greek
1   Existence
2   Reason
3   Knowledge
4   Value (ethics)
5   Mind
6   Language
7   Pythagoras
8   Philosophical methodology
9   Questioning
10   Philosopher
11   Ancient Greek
12   Aristotle
13   Natural philosophy
14   Astronomy
15   Medicine
16   Physics
17   Newton
18   Philosophiæ Naturalis Principia Mathematica
19   Research university
```

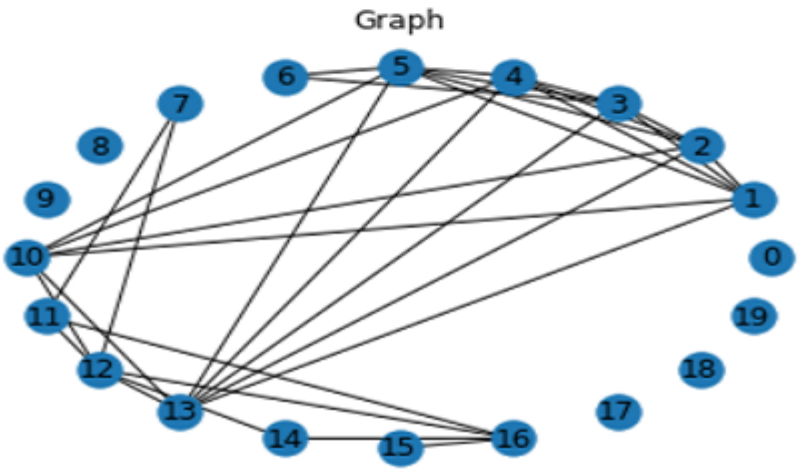


Figure 4.3: Graph representation for the articles in Philosophy cluster

4.4 Tree

The adjacency matrix created in the last module is modified to replace all the 0s to infinity (a value equivalent to it.) Minimum Spanning Tree algorithm is applied to this adjacency matrix resulting in another matrix including all the nodes without any cycles present in the graph created formerly. A tree representation is shown in the figure 4.4.

```
df1 = pd.DataFrame(data=tree)
f=plt.figure()

G1 = nx.OrderedGraph(df1.values)
nx.draw_planar(G1, with_labels=True)
```

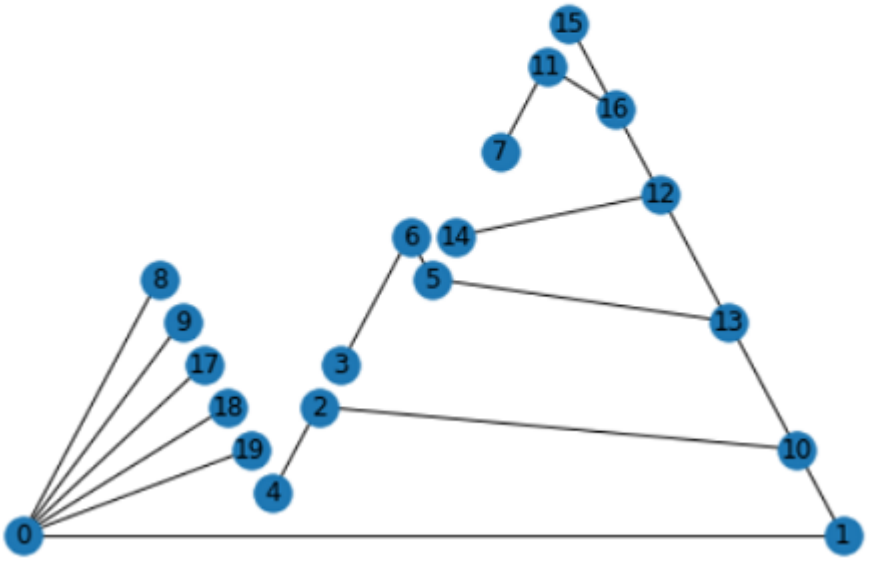


Figure 4.4: Tree representation for the articles in Philosophy cluster

Another attempt was also made to check the relationship between all the 100 articles extracted to see how each of the articles are related to each other across the five clusters chosen in this project. The graph connection is as shown below in Figure 4.5. The results are pretty intuitive. 'Networkx API' has been used here to obtain the graph, after generating the adjacency matrix. Further minimum spanning tree algorithm has been applied on this graph to obtain a spanning tree representation of this graph, as shown in figure 4.6.

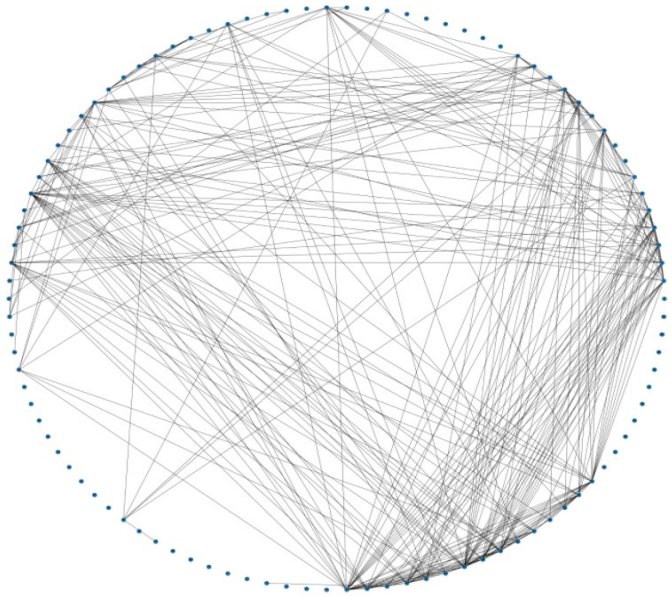


Figure 4.5: Graph representation of all 100 articles across five clusters

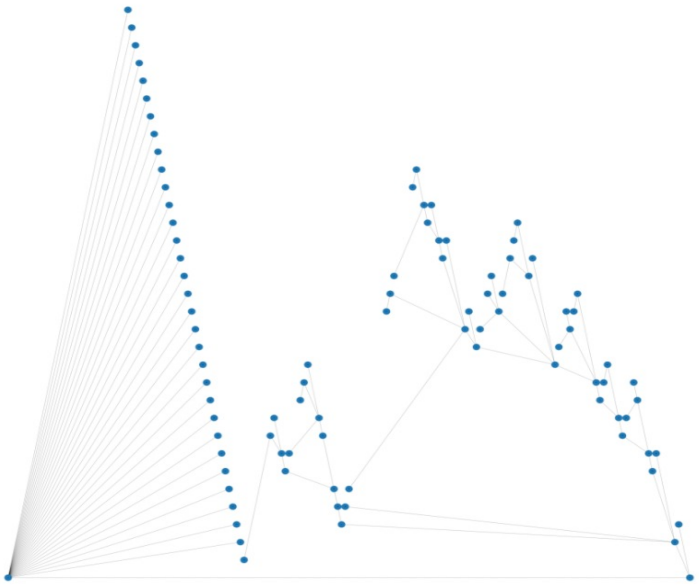


Figure 4.6: Tree representation of all 100 articles across five clusters

4.5 Annotation

It has been observed that Wikipedia articles provide excellent information about almost all the topics in the world. But in this age of fast paced knowledge a 1000+ word article is discouraged. In order to bring this amount of information down to just a few lines while maintaining the purity of the information our module "Annotating Wikipedia articles" has been built. The major challenge in annotating articles on Wikipedia is the versatility of the articles on the site. The topics are of wide range and a generic method of annotating an article might work perfectly for one topic while it would look improper for others. Therefore, this module has been built keeping in mind the variety of topics on Wikipedia and the need to briefly describe them. This module takes the name of the article as its only input and gives out the annotations in three different ways. First one being the introductory line of the article on Wikipedia. The first line (after cleaning the data) always answers to the question of 'what this topic actually is'. After this brief explanation the next output answers to the question of under which sub section does this article come under. To achieve this target, we need to effectively categories articles on Wikipedia under different umbrella terms. The sections that have been identified are "Philosophy", "Mathematics", "Philosophy of Science", "Biography", "Geology" and "History". To categories the topics we calculate the similarity in the article references in each topic and the various sub-section headings. The sub-section with which the topic has the most similarity is added to that sub-section. In this way we are able to tell the user where the article has evolved from. The last output for each annotation will be, the most similar topics to the current topic. By this we are giving the user tags of similar articles he/she would like to further read about to have a comprehensive understanding of all the correlated topics. To meet this objective, we are checking all the articles referenced in the current article and the top three most similar topics among these referenced articles.

By providing these annotations we are able to tell the user 'What the topic is?', 'Where does it come from?' and 'What are the topics related to it?'. And these questions are answered in an unbiased way by the algorithm without much changes over different topics. Some examples can be seen in the Figure 4.7.

```
[ ]  annotate("Albert Einstein")

['Albert Einstein  was a German-born theoretical physicist, widely acknowledged to be one of the greatest physicists of all time',
 '< Albert Einstein , Philosophy of Science>',
 ['<theoretical physicist>', '<quantum mechanics>', '<modern physics>']]

[ ]  annotate("World war")

['A world war is "a war engaged in by all or most of the principal nations of the world"',
 '< World war , History>',
 ['<war>', '<Cold War>', '<War on Terror>']]

[ ]  annotate("Artificial Intelligence")

['Artificial intelligence  is intelligence demonstrated by machines, unlike the natural intelligence displayed by humans and animals,
 '< Artificial Intelligence , Philosophy of Science>',
 ['<intelligence>', '<machines>', '<displayed by humans>']]
```

Figure 4.7: Annotations

Chapter 5

RESULTS AND DISCUSSIONS

The tree representation and annotations together speak volumes about an article under consideration. Dividing the data into clusters which is also a method of Wikipedia helped us not only to understand the relationship between the clusters but also the ones that act as stepping stones to transition from one cluster to another. These clusters are as shown in figure 5.1 below. The overlap between these distinctions can be seen in a more fundamental way for some cases, for example, the relation between Philosophy and Science can be seen with the article Ambiguity. For others, a bit more research is needed, again taking a parallel between Philosophy and Science cluster; a strong relationship is established between Benjamin Franklin and Animism, a topic that can only be understood when proper research on the belief and Franklin's life story is done. While the graph goes for the most mysterious and concealed-answers approach, annotations are straight forward. These provide 3 tags, namely, the cluster to which they belong, three closely related articles (for further exploration), and the first line of the article to give a general idea of what the article is about.

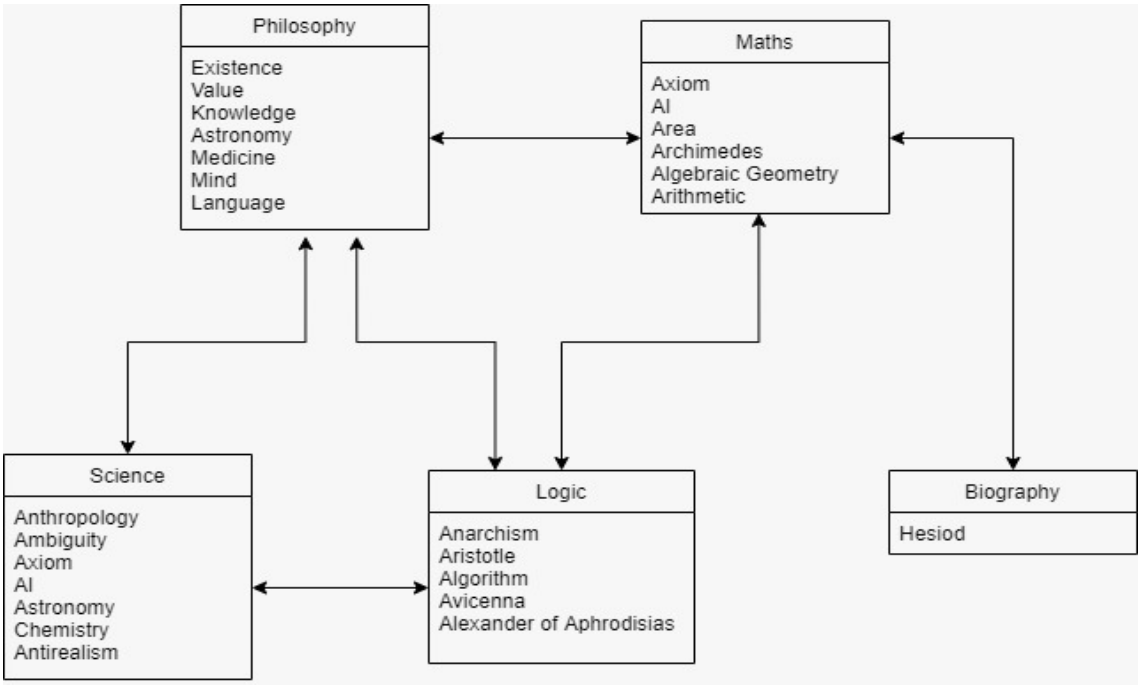


Figure 5.1: Interlinking of multiple domains

Figure 5.2 gives a comparative vision of how the relationship between the articles within each of the clusters. As can be seen in the figure, the connectivity graphs while having a bit more significance in finding more than one paths to travel to other nodes within the cluster, the tree structure condenses it to a minimum spanning tree taking the number of common links between them as weights. Focus should also be on the domain of Biography which has very little correlation for the articles within it but exhibits a diverse relationship with the articles across the cluster as shown in Figure 4.5

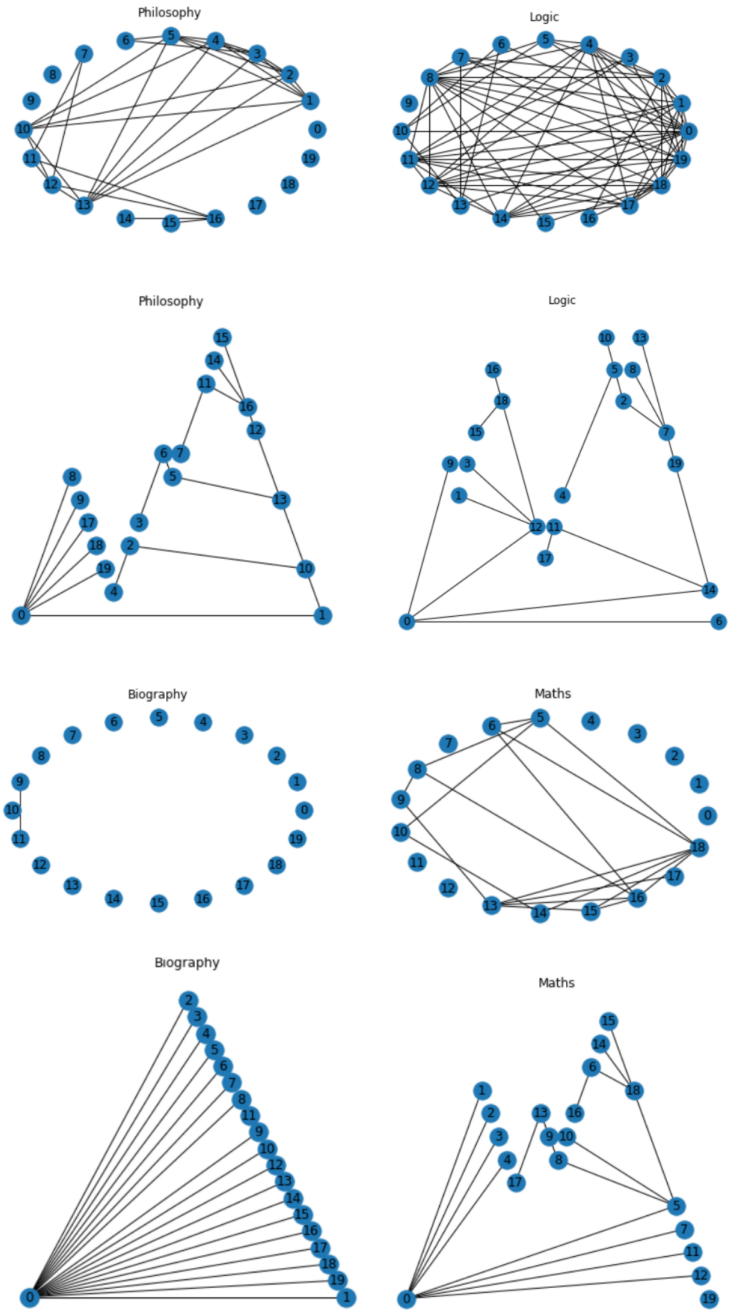


Figure 5.2: Comparative study of clusters

Some of the articles that have relationship across the clusters are consolidated in Figure 5.3.

Existence	Axiom
Existence	Ambiguity
Value(Ethics)	Axiom
Value(Ethics)	Augustine of Hippo
Axiom	Avicenna
Medicine	Avicenna
Astronomy	Astrology
Medicine	Astrology
Knowledge	Artificial Intelligence
Knowledge	Benjamin Franklin
Knowledge	Logic
Mind	Science
Mind	Afterlife
Mind	Aesthetics
Language	Bulgarian Language
Philosopher	Ancient Philosophy
Natural Philosophy	Albertus Magnus
Astronomy	Archaeoastronomy
Medicine	Chemistry
Physics	Logic
Physics	Computer Science
Axiom	Alan Turing
Anthropology	Afterlife
Anthropology	Anarchism
Anti-realism	Albertus Magnus
Chemistry	Logic
Benjamin Franklin	Animism
Pythagoras	Alexander of Aphrodisias
Archimedes	Alexander of Aphrodisias
Archimedes	Algorithm
Knowledge	Arthur Schopenhaur
Pythagoras	Area
Asia	Baghdad
Asia	Hesiod

Figure 5.3: Related articles across clusters

The following snapshot tells how each annotation is displayed. The article under consideration is Alan Turing. The first line of the annotation gives a one line summary of the article. This is usually taken from the first line of the article itself because most of the times the first line gives a definition, if it is a concept, or a brief description, if it is a person or a place. The second line gives a name value pair. The value is the domain to which the article closely belongs to. In this case it is Mathematics. As is mentioned before that there are primary domains created namely, Philosophy, Mathematics, Science, Logic, and Biography and Alan Turing closely relates with Mathematics which makes very good sense considering the work he did in the field of mathematics which led to the testing various other works and became a sort of litmus test in the field of Computer Science later on. The third line of the annotation tells us the name of the articles that are very closely related to the article under consideration. The number of articles that are to be displayed can be limited and here the limit is three.

['Alan Mathison Turing was an English mathematician, computer scientist, logician, cryptanalyst, philosopher, and theoretical biologist', '< alan tuning , Mathematics>', ['<mathematician>', '<computer scientist>', '<cryptanalyst>']]

```
annotate("Alan Turing")
```

```
['Alan Mathison Turing   was an English mathematician, computer scientist, logician, cryptanalyst, philosopher, and theoretical
biologist',
 '< alan tuning , Mathematics>',
 ['<mathematician>', '<computer scientist>', '<cryptanalyst>']]
```

Figure 5.4: Annotation for Alan Turing

Chapter 6

CONCLUSION AND FUTURE SCOPE

Knowing the fact that Wikipedia provides all their archives to anyone who wishes to use it in any form or way they wish, we chose this problem to solve thinking that the data is readily available and is easy to access. Learning the fact that it has to scraped individually for every needed page restricted the model to some extent but also helped us in achieving the same objectives in a modular fashion. The most obvious and evident conclusion is that most articles do not look like they belong to a certain category on the surface but have more to do than expected. The plethora of anchor links present in each article of a Wikipedia page takes directions totally unheard of and that is proved by our model. A certain strength metric of a relationship can be modelled between any two articles and a path can be drawn out that reveals more about the particular topic.

The main objective of this project was to organize Wikipedia data in a much more structured format. Wikipedia is known for being a source of information on almost all topics in the world. But in our own experience we found out the grouping of similar topics and proper categorizing of the vast encyclopedia is not an attribute the website is known for. Even though we have been able to take significant steps in achieving these targets the categories chosen are numbered. And because of the scale of the project, the number of articles studied and parsed into the knowledge system is close to a hundred.

There is a lot of scope to this project and the implementation of those ideas will make it more useful for the target users as well. Efforts can be made to run the modules on much larger samples to have a knowledge system that is much more robust than the current model. With a larger sample size, there will be a need for a larger group of categories to divide the articles into. A GUI for the system will make the experience of utilizing the various modules much simpler and enjoyable. These changes would make the system much more functional.

Plagiarism Report

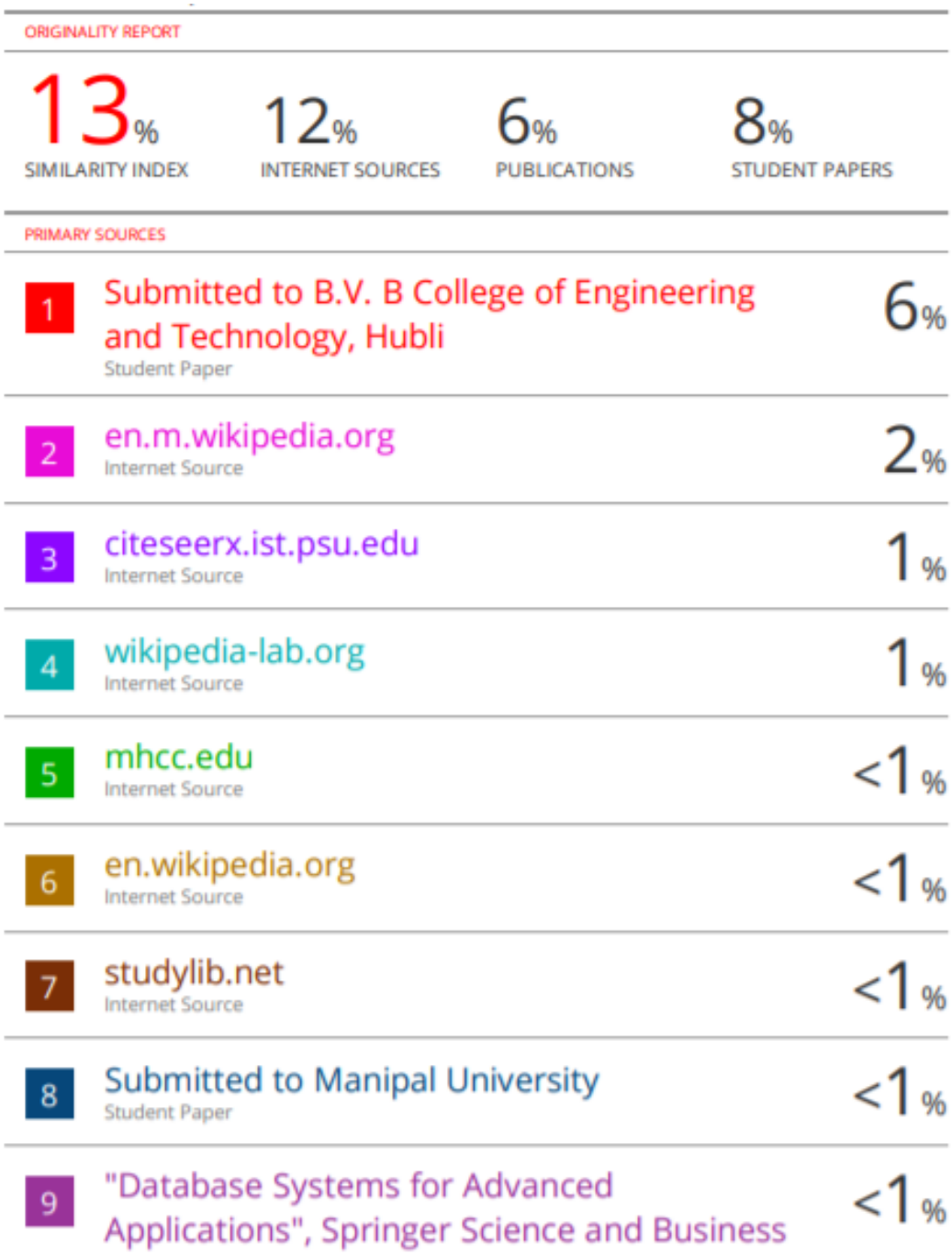
Project Title: Wiki-Tree – A Concept Relation Tree

Team Number: I1

Project Domain: Data Engineering (Data Analysis and Representation)

Category: Industry

Industry name: Knit Arena Software Research and Services Private Limited



ORIGINALITY REPORT

13%	12%	6%	8%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to B.V. B College of Engineering and Technology, Hubli Student Paper	6%
2	en.m.wikipedia.org Internet Source	2%
3	citeseerx.ist.psu.edu Internet Source	1%
4	wikipedia-lab.org Internet Source	1%
5	mhcc.edu Internet Source	<1%
6	en.wikipedia.org Internet Source	<1%
7	studylib.net Internet Source	<1%
8	Submitted to Manipal University Student Paper	<1%
9	"Database Systems for Advanced Applications", Springer Science and Business	<1%

	Media LLC, 2008 Publication	
10	Submitted to Bilkent University Student Paper	<1 %
11	www.scribd.com Internet Source	<1 %
12	thereaderwiki.com Internet Source	<1 %
13	d-nb.info Internet Source	<1 %
14	Submitted to Western Governors University Student Paper	<1 %
15	docs.microsoft.com Internet Source	<1 %
16	eprints.maynoothuniversity.ie Internet Source	<1 %
17	"The Semantic Web – ISWC 2017", Springer Science and Business Media LLC, 2017 Publication	<1 %
18	www.x-mol.com Internet Source	<1 %
19	aaltodoc.aalto.fi Internet Source	<1 %
20	Ben Aouicha, Mohamed, Mohamed Ali Hadj Taieb, and Malek Ezzeddine. "Derivation of "is a" taxonomy from Wikipedia Category Graph", Engineering Applications of Artificial Intelligence, 2016. Publication	<1 %

REFERENCES

[1] Vivi Nastase and Michael Strube. Decoding wikipedia categories for knowledge acquisition. In *AAAI*, volume 8, pages 1219–1224, 2008.

[2] Silviu Cucerzan. Large-scale named entity disambiguation based on wikipedia data. In *Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL)*, pages 708–716, 2007.

[3] Dat PT Nguyen, Yutaka Matsuo, and Mitsuru Ishizuka. Relation extraction from wikipedia using subtree mining. In *Proceedings of the National Conference on Artificial Intelligence*, volume 22, page 1414. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2007.

[4] Kotaro Nakayama, Takahiro Hara, and Shojiro Nishio. Wikipedia link structure and text mining for semantic relation extraction. In *SemSearch*, pages 59–73, 2008.

[5] Kotaro Nakayama, Takahiro Hara, and Shojiro Nishio. Wikipedia mining for an association web thesaurus construction. In *International Conference on Web Information Systems Engineering*, pages 322–334. Springer, 2007.

[6] Wang, Zhigang, et al. "Transfer learning based cross-lingual knowledge extraction for wikipedia." Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). 2013

[7] Wang, Zhi-chun, et al. "Knowledge extraction from Chinese wiki encyclopedias." Journal of Zhejiang University SCIENCE C 13.4 (2012)

[8] Wu, Fei, and Daniel S. Weld. "Open information extraction using wikipedia." Proceedings of the 48th annual meeting of the association for computational linguistics. 2010

[9] Yu, Lishan, and Sheng Yu. "Developing an automated mechanism to identify medical articles from wikipedia for knowledge extraction." International journal of medical informatics 141 (2020)

[10] Nakayama, Kotaro. "Extracting Structured Knowledge for Semantic Web by Mining Wikipedia." International Semantic Web Conference (Posters & Demos). 2008

[11] Ito, Masahiro, et al. "Association thesaurus construction methods based on link co-occurrence analysis for wikipedia." Proceedings of the 17th ACM conference on Information and knowledge management. 2008

[12] Hoffart, Johannes, et al. "Yago2: A spatially and temporally enhanced knowledge base from wikipedia." Commun. ACM 52.4 (2009)

[13] Muchnik, Lev, et al. "Self-emergence of knowledge trees: Extraction of the Wikipedia hierarchies." Physical review E 76.1 (2007)

[14] Nakayama, Kotaro, "Wikipedia mining", Wikimania. Wikimedia , 2008