

WORD TO WORD SIMILARITY CHECKER OF QUESTION DATA

Team: Neo Jarvis



June 13, 2022

Iamneo.ai

Coimbatore

HIGH LEVEL DOCUMENTATION

Revision Number: 03

Last Date of Revision: 01 Aug. 2022

Team Member:

* Suresh Shanmugam – Caretaker
* Biswajit Rajaguru Mohapatra – Data Science Engineer
* Swati Kahar – Data Science Engineer
* Gautham Sevarkodian – Business Analyst

Document Version Control

|  |  |  |  |
| --- | --- | --- | --- |
| **Date Issued** | **Version** | **Description** | **Author** |
| 18 May 2022 | 0.0 | Problem Statement | Biswajit, Swati |
| 14 June 2022 | 0.0 | POC | Biswajit, Swati |
| 17 June 2022 | 0.0 | Architecture Design | Biswajit |
| 25 June 2022 | 1.0 | Question Duplication and API integration | Biswajit, Swati |
| 10 July 2022 | 1.0 | POC of Filters | Biswajit, Swati |
| 15 July 2022 | 2.0 | Documentation | Gautham, Biswajit |
| 18 July 2022 | 2.0 | Final Code as per Filters | Biswajit, Swati |
| 28 JUL 2022 | 3.0 | Update Architecture | Biswajit |
| 29 JUL 2022 | 3.0 | Integrated Cloud Storage Module | Biswajit |
| 01 AUG 2022 | 3.0 | Data Flow Design | Biswajit |

Contents

1. Abstract
2. Introduction
3. General description
4. Design details
5. Performance
6. Conclusion

**Abstract**

Neo exam is one of the important products of iamneo.ai. This is a B2B product which helps organizations to conduct exams to fulfil their requirements as per their needs. The preparation of the questions is being done by the Neo Elon team in our organization. Neo Elon team is responsible for preparing questions, question banks and solutions for the prepared question for evaluation purpose. The prepared question is then being displayed to the client. The team prepared almost 20 kinds of questions as of now.

However, sometimes the question being prepared gets repeated and stored in the database and the same gets displayed on the client side i.e., the repeated questions. This results in duplicated or exact copies of the question being present in the question bank that the client receives. This is also responsible of taking up extra spaces in our database.

This is a project as a part of our internal tech optimization and helping Neo Elon team not facing the duplicate question problem.

Introduction

Why this High-Level Documentation?

The purpose of this High-Level Design (HLD) document is to add necessary details to the current project description to represent a suitable model for coding. This document is also intended to help detect contradictions prior to coding, and can be used as a reference manual for how modules interact at a high level.

The HLD will:

* Present all the design aspects and define them in detail
* Describe the user interface being implemented
* Describe the hardware and software interfaces
* Describe the design features and architecture of the project.
* List and describe the non-functional attributes like:
* Reliability
* Maintainability
* Portability
* Reusability
* Application Compatibility
* Resource utilization
* Serviceability

**Scope**

The HLD documentation present the structure of the system, such as the database architecture, application architecture (layers), application flow (Navigation), and technology architecture. The HLD uses non-technical to mildly technical terms which should the understandable to the administrator of the system.

**Definitions**

|  |  |
| --- | --- |
| Term | Description |
| Database | Collection of all the information monitored by this system |
| IDE | Integrated Development Environment |
| BQ | Big Query |
| GCP | Google Cloud Platform |
| Cloud Storage | Google Cloud Storage |

General Description

Product Perspective

Question duplicate filtering application is intended to do the following things:

1. Will receive a new question as an user input and check if the exact question is already present in the database.
2. In case of presence of an exact match of the input question the application will throw an response of number of duplicate and details of the duplicated questions.

This product is intended to solve the following problem the organization as a whole is facing:

1. More than 70% questions in the database are a duplicate created by the concerned team through various processes, this results in having duplicated questions in the question bank prepared by the Neo Elon team for the Neo Exam and PAT platforms. These question banks are directly passed to the client, presence of repeated questions result in lowering the credibility of the particular platform. Cleaning the database from duplicate questions and checking for presence of the newly entered question in the database will restrict the user to enter duplicate question and will optimize the question bank accordingly.
2. This product will help in generating expected question banks i.e, question bank without duplicates.

Problem Statement:

To create an end-to-end application that could fetch the updated question data along with its details and compare a given new question with the questions present in database to check an exact match. On a successful match, the application should be able to generate and show the details of the duplicated questions on the platform.

Proposed solution:

The solution proposed here is to build an application that could accept a question and school code. Using these inputs, the application will fetch the data from the database and process it to find an exact match for the accepted question in our database. If found, the application will give an output of the details of the matched question along with their count. This will ease the efforts of the Neo Elon team to find the presence of duplicate question in the database manually and duplication of the questions in the database could be easily avoided.

Further Improvements:

The same model can be further enhanced to have a feature of receiving bulk-uploaded question file and process them to check for duplication in the database. The same application can also have an integration to check for the semantic matching of the exact matched question for having a filtration based on the meaning of the questions.

Data Requirements:

Data requirements are completely dependent on the problem statement.

* The particular application needs to have data that is present in the database.
* The application demands an authentication file that establishes a connection with the google big-query.

Tools Used:

Python as a programming language, Pandas, NumPy and google-cloud as libraries are used for this particular program.

* VS-Code is used as IDE
* GitHub for versioning of the application.
* For front-end development HTML/CSS is used.
* Python FLASK is used for API creation.
* Google big-query as data source
* Google Cloud storage for file storage

Constraints:

The word-word question duplicate application should be user friendly, as automated as possible and user shouldn’t need to know any of the internal working of the application.

Assumptions:

The main objective of the project is to implement the use cases as previously mentioned problem statement for each time the database and input question is updated with reference to duplicate question.

Design Details

Process Flow:

For the application to work efficiently on finding the duplicated questions the following process-flow need to be followed.

FETCHING DATA

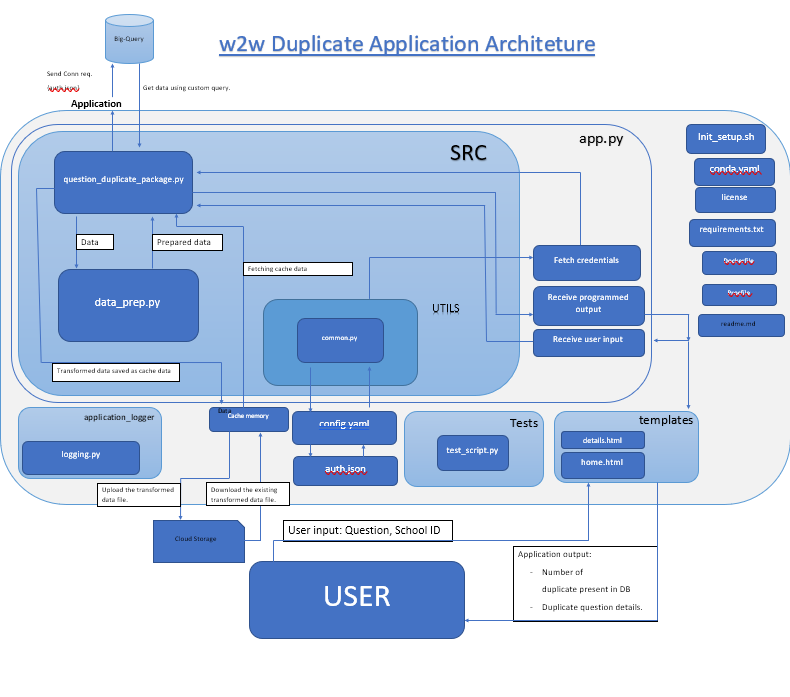
ANNONATION CHECK

DATA PREPROCESSEING

FILTER THE DUPLICATES

CHECK THE INPUT QUESTION’S PRESENCE IN FILTERED DATA

TAKE NECESSARY ACTION

 Architecture Design

Event log:

The system should log every event so that the user will know what process is running internally.

**Initial step-by-step Description:**

1. The system identifies what step logging is required.
2. The System should be able to log each and every system flow
3. Developer can choose logging method, either database logging or file logging as well
4. System shouldn’t hang even after using so many loggings. Logging is done so that we can debug issues at process level, so it is mandatory.

Error Handling:

System should be able to handle the detected error and an explanation should be displayed is to what went wrong? An error will be defined as anything that falls outside the normal and intended usage.

Data Flow:

The flow of the data is made as simpler as possible considering the file management efficiency and processing efficiency. The following link can be used to visualize the data flow of the application.

[Data-Flow](https://www.figma.com/file/JrsbruMXVhNl7z7g4fA38I/W2W_Duplicate_Data_Flow?node-id=0%3A1)

Performance

The word-word duplicate application is used for filtering out the duplicated questions in the database along with this the application is also responsible for checking weather the newly input question is already present in the database or not.

Reusability:

The code written and the components used should have the ability to be reused with no problems.

Application Compatibility:

The different components for this project will use python as an interface between them. Each component and module will have its own task to perform, it is the job of the python to ensure the proper transfer of information.

Resource Utilization

When any task is performed, it will likely use all the processing power available until that function is finished. The application will interact with data storage and data fetching platforms will all its capacity till the process is finished.

Deployment:

Conclusion

The designed word-word duplicate model will efficiently check if the given question has a duplicate present in the database or not provided the school code is also provided by the user along with the question. Using this application, the organization will conclude the following points:

1. Avoiding the issue of generating question bank with duplicated question
2. Avoiding the issue of duplicated questions being updated in the database.