Application of Ontology-Driven NLP for the Analysis of Text

Submitted in partial fulfilment of the requirements
of the degree of

BACHELOR OF ENGINEERING

In

COMPUTER ENGINEERING

By

Group No: 35

Roll No. Name

1604004 Eashan Bajaj

1604092 Aakash Ramchandani

1604122 Omkar Yadav

Guide:

PROF. UJWALA H. BHARAMBE
(Assistant Professor, Department of Computer Engineering, TSEC)

Computer Engineering Department
Thadomal Shahani Engineering College
University of Mumbai

2019-2020

CERTIFICATE

This is to certify that the project entitled **"Application of Ontology-Driven NLP for the Analysis of Text"** is a bonafide work of

Roll No.	Name
1604004	Eashan Bajaj
1604092	Aakash Ramchandani
1604122	Omkar Yadav

Submitted to the University of Mumbai in partial fulfilment of the requirement for the award of the degree of **"BACHELOR OF ENGINEERING"** in **"COMPUTER ENGINEERING"**.

Prof. Ujwala H. Bharambe
Guide

Dr. Tanuja Sarode
Head of Department

Dr. G.T. Thampi
Principal

Project Report Approval for B.E

Project report entitled (***Application of Ontology-Driven NLP for the Analysis of Text***) by

Roll No.	Name
1604004	Eashan Bajaj
1604092	Aakash Ramchandani
1604122	Omkar Yadav

is approved for the degree of ***"BACHELOR OF ENGINEERING"*** in ***"COMPUTER ENGINEERING"***.

Examiners

1.----------

2.----------

Date:

Place:

Declaration

We declare that this written submission represents my ideas in my own words and where others 'ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will because for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

1) _____

1604004, Eashan Bajaj

2) _____

1604092, Aakash Ramchandani

3) _____

1604122, Omkar Yadav

Date:

Application of Ontology-Driven NLP for the Analysis of Text

Abstract

Examination being the lone measure of competence in the current education system of our country, has a decisive role in building of student's career. Hence, utmost care has to be taken in framing the question paper. However, setting up a good question paper for assessment is a challenging task. Fairness, accuracy, consistency and elimination of bias are very important while selecting questions into the paper. We address the issue of providing some automated framework to judge the fairness of a question paper with respect to a given syllabus. We explored the possibility of using the semantic web technology and ontology in particular in addressing the issue of syllabus fairness. We have introduced the notion of syllabus ontology that forms the semantically connected network of concepts from the domain. We identified various syllabus fairness issues and proposed an initial model for measuring the syllabus fairness of a question paper.

TABLE OF CONTENTS

List of Figures

Chapter 1

Introduction

1.1 Introduction

The most important activity in an educational environment is the examination being conducted. Setting up a good question paper for assessment on the other hand is a challenging task. There are no standardized methods and thus depends completely on the expertise and expectations of the individual examiner responsible for doing so. The question paper, minimally, consists of a set of questions relevant to the syllabus of a particular course. But there is more which goes into the making of a question paper. The questions need to span the syllabus in a fair manner. They must be of varying difficulty levels, etc. The generation of question paper depends on the syntactic and semantic issues which need to be addressed. The syntactic issues deal with the structure of the question paper which may be defined by a pre-set template. Every university defines its own template for the question paper. The template includes the header information such as course name, for which class, total marks, duration of the paper, etc. It also contains information about number of questions, number of sub questions within a question, marks distribution among the questions and rules of choosing questions to be attempted by students. The examiner has to strictly follow the template structure while setting up a question paper. The semantic issues are for assurance of the quality of question paper generated. Fairness, accuracy, consistency and the elimination of bias are very important while selecting questions into the paper. The quality of a question paper is a function of all these aspects. One of these criteria which we are addressing in this paper is to find how fair the paper is to given syllabus. Intuitively, a good paper should have questions covering the entire syllabus. In this paper, we address the issue of providing some automated framework to judge the fairness of a paper regarding the syllabus coverage. The major topics in the syllabus are given with a weightage that correspond to the number of lecture hours to be used to teach that topic. Intuitively this is also an indication of what fraction of the exam questions will be from this part. There are dependencies and relationships among the different topics in the syllabus which

may not be explicit. Keeping this in mind we've explored the possibility of using an ontological structure for representing syllabus which we call as syllabus ontology.

1.2 Aim & Objectives

Objective of study is to propose a feasible method, which contains several aspects to accurately tackle the problem of judging syllabus fairness. With this the university will be able to select a more accurate question paper for the examination. Concepts associated with each question from the question paper are mapped to the syllabus ontology that we've generated. Depending on how many nodes in the ontology are getting covered and how many times each node is referred by the questions, syllabus coverage in terms of percentage can be calculated. Every question in the question paper is processed to extract the concepts/ keywords so that it can be mapped to the nodes in the syllabus ontology. Once all the concepts / keywords are identified they have to be mapped to the syllabus ontology. Most of the keywords in the question will get mapped to the lower level concepts in the ontology. After doing so we can therefore judge the paper fairness based on the syllabus and therefore conclude how fair it is.

1.3 Scope

Examination being the lone measure of competence in the current education system of our country, has a decisive role in building of student's career. Hence, utmost care has to be taken in framing the question paper. However, setting up a good question paper for assessment is a challenging task. Fairness, accuracy, consistency and elimination of bias are very important while selecting questions into the paper. A given question paper is not fair to the syllabus if there are questions which are from topics not prescribed in the syllabus and the marks assigned to questions from a particular topic is not according to the weightage given in the syllabus for that topic. Every question from the paper should have a direct mapping to the topics given in the syllabus. The question paper should also cover the syllabus in a reasonably uniform way. We can therefore focus on various syllabus fairness issues and propose a method for measuring the syllabus fairness of the question paper.

Chapter 2

Review of Literature

2.1 Domain Explanation

Ontology is a data model that represents knowledge as a set of concepts within a domain and the relationships between these concepts.

The basic idea is using ontology to represent domain knowledge, extracting concepts from natural language sentence and mapping to ontology to get a correct answer.

Ontologies provide a semantic context. Identifying entities in unstructured text is a picture only half complete. Ontology models complete the picture by showing how these entities relate to other entities, whether in the document or in the wider world.

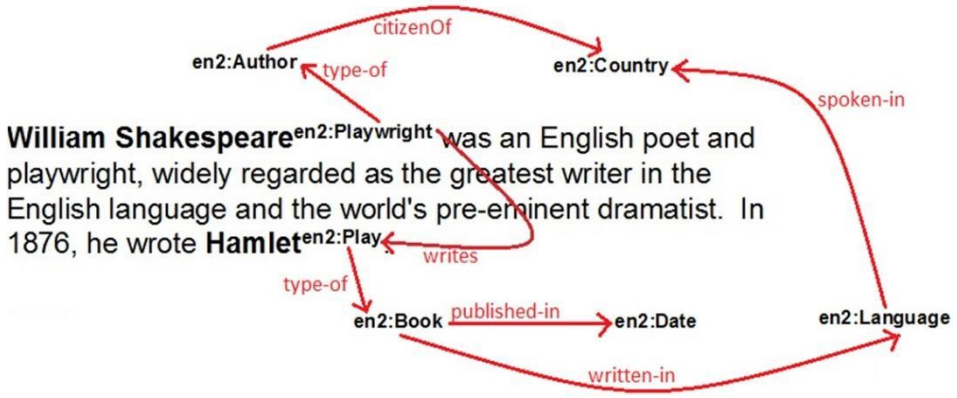

Fig 2.1 Ontology

We've only recognized (e.g. annotated) two words in this entire sentence: William Shakespeare as a Playwright and Hamlet as a Play. There are a total of 6 annotations represented on the diagram with arrows flowing between them. These annotations are produced by the NLP parser, and modelled (here's the key point), they are modelled in the Ontology. It's in the Ontology that we specify how a Book is related to a Date, or to a Language, and Language to a Country to

an Author, to a work produced by that Author, and so on. Each annotation is backed by a dictionary. The data for that dictionary is generated out of the triple store that conforms to the Ontology. The Ontology shows the relationship of all the annotations to each other.

The annotation of "William Shakespeare" as an Author is an implicit triple:
William Shakespeare an Author

We are now beginning to transition from unstructured data into the realm of structured data; if we know that William Shakespeare is an Author, we also know that Authors live in Countries; that Authors write books that are published on certain dates and written in certain languages, etc. There's an entire semantic chain of information that can be derived from this sentence. Further, the Ontology helps us to understand what data we're missing. It appears that all books are published on a date. Further, it appears that a language is involved too.

To summarize, the Ontology gives us the relations that exist between annotations. It helps us to understand each annotated token in a larger context (the context of a semantic chain and semantic network). It also helps us to understand what information we are missing, and what else we need to look for.

An Ontology model can help clarify a large corpus of unstructured data. The Ontology is your link into the real world. Without an Ontology, the annotations used by the NLP parser can become somewhat random. Who decides what an annotation should be named? Are they making this decision in coordination with what already exists? What modelling discipline exists? In past projects without an Ontology model, the NLP annotations over time had no link to the real world. Someone joining the project wouldn't know what a "Remaining Useful Word" or a "Power Action Word" was - there's no just way. If these had been designed in the discipline of an Ontology model, this discipline would have enforced a better standard in terms of naming, and likewise provided a link to the real world.

Consider the diagram above. We may never annotate the source text for Language, Date or Country. These concepts still provide value, because they give us the context and domain understanding of the concepts that we use as annotations in our NLP parser (like Book, Play and Author). This is an important point: Not every Ontology class needs to be associated with an annotation/dictionary in your NLP parser. In an extreme example, you might have an Ontology model with 15 classes and only one of them is used in the NLP parser.

Also note: There is no constraint toward a single Ontology model. Multiple ontology models can be used. It is likewise not a necessity that Ontology models must be related, either integrated peer-to-peer or via an "Upper Ontology". The need may exist, but it depends on the circumstances. Maintaining multiple models, each as a context around a particular annotation, or annotation set, is a valid solution. It may even make collaborative team efforts simpler.

2.2 Existing Solution

The existing solution is for examiners to manually check if the question paper is modelled according to the syllabus or not. There are no standardized methods and thus depends completely on the expertise and expectations of the individual examiner responsible for doing so.

2.3 Hardware and Software Requirements

2.3.1 Hardware Requirements

- Fluently working laptops
- RAM minimum 4Gb
- The system will use the standard hardware and data communication resources.

2.3.2 Software Requirements

- Python 3.6
- Pycharm IDE
- Libraries (nltk, pandas, copy, difflib, fuzzywuzzy, networkx, re, matplotlib, tqdm)
- Linux/Windows

Chapter 3

Analysis

3.1 Functional Requirements

The system shall abide by the following functional requirements:

1. Question Paper will be accepted in the form of text input to the system. The system will already hold the syllabus ontology covered for the given subject.

2. The system will accurately predict the percentage of syllabus covered by the questions accepted as input.

3. The system will display correctly the prediction results on the UI.

3.2 Non-Functional Requirements

1. Accessibility: The software system can be used by examiners as well as students. It would be made accessible to everyone in the form of a live website.

2. Availability: The website will be available at all times, unless the system is non-operational in which case notification will be provided on the website.

3. Flexibility: Provisions shall be made to add support for additional features like OCR to read input in image format to support images of question papers.

4. Scalability: We intend to make provisions such that the system could be scaled up to include all major Mumbai University engineering branches.

Chapter 4

Design

4.1 System Architecture

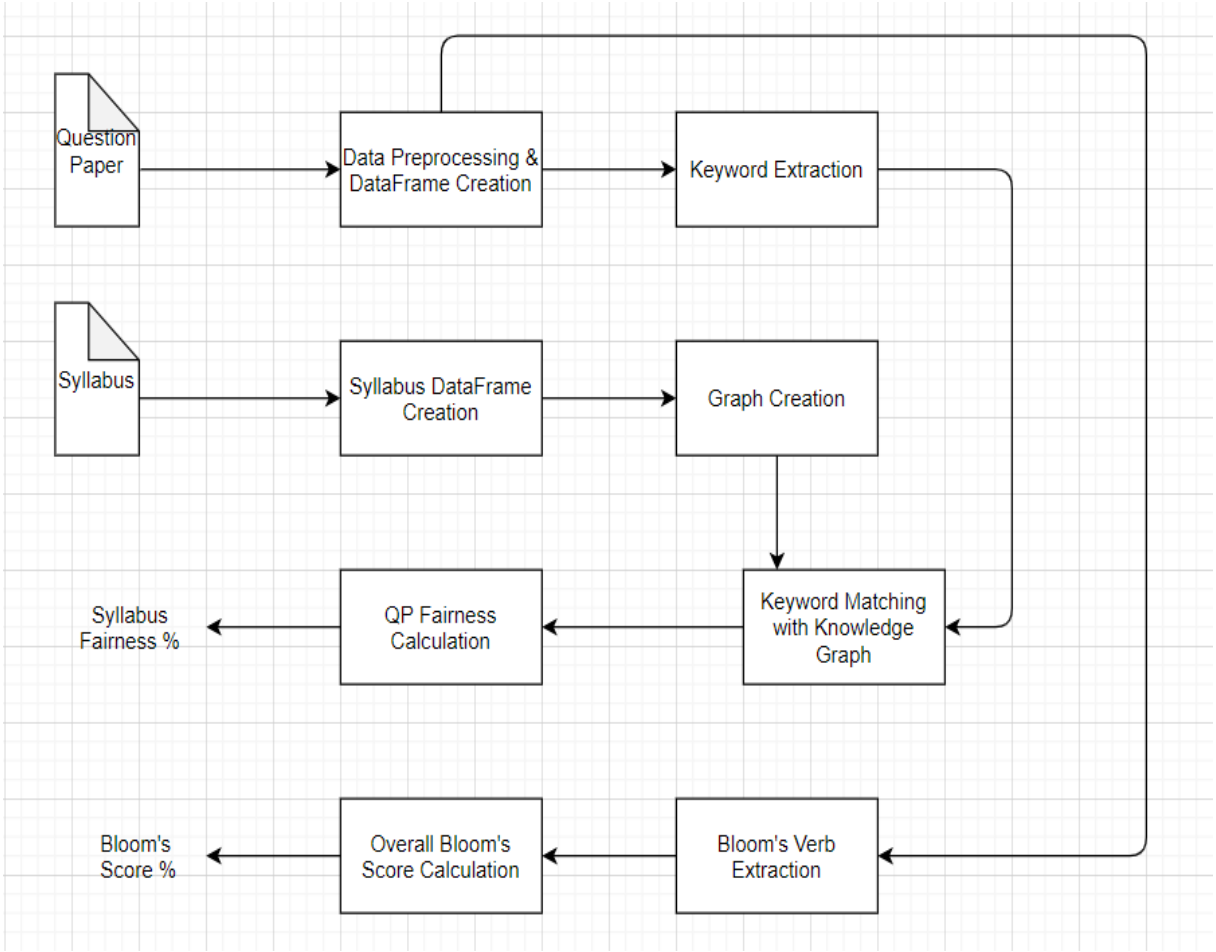

Fig 4.1 System Architecture

Architecture diagram gives the graphical representation of the system. This system will involve components like:

4.1.1 QP Data Preprocessing and DataFrame Creation

The probability of anomalous data has increased in today's data due to its humongous size and its origin for heterogenous sources. Considering the fact that high quality data leads to better models and predictions, data preprocessing has become vital–and the fundamental step in the data science/machine learning/AI pipeline.

While gathering data, one might come across three main factors that would contribute to the quality of data:

a. Accuracy - Erroneous values that deviate from the expected.

b. Completeness: Lacking attribute/feature values or values of interest.

c. Consistency: Aggregation of data is inconsistent.

To ensure high quality data, it's crucial to pre-process it. To make the process easier, data pre-processing is divided into four stages: data cleaning, data integration, data reduction, and data transformation.

a. **Data Cleaning:** Data cleaning refers to techniques to 'clean' data by removing outliers, replacing missing values, smoothing noisy data, and correcting inconsistent data. Many techniques are used to perform each of these tasks, where each technique is specific to user's preference or problem set. Below, each task is explained in terms of the techniques used to overcome it. In order to deal with missing data, multiple approaches can be used. Let's look at each of them.

Removing the training example: You can ignore the training example if the output label is missing (if it is a classification problem). This is usually discouraged as it leads to loss of data, as you are removing the attribute values that can add value to data set as well.

Filling in missing value manually: This approach is time consuming, and not recommended for huge data sets.

Using a standard value to replace the missing value: The missing value can be replaced by a global constant such as 'N/A' or 'Unknown'. This is a simple approach, but not fool proof.

Using central tendency (mean, median, mode) for attribute to replace the missing value: Based on data distribution, mean (in case of normal distribution) or median (for non-normal distribution) can be used to fill in for the missing value.

Using central tendency (mean, median, mode) for attribute belonging to same class to replace the missing value: This is the same as method 4, except that the measures of central tendency are specific to each class.

Using the most probable value to fill in the missing value: Using algorithms like regression and decision tree, the missing values can be predicted and replaced.

Noise is defined as a random variance in a measured variable. For numeric values, boxplots and scatter plots can be used to identify outliers. To deal with these anomalous values, data smoothing techniques are applied, which are described below.

Binning: Using binning methods smooths sorted value by using the values around it. The

sorted values are then divided into ‘bins’. There are various approaches to binning. Two of them are smoothing by bin means where each bin is replaced by the mean of bin’s values, and smoothing by bin medians where each bin is replaced by the median of bin’s values.

Regression: Linear regression and multiple linear regression can be used to smooth the data, where the values are conformed to a function.

Outlier analysis: Approaches such as clustering can be used to detect outliers and deal with them.

b. **Data Integration:** Since data is being collected from multiple sources, data integration has become a vital part of the process. This may lead to redundant and inconsistent data, which could result in poor accuracy and speed of data model. To deal with these issues and maintain the data integrity, approaches such as tuple duplication detection and data conflict detection are sought after. The most common approaches to integrate data are explained below.

Data consolidation: The data is physically bought together to one data store. This usually involves Data Warehousing.

Data propagation: Copying data from one location to another using applications is called data propagation. It can be synchronous or asynchronous and is event-driven.

Data virtualization: An interface is used to provide a real-time and unified view of data from multiple sources. The data can be viewed from a single point of access.

c. **Data Reduction:** The purpose of data reduction is to have a condensed representation of the data set which is smaller in volume, while maintaining the integrity of original. This results in efficient yet similar results. A few methods to reduce the volume of data are:

Missing values ratio: Attributes that have more missing values than a threshold are removed.

Low variance filter: Normalized attributes that have variance (distribution) less than a threshold are also removed, since little changes in data means less information.

High correlation filter: Normalized attributes that have correlation coefficient more than a threshold are also removed, since similar trends means similar information is carried. Correlation coefficient is usually calculates using statistical methods such as Pearson’s chi-square value etc.

Principal component analysis: Principal component analysis, or PCA, is a statistical method which reduces the numbers of attributes by lumping highly correlated attributes together. With each iteration, the initial features are reduced to principal components, with greater variance than the original set on the condition that they are uncorrelated with the preceding components. This method, however, only works for features with numerical values.

d. **Data Transformation:** The final step of data pre-processing is transforming the data into form appropriate for Data Modelling. Strategies that enable data transformation include:

Smoothing

Attribute/feature construction: New attributes are constructed from the given set of

attributes.

Aggregation: Summary and Aggregation operations are applied on the given set of attributes to come up with new attributes.

Normalization: The data in each attribute is scaled between a smaller range e.g. 0 to 1 or -1 to 1.

Discretization: Raw values of the numeric attributes are replaced by discrete or conceptual intervals, which can in return be further organized into higher level intervals.

Concept hierarchy generation for nominal data: Values for nominal data are generalized to higher order concepts.

A **data frame** is a table or a two-dimensional array-like structure in which each column contains values of one variable and each row contains one set of values from each column.Pandas DataFrame is two-dimensional size-mutable, potentially heterogeneous tabular data structure with labelled axes (rows and columns). DataFrame consists of three principal components, the data, rows, and columns. DataFrame will be created by loading the datasets from existing storage, storage can be SQL Database, CSV file, and Excel file. Pandas DataFrame can be created from the lists, dictionary, and from a list of dictionaries etc. We can perform basic operations on rows/columns like selecting, deleting, adding, and renaming.

4.1.2 Graph Creation

A knowledge graph acquires and integrates information into an ontology and applies a reasoner to derive new knowledge. In other words, a knowledge graph is a programmatic way to model a knowledge domain with the help of subject-matter experts, data interlinking, and machine learning algorithms. A knowledge graph is typically built on top of the existing databases to link all data together at web-scale combining both structured information or unstructured.

4.1.3 Keyword Extraction

Keywords form an important component since they provide a concise representation of the article's content. Keywords also play a crucial role in locating the article from information retrieval systems, bibliographic databases and for search engine optimization. Keywords also help to categorize the article into the relevant subject or discipline. Keyword extraction (also known as keyword detection or keyword analysis) is a text analysis technique that consists of

automatically extracting the most important words and expressions in a text. It helps summarize the content of a text and recognize the main topics.

You can use a keyword extractor to pull out single words (keywords) or groups of two or more words that create a phrase (key phrases). Keyword extraction will help us identify and extract the keywords from the question papers. These keywords will then be compared with the syllabus ontology of the respective paper.

4.1.4 Keyword Matching with Knowledge Graph

Keyword matching is done to identify responsible nodes while traversing the graph. Once the keywords are extracted, we match them with the knowledge graph created from the syllabus.

4.1.5 QP Fairness Calculation

We calculate the fairness of each node and then the parent nodes till the root node is reached.

4.1.6 Bloom’s Verb Extraction

Bloom's taxonomy is a taxonomy of measurable verbs to help us describe and classify observable knowledge, skills, attitudes, behaviours and abilities. In order for an objective to give maximum structure to instruction, it should be free of vague or ambiguous words or phrases. Bloom's taxonomy is a set of three hierarchical models used to classify educational learning objectives into levels of complexity and specificity. The three lists cover the learning objectives in cognitive, affective and sensory domains. The cognitive domain list has been the primary focus of most traditional education and is frequently used to structure curriculum learning objectives, assessments and activities. Bloom’s taxonomy is a powerful tool to help develop learning objectives because it explains the process of learning. Before you can understand a concept, you must remember it. To apply a concept, you must first understand it. In order to evaluate a process, you must have analysed it. To create an accurate conclusion, you must have completed a thorough evaluation.

4.1.7 Overall Bloom's Score Calculation

Bloom's taxonomy helps the teachers in setting a suitable question paper. With the provided hierarchy, the depth of the questions can be perceived. The lower words in the hierarchy prompt memorization while the higher words promote a much deeper understanding. The overall Bloom's score calculated tells us how well the question paper has been set.

4.2 Flowcharts

4.2.1 Ontology Tree

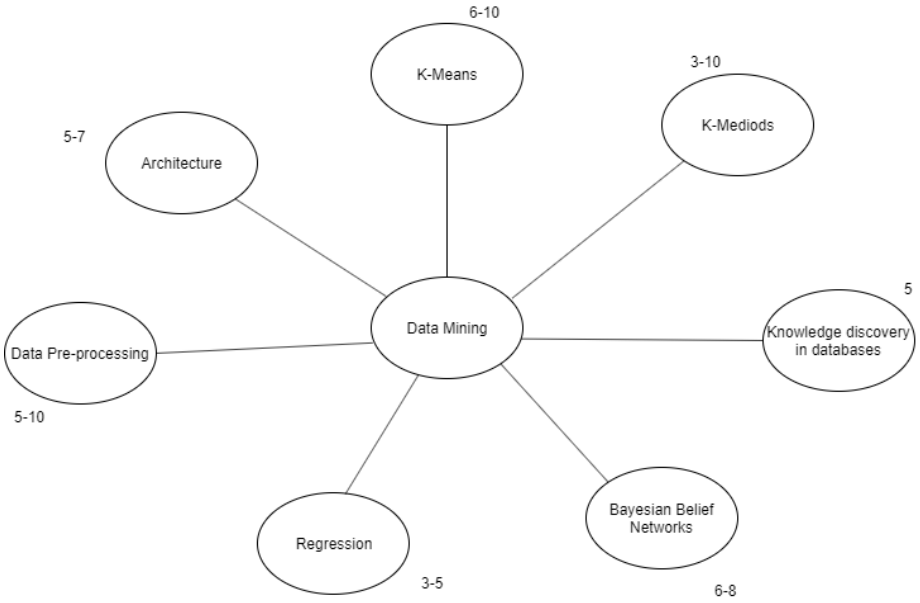

Fig 4.2 Ontology Tree

An ontology is a way of showing the properties of a subject area and how they are related, by defining a set of concepts and categories that represent the subject. Ontology encompasses a representation, formal naming and definition of the categories, properties and relations between the concepts, data and entities that substantiate one, many or all domains of discourse.

4.2.2 Use Case Diagram

A UML use case diagram is the primary form of system/software requirements for a new software program underdeveloped. Use cases specify the expected behaviour (what), and not the exact method of making it happen (how). The Use Case model shows how the user interacts with the system. The user can enter the question paper and the syllabus to acquire the desired results.

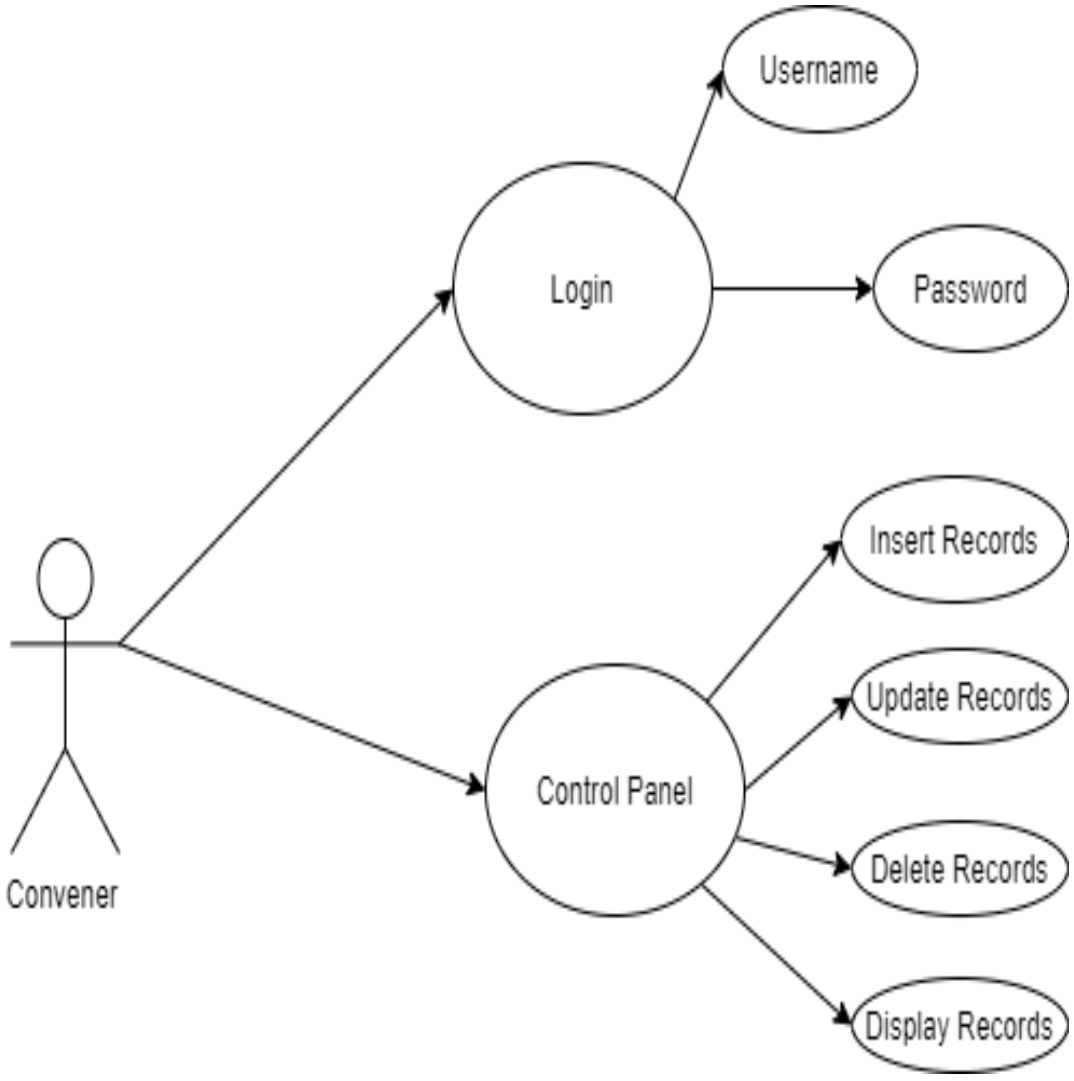

Fig 4.3 Exam Convener

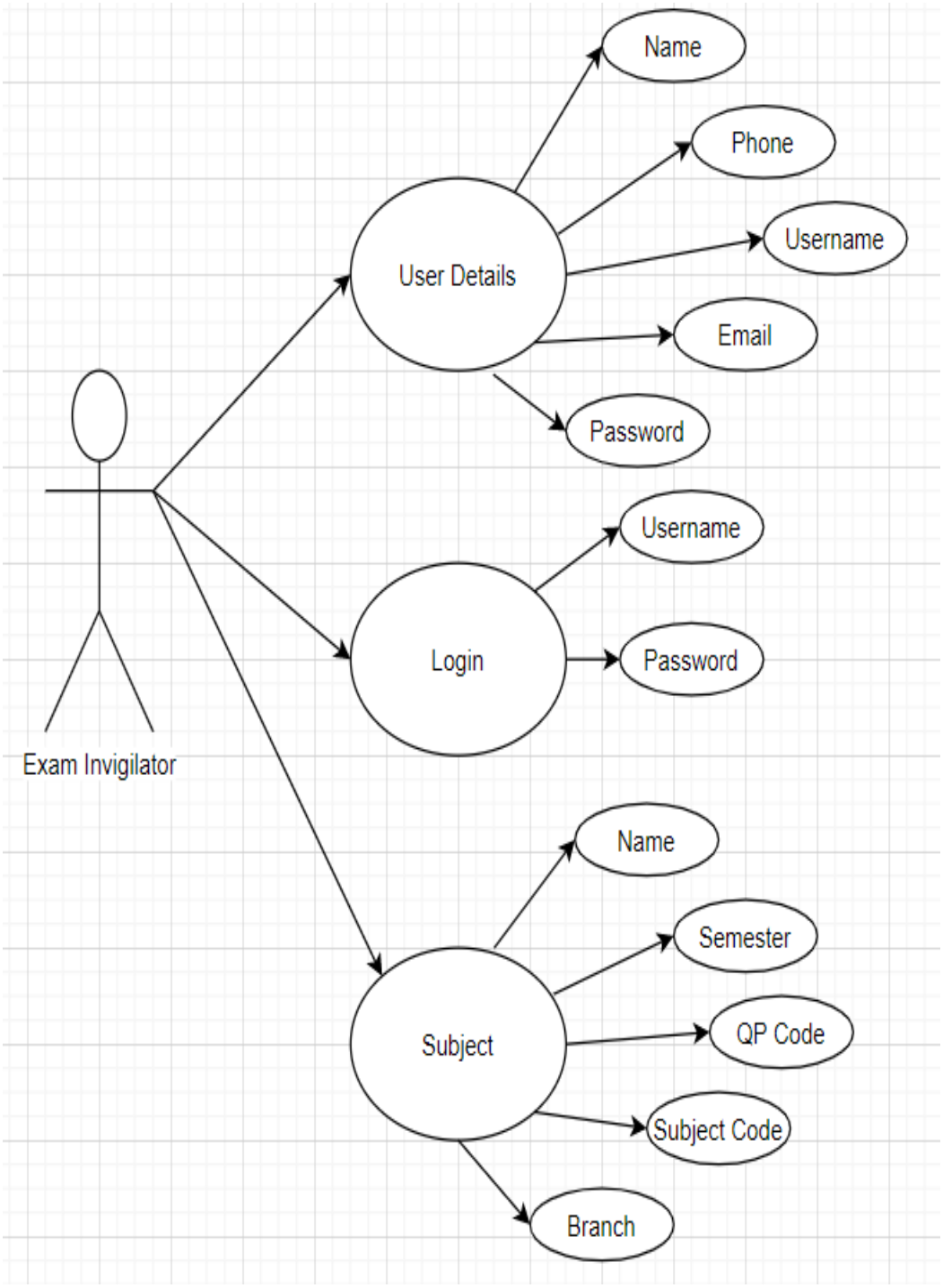

Fig 4.4 Exam Invigilator

4.2.3 Data Flow Diagrams

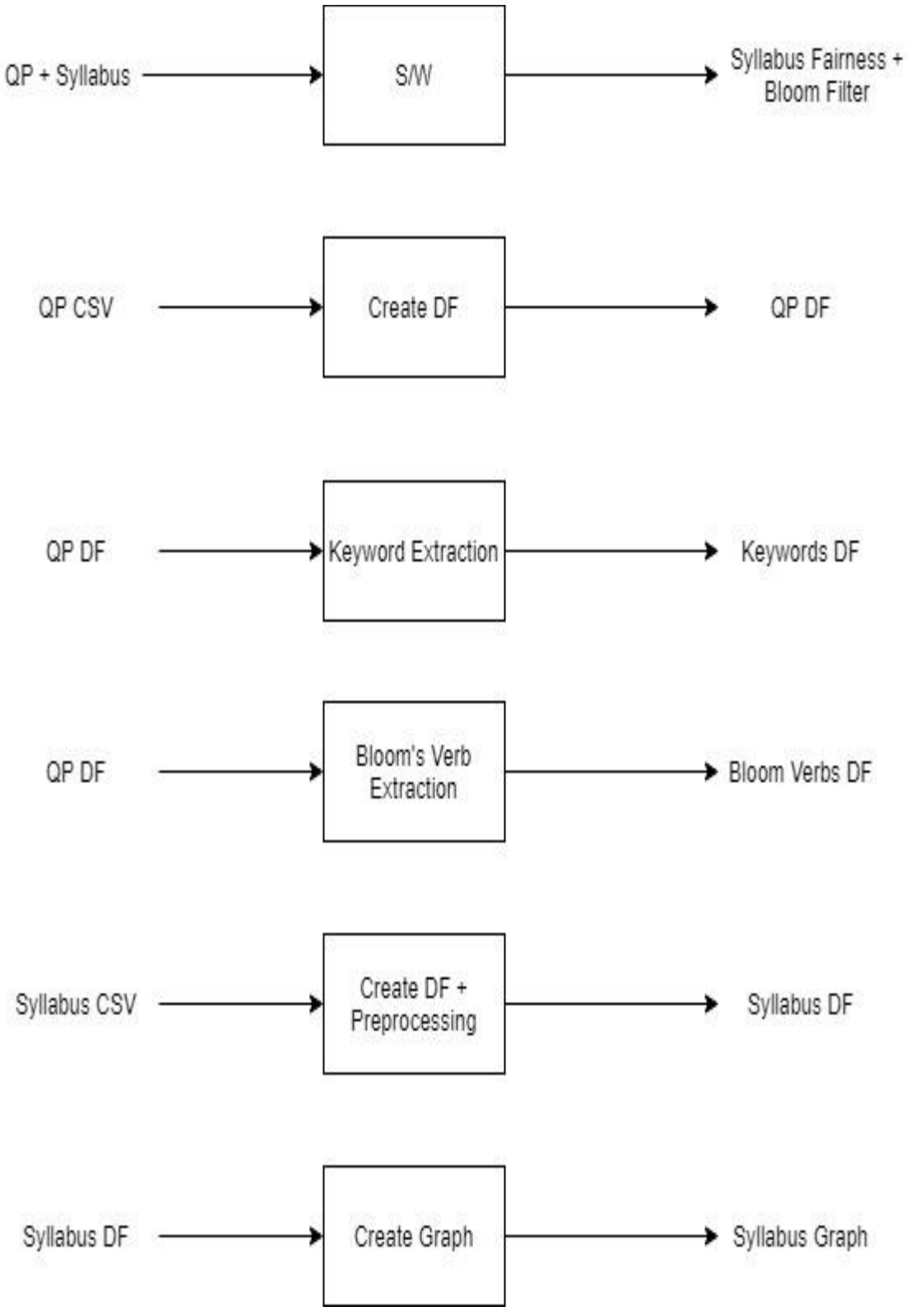
QP + Syllabus
S/W
Syllabus Fairness + Bloom Filter
QP CSV
Create DF
QP DF
QP DF
Keyword Extraction
Keywords DF
QP DF
Bloom's Verb Extraction
Bloom Verbs DF
Syllabus CSV
Create DF + Preprocessing
Syllabus DF
Syllabus DF
Create Graph
Syllabus Graph

Fig 4.5 DFD 1

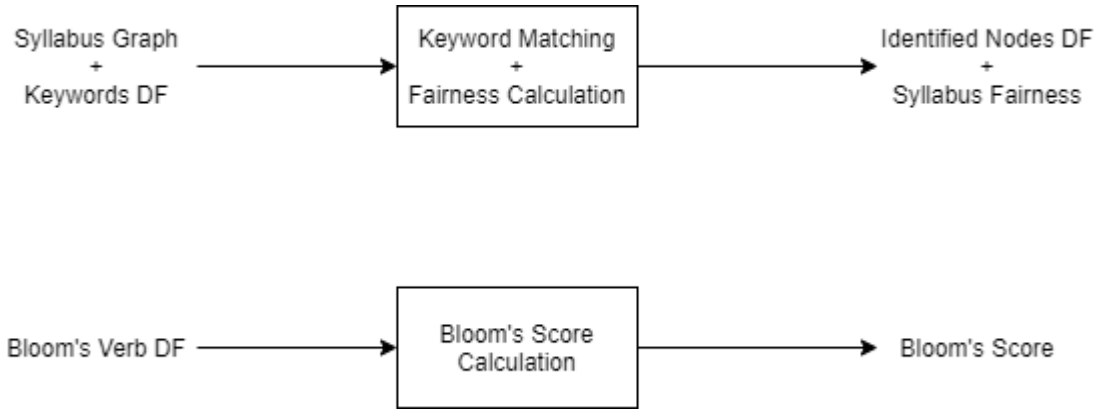

Fig 4.6 DFD 2

4.3 GUI Design

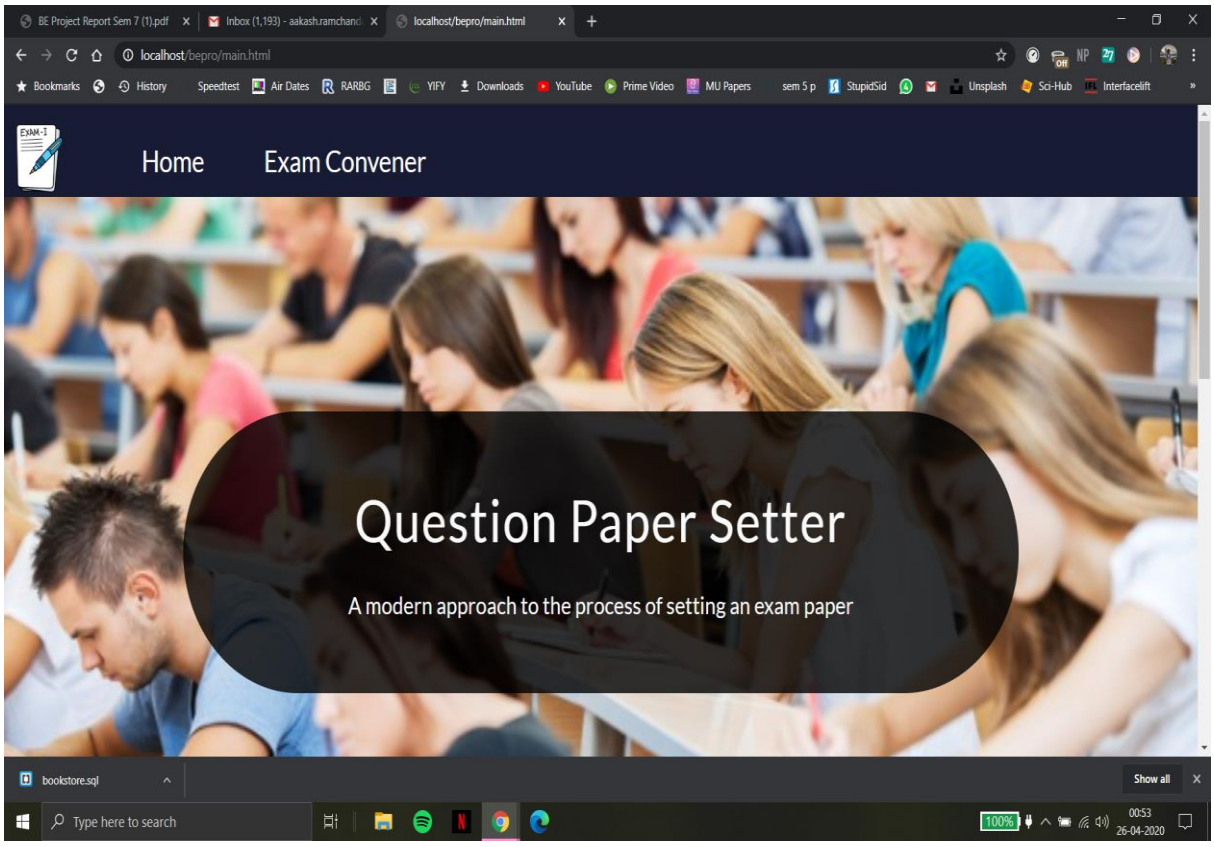

Fig 4.7 User 1

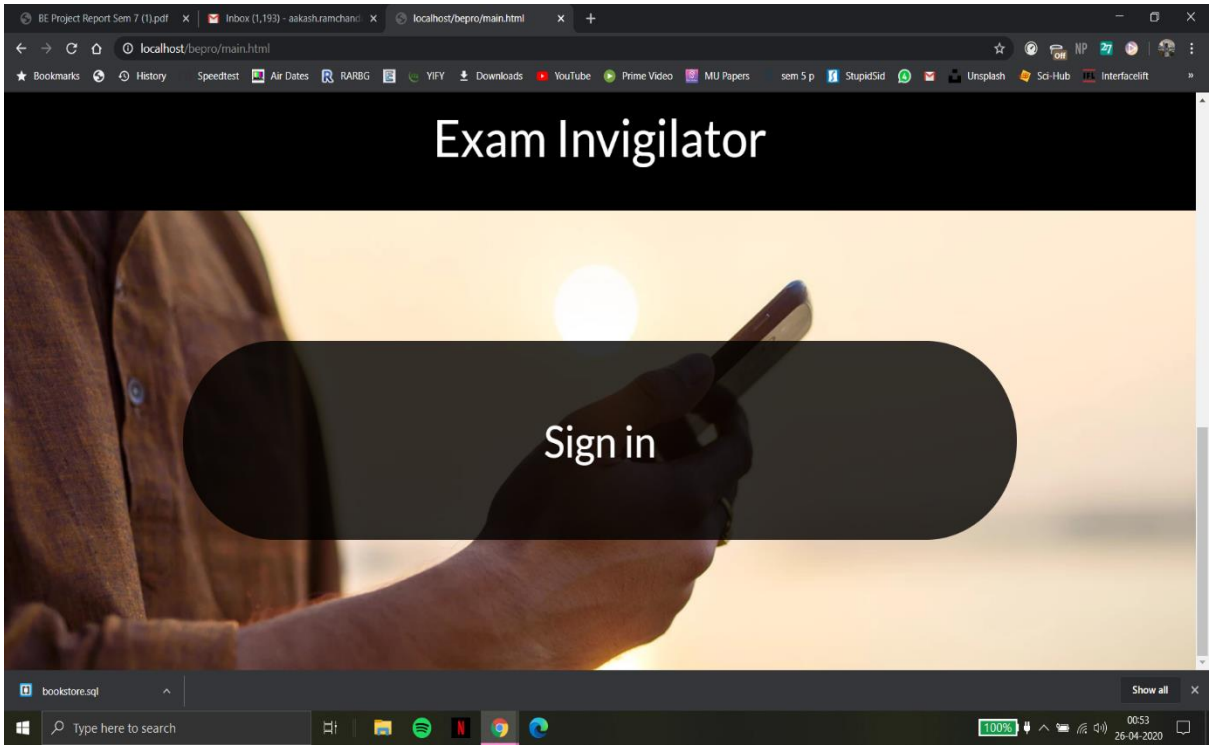

Fig 4.8 User 2

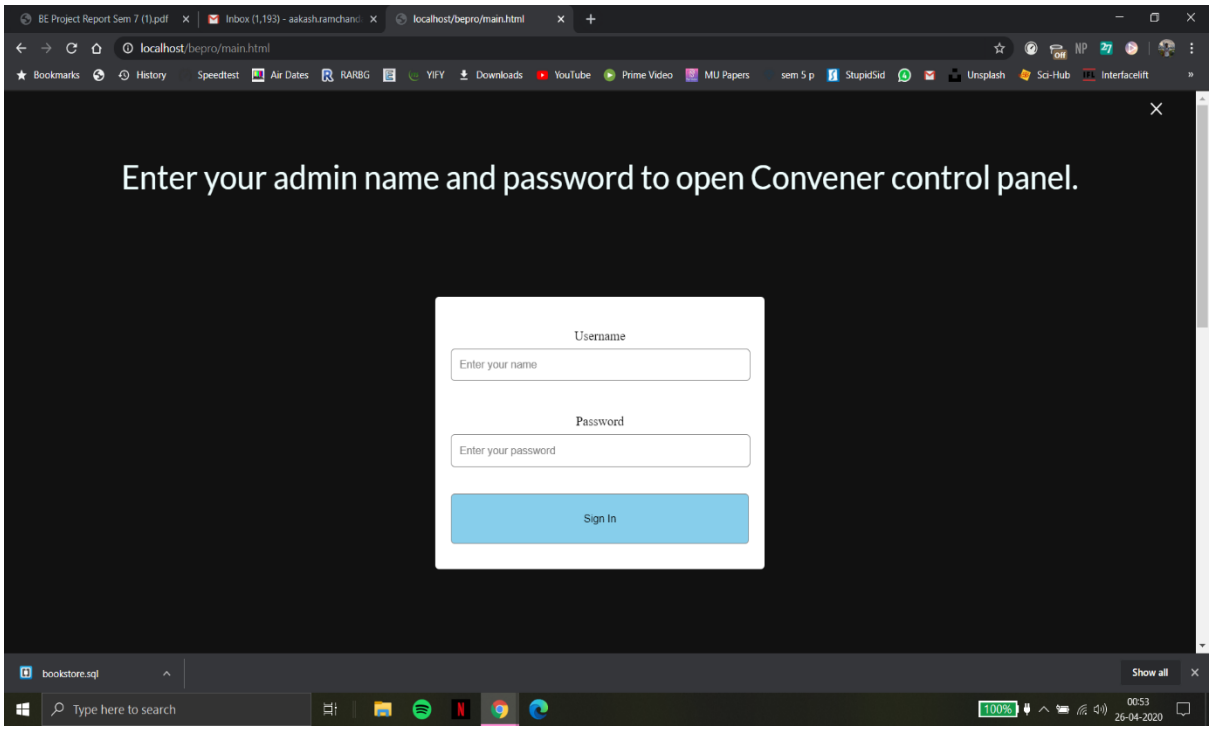

Fig 4.9 Login Page

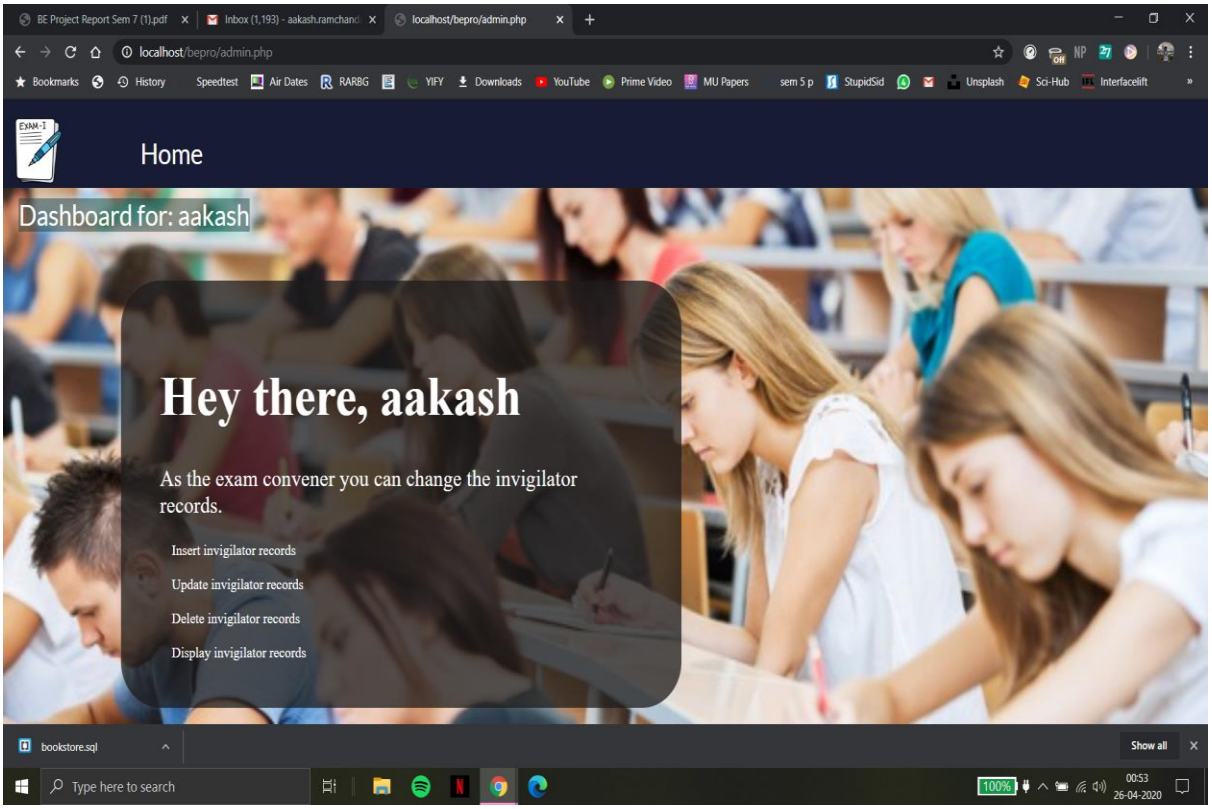

Fig 4.10 Dashboard

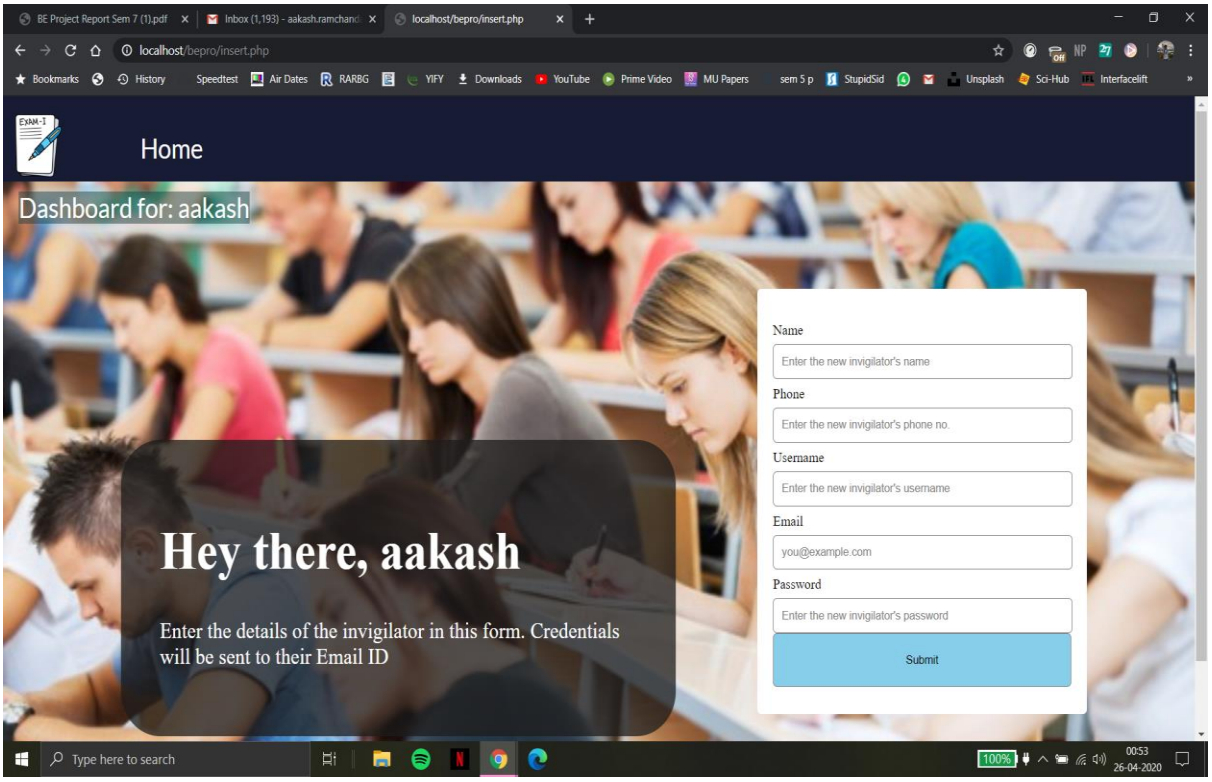

Fig 4.11 User Details

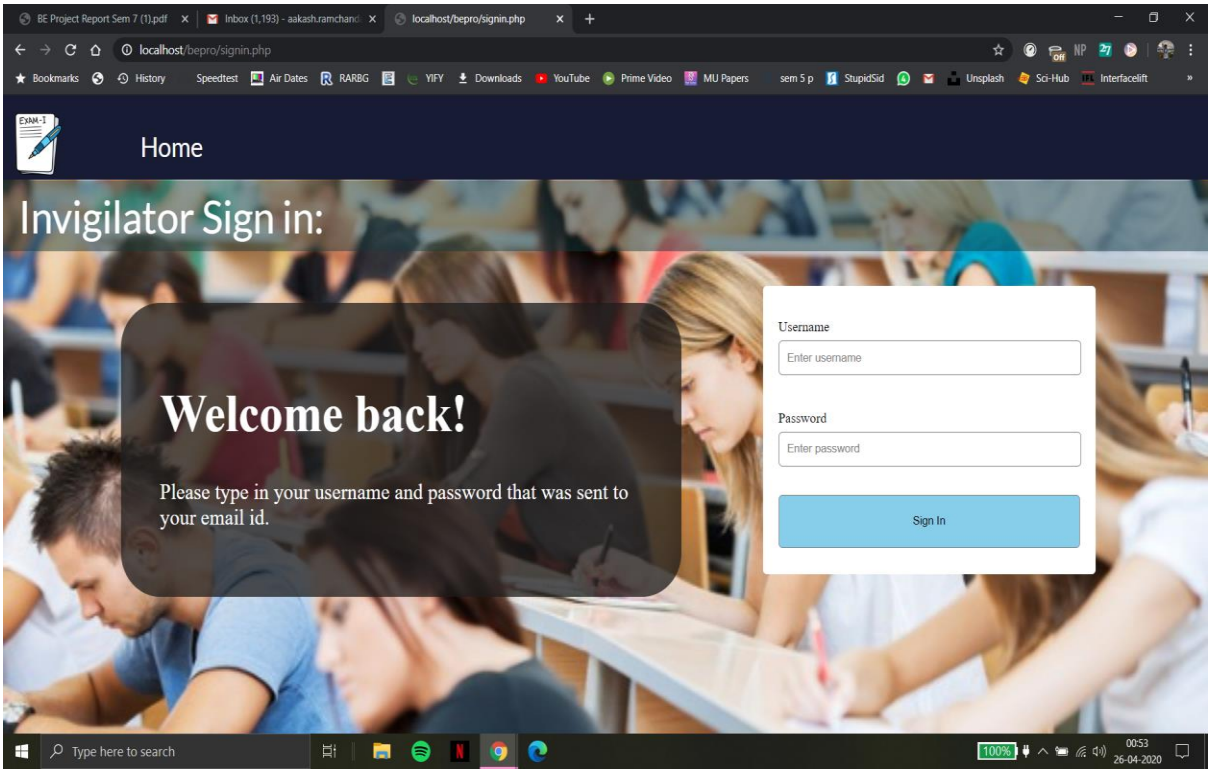

Fig 4.12 Invigilator

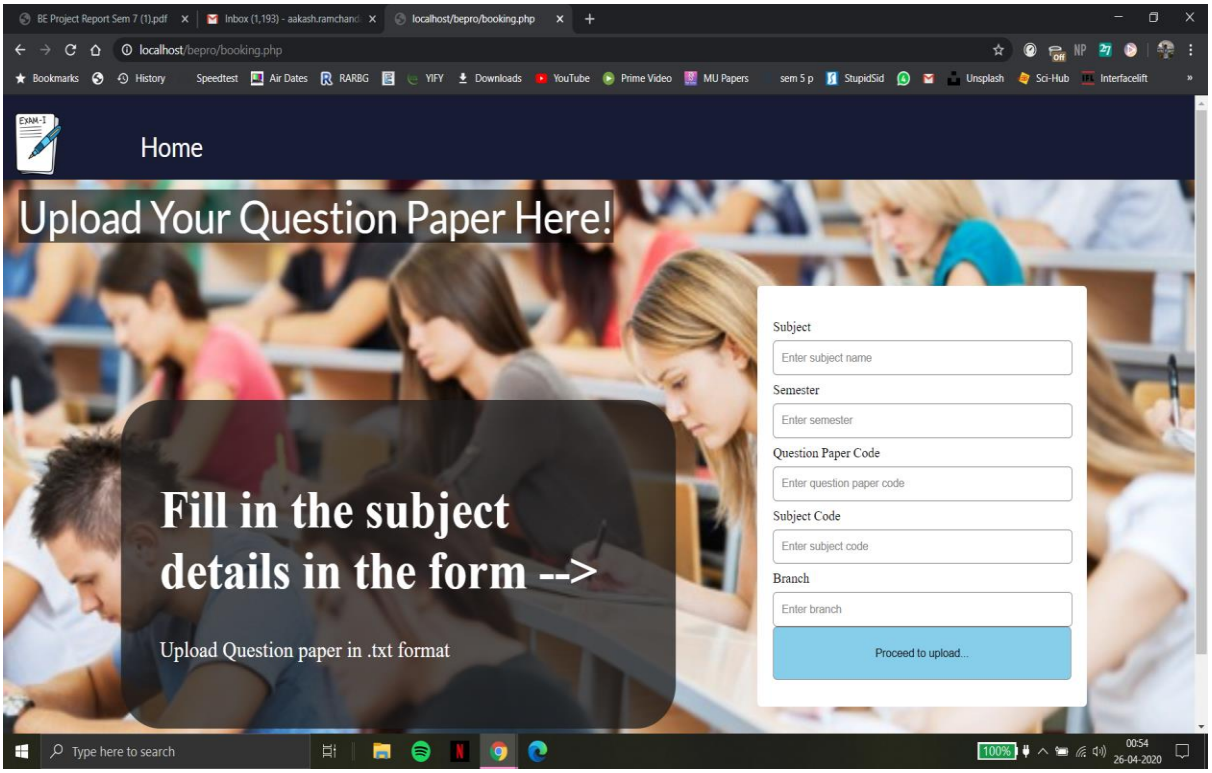

Fig 4.13 Subject Details

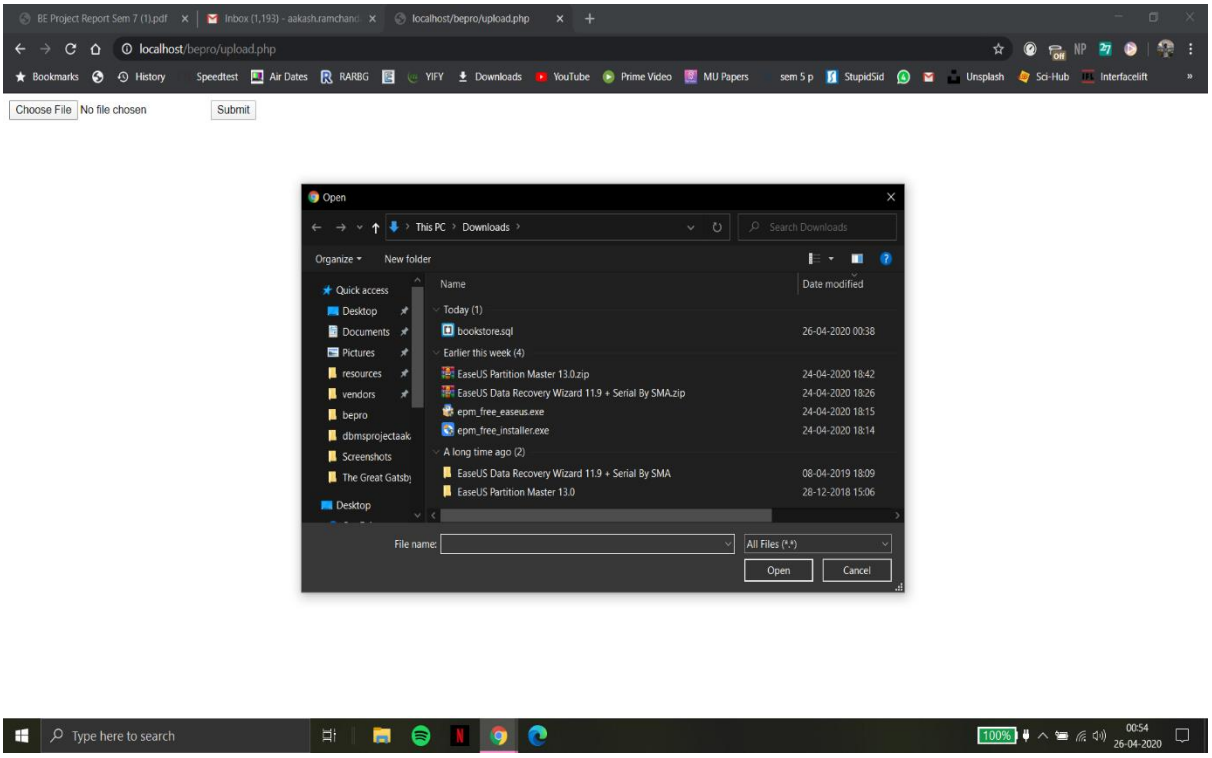

Fig 4.14 Upload

Chapter 5

Implementation

5.1 Architecture for Implementation

The broad approach to assessing fairness is as follows. Concepts associated with each question from the question paper are mapped to the syllabus ontology. Depending on how many nodes in the ontology are getting covered and how many times each node is referred by the questions, syllabus coverage in terms of percentage can be calculated.

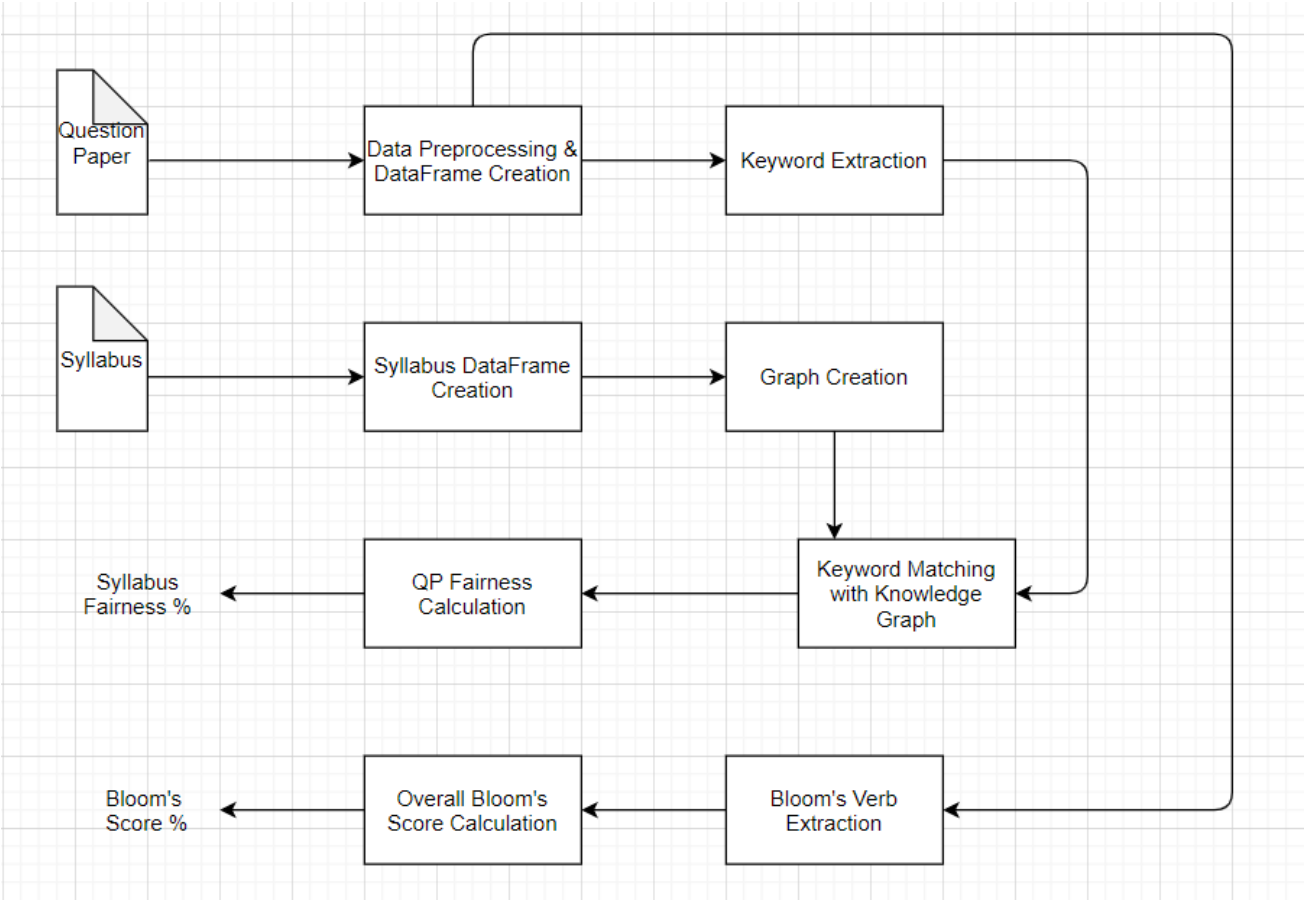

Fig 5.1 Architecture for Implementation

5.1.1 QP Data Preprocessing and DataFrame Creation

Text pre-processing can be divided into two broad categories: noise removal & normalization. Data components that are redundant to the core text analytics can be considered as noise. Handling multiple occurrences / representations of the same word is called normalization. There are two types of normalization: stemming and lemmatization. Let us consider an example of various versions of the word learn: learn, learned, learning, learner. Normalisation will convert all these words to a single normalised version: “learn”. Libraries used: Regular Expression Operations (re), Natural Language Toolkit (nltk) and Pandas.

The pandas library is required to convert the csv file where the papers are stored into the dataframe format.

	Num	Question	Marks
0	2a	Write a program to implement Circular Linked L...	10
1	4b	Explain different cases for deletion of a node...	10
2	5a	Write a program in ‘C’ to implement Stack usin...	10
3	5b	Explain Depth First search (DFS) Traversal wit...	10
4	6a	Application of Linked-List –Polynomial addition	10
5	6d	Topological Sorting	10

Fig 5.2 Preprocessed Data

5.1.2 Graph Creation

The knowledge graph we require is created from the syllabus DataFrame. The graph consists of Vertices or Nodes and edges showing the hierarchy of relation between them. The graph is created using the NetworkX library as well as pandas for representing the nodes and attributes. Each node has these attributes:

- Max Marks

- Node Bucket
- Overflow Bucket

  It is exported as pickle using the Pickk library.

Fig 5.3 Graph Created

5.1.3 Keyword Extraction

The dataframe acquired from the question paper now requires keywords to be identified. In order to do so we perform the following steps:

1. Convert to lowercase.
2. Remove the punctuations using the library RegEx.
3. We then remove the tags, special characters and digits.
4. Lemmatization is then carried out using the WordNet Corpus library.
5. Stopwords are then removed using the nltk library.

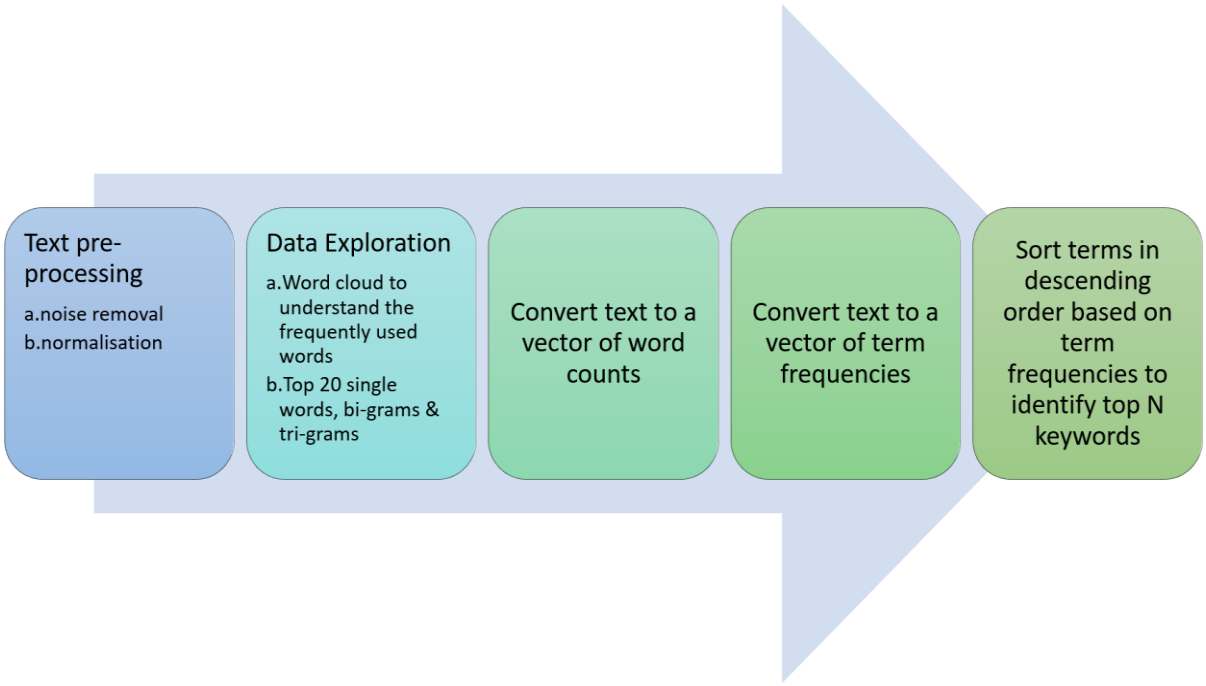

Fig 5.4 Extraction Process

	Num	Question	Marks	Extracted Keywords
0	2a	Write a program to implement Circular Linked L...	10	circular linked
1	4b	Explain different cases for deletion of a node...	10	case deletion node binary search tree function
2	5a	Write a program in 'C' to implement Stack usin...	10	c stack linked
3	5b	Explain Depth First search (DFS) Traversal wit...	10	depth first search dfs traversal recursive fun...
4	6a	Application of Linked-List –Polynomial addition	10	application linked polynomial addition
5	6d	Topological Sorting	10	topological sorting

Fig 5.5 Keywords Extracted

5.1.4 Keyword Matching with Knowledge Graph

Once the keywords are identified from the question paper, we match them with the knowledge graph nodes. We do this by traversing the graph and store the successors and predecessors of each node. We use string matching algorithms to match the keywords extracted,

with the nodes. This is done using the Difflib Sequence Matcher and FuzzyWuzzy python libraries.

	Num	Question	Marks	Extracted Keywords	Identified Node
0	2a	Write a program to implement Circular Linked L...	10	circular linked	circular linked list
1	4b	Explain different cases for deletion of a node...	10	case deletion node binary search tree function	None
2	5a	Write a program in 'C' to implement Stack usin...	10	c stack linked	linked list stack
3	5b	Explain Depth First search (DFS) Traversal wit...	10	depth first search dfs traversal recursive fun...	depth first search dfs
4	6a	Application of Linked-List – Polynomial addition	10	application linked polynomial addition	linked list polynomial addition
5	6d	Topological Sorting	10	topological sorting	topological sorting

Fig 5.6 Keywords Matched with Knowledge Graph

Once all the concepts / keywords are identified they have to be mapped to the syllabus ontology. Most of the keywords in the question will get mapped to the lower level concepts in the ontology. The keywords are first searched in the concept list. Every concept in the list has pointers directly to the nodes in the ontology. In this way all the keywords from a question gets mapped to the corresponding nodes in the ontology.

5.1.5 QP Fairness Calculation

The ontology contains a few details about the various keywords present in the syllabus. The NAME is the name of a concept which it is representing. WEIGHTS (r1-r2) provide information of the range of weightage assigned to that node in the ontology. FAIR NODE (FN) stores the fair marks associated with that particular question. The maximum capacity of FN is equal to the upper threshold r2 of node. OVERFLOW NODE (ON) stores the excess marks associated with that question in the paper being analysed. FAIR BUCKET (FB) stores the overall value of fair weightage marks and OVERFLOW BUCKET (OB) stores the unfair marks associated with the question paper being analysed. While mapping questions, fair node collects the marks assigned

to that concept in the QP. When a bucket is filled with more than its capacity, the excess marks are added in the overflow bucket.

We calculate for each node, then move on to their parent nodes and so on until the root node is reached in the graph. The traversal of nodes in the graph is done using the Breadth First Search(BFS) method of traversal. The graph had been created using the networkX library which also aids in the traversal of its nodes.

For the above example taken by us the value we have acquired is: **83.3%**

5.1.6 Bloom's Verb Extraction & Score Calculation

In order to carry out Bloom's verb extractions, from the questions:

1. We remove punctuations using the RegEx python library.
2. We convert all the questions to lowercase.
3. We remove all tags, special cases and digits.
4. Perform tokenization to separate individual words from the questions using the nltk python library.
5. We then carry out Tagging using the nltk library.

The custom Unigram tagger is responsible for tagging Bloom verbs.

	Num	Question	Marks	Extracted Keywords	Identified Node	Bloom's Score
0	2a	Write a program to implement Circular Linked L...	10	circular linked	circular linked list	0.380952
1	4b	Explain different cases for deletion of a node...	10	case deletion node binary search tree function	None	0.333333
2	5a	Write a program in 'C' to implement Stack usin...	10	c stack linked	linked list stack	0.380952
3	5b	Explain Depth First search (DFS) Traversal wit...	10	depth first search dfs traversal recursive fun...	depth first search dfs	0.428571
4	6a	Application of Linked-List – Polynomial addition	10	application linked polynomial addition	linked list polynomial addition	0.047619
5	6d	Topological Sorting	10	topological sorting	topological sorting	0.047619

Fig 5.7 Bloom's Score Calculation

The bloom score is then acquired for each question based on Bloom's taxonomy from extracted verbs on a scale: 0-1.

We then take the average of all scores in the entire paper to calculate the overall score. From the above example we've calculated it to be **26.98.**

5.2 Results and Evaluation

From the above performed process we can now conclude that the we've successfully constructed a knowledge graph which gives the relation between topics in the syllabus. We've devised a method of traversing this graph and have performed keyword matching to compare the questions of the paper set along with this knowledge graph to judge the syllabus fairness and thus conclude our result. For the above sample paper questions and syllabus we can successfully say that the paper is 83.3% fair. We also applied Bloom's Taxonomy to determine the depth of the

questions and based on the above selected example questions we can say conclude the overall bloom score to be 26.98 which is the calculated average of individual scores of the used questions.

Chapter 6

Conclusion

In this report, we discussed the issues of providing an automated framework to judge the fairness of a question paper towards a given syllabus. We have made use of ontology and knowledge graphs that forms the semantically connected network of concepts (topics) from the domain to represent a syllabus. We've devised a method of traversing this graph and have performed keyword matching to compare the questions of the paper set along with this knowledge graph to judge the syllabus fairness and thus conclude our result. We've identified various syllabus fairness issues and developed a model to resolve them as well as to judge the depth of the questions in the paper as set by using Bloom's Taxonomy. Experimental results for the domain of Data Structures have been reported as observed from our implementation. We have created a GUI in the form of a website so as to display our project and to provide the user of our website access to it for his purpose, may he be an invigilator or the exam convener. The result obtained by us is as desired and we can say that our project has been successfully been implemented.

References

1. Natural Language Processing :
https://en.wikipedia.org/wiki/Natural_language_processing

2. Ontology:
https://en.wikipedia.org/wiki/Ontology_(information_science)

3. Ontology driven NLP :
https://www.ibm.com/developerworks/community/blogs/nlp/entry/ontology_driven_nlp?lang=en

4. Keyword Extraction using NLP:
https://medium.com/analytics-vidhya/automated-keyword-extraction-from-articles-using-nlp-bfd864f41b34

5. Grigoris, A. and Frank, van, H. A Semantic Web Primer. The MIT Press, Cambridge, Massachusetts, London.

6. M. H Dunham, "Data Mining: Introductory and Advanced Topics',

7. G Ducatel, Z Cui, B Azwine, "Hybrid ontology and keyword matching indexing system" IntraWebs Workshop at WWW2006.

8. B. Katz J. Lin, " Annotating the Semantic Web Using Natural Language", proc. workshop on NLP and XML (NLPXML 2002), COLING 2002, Taipie, Taiwan September 2002.

9. Pandas:

https://pandas.pydata.org/pandas-docs/version/0.15/tutorials.html

10. Pickle:

https://docs.python.org/3/library/pickle.html

11. NetworkX:

https://networkx.github.io/documentation/stable/

12. Python RegEx:

https://docs.python.org/3/howto/regex.html

13. Wordnet Corpus:

https://www.nltk.org/howto/wordnet.html

14. NLTK:

https://pythonprogramming.net/stop-words-nltk-tutorial/

15. Difflib Sequence Matcher:

https://docs.python.org/3/library/difflib.html

16. FuzzyWuzzy:

https://pypi.org/project/fuzzywuzzy/

17. Bloom's Taxonomy:

https://cft.vanderbilt.edu/guides-sub-pages/blooms-taxonomy/

ACKNOWLEDGEMENTS

Our team is grateful for being given an opportunity to build a project in the domain of NLP and would like to thank all the people who have given us their cooperation in making this project a success. We extend our sincere thanks to Mrs. Ujwala H. Bharambe, our guide and our project co-ordinator for providing sufficient infrastructure and a good environment in the college to complete our project and with several pointers to make our project better with her constant encouragement and providing us with the chance to make a project and work on domains we wanted to explore. We also would like to show our appreciation for our college Thadomal Shahani Engineering College and our entire faculty for their constant guidance and support which has helped us reach where we are and enabling us to use our theoretical knowledge in practical applications in various domains.

-Mr. Eashan Bajaj

-Mr. Aakash Ramchandani

-Mr. Omkar Yadav