

Automatic Modeling and Recommendation of Topics in Online Pedagogy using Metro Maps

*A Report Submitted
in Fulfillment of the Requirements for the Course*

B.Tech. Project (CS399)

by

Gourav Patidar
(B18CSE015)

and

Utpal Gupta
(B18CSE058)

under the guidance of

Dr. Chiranjoy Chattopadhyay
[Assistant Professor]



॥ त्वं ज्ञानमयो विज्ञानमयोऽसि ॥

to the

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY JODHPUR**

CERTIFICATE

*This is to certify that the work contained in this thesis entitled “**Automatic Modeling and Recommendation of Topics in Online Pedagogy using Metro Maps**” is a bonafide work of **Gourav Patidar (Roll No. B18CSE015)** and **Utpal Gupta (Roll No. B18CSE058)**, carried out in the Department of Computer Science and Engineering, Indian Institute of Technology Jodhpur under my supervision and that it has not been submitted elsewhere for a degree.*

Supervisor: **Dr. Chiranjoy Chattopadhyay**

Assistant/Associate Professor,

April, 2021

Jodhpur.

Department of Computer Science & Engineering,
Indian Institute of Technology Jodhpur, Rajasthan.

Contents

1	Introduction	1
2	Problem Statement	1
2.1	Background Survey	1
2.2	Solution Proposed	2
3	Methods Used	3
3.1	Parsing	3
3.2	Word Embedding: Using Word2Vec	4
3.3	Clustering	6
3.4	Heuristics Approach	6
4	Process Flow	7
4.1	Database creation of XML files of books	7
4.2	Selecting book-set through user's query	8
4.3	Chapter Title extraction from XML parsing	8
4.4	Word embedding of obtained chapter titles	9
4.5	Clustering of Chapters	10
4.6	Metro-Map Construction	10
5	Results	11
5.1	Selecting book through user's query	11

5.2	Chapter Title extraction from XML parsing	11
5.3	Word embedding of obtained chapter titles	12
5.4	Clustering of Chapters	14
5.5	Metro-Map Construction	15
5.6	Conclusion	15
6	Future Work	16
6.1	Data-set and Efficiency	16
6.2	Algorithm and Metrics	16
6.3	API	17
	References	18

Chapter 1

Introduction

Over the years we have seen world going through digitization. Technology has made our life easy but some things are still indispensable. Textbooks were the main sources for providing information to students and teachers. In today's time to learn anything, people need to refer to several web sources and E-books to gain knowledge of their fields i.e. knowledge learning. To learn one particular topic a person should have adequate prerequisite knowledge of that topic. A reader should be aware of the path he should take in order to get a better understanding of the topic and cover the prerequisite knowledge in the most optimal way possible. Hence, it will be very helpful if we can represent these books in a concise yet comprehensive image.

So to overcome this issue we have proposed a method called the metro map construction that would suggest a path from different books to gain efficient knowledge learning by summarizing massive electronic textbooks to free ourselves from this difficulty. A metro map would help us in learning efficient knowledge which is highly informative i.e. the constructed map would contain knowledge-concentrated learning paths and would also have a good fluency i.e. the constructed metro map would be fluent, so that people can gain the knowledge easily and also the constructed map would have high coverage i.e. the constructed map would enclose large knowledge with less redundancy with optimal path.



Fig. 1.1 Example of a Metro Map from massive electronic textbooks for learning Mathematics.

Chapter 2

Problem Statement

2.1 Background Survey

Several solutions have been proposed earlier for knowledge learning. Some of them are as follows:

- Linking encyclopedic information to educational material
- Augmenting a section of the textbook by the images which are most relevant
- Identifying sections of a book that are not well-written, and augmenting the sections of some other books along-with some articles and images from web for better understanding

As we can see that in above proposed solutions nobody has proposed a solution to represent an E-book in form of a comprehensive image that will help in Knowledge learning.

2.2 Solution Proposed

Solution that we have proposed is different from the solution that were proposed later as we implemented some methods to facilitate learning through summarising a textbook in an image. Our procedure was based on Natural language processing algorithms and chapters of books were used as the initial data. We used NLP algorithms to find the similarity and relevance of the chapter titles of books of particular subject.

Features that we decided to incorporate in our project were:

- Create an comprehensive image for books in all field through a freshly designed procedure
- Implement our method for some digital library such as National Digital Library of India (NDLI)
- To make an API which can take field name in which user is interested along-with the chapters from which user want to learn and the topic which user wants to learn. API will return a path user should take to learn a particular topic; also the amount of time that user have to learn a that topic
- To implement some metrics such as:
 1. Informativeness: It will check whether a path is covering all the chapters that are required to learn a topic or not.
 2. Fluency: It will tell us how easy it is to follow a particular learning path.
 3. Coverage: This parameter will be used to cover maximum node in learning graph. For example, if a user doesn't have enough time to complete the whole book, then informativeness will be more precedent

Chapter 3

Methods Used

To achieve our results, we have implemented the following methods in our project described below.

3.1 Parsing

1. First Chapter	1
1.1. First Heading	1
1.2. Second Heading	1
1.3. Third Heading	1
2. Second Chapter	2
2.1. Second Chapter First Heading	2
2.2. Second Chapter Second Heading	2
2.3. Second Chapter Third Heading	2

Fig. 3.1 Example of table of content of an XML format E-Book

Digital libraries containing the electronic book usually follows the XML format standard where the metadata such as title, authors and publishers of a book are encoded and the content of the chapters of the book is organized in a hierarchical structure. In the above figure we can see the table of content of any E-Book. To understanding the concept behind parsing the chapter titles from the e-books we have to see the XML code that will lead to

such table of content.

```
<?xml version="1.0" encoding="utf-8"?>

<content>

<chapter name="First Chapter">

    <heading>First Heading</heading>

    <heading>Second Heading</heading>

    <heading>Third Heading</heading>

</chapter>

<chapter name="Second Chapter">

    <heading>Second Chapter First Heading </heading>

    <heading>Second Chapter Second Heading</heading>

    <heading>Second Chapter Third Heading</heading>

</chapter>

</content>
```

In the above code we can see that tag ***chapter*** has an attribute ***name***. So we have used Element tree module in python to parse name of all the chapters in an e-book data set of a particular subject. After parsing we stored the chapter in a CSV files so that the chapter titles can be used later.

3.2 Word Embedding: Using Word2Vec

From the XML files of the books we extract the titles of all chapters i.e. the books belonging the relevant query Q. Now our target was to represent each chapter in form of vectors in 2-D. For better accuracy, we decided to use word embedding technique Word2Vec instead of Paragraph2Vec. Word2Vec has two kind of training algorithms, ***ContinuousBag – Of – Words(CBOW)*** which treat all words as if all words are there in a bag and treating these words as context words we have to find a focus word. On

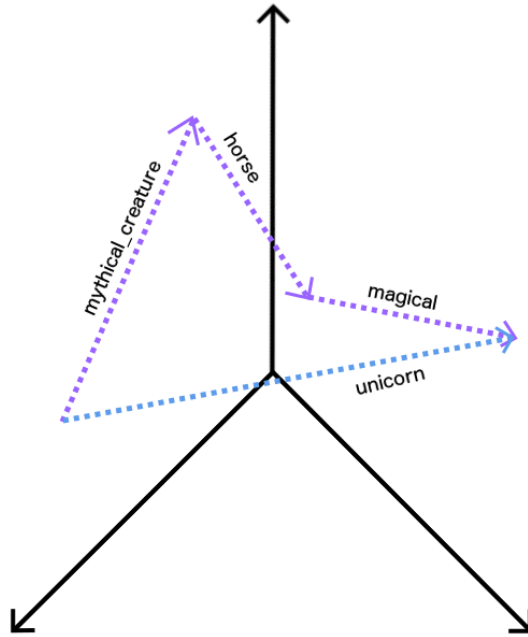


Fig. 3.2 Example of Word embedding using Word2Vec algorithm in NLP

the other hand in *Skip – Gram* training algorithm our target is to find context words out of one focus words.

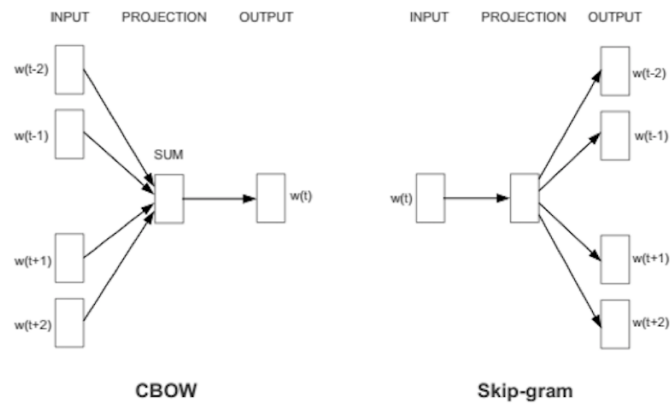


Fig. 3.3 CBOW and Skip-Gram

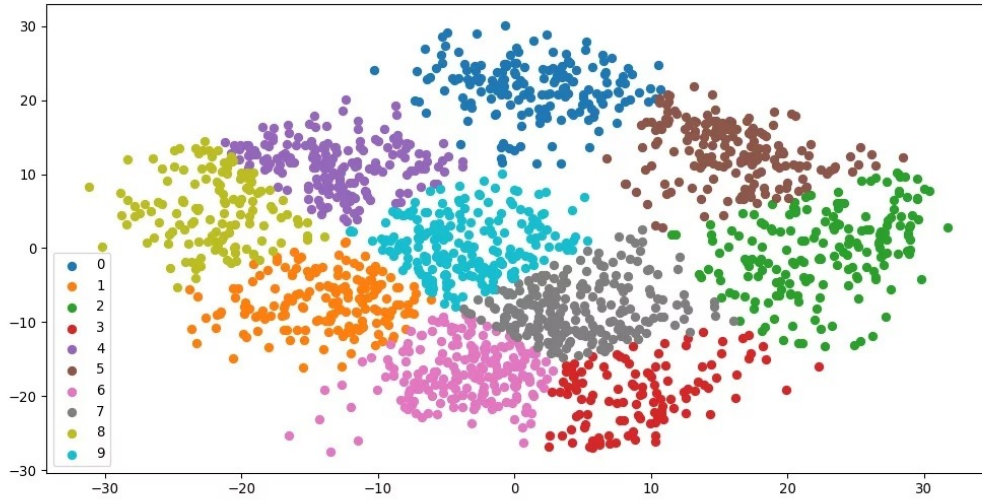


Fig. 3.4 Example of vector clustering through K-means clustering algorithm

3.3 Clustering

To cluster the final vectors of chapter titles we used K-means clustering algorithm. Through this algorithm N vectors can be clustered in K different groups. A cluster here refers to a set of vectors aggregated together to form a set of space.

3.4 Heuristics Approach

To find the most optimal learning path we decided to use a utility function or heuristics. This function will return a value that will tell us the similarity between two clusters of chapter. An equivalent vector for a cluster will be the average of all vectors that lie within a cluster or in we can say the **centroid** of the cluster. We have use cosine similarity between two clusters as our heuristics. More the cosine similarity better the learning path is.

Chapter 4

Process Flow

4.1 Database creation of XML files of books

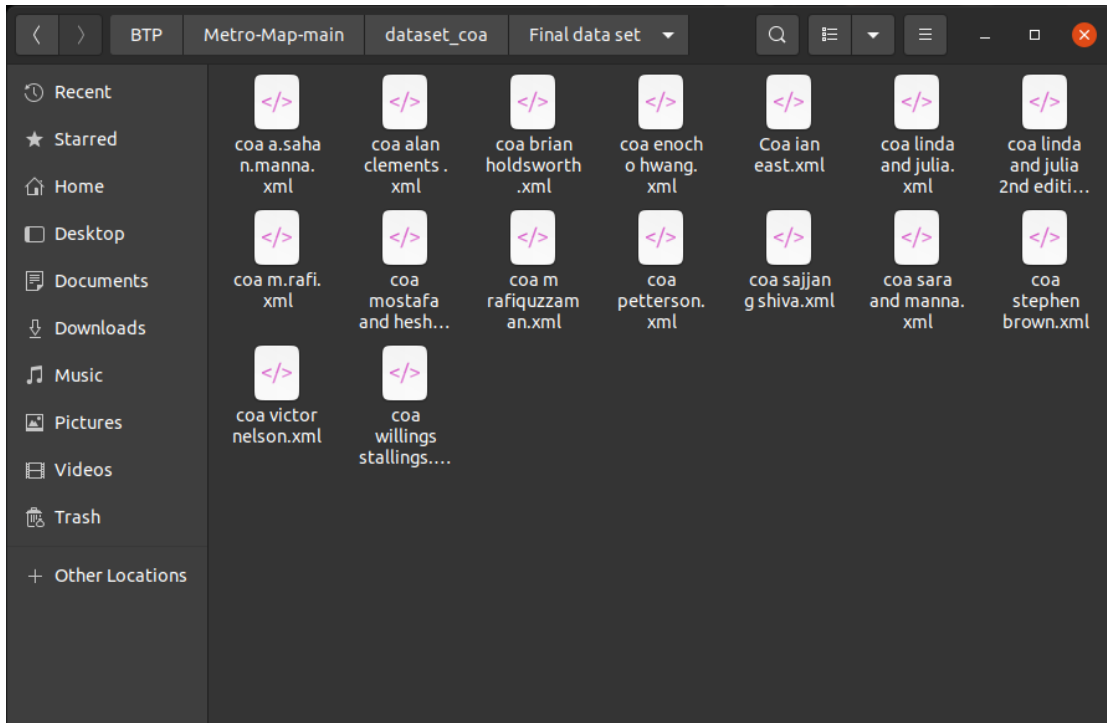
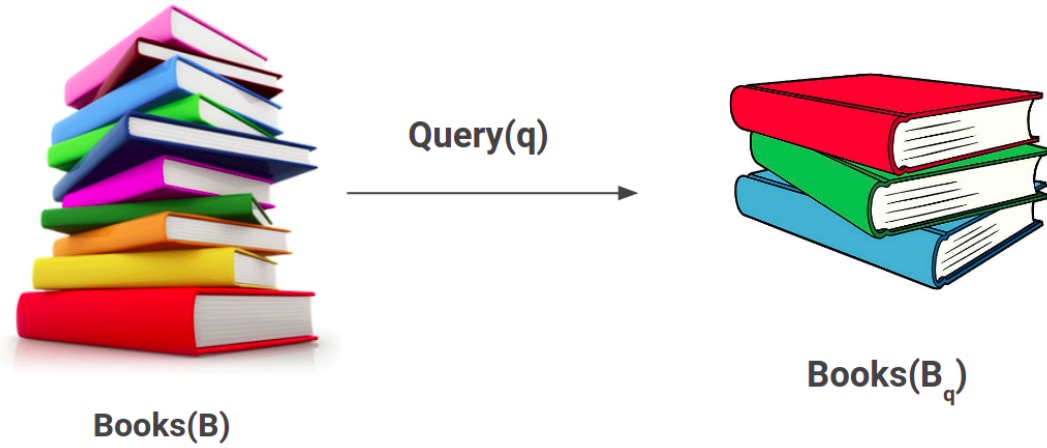


Fig. 4.1 Picture of our data-set for Books on Computer Organisation and Architecture

Since there was no public data-set, we created a mini-data-set of XML files. Chapters from different books of the same subject from different authors we encoded in XML files

with suitable attributes. Those attributes were further used in parsing.

4.2 Selecting book-set through user's query



User enters a query Q through which we select most relevant directory in which books of the subject related to query are stored in XML format. A Python extension for accessing Java Lucene i.e. PyLucene was used to search for those books. Indexing of these books were done and the query search is done over this indexed books to get the required data set B_q .

4.3 Chapter Title extraction from XML parsing

In section 3.1 we mentioned about parsing. A parse tree was constructed using element tree module in python and all the chapter of relevant textbooks were extracted and stored in a CSV file for further usage.

Fig. 4.2 Extracted Chapter titles through XML parsing

4.4 Word embedding of obtained chapter titles

Through parsing chapter titles were obtained. Now task was to first removing the aspect word from the chapter titles such as a, an, the, is, and etc. That was done using the stop words list in NLTK library. After removing these stopwords task was to convert the titles into vectors. But some words in the titles were more significant than the others. For example, if there is a chapter in mathematics book **Introduction to Game Theory**, Game theory has more significance than Introduction(to will be removed as it is a stop word). So we define words in two type: Topic words and Aspect words. Words 'Game' and 'Theory' are topic words, and word 'Introduction' is an aspect word. Reason behind this division is to remove the ambiguity between chapter titles such as **Introduction to Game Theory** and **Introduction to Probability Theory**. Now to find the resultant vector of a chapter title we will be the average vector of the vectors of aspect words and topic words.

4.5 Clustering of Chapters

As mentioned in section 3.2, We used Word2Vec model to represent our words(both aspect and topic) into vectors. Word2Vec returns a vector of 100 dimensions for each word. To find the resultant vector of any chapter title, we took average of the vector of topic and aspect words in that chapter title. After getting the resultant vector for each chapter title we clustered them into K clusters using K-means clustering algorithm in scikit-learn library in python. Centroid of each cluster was also calculated so that we can represent each cluster by them. To plot the vectors of each chapters or word, we have used an inbuilt function in python called Principal Component Analysis (PCA) which is a technique that can be utilized for extracting information from a high-dimensional space by projecting it into a lower-dimensional sub-space.

4.6 Metro-Map Construction

This is the most tricky and most important part of this project. First of all the cluster which contain the starting chapter(say S) which user entered was selected. Then cosine similarity was calculated for all the close clusters and then utility function was used to decide which will be the next cluster in learning path. We can consider this problem as search problem in artificial intelligence. where search space is all the nodes(or centroids of the clusters) in the graph. Start state is the starting cluster S and Goal test will give true when destination state(say D) will be reached or when total number of clusters will be equal to K. Another way to find out the most optimal path is to first find all the possible path from S to D. For that we can use Depth-First Search algorithm using dynamic programming. On those paths we can implement our previously mentioned metrics and find out the most optimal path according to all given parameters such as time in hand etc.

Chapter 5

Results

5.1 Selecting book through user's query

```
C:\Users\Gourav\Desktop>python lucene.py
<Hit {'content': 'limits ', 'path': '/b', 'title': 'Second try'}>
<Hit {'content': 'limits ', 'path': '/b', 'title': 'Second try'}> 1.8860373656870277
{('content', b'limits')}
<Hit {'content': 'conclusion and limits', 'path': '/a', 'title': 'My document'}> 1.405631621596936
{('content', b'limits')}
matched terms
[('content', b'limits')]
[('content', b'limits')]
more_results
<Top 1 Results for Or([Term('content', 'limits', boost=0.4992525323447402)]) runtime=0.000260800000000061>
Scored 2 of exactly 2 documents
```

Fig. 5.1 Result of query processing

The above figure shows the result of query processing over the data set for finding the books referring to limits section.

5.2 Chapter Title extraction from XML parsing

Above figure show the words that we get after parsing the XML file, extracting the titles and removing the stop words.

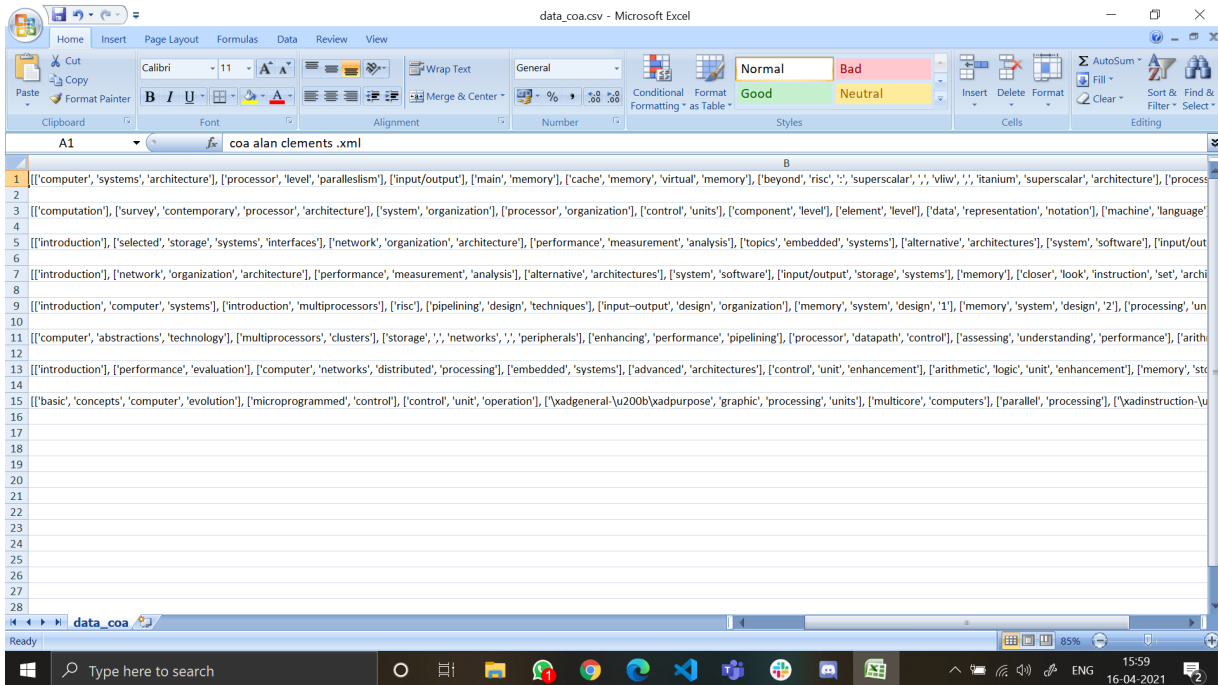


Fig. 5.2 Extracted chapters after removing stop words

5.3 Word embedding of obtained chapter titles

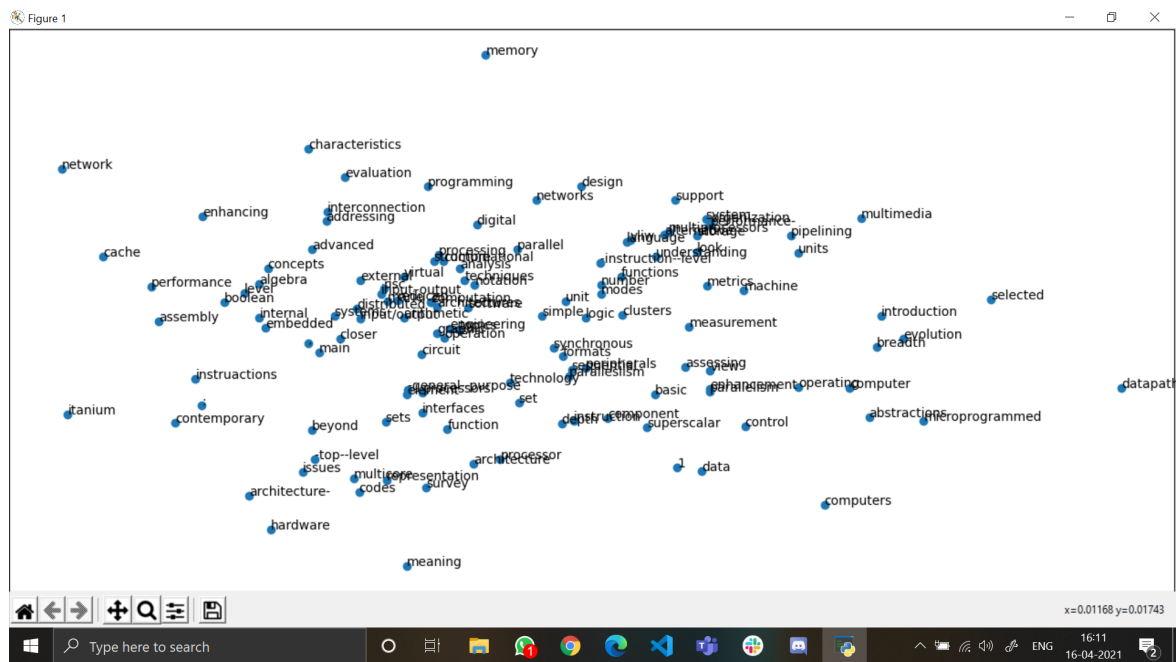


Fig. 5.3 Vector of each word individually

5.4 Clustering of Chapters

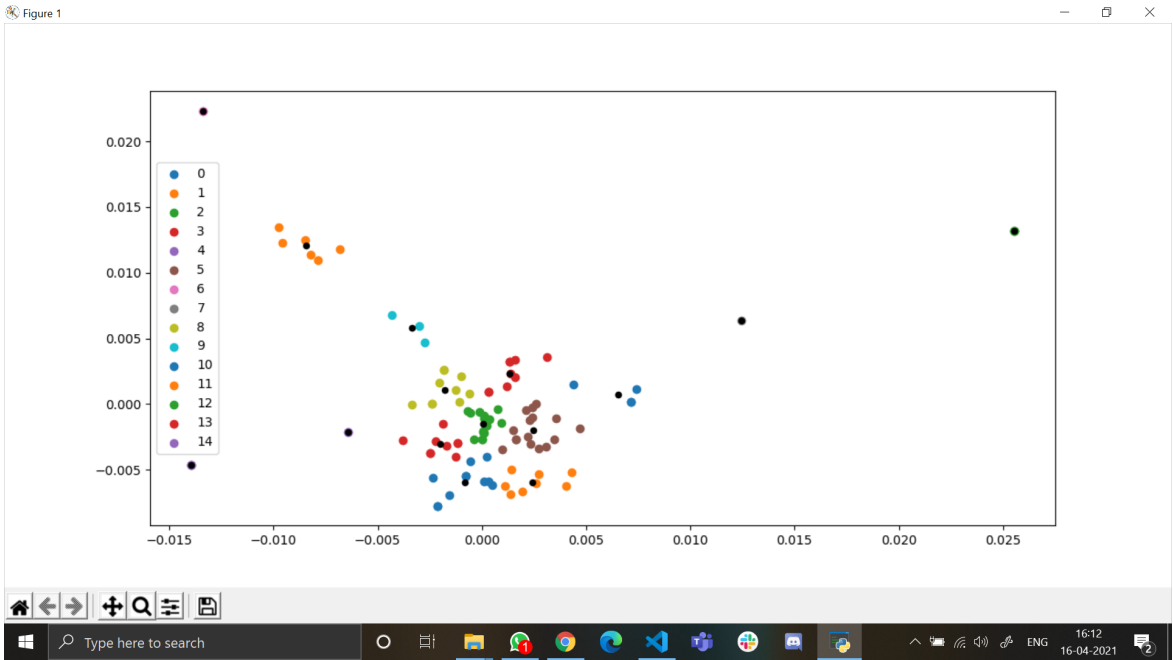


Fig. 5.5 Plot after applying K-means algorithm on vectors of chapter titles

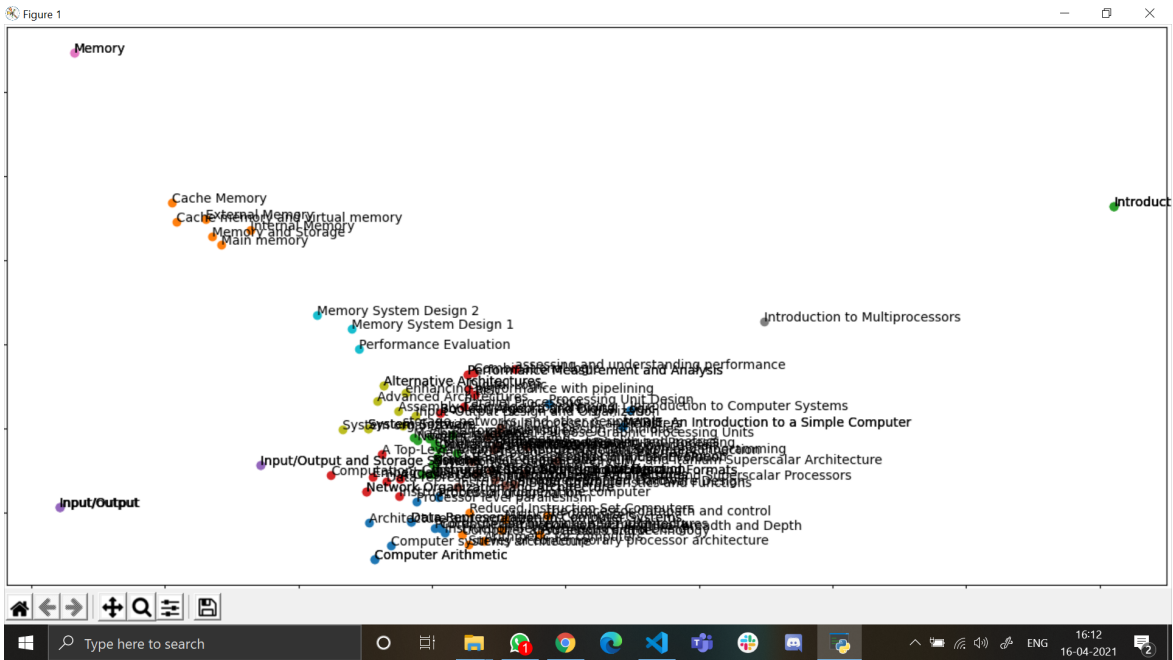
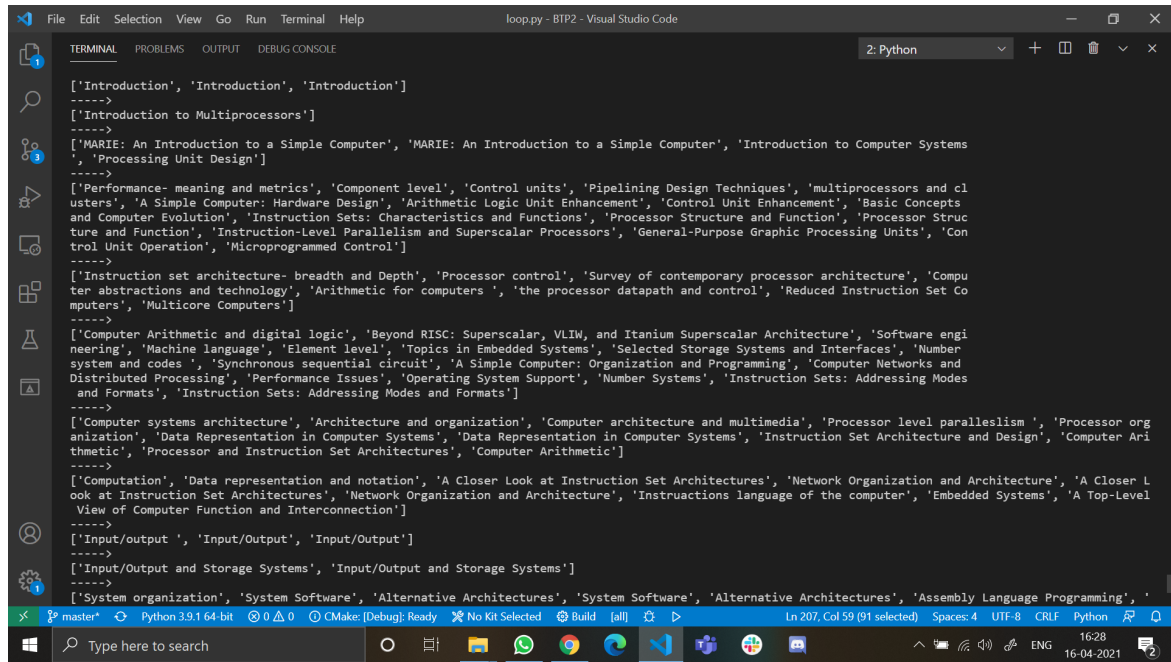


Fig. 5.6 Plot with labels

5.5 Metro-Map Construction



```
File Edit Selection View Go Run Terminal Help loop.py - BTP2 - Visual Studio Code
TERMINAL PROBLEMS OUTPUT DEBUG CONSOLE 2: Python
['Introduction', 'Introduction', 'Introduction']
----->
['Introduction to Multiprocessors']
----->
['MARIE: An Introduction to a Simple Computer', 'MARIE: An Introduction to a Simple Computer', 'Introduction to Computer Systems', 'Processing Unit Design']
----->
['Performance- meaning and metrics', 'Component level', 'Control units', 'Pipelining Design Techniques', 'multiprocessors and clusters', 'A Simple Computer: Hardware Design', 'Arithmetic Logic Unit Enhancement', 'Control Unit Enhancement', 'Basic Concepts and Computer Evolution', 'Instruction Sets: Characteristics and Functions', 'Processor Structure and Function', 'Processor Structure and Function', 'Instruction-Level Parallelism and Superscalar Processors', 'General-Purpose Graphic Processing Units', 'Control Unit Operation', 'Microprogrammed Control']
----->
['Instruction set architecture- breadth and Depth', 'Processor control', 'Survey of contemporary processor architecture', 'Computer abstractions and technology', 'Arithmetic for computers', 'the processor datapath and control', 'Reduced Instruction Set Computers', 'Multicore Computers']
----->
['Computer Arithmetic and digital logic', 'Beyond RISC: Superscalar, VLIW, and Itanium Superscalar Architecture', 'Software engineering', 'Machine language', 'Element level', 'Topics in Embedded Systems', 'Selected Storage Systems and Interfaces', 'Number system and codes', 'Synchronous sequential circuit', 'A Simple Computer: Organization and Programming', 'Computer Networks and Distributed Processing', 'Performance Issues', 'Operating System Support', 'Number Systems', 'Instruction Sets: Addressing Modes and Formats', 'Instruction Sets: Addressing Modes and Formats']
----->
['Computer systems architecture', 'Architecture and organization', 'Computer architecture and multimedia', 'Processor level parallelism', 'Processor organization', 'Data Representation in Computer Systems', 'Data Representation in Computer Systems', 'Instruction Set Architecture and Design', 'Computer Arithmetic', 'Processor and Instruction Set Architectures', 'Computer Arithmetic']
----->
['Computation', 'Data representation and notation', 'A Closer Look at Instruction Set Architectures', 'Network Organization and Architecture', 'A Closer Look at Instruction Set Architectures', 'Network Organization and Architecture', 'Instructions language of the computer', 'Embedded Systems', 'A Top-Level View of Computer Function and Interconnection']
----->
['Input/output', 'Input/Output', 'Input/Output']
----->
['Input/Output and Storage Systems', 'Input/Output and Storage Systems']
----->
['System organization', 'System Software', 'Alternative Architectures', 'System Software', 'Alternative Architectures', 'Assembly Language Programming', 'System Software', 'Alternative Architectures', 'Assembly Language Programming', 'System Software', 'Alternative Architectures', 'Assembly Language Programming']
```

Fig. 5.7 Final path with clusters represented in brackets and arrow represents the move

5.6 Conclusion

We tried to represent books in a concise but comprehensive path using a self-designed algorithm. We were not able to implement the DFS method to find the most optimal path because number of permutations were very high. Some cluster had more chapters than the other as during reduction of dimensions by PCA some important information was being lost. We were able to find one optimal path but we couldn't implement the case where starting point and the destination chapter will be given as input.

Chapter 6

Future Work

The work done by us till now faced quite some challenges which can be improved in the future versions of the work to make it more accurate and efficient.

6.1 Data-set and Efficiency

We would work on the data set of the project, currently a small data set is been used to work on, in future we would work on larger data set and accordingly we need to work on efficiency of clustering and other parts of project as we were losing some important information because of dimension reduction. Also generalizing a generating function for vectorizing and clustering together.

6.2 Algorithm and Metrics

Work on different algorithms for clustering and map construction and increase the efficiency which would result in high valued in results.

6.3 API

Work on creating a API which help in interactive visualization of metro maps depending upon users requirements and return the appropriate solutions and maps. That API will take starting chapter from where a user believes he want to start learning and the destination chapter till where he wants to learn. To find the optimal path for the user we will have to implement dynamic programming algorithms to find all the possible path and then we will have to check for the metrics that we decided beforehand.

References

- [1] Parsing. <https://www.xmlpdf.com/tableofcontents.html>.
- [2] Clustering. <https://www.askpython.com/python/examples/plot-k-means-clusters-python>.
- [DS] Eric Horvitz Dafna Shahaf, Carlos Guestrin. Metro maps of science. <https://www.cs.cmu.edu/~dshahaf/kdd2012-shahaf-guestrin-horvitz.pdf>.
- [IFJ] Iztok Fister Iztok Fister Jr. *INFORMATION CARTOGRAPHY IN ASSOCIATION RULE MINING*. <https://arxiv.org/pdf/2003.00348v1.pdf>.
- [WL] Jiale Yu Yangfan Zhou Baogang Wei Weiming Lu, Pengkun Ma. Metro maps for efficient knowledge learning by summarizing massive electronic textbooks. <https://doi.org/10.1007/s10032-019-00319-y>.