### Project Report On

**“COLLEGE EVENTS HUB”**

Submitted in partial fulfillment for the award of the degree of

# Bachelor of Technology in

**Information Technology By**

## TIRUMALA SANDEEP KUMAR (20R21A12A9)

## KURUVA RITHEN KUMAR (20R21A1276)

## GV. SUMANTH (20R21A1223)

Under the guidance of

## Mrs.Jeethu Philip (Assistant Professor)



**2024**

# Department of Information Technology



**Department of Information Technology**

CERTIFICATE

This is to certify that the project entitled **“College Events Hub”** has been submitted by **TIRUMALA SANDEEP KUMAR (20R21A12A9), KURUVA RITHEN KUMAR (20R21A1276), GV. SUMANTH (20R21A1223)** in the partial fulfillment of the requirements for the award of degree of Bachelor of Technology in Information Technology from Jawaharlal Nehru Technology University, Hyderabad. The results embodied in this project have not been submitted to any other University or Institution for the award of any degree or diploma.

Supervisor Head of the Department

Mrs. Jeethu Philip Dr. N.V Rajashekar Reddy

External Examiner

# DECLARATION

I hereby declare that the project entitled **“College Events Hub”** is the work done during the period from **August 2023 to April 2024** and is submitted in the partial fulfillment of the requirements for the award of degree of Bachelor of technology in Information Technology from Jawaharlal Nehru Technology University, Hyderabad. The results embodied in this project have not been submitted to any other University or Institution for the award of any degree or diploma.

**Tirumala Sandeep Kumar (20R21A12A9)**

**Kuruva Rithen Kumar (20R21A1276)**

**GV. Sumanth (20R21A1223)**

# ACKNOWLEDGEMENT

There are many people who helped us directly and indirectly to complete our project successfully. We would like to take this opportunity to thank one and all. First of all we would like to express our deep gratitude towards our supervisor **Mrs. Jeethu Philip, Assistant Professor, Department of IT** for her support in the completion of our dissertation. We wish to express our sincere thanks to **Dr. N.V RAJASHEKAR REDDY, HOD, Department of IT** and also to Principal **Dr.K.SRINIVASA RAO** for providing the facilities to complete the dissertation.

We would like to thank all our faculty, coordinators and friends for their help and constructive criticism during the project period. Finally, we are very much indebted to our parents for their moral support and encouragement to achieve goals.

**Tirumala Sandeep Kumar (20R21A12A9)**

**Kuruva Rithen Kumar (20R21A1276)**

**Gv. Susmanth (20R21A12A9)**

# ABSTRACT

College Events Hub is a cutting-edge platform designed to revolutionize how college clubs and events are managed and experienced. Serving as a comprehensive solution, it bridges communication divides between club members, leaders, coordinators, and the broader student body. Through its intuitive interface, users gain access to a wealth of information about both college-hosted and external events, ensuring they stay up-to-date with the latest announcements and schedules. Administrators enjoy secure access through a robust authentication system, safeguarding sensitive data and allowing for effective management of the platform.

At its core, College Events Hub streamlines event management by offering detailed information and facilitating smooth registration with an integrated payment system. Organizers can effortlessly coordinate activities while attendees enjoy a hassle-free registration process, promoting greater participation and engagement in college activities. Furthermore, the platform fosters interaction and feedback through its event-specific comment sections, empowering users to connect, share insights, and build community. By incorporating student ratings and reviews, College Events Hub not only informs future event decisions but also facilitates continuous improvement, ultimately enhancing the overall college experience.

# CONTENTS

|  |  |  |
| --- | --- | --- |
| **S. No** | **Contents** | **Page No** |
|  | Certificate | 2 |
|  | Declaration | 3 |
|  | Acknowledgement | 4 |
|  | Abstract | 5 |
| 1 | Introduction | 8-9 |
| 1.1 | Overview |  |
| 1.2 | Purpose of the project |  |
| 1.3 | Motivation |  |
| 2 | Literature Survey | 10-12 |
| 2.1 | Existing System |  |
| 2.2 | Disadvantages of Existing System |  |
| 3 | Proposed System | 13-15 |
| 3.1 | Advantages of Proposed System |  |
| 3.2 | System requirements |  |
| 4 | System Design | 16-18 |
| 4.1 | Proposed System Architecture |  |
| 4.2 | Modules |  |
| 5 | Implementation and Installation | 19-30 |
| 5.1 | Algorithm |  |
| 5.2 | Implementation Steps |  |
| 5.3 | Source Code |  |
| 6 | DFD Diagrams | 31-32 |
| 7 | Testing | 33-37 |
| 7.1 | Introduction |  |
| 7.2 | Unit Testing |  |
| 7.3  7.4  7.5 | Integration Testing Validation  Closure of Testing and Validation |  |
| 8 | Results | 38 |
| 8.1 | Accuracy |  |
| 8.2 | Cross-Validation |  |
| 9 | Screenshots | 39-43 |
| 10 | Conclusion | 44-45 |
| 11 | References | 46-48 |

**List of Figures**

|  |  |  |
| --- | --- | --- |
| **Figure No** | **Name of the Figure** | **Page no** |
| 2 | NLP Stages | 04 |
| 2 | Working Mechanism of NLP | 04 |
| 4.1 | Architecture Diagram | 09 |
| 4.1 | Proposed System Architecture | 10 |
| 6.1 | DFD diagram | 25 |
| 9 | Screenshots | 33 |

# INTRODUCTION

## OVERVIEW

In the vibrant realm of college life, managing events effectively plays a pivotal role in fostering community engagement, collaboration, and student enrichment. To address this essential need, College Events Hub emerges as a powerful, full-stack platform tailored specifically for organizing and participating in college events with ease.

Utilizing Next.js 14 as its foundation, College Events Hub combines advanced technology with user-friendly design to offer a comprehensive solution perfectly suited for the collegiate environment. It goes beyond simple event organization, serving as a central hub for communication and interaction among various clubs, leaders, coordinators, and the student body.

Powered by a diverse tech stack including Node.js, TypeScript, and TailwindCSS, College Events Hub integrates seamlessly with essential tools like Stripe for secure payment processing and React Hook Form for smooth data management. This ensures a seamless experience for administrators and users alike.

At its core, College Events Hub empowers administrators to manage events effortlessly, enabling them to create, edit, and remove events as needed. Meanwhile, students benefit from an intuitive interface that provides detailed event information, facilitates easy registration, and even allows for event ratings and feedback submission.

In essence, College Events Hub transcends traditional event management on college campuses. It enriches the college experience by promoting efficient communication, engagement, and community building. With its innovative features and user-friendly design, College Events Hub redefines the way colleges approach event organization, fostering a more vibrant and connected campus culture.

## PURPOSE OF THE PROJECT

The primary purpose of the College Events Hub project is to revolutionize event management within the college setting. By consolidating club activities and event information into a centralized platform, the project aims to simplify communication between club members, leaders, coordinators, and students. This platform serves as a comprehensive solution, offering up-to-date event details, secure registration processes, and avenues for feedback and interaction, ultimately fostering a more connected and engaged college community.

Through its user-friendly interface and robust features, College Events Hub seeks to streamline event planning and execution, making it easier for organizers to coordinate activities and for attendees to participate. By promoting efficient communication and facilitating seamless event registration, the project aims to enhance the overall college experience, encouraging greater involvement in campus events and fostering a sense of belonging among students. Overall, the purpose of the project is to create a dynamic and inclusive environment where students can easily discover, participate in, and contribute to a diverse array of college activities and Events.

Furthermore, the overarching purpose of the College Events Hub project extends beyond mere event management to encompass the broader goal of enriching the college experience. By promoting efficient communication, facilitating seamless event registration, and fostering interaction and feedback, the project aims to cultivate a vibrant and inclusive campus culture. Ultimately, the project seeks to create an environment where students feel empowered to explore their interests, engage with their peers, and actively contribute to the dynamic tapestry of college life. Through its comprehensive approach to event management and community building, College Events Hub endeavors to elevate the college experience and leave a lasting impact on the campus.

## MOTIVATION

The motivation behind the development of the College Events Hub project stems from a recognition of the inherent challenges and inefficiencies associated with managing club activities and events within the college ecosystem. Traditional methods of communication and event organization often rely on fragmented systems, leading to confusion, missed opportunities, and administrative overhead. Understanding the importance of fostering a vibrant and engaged campus community, there was a clear motivation to create a centralized platform that addresses these shortcomings and streamlines the entire event management process.

Moreover, the desire to enhance the overall college experience for students served as a significant motivational factor. Recognizing that extracurricular activities and events play a vital role in shaping students' holistic development, there was a strong impetus to provide a solution that not only facilitates easier event coordination but also promotes greater participation and engagement. By offering a user-friendly platform that empowers students to discover, register for, and provide feedback on events, the project aims to create a more dynamic and inclusive campus environment where students feel connected and supported in their extracurricular endeavors.

# LITERATURE SURVEY

### INTRODCTION

Event management within educational institutions has garnered significant attention in academic research due to its vital role in enhancing student engagement, fostering community spirit, and promoting campus culture. Existing literature offers insights into various aspects of event management within colleges and universities, shedding light on best practices, challenges, and opportunities for improvement.

Communication and Collaboration: Studies highlight the importance of effective communication and collaboration among students, faculty, and administrators in the planning and execution of college events (Reinhardt et al., 2015; Kirillova et al., 2017). Fragmented communication channels and lack of collaboration platforms have been identified as barriers to successful event management, emphasizing the need for centralized communication tools and collaborative platforms to streamline coordination efforts (Ho and McKechnie, 2016; Ratten, 2018).

Technology Integration: The integration of technology in event management has emerged as a key trend, enabling colleges to automate administrative tasks, enhance communication, and improve participant engagement (Al-Ansari and Ismail, 2017; Kononova et al., 2019). Studies emphasize the importance of user-friendly interfaces, mobile accessibility, and integrated payment systems in enhancing the effectiveness of event management platforms (Bowen et al., 2016; Yeh et al., 2018).

Student Engagement and Satisfaction: Research indicates a strong correlation between student engagement in campus events and overall satisfaction with the college experience (Astin, 1984; Pascarella and Terenzini, 2005). Engaging events that cater to diverse interests and preferences contribute to a sense of belonging and connectedness among students, thereby enhancing retention rates and academic success (Kuh, 2001; Tinto, 2012).

Payment Security and Trust: The security of online payment systems in event management is a growing concern, with studies highlighting the importance of trust, transparency, and data security in fostering participant confidence (Wang et al., 2019; Cheung et al., 2020). Secure payment gateways, encryption protocols, and compliance with data protection regulations are essential for mitigating risks and ensuring participant trust in online transactions (Chen and Lu, 2015; Tan and Teo, 2016).

Overall, the literature underscores the importance of effective communication, technology integration, student engagement, payment security, and feedback mechanisms in successful event management within educational institutions. By drawing upon these insights, the College Events Hub project aims to address the key challenges identified in the literature and contribute to the advancement of event management practices within college campuses.

* 1. **EXISTING SYSTEM**

Before the implementation of the College Events Hub project, the existing system for managing club activities and events within the college was often fragmented and inefficient. Communication among club members, leaders, coordinators, and students typically relied on disparate channels such as email threads, social media groups, and physical notice boards. This decentralized approach often led to miscommunication, missed deadlines, and a lack of centralized information, resulting in frustration and confusion for all parties involved.

Event coordination and registration processes were also cumbersome and time-consuming under the existing system. Organizers had to manually collect registrations, manage attendee lists, and handle payment transactions through offline methods, leading to administrative overhead and potential errors. Additionally, there was limited opportunity for students to provide feedback or engage with event organizers beyond the initial registration process, resulting in a missed opportunity to enhance the overall event experience and foster community interaction.

Overall, the existing system for managing college events lacked cohesion and efficiency, hindering effective communication, event coordination, and student engagement. Recognizing these shortcomings, there was a clear need for a centralized platform that could streamline event management processes, improve communication channels, and enhance the overall college experience for students. This realization served as a driving force behind the development and implementation of the College Events Hub project.

## DISADVANTAGES OF EXISTING SYSTEM:

**Fragmented Communication Channels:**

* Communication is spread across various platforms such as email, social media, and physical notices.
* Lack of centralized communication leads to information getting lost or overlooked.
* Results in miscommunication, confusion, and delays in event planning and execution.
* Manual Event Coordination Processes:
* Organizers rely on paper-based forms or excel sheets for event coordination.
* Collecting registrations, managing attendee lists, and tracking payments are labor-intensive.
* Time-consuming process, especially for large-scale events, leading to inefficiencies and logistical challenges.

**Limited Student Engagement and Feedback:**

* Once registration is complete, there's limited opportunity for students to engage further with events.
* Lack of mechanisms for students to provide feedback, ask questions, or interact beyond the initial.
* Missed opportunity to enhance the event experience, gather valuable insights, and foster a sense of community among participants.

# PROPOSED SYSTEM

The proposed College Events Hub system represents a significant advancement over the existing methods of managing club activities and events within the college. At its core, the system offers a centralized communication platform that aggregates all information, announcements, and updates related to club activities and events. By consolidating communication channels into a single hub, the system reduces the risk of miscommunication and confusion among club members, leaders, coordinators, and students. This centralized approach ensures that all stakeholders have easy access to the latest information, facilitating smoother coordination and organization of events.

One of the key features of the proposed system is its automation of event coordination processes. Through automated registration, attendee management, and payment processing functionalities, the system streamlines event planning and execution. Organizers can set up events efficiently, manage attendee lists effortlessly, and track attendance accurately—all within the platform. By reducing manual effort and minimizing errors associated with traditional paper-based methods, the system enables organizers to focus more on delivering high-quality events that enrich the college experience for participants.

Moreover, the proposed system aims to enhance student engagement and interaction through its interactive features. By incorporating event ratings, reviews, and comments, the system encourages students to provide feedback, ask questions, and interact with organizers and peers. This fosters a sense of community and collaboration within the college, enriching the overall event experience and promoting greater participation. With its comprehensive event information, secure authentication and payment systems, and efficient event planning tools, the College Events Hub system offers a holistic solution for managing club activities and events, paving the way for a more vibrant and connected college community.

.

## ADVANTAGES OF PROPOSED SYSTEM

**Do Centralized Communication Hub:**

* + - The system provides centralized platform for all communication related to club activities

and events.

* + - Users can access announcements, updates, and event details in one place, reducing

confusion and ensuring consistent information dissemination.

**Automated Event Coordination:**

* + - With automated registration, attendee management, and payment processing features, the

system streamlines event coordination processes.

* + - Organizers can set up events efficiently and manage registrations seamlessly, saving time

and minimizing errors associated with manual methods.

**Enhanced Student Engagement:**

* + - The system promotes student engagement through interactive features such as event

ratings, reviews, and comments.

* + - Students can provide feedback, ask questions, and interact with organizers and peers,

fostering a sense of community and collaboration.

**Comprehensive Event Information:**

* + - College Events Hub offers comprehensive event information, including schedules,

descriptions, locations, and attendee lists.

* + - Users have easy access to all relevant details, facilitating informed decision-making and encouraging greater participation in events.

**Secure Authentication and Payment System:**

* + - The system employs secure authentication mechanisms to ensure only authorized users

have access to sensitive information.

* + - Integrated payment systems facilitate secure and convenient event registration, enhancing

user experience and increasing participation.

**Efficient Event Planning and Execution:**

* + - By streamlining processes and centralizing information, the system enables efficient event planning and execution.

.

## SYSTEM REQUIREMENTS SOFTWARE REQUIREMENTS

## 

**Node.js:**

**Version:** Node.js 14.x or higher.

**Description:** Node.js is a JavaScript runtime built on Chrome's V8 JavaScript engine. It allows for server-side execution of JavaScript code and is essential for running Next.js applications.

**Visual Studio Code (VS Code):**

Visual Studio Code is a lightweight, open-source code editor developed by Microsoft.

It provides features such as syntax highlighting, code completion, debugging support, and integrated version control.

VS Code will be used by developers to write, debug, and manage the codebase of the College Events Hub application, providing a seamless development experience.

**Next.js:**

**Version:** Next.js 14.x or higher.

**Description:** Next.js is a React framework that enables server-side rendering (SSR), static site generation (SSG), and client-side rendering (CSR) for React applications. It provides features like routing, image optimization, and code splitting out of the box.

**TypeScript:**

**Version:** TypeScript 4.x or higher.

**Description:** TypeScript is a statically typed superset of JavaScript that adds optional static typing to the language. It improves developer productivity by catching type-related errors early in the development process and enabling better code organization and documentation.

**TailwindCSS:**

**Version:** TailwindCSS 3.x or higher.

**Description:** TailwindCSS is a utility-first CSS framework that provides pre-built utility classes for styling user interfaces. It allows developers to rapidly build custom designs without writing custom CSS by hand, promoting a highly customizable and maintainable codebase.

**Stripe:**

**Version**: Stripe API.

**Description:** Stripe is a popular payment processing platform that allows businesses to accept online payments securely. It provides APIs for handling payment transactions, managing customer data, and integrating with various e-commerce platforms.

**Shadcn:**

Further clarification needed. If it's related to Shadow DOM or a specific library, additional details are required.

**Uploadthing:**

Further clarification needed. If it's a specific library or tool for handling file uploads, additional details are required.

## HARDWARE REQUIREMENTS:

## Server:

**CPU:** Multi-core processor (e.g., Intel Core i5 or equivalent)

**RAM:** Minimum 4GB (8GB or more recommended for better performance)

**Storage:** SSD storage recommended for faster read/write operations

**Client Devices:**

**Desktop/Laptop:**

**CPU:** Dual-core processor (e.g., Intel Core i3 or equivalent)

**RAM:** Minimum 4GB

**Storage:** HDD or SSD

**Display:** Minimum resolution of 1366x768 pixels

**Mobile:**

Compatible with smartphones and tablets running modern web browsers

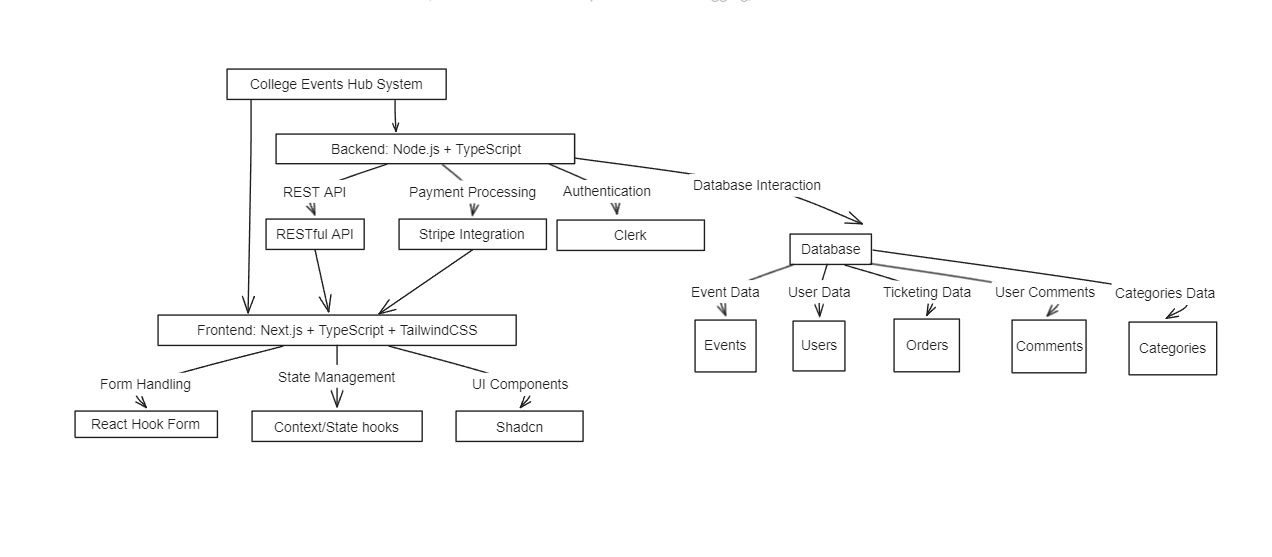
(e.g., Safari, Chrome).

Responsive design for optimal viewing and interaction on various

screen sizes

# 4 SYSTEM DESIGN

## 4.1 ARCHITECTURE

****

**Fig 3: Architecture Diagram**

In the system architecture for College Events Hub, MongoDB serves as the core database component responsible for storing and managing event-related data. The architecture comprises several interconnected elements:

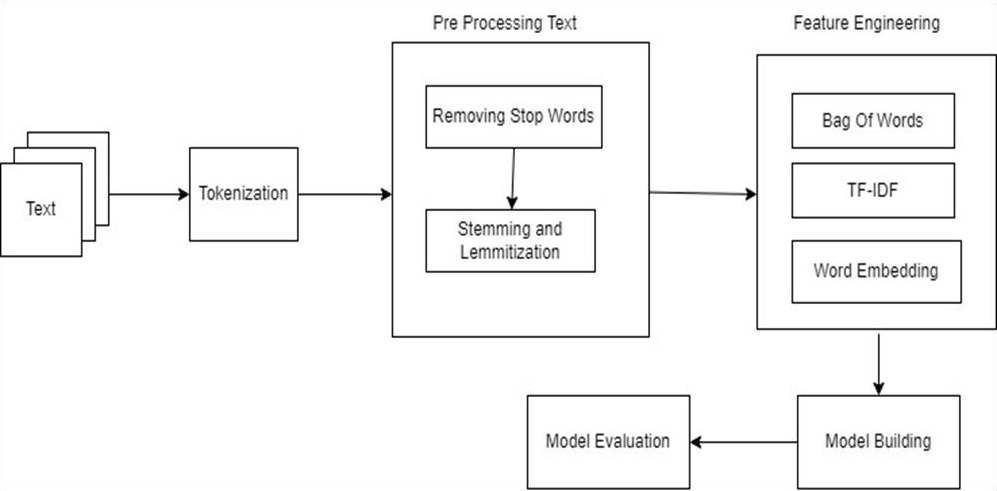
The frontend, developed using Next.js, integrates React Hook Form for efficient data management and TailwindCSS for sleek styling, ensuring a user-friendly interface. This frontend communicates with the backend, powered by Node.js with TypeScript for enhanced reliability and scalability.

Authentication is managed through Clerk, ensuring secure user access across both frontend and backend components. Stripe is integrated into the backend for seamless payment processing, while Uploadthing handles file upload tasks efficiently.

The MongoDB database component stores various types of data essential for event management, including event details, user information, and feedback submissions. Node.js facilitates communication with MongoDB, handling data storage and retrieval operations effectively.

This cohesive architecture enables College Events Hub to provide robust event management capabilities, secure authentication, seamless payment processing, and efficient file management. MongoDB's scalability and flexibility complement the other components, contributing to the platform's overall functionality and performance

## PROPOSED SYSTEM ARCHITECTURE



**Fig 4: Proposed System Architecture**

The first step is text preprocessing, which involves cleaning and transforming raw text data into a format that can be analyzed by a machine learning model. This often includes steps such as tokenization, where the text is divided into Individual words or phrases, and normalization, where the text is converted to lowercase and any unnecessary characters are removed.

The second step is feature engineering, which involves selecting or creating features that will be used by the machine learning model to make predictions. This often Includes techniques such as bag-of-words, where each word or phrase in the text is treated as a separate feature, and TF-IDF, which calculates the Importance of each word in the text.

The third step is model building, where the machine learning model is trained using the preprocessed text data and selected features. The flowchart outlines several common types of models, such as decision trees and support vector machines, and highlights the importance of hyper parameter tuning to optimize the performance of the model.

The final step is model evaluation, where the performance of the machine learning model is measured using metrics such as accuracy and F1 score. The flowchart also emphasizes the importance of cross-validation, where the model is tested on multiple subsets of the data to ensure that it can generalize to new data.

Overall, the flowchart outlines the important steps involved in a typical machine learning workflow for text data, from preprocessing and feature engineering to model building and evaluation

## MODULES

* **User Module:** In this module the User loads the dataset and take cares of theother input to feed the machine learning model.
* **Tokenization Module:** In this module the input text is split and Tokenized with a Count Vectorizer of Scikit-Learn module which eases the work of MLmodel to Learn Text.
* **Pre-Processing Module:** This module is all about removing the Stop-words (is, a, the…etc.), Which makes the ML model efficient and uses NLP methods like Stemming and Lemmatization which helps the model in the generation of test Data Output.
* **Model Building:** In this module we build the GENSIM’s Doc2Vec model, a NLP based module with Train dataset as input.
* **Result Module:** In this module the result is obtained as 3 Distracters and there Cosine relation between the answer-text and each Distracter stored in Results.csv file.

## 5. IMPLENTATION AND INSTALLATION

* 1. **ALGORITHM**

### Bag of Words

"Bag of Words" (BOW) is not an algorithm per se, but rather a simple technique used in natural language processing (NLP) and information retrieval (IR) to represent text data. Bag of Words represents a document as a "bag" or a collection of words, disregarding grammar and word order but considering the frequency of occurrence of each word. In this approach, the text is tokenized and all words are assigned a unique ID, and then a vector is constructed where each dimension represents a word ID, and the value of the dimension represents the frequency of the corresponding word in the document. The Bag of Words representation can be used as an input for many different algorithms, such as classification algorithms like Naive Bayes or SVMs, or clustering algorithms like k-means or used technique in conjunction with other algorithms for text analysis.

There are various algorithms and techniques that can be used for this task, such as rule-based methods, neural network-based methods, and statistical methods. Some popular algorithms and techniques for distracter generation include:

**Rule-based methods:** These methods rely on manually defined rules and heuristics to generate distracters. For example, a rule-based method may generate distracters by replacing certain words in the correct answer with synonyms, or by generating incorrect answers based on common misconceptions or errors.

**Neural network-based methods:** These methods use neural networks, such as sequence-to-sequence models or transformer-based models, to generate distracters. These models are trained on large datasets of question-answer pairs and use the context of the question and answer to generate plausible distracters.

Here are some examples of neural network-based methods for question and answer guided distracter generation:

* + 1. Seq2Seq Model
    2. GPT-2: GPT-2 (Generative Pre-trained Transformer 2)
    3. Transformer-XL

**Statistical methods:** These methods use statistical techniques, such as language models and probability distributions, to generate distracters. For example, a statistical method may generate distracters by sampling from a language model that has been trained on a large corpus of text. There is no one "best" algorithm for question and answer guided distracter generation, as different methods may work better for different types of questions and datasets. The choice of algorithm will

depend on factors such as the size and complexity of the dataset, the type of questions being asked, and the available computing resources.

Overall, neural network-based methods for distracter generation have shown promising results, especially for complex questions that require a deep understanding of the context. However, these methods can be computationally expensive and require large amounts of data for training.

## IMPLEMENTATION STEPS:

### STEP 1: Importing necessary modules

The first and foremost step involves importing necessary modules, libraries and packages and loading the dataset as a pandas data frame.

### STEP 2: Loading dataset using Pandas

When you load a dataset using Pandas, you read the data from a file or URL and use the Pandas library to create a Python Data-Frame that can be inspected and altered using different Pandas functions. Additionally, it entails loading training data, testing data, and result data with questions and other distractions.

### STEP 3: Creating a function to remove Punctuations

Creating a function to remove all the punctuations from the test data and making it clean. Since punctuation is difficult to process in natural English strings, we must remove it before using the strings for additional processing. Removing punctuation's from a string using translation function, using loop and replace function, using regex, the filter function.

### STEP 4: Applying the function on Train data Distracters to remove the punctuations

To remove punctuation from the train data distracters using Python, you can apply a function using the string method translate(). This function can be used to remove any specified characters, including punctuation, from a string

### STEP 5: Applying the same function on Questions of Train and Test Data

To remove punctuation from the questions in both train and test data using Python, you can apply a lambda function using the apply() method in Pandas, which uses the translate() method to remove all punctuation characters from the string. The cleaned values are then saved back to the respective 'Question' column in each Data-Frame. Performance of machine learning algorithms when they are

used to make predictions on data not used to train the model.

### STEP 6: Creating a TaggedIterator Function

By applying the function and getting the individual words from the data.

### STEP 7: Applying the Functions on Cleaned Questions and Answers of Training and Testing Data

To apply a function to the cleaned questions and answers of training and testing data, you can use the apply() method in Pandas. For example, to convert all text to lowercase in the 'Question' and 'Answer' columns of both the training and testing data, you can use the apply() method with a lambda function that calls the lower() method, and the cleaned values are saved back to the respective columns in each Data-Frame.

### STEP 8: Generating Doc2Vec model

By using the parameters vector size, window, min count, workers, alpha, epochs we built a doc2vec model.. Doc2Vec is a Model that represents each Document as a Vector. In contrast to the Word2Vec model, the Doc2Vec model gives the vector representation for an entire document or group of words. With the help of this model, we can find the relationship among different documents

### STEP 9: Building the Vocabulary of the model

Building the vocabulary such that it takes sentences built from tagged iterator.vocabulary builder is the first building block of the word2vec model. It takes raw text data, mostly in the form of sentences. The vocabulary builder is used to build vocabulary from your given text corpus. It will collect all the unique words from your corpus and build the vocabulary.

### STEP 10: Training the Model with Training Data

Giving the question, answer and distracter to the model and training it.

### STEP 11: Pickling the Trained Model

Pickling a trained language model refers to the process of serializing the model object into a binary format that can be stored in a file. This is achieved using the built-in pickle module in Python. The pickled model can later be loaded back into memory and used for various language-related tasks

### STEP 12: Creating a Function to Generate Distracter

A function to generate distracters is a piece of code that takes a question and a correct answer as input, and outputs a list of incorrect answer choices that are similar in meaning or context to the correct answer. This function can use techniques such as synonym replacement, antonym replacement, paraphrasing, or using related concepts to generate the distracters.

### STEP 13: Generating the distracters for Test Data

The knowledge used from training data and using it on test data. The function can be applied to a set of test data, typically in the form of a list of tuples containing questions and their correct answers.

The resulting distracters can be used as multiple-choice options for the corresponding questions.

### STEP 14: Appending the Distracters to Result Data

A list of resultant distracters is generated by the function.

### STEP 15: Appending answers and Distracters together

After generating distracters for a set of questions, you may want to append both the correct answer and the distracters to create a list of answer choices for each question. This can be done by combining the correct answer and distracters into a single list.

### STEP 16: Using Sklearn Count Vectorizer to Vectorize Data

Using vectorizer() method for converting text to vector form.

### STEP 17: Creating a Function to Produce a Cosine Similarity Metric

A function is created to check the similarity between the distracters.

### STEP 18: Creating another Function for Cosine Similarity which also uses above function

Using the function and generating the Cosine Similarity score for every question.

### STEP 19: Generating the Cosine Score for Test Data

Generating the cosine score for test data involves computing the cosine similarity between each test question and its corresponding set of distracters. The cosine similarity measures the similarity between two vectors in a multi-dimensional space and is commonly used in natural language processing and information retrieval tasks.

### STEP 20: Generating the Cosine Score for Result Data

Generating the Cosine score for all questions in the result data and taking the top 3 distracters. Generating the cosine score for result data involves computing the cosine similarity between each question-answer pair and its corresponding set of distracters. This is typically done using a vectorizer and a cosine similarity function.

### STEP 21: Appending the Test Score to the Test Data

Appending the test score to the test data involves adding the cosine similarity scores generated for each test question and its corresponding set of distracters to the original test data. This can be done by iterating through the test data and its corresponding cosine similarity scores and appending the scores to the appropriate question in the test data. The resulting augmented test data can be used for further analysis or evaluation of the distracter generation algorithm.

### STEP 22: Converting the Result Data into appropriate Format

Converting the result data into appropriate format involves transforming the data into a suitable format for downstream analysis or processing, such as vectorizing text data, encoding categorical variables, or scaling numerical features. The specific format and preprocessing steps depend on the requirements of the downstream application.

### STEP 23: Converting the Test Data into appropriate Format

Converting the test data into appropriate format involves preparing the data for downstream analysis or processing, such as encoding categorical variables, scaling numerical features, or vectorizing text data. The specific format and preprocessing steps depend on the requirements of the downstream application.

### STEP 24: Saving the Test Data into CSV file

Saving test data into a CSV file involves storing data in a structured format that is easy to read and analyze. Using a programming language such as Python and a library such as Pandas to create a

Data-Frame from the test data and then save it to a CSV file using the to\_csv() method. This is useful for sharing data, backing up data, or analyzing data using other software tools.

### STEP 25: Saving the Result Data into CSV file

To save result data into a CSV file, you can use Pandas in Python to create a Data-Frame and then use to\_csv() method to save it. This allows you to store the data in a structured format that is easy to share and analyze, using a standard format that is compatible with many other applications and systems.

## SOURCE CODE:

* + 1. **Importing necessary modules**

import numpy as np import pandas as pd

from gensim.models import Doc2Vec

from gensim.models.doc2vec import TaggedDocument from nltk import word\_tokenize

import re

from sklearn.feature\_extraction.text import CountVectorizer from scipy.spatial import distance

import pickle

* + 1. **Loading Dataset using pandas** train\_data=pd.read\_csv("Train.csv") test\_data=pd.read\_csv('Test.csv') result\_data=pd.read\_csv('Results.csv')
    2. **Creating a function to Remove Punctuations** punctuations = '''!()-[]{};:'"\,<>./?@#$%^&\*\_~''' def sent\_clean(my\_str): my\_str=my\_str.lower()

my\_str= re.sub("n't", "not",my\_str )

my\_str = re.sub("'s", "", my\_str)

my\_str = re.sub("'ll", "", my\_str)

my\_str=re.sub("p.m","pm",my\_str)

my\_str=re.sub("a.m","am",my\_str) no\_punct = ""

for char in my\_str:

if char not in punctuations: no\_punct = no\_punct + char

return(no\_punct)

## Applying the function on Train data Distracters to remove the punctuations

distracters=[]

for i in train\_data['distracter']: j=i.split(',')

for k in j:

l=sent\_clean(k) distracters.append(l)

* + 1. **Applying the same function on Questions of Train and Test Data** train\_data['question']=train\_data['question'].apply(sent\_clean) result\_data\_answer=result\_data['answer\_text'].apply(sent\_clean) test\_data\_answer=test\_data['answer\_text'].apply(sent\_clean) train\_data\_answer=train\_data['answer\_text'].apply(sent\_clean)

## Creating a TaggedIterator Function

class TaggedDocumentIterator(object): def init (self, doc\_list, labels\_list):

self.labels\_list = labels\_list self.doc\_list = doc\_list def iter (self):

for idx, doc in enumerate(self.doc\_list):

yield TaggedDocument(words=doc.split(), tags=[self.labels\_list[idx]])

## Applying the Functions on Cleaned Questions and Answers of Training and Testing Data

dt = list(train\_data\_answer) print(len(dt)) print(dt.extend(distracters)) print(len(dt)) total\_indx=len(dt)

docLabels = list(range(total\_indx)) print(len(docLabels))

sentences = TaggedDocumentIterator(dt, docLabels)

## Generating Doc2Vec model

model = Doc2Vec(vector\_size=100, window=10, min\_count=5, workers=11,alpha=0.025, epochs=20)

## Building the Vocabulary of the model

model.build\_vocab(sentences)

## Training the Model with Training Data

model.train(sentences,total\_examples=model.corpus\_count, epochs=model.epochs)

## Pickling the Trained Model

pickle.dump(model, open('team\_09.pkl', 'wb'))

## Creating a Function to Generate Distracter

def get\_distracter(strng): tokens = strng.split()

new\_vector = model.infer\_vector(tokens) most\_similar\_docs = []

for d inmodel.docvecs.most\_similar([new\_vector],topn=3): most\_similar\_docs.append(dt[d[0]])

return(most\_similar\_docs)

## Generating the distracters for Test Data

result\_distracters=[]

for i in result\_data\_answer: distrct=get\_distracter(i)

for et in distrct:

if et=='' or et==' ': dindx=distrct.index(et) distrct[dindx]=i

result\_distracters.append(distrct)

## Appending the Distracters to Result Data

result\_data['distracter']=result\_distracters

## Appending answers and Distracters together

allsentences = dt allsentences.extend(list(test\_data\_answer))

## Using Sklearn Count Vectorizer to Vectorize Data

vectorizer = CountVectorizer() all\_sentences\_to\_vector = vectorizer.fit\_transform(allsentences)

* + 1. **Creating a Function to Produce a Cosine Similarity Metric** def cosine\_distance\_countvectorizer\_method(s1, s2): text\_to\_vector\_v1=vectorizer.transform([s1]).toarray().to list() text\_to\_vector\_v2=vectorizer.transform([s2]).toarray().to list()

cosine = distance.cosine(text\_to\_vector\_v1, text\_to\_vector\_v2) print('Similarity of two sentences are equal to ',round((1-cosine)\*100,2),'%')

return (round((1-cosine)\*100,2))

## Creating another Function for Cosine Similarity which also uses above function

def cosine\_score(dff,df\_dstr): cosine\_scores=[]

for i in range(len(dff)):

rvw=dff[i] temp=[]

for j in df\_dstr[i]: scr=cosine\_distance\_countvectorizer\_method( rvw,j) temp.append(scr) cosine\_scores.append(temp)

return(cosine\_scores)

## Generating the Cosine Score for Test Data

test\_distracter\_scores=cosine\_score(test\_data\_answer,test\_distracters)

## Generating the Cosine Score for Result Data

result\_distracter\_scores=cosine\_score(result\_data\_answer, result\_distracters)

## Appending the Test Score to the Test Data

test\_data['cosine score %']=test\_distracter\_scores

## Converting the Result Data into appropriate Format

for i in range(len(result\_data['distracter'])): tx=result\_data['distracter'][i] tx=re.sub("\[",'',str(tx))

tx=re.sub("\]",'',tx) result\_data['distracter'][i]=tx

## Converting the Test Data into appropriate Format

for i in range(len(test\_data['distracter'])): tx=test\_data['distracter'][i] tx=re.sub("\[",'',str(tx))

tx=re.sub("\]",'',tx) test\_data['distracter'][i]=tx

## Saving the Test Data into CSV file

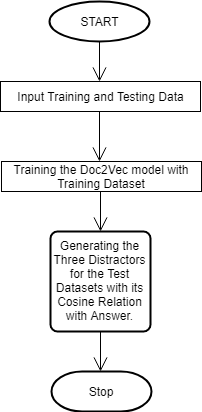
test\_data.to\_csv('Test.csv',index=False)

## Saving the Result Data into CSV file

test\_data.to\_csv('Test.csv',index=False)

## 6 DFD DIAGRAMS

* 1. **Flow chart**



### Fig 5: Flowchart of Doc2Vec classifier for Generating MCQ’s Distracters.

The first step is to gather the necessary data and split it into training and testing datasets. The training data is used to train the Doc2Vec model, while the testing data is used to generate distracter sentences.

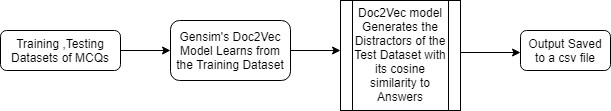
The second step is to input the training data into the Doc2Vec model and train the model. Doc2Vec is an algorithm that can be used to generate embeddings, or vector representations, of text documents.

The third step is to generate distracter sentences for the test dataset using the trained Doc2Vec model. This involves selecting the correct answer sentence for each question in the test dataset, and then generating several distracter sentences that are related to the correct answer based on their cosine similarity with the answer sentence.

The fourth step is to evaluate the performance of the distracter generation process. This Involves comparing the generated distracter sentences to the correct answer sentence and assessing their quality based on criteria such as coherence and relevance.

Overall, the flowchart outlines the important steps involved in training a Doc2Vec model and using it to generate distracter sentences for a test dataset. This process can be used to create multiple-choice questions for educational or assessment purposes, or to generate plausible alternatives for natural language generation tasks.

## Data Flow Diagram



**Fig 6: Data Flow Diagram**

### GOALS:

The Primary goals in the design of the DFD are as follows:

* + 1. Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.
    2. Provide extendibility and specialization mechanisms to extend the core concepts.
    3. Be independent of particular programming languages and development process.
    4. Support higher level development concepts such as collaborations, framework, patterns and components.
    5. Integrate best practices and provide a formal basis for understanding the modeling

## 7. TESTING

* 1. **INTRODUCTION**

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

### TYPES OF TESTS

**Unit testing**

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensurethat each unique path of a business process performs accurately to the documentedspecifications and contains clearly defined inputs and expected results.

### Integration testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the

combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

### System Testing

System testing ensures that the entire integrated software system meetsrequirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration-oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

### White Box Testing

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

### Black Box Testing

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box. you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

## Unit Testing

Unit testing is usually conducted as part of a combined code and unit test phase ofthe software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

### Test strategy and approach

Field testing will be performed manually and functional tests will be written in detail.

### Test objectives

* + - * To test every part of the code.
      * Check every part of the code is running without any compile time error.

### Features to be tested

* + - * To check the importing of modules are doing good.
      * To check the dataset is loaded perfectly.
      * To check the Visualization is done good.
      * To check for any compile time and run time errors.
      * To check every requirement is installed.

## Integration Testing

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that part of code or the piece of code,

e.g. components in a software system or – one step up – software applications at the company level – interact without error.

**Test Results:** All the test cases mentioned above passed successfully. No defects encountered

## VALIDATION

The testing process is a part of broader subject referring to verification and validation. We have acknowledged the system specification and tried to meet the customers’ requirements and for this sole purpose, we have to verify and validate the product to make sure everything is in place. Verification and Validation are two different things. One is performed to ensure that the software correctly implements a specific functionality and other is due to ensure if the customer requirements and properly met or not by the end product.

## 7.4 Closure of Testing and validation

At the end of this phase, the Model is ready to be deployed or to be given to the user for use. In other words, the Model is ready to use at the end of this phase. This means that there are no loopholes and errors in the Model and it operates without any problems. It also means that the system and its operating without any problems. It also means that the Model is assured of quality and ensured of its functionality and performance.

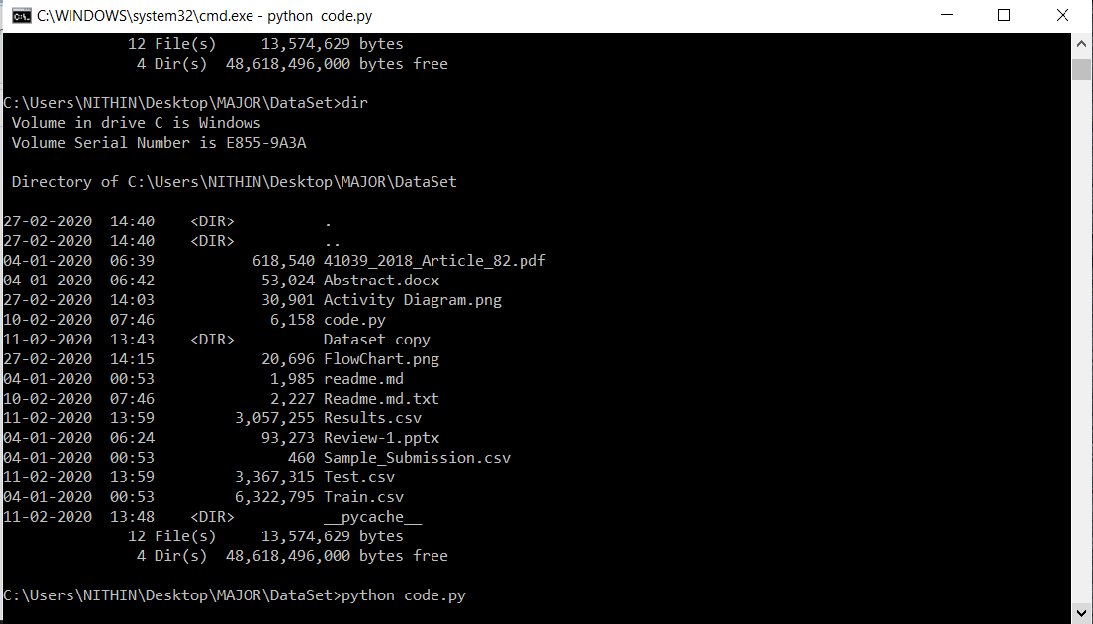
## 8 RESULTS

Here shows all the possible top 3 distracters with their near cosine similarity metric values compared with the answer given. By applying the implementation methods and techniques in order to train and test the data with the near grammar question. The distracters were created for multiple- choice questions for English-speaking students by applying semantic analysis to texts.

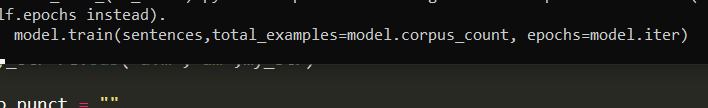
## 8.1 ACCURACY

Machine Learning model accuracy is the measurement used to determine which model is best at finding the most and the nearest distracters generated when the question and answer is answer is given. The accuracy of this model is based on the highest cosine value of the distracters generated. Most of the grammar questions produce at most 65% of accurate distracters according to the model we trained.

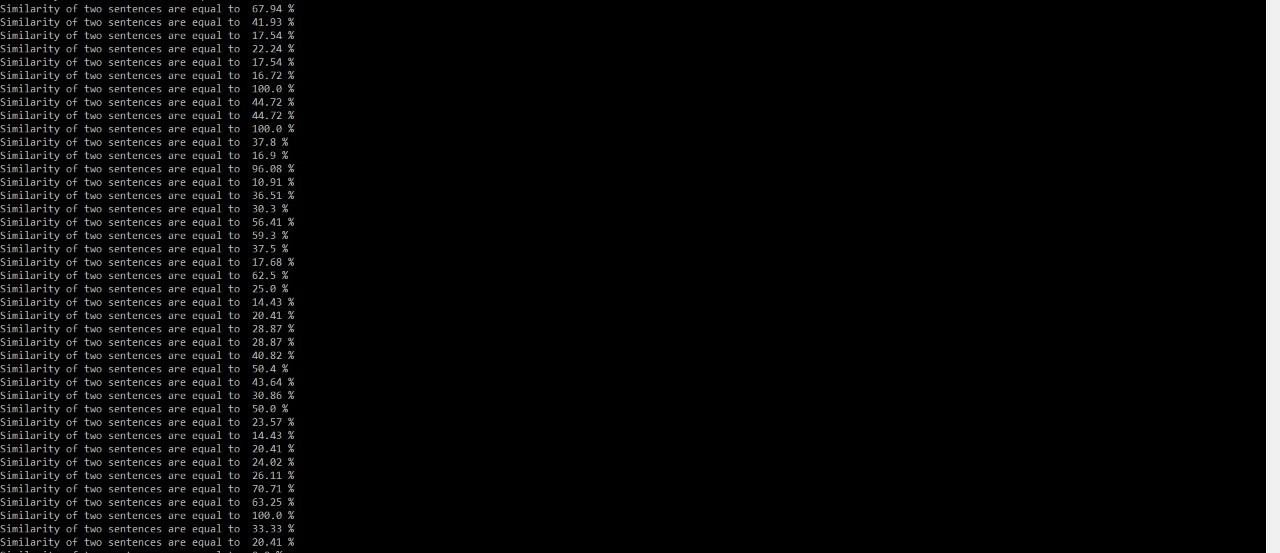
## 9 SCREEN SHOTS



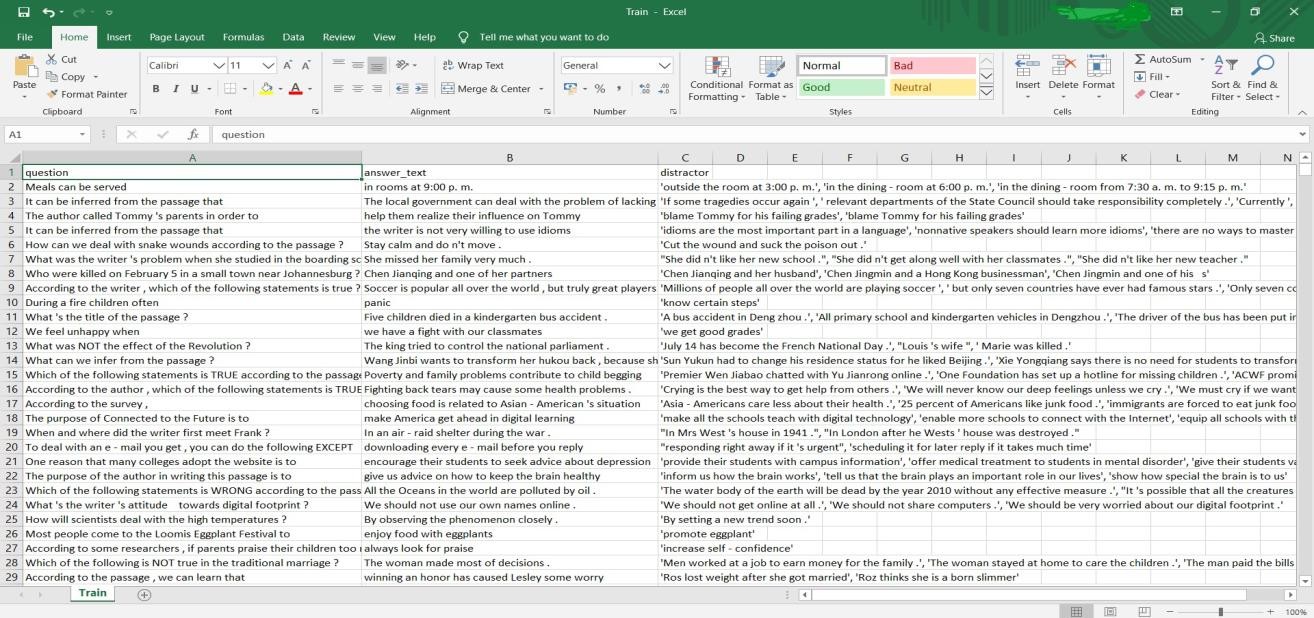
**Fig 9.a:** Running the python code in command line



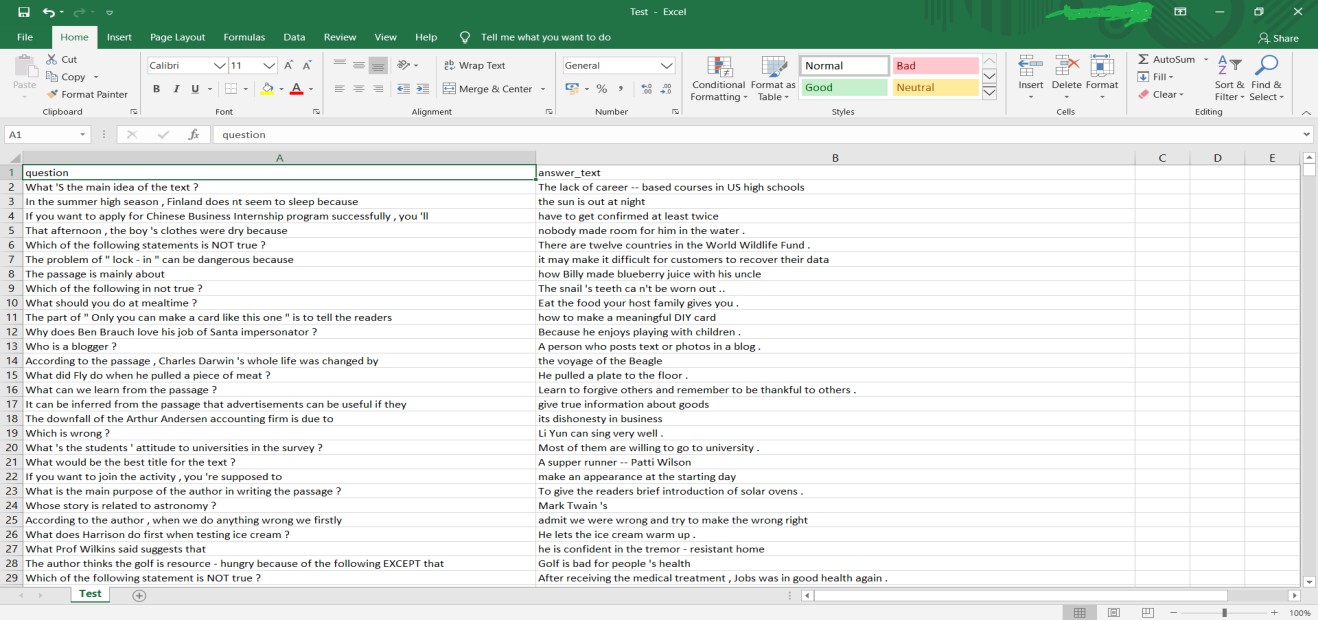
**Fig 9.b:** The above line in the Fig shows that the model is getting trained.



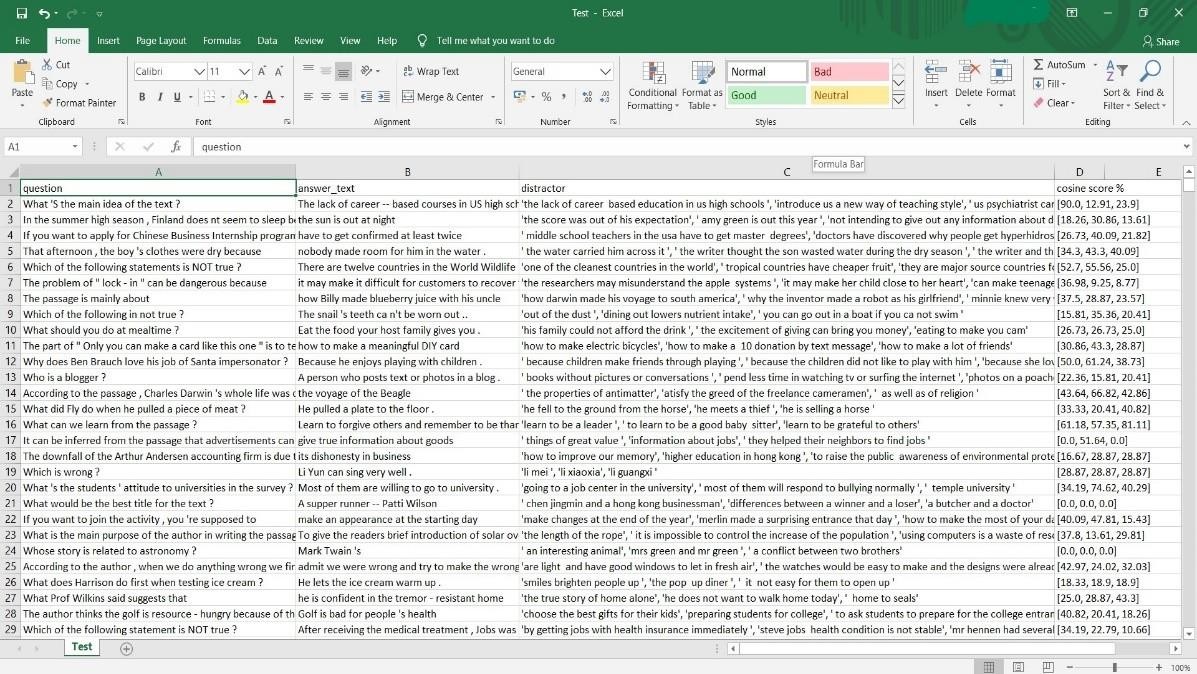
**Fig 9.c:** The above Fig shows the Cosine Similarity between Answer and generated Distracter.



**Fig 9.d:** The picture of our Training data which consists of Questions, Answers,Three Distracters separated by comma.



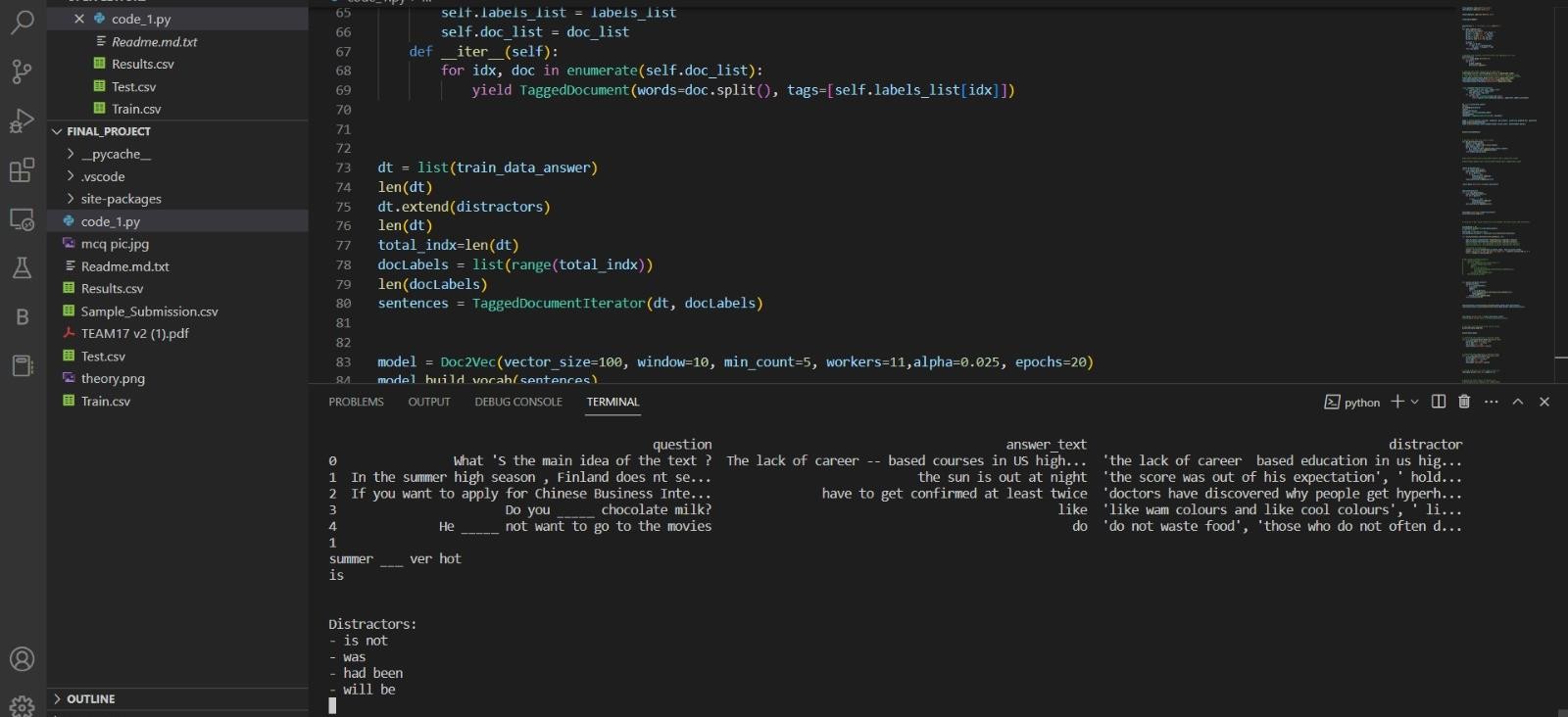
**Fig 9.e:** Test data before Processing to the model, which contains only Question and Answers.



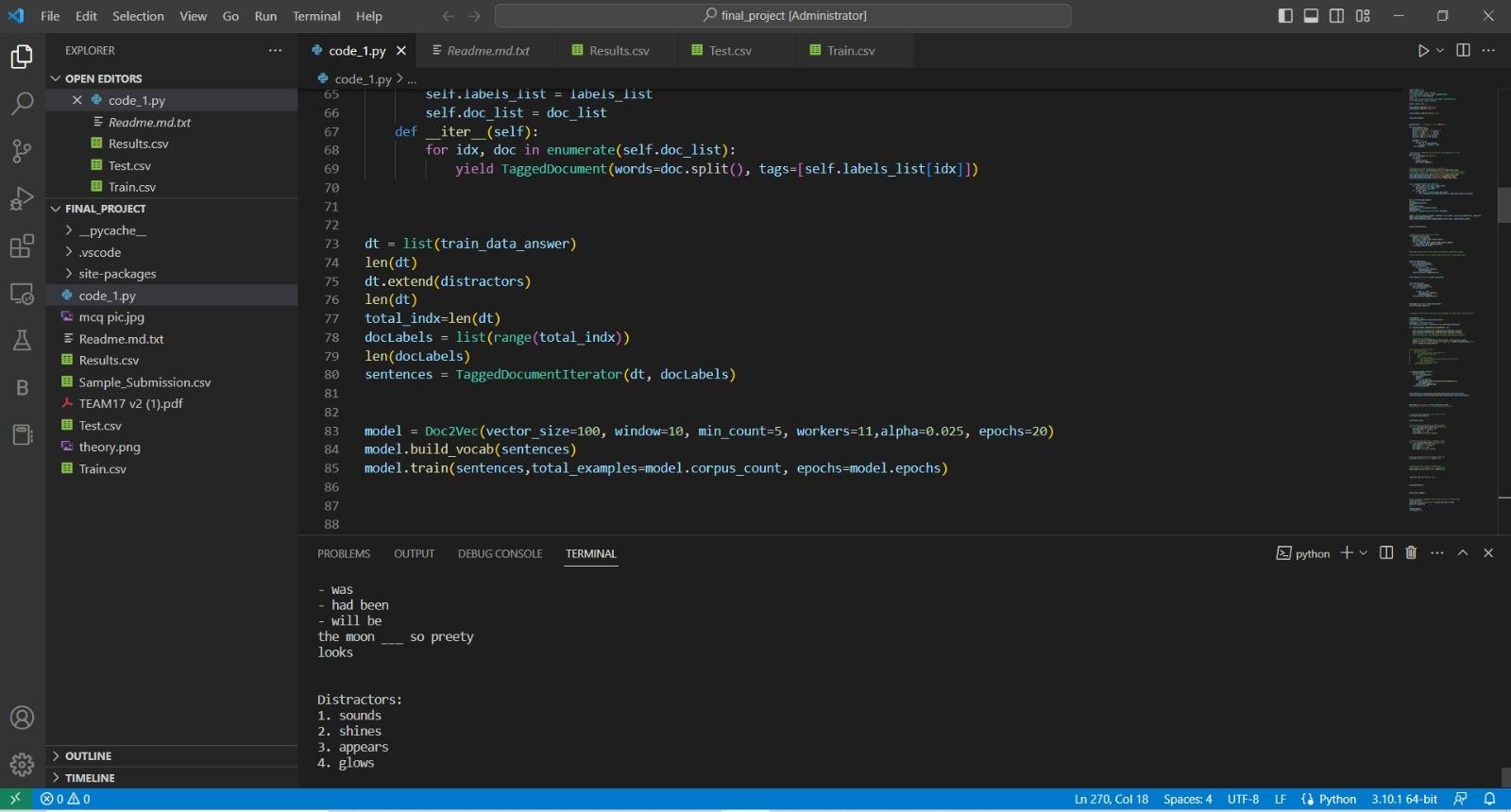
**Fig 9.f:** After Processing Test data to model, The Generated Distracters with the CosineSimilarity is Stored Back into the Test.csv file.

* Manually we can run this code my adding the question and the answers so that the distracters are generated.

Here are some of the examples.



**Fig 9.g:** Distracters generated for the question “Summer very hot.” With the answer “is”



**Fig 9.h:** Another example of generating distracters for the question “The moon so pretty.” For the answer “looks”

## 10 CONCLUSION

A study on MCQ’s Distracters Generator using NLP algorithms. A Hackathon available MCQ’s data set has been used for evaluation using individual model using GENSIM’s Doc2Vec model. The Cosine Similarity metric has been adopted as a performance measures, as it takes into account the Similarity of the given answer to the generated distracters. We have used a English language MCQ’s dataset. We made the model more efficient by removing the stop words and punctuations. The Doc2Vec model is more boon than other NLP models like Word2Vec. As our Doc2vec model came to existence to overcome the drawbacks of Word2Vec.This shows that the Doc2Vec method is stable in performance. For future work, the methods studied in this paper will be extended to online learning models. In addition, other online learning models we can also implement the model in web applications and also in android application using Tensorflow lite. This in turn will help in generating the distracters and also helps in Teaching Sector by decreasing the burden of the examination branch.

## 10.1 Future Scope

We believe that our study will have profound impact on both research and Educational communities. In the future, more advanced NLP models overcoming the Doc2Vec models can be explored for feature learning. We can automate the Model by overcoming the Drawback of our Model i.e. It Should be trained by the similar subject MCQ’S, by Fetching the Data from internet by web Crawling.

We can deploy our Machine Learning Model

 Web Application

 Mobile Application

## 11 REFERENCES

1. https://machinelearningmastery.com/natural-language-processing , Rada Mihalcea and Dragomir Radev, “Graph based natural language processing and information retrieval” [www.cse.unt.edu/~rada/CSCE5290/Lectures/Intro,](http://www.cse.unt.edu/~rada/CSCE5290/Lectures/Intro) Roxana Girju, “Introduction to Syntactic Parsing”, 2004.Geoffrey N. Leech “Semantics : The Study of Meaning” by (1974, Paperback)ISBN-10:- 13:9780140216943 , Gargesh, R.”Linguistic Perspective on Literary Style”. Delhi: Delhi University Press.1990 ,[ Wood, G.C.,” Lecture on Introduction to Semantics at the University of Sheffield”.
2. https:/[/www](http://www.geeksforgeeks.org/nlp-how-tokenizing-text-sentence-words-works).[geeksforgeeks.org/nlp-how-tokenizing-text-sentence-words-works](http://www.geeksforgeeks.org/nlp-how-tokenizing-text-sentence-words-works)
3. https:/[/www](http://www.geeksforgeeks.org/nlp-part-of-speech-default-tagging).[geeksforgeeks.org/nlp-part-of-speech-default-tagging](http://www.geeksforgeeks.org/nlp-part-of-speech-default-tagging)
4. https://towardsdatascience.com/named-entity-recognition-with-nltk-and- 8c4a7d88e7da
5. <http://nlpprogress.com/english/constituency_parsing>
6. Chang Ming-Wei, Quang Do, Dan Roth. Multilingual Dependency Parsing: A Pipeline Approach. 2006
7. https://radimrehurek.com/gensim/models/doc2vec
8. https:/[/www](http://www.educba.com/machine-learning-life-cycle).[educba.com/machine-learning-life-cycle](http://www.educba.com/machine-learning-life-cycle)
9. https:/[/www](http://www.kdnuggets.com/2018/06/top-20-python-libraries-data-science-2018).[kdnuggets.com/2018/06/top-20-python-libraries-data-science-2018](http://www.kdnuggets.com/2018/06/top-20-python-libraries-data-science-2018)
10. https:/[/www](http://www.guru99.com/nltk-tutoria).[guru99.com/nltk-tutoria](http://www.guru99.com/nltk-tutoria)
11. https://towardsdatascience.com/a-quick-introduction-to-the-pandas-python-library- f1b678f34673
12. https:/[/www](http://www.geeksforgeeks.org/regular-expression-python).[geeksforgeeks.org/regular-expression-python](http://www.geeksforgeeks.org/regular-expression-python)
13. https://machinelearningmastery.com/a-gentle-introduction-to-scikit-learn-a-python- machine- learning-library
14. https:/[/www](http://www.geeksforgeeks.org/understanding-python-pickling-example).[geeksforgeeks.org/understanding-python-pickling-example](http://www.geeksforgeeks.org/understanding-python-pickling-example)
15. Araki, J.Rajagopal, D.Sankaranarayanan, S.Holm, S.Yamakawa, Y.Mitamura, T.(2016). Generating questions and multiple choice answers using semantic analysis of texts, In Proceedings of COLING 2016, the26th International Conference on Computational Linguistics: Technical Papers,

Osaka, Japan. <http://aclweb.org/anthology/C16-1107>(pp.1125–1136).

1. Mitkov R., & Ha, L. A.(2003).Computer-aided generation of multiple-choice tests, In Proceedings of the HLT-NAACL03 Workshop on Building Educational Applications Using Natural Language Processing -Volume2, Association for Computational Linguistics, Stroudsburg, PA, USA, HLT-NAACL-EDUC’03. <http://dx.doi.org/10.3115/1118894.1118897>(pp.17–22).
2. Downing, S. M.(2006). Selected-response item formats in test development, In Handbook of Test Development (pp.287–301). Mahwah: Lawrence Erlbaum Associates, Inc.
3. Moser,J.R.,Gütl,C.,Liu,W.(2012). Refined distracters generation with lsa and stylometry for automated multiple choice question generation. In M. Thielscher & D. Zhang (Eds.), AI2012: Advances in Articial Intelligence (pp.95–106). Berlin: Springer.
4. Quoc Le, [QVL@GOOGLE.COM,](mailto:QVL@GOOGLE.COM) Tomas Mikolov, [TMIKOLOV@GOOGLE.COM,](mailto:TMIKOLOV@GOOGLE.COM) Google Inc, 1600 Amphitheatre Parkway, Mountain View, CA 94043.
5. Collobert, Ronan and Weston, Jason. A unified architecture for natural language processing: Deep neural networks with multitask learning. In Proceedings of the 25th International Conference on Machine Learning, pp. 160–167. ACM, 2008.
6. Satria, A.Y., & Tokunaga, T. (2017). Automatic generation of English reference question by utilizing nonrestrictive relative clause, In Proceedings of the 9th International Conference on Computer Supported Education (pp. 379–386).
7. Sumita, E., Sugaya, F., Yamamoto, S. (2005). Measuring non-native speakers’ proficiency of English by using a test with automatically-generated fill-in-the-blank questions, In Proceedings of the Second Work shop on Building Educational Applications Using NLP (pp. 61–68). Ann Arbor: Association for Computational Linguistics.
8. Sakaguchi, K., Arase, Y., Komachi, M. (2013). Discriminative approach to fill-in- the-blank quiz generation for language learners, In Proceedings of the 51st Annual Meeting of the Association for Computational Linguistic (pp. 238–242). Sofia: Association for Computational Linguistics.
9. Zesch, T., &Melamud, O. (2014). Automatic generation of challenging distracters using context sensitive inference rules, In Proceedings of the Ninth Workshop on Innovative Use of NLP for Building Educational Applications (pp. 143–148).Baltimore: Association for Computational Linguistics.
10. Susanti, Y., Tokunaga, T., Nishikawa, H., Obari, H. (2017). Evaluation of automatically generated English vocabulary questions. Research and Practice in Technology Enhanced Learning, 12(1), 22. [26]https:/[/www](http://www.hackerearth.com/challenges/hiring/valuelabs-ml-hiring-2019).[hackerearth.com/challenges/hiring/valuelabs-ml-hiring-2019](http://www.hackerearth.com/challenges/hiring/valuelabs-ml-hiring-2019)