*Dissertation on*

# "Automated MCQ Generation"

*Submitted in partial fulfillment of the requirements for the award of degree of*

**Bachelor of Technology**
**in**
**Computer Science & Engineering**

**UE19CS390B – Capstone Project Phase - 2**

***Submitted by:***

|  |  |
|---|---|
| **Sai Charan** | **PES1UG19CS329** |
| **Abhishek** | **PES1UG19CS022** |
| **Kundansai** | **PES1UG19CS049** |
| **Rashi** | **PES1UG20CS827** |

*Under the guidance of*

**Prof. Dinesh Singh**
PES University

**August - December 2022**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**
FACULTY OF ENGINEERING
**PES UNIVERSITY**
(Established under Karnataka Act No. 16 of 2013)
100ft Ring Road, Bengaluru – 560 085, Karnataka, India

**PES UNIVERSITY**

(Established under Karnataka Act No. 16 of 2013)

100ft Ring Road, Bengaluru – 560 085, Karnataka, India

**FACULTY OF ENGINEERING**

# CERTIFICATE

*This is to certify that the dissertation entitled*

**'Automated MCQ Generation'**

*is a bonafide work carried out by*

| | |
|---|---|
| **Sai Charan** | **PES1UG19CS329** |
| **Abhishek** | **PES1UG19CS022** |
| **Kundansai** | **PES1UG19CS049** |
| **Rashi** | **PES1UG20CS827** |

in partial fulfillment for the completion of seventh semester Capstone Project Phase - 2 (UE19CS390B) in the Program of Study - Bachelor of Technology in Computer Science and Engineering under rules and regulations of PES University, Bengaluru during the period Aug. 2022 – Dec. 2022. It is certified that all corrections / suggestions indicated for internal assessment have been incorporated in the report. The dissertation has been approved as it satisfies the 8th semester academic requirements in respect of project work.

| | | |
|---|---|---|
| Signature | Signature | Signature |
| Dinesh Singh. Prof | Dr. Shylaja S S Chairperson | Dr. B K Keshavan Dean of Faculty |

**External Viva**

| **Name of the Examiners** | **Signature with Date** |
|---|---|
| **1.** ____________________________ | ____________________________ |
| **2.** ____________________________ | ____________________________ |

# DECLARATION

We hereby declare that the Capstone Project Phase - 2 entitled **“Automated MCQ Generation”** has been carried out by us under the guidance of Prof. Dinesh Singh and submitted in partial fulfillment of the course requirements for the award of degree of **Bachelor of Technology** in **Computer Science and Engineering** of **PES University, Bengaluru** during the academic semester August – December 2022. The matter embodied in this report has not been submitted to any other university or institution for the award of any degree.

| | | |
|---|---|---|
| **PES1UG19CS329** | **Sai Charan** | [illegible] |
| **PES1UG19CS022** | **Abhishek** | Abhishek |
| **PES1UG19CS049** | **Kundansai** | A. KundanSai |
| **PES1UG20CS827** | **Rashi** | Rashi |

## ACKNOWLEDGEMENT

We would like to express our gratitude to Prof. Dinesh Singh, Department of Computer Science and Engineering, PES University, for her continuous guidance, assistance, and encouragement throughout the development of this UE19CS390B - Capstone Project Phase – 2.

We are grateful to the project coordinator, Prof. Mahesh H.B, for organizing, managing, and helping with the entire process.

We take this opportunity to thank Dr. Shylaja S S, Chairperson, Department of Computer Science and Engineering, PES University, for all the knowledge and support we have received from the department. We would like to thank Dr. B.K. Keshavan, Dean of Faculty, PES University for his help.

We are deeply grateful to Dr. M. R. Doreswamy, Chancellor, PES University, Prof. Jawahar Doreswamy, Pro Chancellor – PES University, Dr. Suryaprasad J, Vice-Chancellor, PES University for providing to me various opportunities and enlightenment every step of the way. Finally, this project could not have been completed without the continual support and encouragement I have received from my family and friends.

## ABSTRACT

The most important thing in learning is assessment and the question is crucial for assessment. Online tests are becoming more common at universities, colleges and other educational institutions. The assessment pattern is largely evolving towards objective assessment, i.e., MCQ’s, Fill in the blanks, Match the following and True or False based. Even though MCQ’s have many advantages like electronic evaluation and less testing time, exam question preparation by hand is a difficult undertaking for educators, when the time is limited. To address this issue in the development of objective assessment, this study offers an automated exam question generator.

Following the reading of an instructive line from the book, which is a true sentence, several fake phrases are created, not only by replacing the true statement’s core terminology with an antonym but also by negating it. The True and False statements are randomly rearranged to create the test questions.

In order to perform well in examinations, students must be able to recall and apply key ideas from their course material. Practice and self-evaluation through questions are essential to accomplishing this. Therefore, we introduce an automated system for Gap-Fill Question generators.

An automated MCQ Generation system is both cost- and time-effective is becoming increasingly necessary. Distractors are created using Wordnet and Sense2Vec in order to generate options for queries.

# TABLE OF CONTENTS

# LIST OF FIGURES

# CHAPTER 1

# INTRODUCTION

In the fast-growing technology today, the user's accessibility of video is growing at an exponential rate. The reason behind this is that users must watch the entire video in order to extract useful information. Due to the temporal nature of the video, the editing and administration of such big data differs from that of text. As a result, detecting anomalous, suspicious, or deviant activity in high-dimensional data is challenging and time consuming.

Exam-style questions are a basic teaching tool that can be used for a variety of objectives. Questions have the capacity to influence student learning in addition to serving as an assessment tool. Learning tools are provided, and students can learn from a variety of online sources. The manual questions from the learning materials, on the other hand, are necessary for the learner's assessment. To our knowledge, no generic assessment approach has been provided in the literature to evaluate learners' learning gaps from e-reading documents. As a result, automated question production and evaluation methodologies can aid in the assessment system's automation.

In the objective test type question, you must select the best response from a list of options or enter a phrase or word to finish a sentence.  It takes a lot of time and effort to create the handcraft assessment item. The active learning framework can, however, be used by the system for automatically generating questions. Questions with alternatives are the greatest option to gauge a learner’s level of understanding in order to make the assessment process simpler and less time-consuming. The focus of the proposed study is on creating fill-in-the-blank questions to assess a learner’s gaps in knowledge. The majority of a text’s sentences are not good candidates for good inquiries. As a result, the job of choosing informative sentences draws our attention and causes us to ask questions. To construct an inquiry or come from an informative sentence, the chosen topic-word is dropped.

Generating False sentences from True sentence, among other educational-purposes questions, can produce valid evaluations directly, easily, and efficiently which is helpful to determine whether the students have any misconceptions about the subject matter. Instead of using an interrogative language to define the T/F question, we will instead use a declarative sentence in this essay. However, the paucity of training data and difficulty in extracting appropriate testing points from a particular passage, as well as the high frequency of grammatical and semantic errors, prevent the quality of the generated questions from being adequate for evaluation.

Given an instructive text, our framework aims to find an effective way to extract summary out of it and generate questions, i.e., MCQ's, Fill in the blanks, Match the following and True or False.

# CHAPTER 2

## PROBLEM STATEMENT

Natural Language Processing is the discipline of AI that allows computers to understand text in the same manner that humans do. Automatically generating multiple-choice questions is an important yet difficult issue in Natural Language Processing. It is the problem of creating correct and relevant questions automatically from textual input. Our problem statement is to make a model making use of NLP to generate relevant MCQ's to test the user's understanding.

Including the Education industries, many organizations were affected during the Covid-19 pandemic. All the educational institutions changed their curriculum, and everything was made online, to make it easier for students and giving access, the live videos of classes are being uploaded on drive. But in some places where the students have network connectivity issues, it's hard to follow up the classes. Student's attention is going to change as the environment is different at home and there's a decorum in a college. It's a time-killing process to go through the videos at the end time for revision instead of having a summary of the video helps.

In Education evaluation, Automated Question Generation plays an important role. Handmade question design requires a lot of time, effort and money, and manual answer evaluation is likewise time consuming. To solve this issue, we build a model which aims to generate relevant questions automatically and it even makes the diversions in the options for MCQ's to test the learners. The objective questions allow students to choose the best response from four alternatives to finish a statement.

Test items can be divided into two general categories in which objective items, which ask students to choose the right response from a range of options or to add a word or phrase to a question or finish a sentence items that allow the learner to express themselves subjectively or in an essay; and organize and deliver a unique response. Multiple choice, True or False, Match the following and completion are examples of objective items, whereas subjective things include prolonged response, brief answer, performance, problem solving testing items in which one or more may be used for instructive reasons.

# CHAPTER 3

# LITERATURE SURVEY

In this chapter, we present the current knowledge of the area and review substantial findings that help shape, inform and reform our study.

## 3.1 Video Summarization

### **3.1.1** Automatic video summarization with Timestamps using NLP text fusion Published [1]

-Ahmed Emad, Fady Bassel, Mark Refaat, Mohamed Abdelhamed, Nada Shorim, Ashraf AbdelRaouf

**Year of Publication:** 2021

**Description**

The main idea of this research paper is to generate summarization and timestamps for videos automatically which reduces time. By using timestamps, we can save a lot of time for users who go through videos that have unwanted data, these timestamps help the users to find and watch clips that have more information.

Main goal is extraction of keywords, extracted keywords helps finding videos with significant videos keywords and summarizing videos depending on frames, emotions and speech. Firstly, the video content appears in the frame and the output is a summarized text for the input. The emotion and how it changes during a specific period merged with the output summarization of the frames. The audio transcribing into text occurs and output is summarization of the audio track. Finally, the fusion happens between all summarizations (audio, video, emotion) using NLP technique, Techniques such as tokenization, sentence segmentation, lemmatization & stemming, and then abstract summarization.

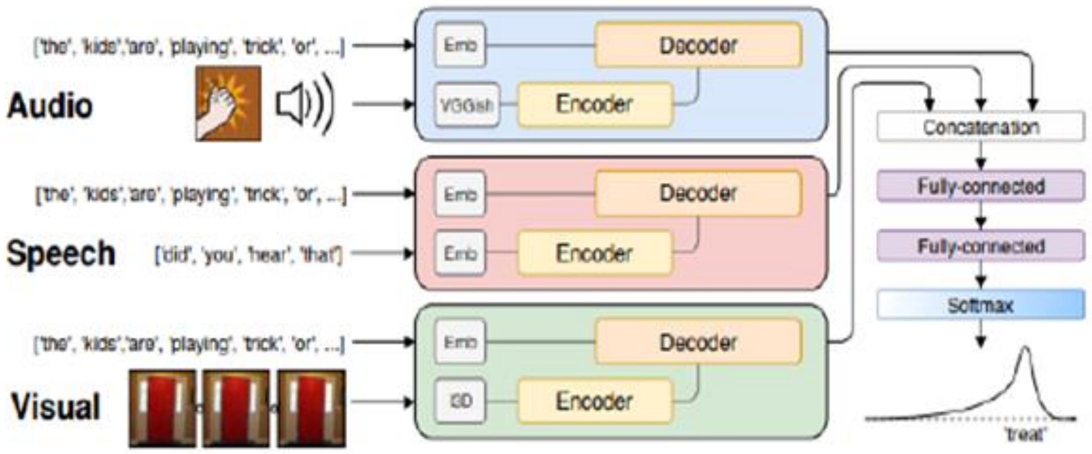

['the', 'kids','are', 'playing', 'trick', 'or', ...]
Audio
Emb
Decoder
VGGish
Encoder
['the', 'kids','are', 'playing', 'trick', 'or', ...]
Speech
['did', 'you', 'hear', 'that']
Emb
Decoder
Emb
Encoder
['the', 'kids','are', 'playing', 'trick', 'or', ...]
Visual
Emb
Decoder
I3D
Encoder
Concatenation
Fully-connected
Fully-connected
Softmax
'treat'

Figure 3.1: System Overview

**Advantages:**

- It reduces time consuming for searching proper content in video.
- It gives a more accurate summarization of the video, nearly 87%.

**Disadvantages:**

More work should be applied in text fusion using NLP techniques for linking words and removing duplicates to give smoother and more accurate sentences.

## **3.1.2** Text summarization and paraphrasing [2]

-Chi Zhang, Shagan Sah, Thang Nguyen, Dheeraj Peri, Alexander Louiy, Carl Salvaggio, Raymond Ptucha

**Year of Publication:** 2017

**Description:**

This research speaks about the sentence to vector encoding frame, the vector representation is extracted from the encoder - decoder model which is trained on sentence paraphrase pairs. The two important aspects of video summarisation are Sentence Paraphrasing and Paragraph summarization which are useful to process text. We pass each frame of video through CNN and RNN techniques which generate vector representation of words.

The approach they used is SoftMax training in which we computed logits of every training example in the word level, we pick a small set of sampled classes according to a chosen sampling function. In our model we use RNN encoder and LSTM cells since it is easy to implement and performs as well. For encoding a sentence, the embedded words are iteratively processed by the LSTM cell. A set of candidates C is created containing the union of the target class and the sampled classes. These vectors are fed into a hierarchical encoder and then the summarized text is generated word by word in an RNN decoder. The first LSTM layer serves as a filter, and it will explore local temporal structure within subsequence. The second LSTM learns the temporal dependencies among the subsequences.

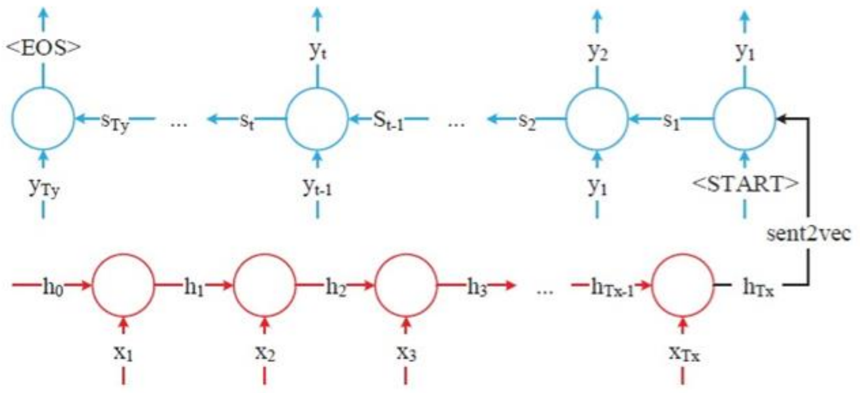


Figure 3.2: The paragraph summarizer.

The red and blue cells represent encoder and decoder respectively. The encoder inputs xi are the vector representation generated using sent2vec of the sentences in the paragraph. The decoder outputs Yi are the words in summarized text. The intermediate vector in black (paragraph2vec) is a vector encoded paragraph.

**Advantages:**

It's applicable to a variety of tasks like in academics, offices, exams.

**Disadvantages**:

Accuracy might drop if we increase the length of the paragraph.

### **3.1.3** Automated Video program summarization using Speech Transcripts [4]

-Cuneyt M. Taskiran, Member, IEEE, Zygmunt Pizlo, Arnon Amir, Senior Member, IEEE, Dulce Ponceleon, Member, IEEE, and Edward J. Delp, Fellow, IEEE

**Year of Publication:** 2006

**Description:**

In this research paper we follow a method to generate automatic video summaries using transcripts given by automatic speech recognition. We take the full program and divide into segments based on pause detection and get a score for each segment, based on the frequencies of the words and bigrams it has. Finally, a summary will be generated by selecting the segments with the highest score to duration ratios simultaneously maximizing the coverage of the summary over the full program.

Our summary approach measures both concepts and maximizes your total amount of both details and coverage function that combine to find the trade between those both. This method allows the user to change the weight and reproduce the video summary of the program with additional details or additional coverage, depending on the specific application. One of the key features of video summaries is that students can skip parts of the video lesson they are familiar with, instead, focusing on new things. This is usually done by first dividing the program into a number of segments. Then, the segments are selected to summarize by obtaining the necessary result for each segment of the program and selecting the high-scoring segments, or by combining the same segments together.

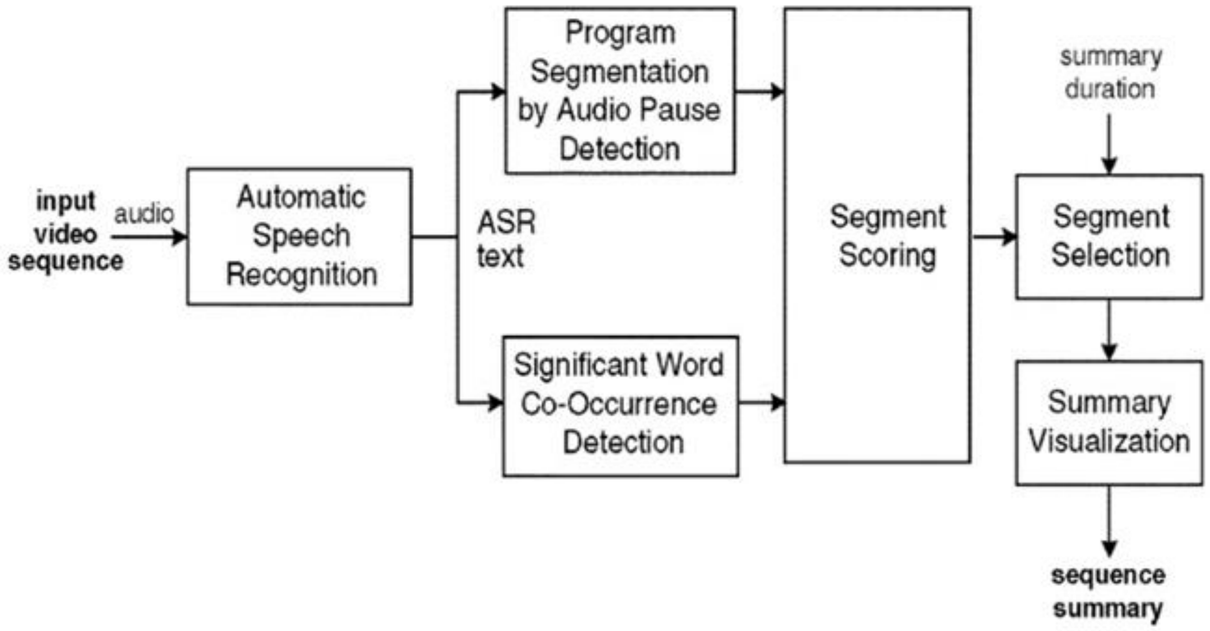


Figure 3.3: Block diagram of the proposed video summarization system.

**Disadvantages:**

The processing is generally domain specific, and a new set of events and event detection rules must be derived for each application domain

### **3.1.4** An Audio-Video Summarization Scheme Based on Audio and Video Analysis [5]

-Marco Furini,Vittorio Ghini

**Year of publication**:2005

**Description:**

In this paper the proposed idea for summarization of videos is done with two main techniques, one uses only video analysis to generate summaries and the other uses both audio and video analysis to generate summaries.

In video summarization we produce summaries with different heuristic algorithms which control the jerkiness of the video. The mean opinion score determines the quality of the produced summaries. Exact scope of the video will not be summarized if we don't consider the vocal instructions given by the lecturer.

The audio-video summarizer is based on 3 important features (temporal reduction to remove unwanted parts, audio continuity and video continuity). We must come up with a good trade-off between temporal reduction and video continuity which is lost due to trimming of video. The algorithms used for this purpose are ALL,3X,2X. We also use silence detectors algorithms (RAT) to remove silent parts of the video. We do this by having a threshold limit, if the voice is below that limit, we remove that audio frames and video frames accordingly. From the three algorithms listed above ALL algorithms work the best.

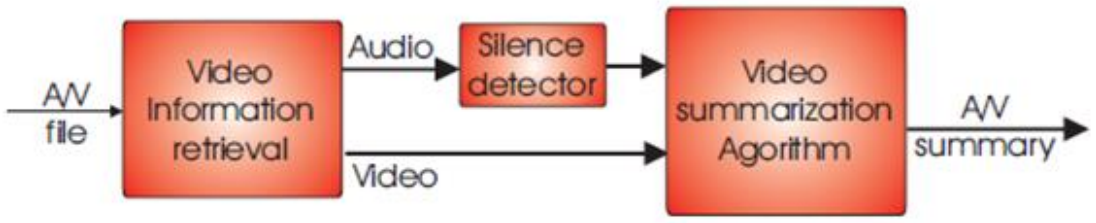


Figure 3.4: Audio Video Summarization Scheme.

**Advantages:**

Based on the user requirements we can adjust between temporal reduction and video continuity

**Disadvantages:**

Video reduction might remove contents which might be crucial or reduce the scope of the video.

### **3.1.5** AUDIO DATA SUMMARIZATION SYSTEM USING NATURAL LANGUAGE PROCESSING [6]

**-**Pravin Khandare, Sanket Gaikwad, Aditya Kukade, Rohit Panicker, Swaraj Thamke

**Year of publication**:2019

**Description:**

This paper provides techniques for converting speech audio file to text file and summarization of the text file. Summarization is performed by assigning weights according to the number of occurrences of each word in the file.

So, in this paper we first do the speech to text conversion and then we do the summarization of text. Speech to text conversion is done with the help of python modules(pyaudio). The final audio is in the format of .wav, as it is easy to process further. The audio file is divided into chunks of processable file using a mathematical formula considering factors like frame rate, duration, bit-rate etc. Text summarization is done using an open-source library called the spaCy. First step of summarization is tokenization of words by breaking off sentences and generating frequency of the words produced and normalizing it. Next, we generate scores of each sentence in the data by using the frequencies. From the data produced we select top-N sentences which have more weight and create a list with sentences according to their sentence scores. Finally, we join all the sentences and produce a finalized summary.

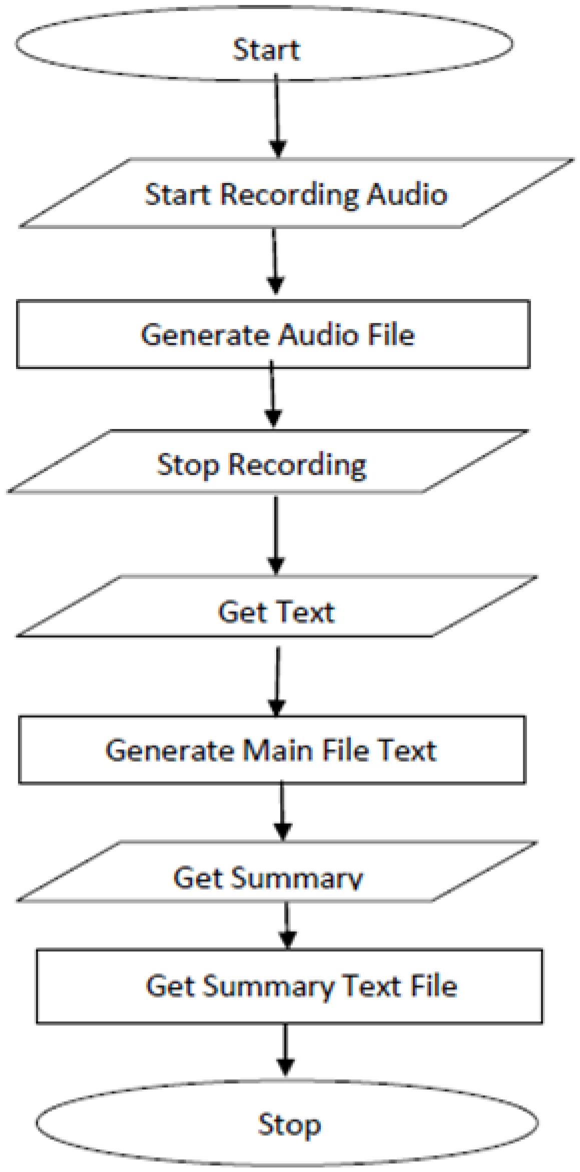


Figure 3.5: Flow chart of data summarization

**Advantages:**

- Time efficient.

- Quickly create a summary compared to manually making notes.

**Disadvantages**

- Using google speech to text recognition requires internet connection for converting to text.
- No lecture audio is 100% clear as they have some amount of deviation which affects the accuracy of the model.

## 3.2 MCQ Generation

### **3.2.1** Automated MCQ generator using NLP [3]

-Pritam Kumar Mehta, Prachi Jain, Chetan Makwana, Dr. C M Rau

**Year of publication**: May 2021

**Description:**

In this paper the text is first summarized using the BERT algorithm and accordingly sentence mapping is done for generating MCQ’s. In order to generate choices for questions distractors are generated using Wordnet which is a lexical database for English.

In the BERTSUM model, at the start of each sentence, a [CLS] token is added, and between every two sentences, a [SEP] token is added to separate the sentences. In the BERTSUM model, each sentence is assigned an embedding of Ea or Eb depending on whether the sentence is even or odd. If the sequence is [s1, s2, s3] then the segment embeddings are [Ea, Eb, Ea]. This way, all sentences are embedded and sent into further layers. BERTSUM assigns scores to each sentence that represents how much value that sentence adds to the overall document. So, [s1, s2, s3] is assigned [score1, score2, score3]. The sentences with the highest scores are then collected and rearranged to summarize the input text. In the output, the precise and summarized text is generated Keyword extraction is done by python key extraction library RAKE Generation of distractors is done by using WordNet API.

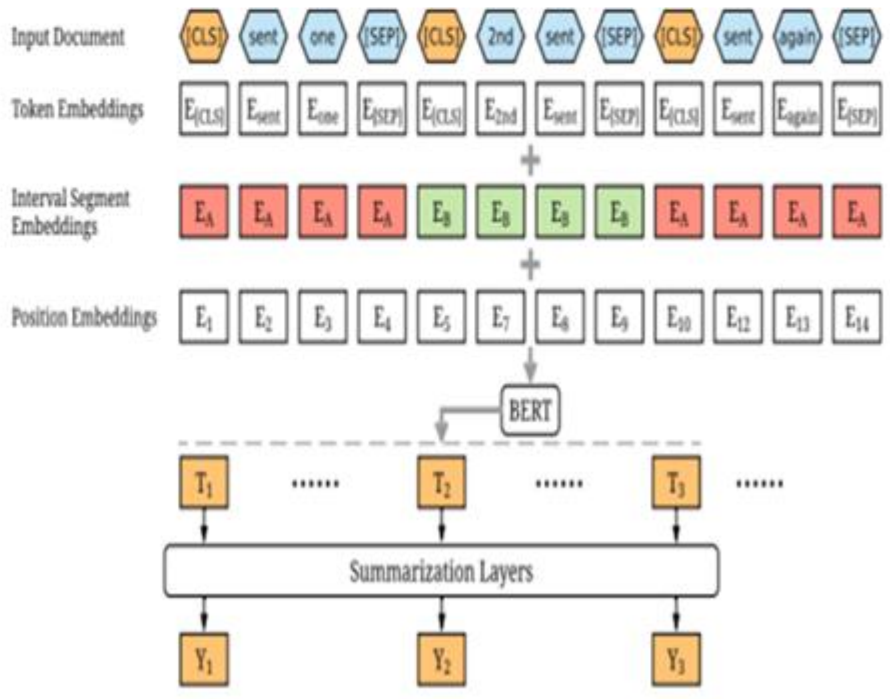


Figure 3.6: Overview Architecture of BERTSUM model

**Advantages:**

BERT algorithm has much better performance over other legacy methods as well as it can process a large amount of data in less time. It will enhance the speed of generating MCQs from the given text

**Disadvantages:**

The model is large because of the training structure and corpus. It is slow to train because it is big and there are a lot of weights to update.

## **3.2.2** Automatic MCQ Generator [7]

-Dr. S.Muthu Sundari, Vishwa A, Srinivasa kiruthik K S, Sukesh Raj R

**Year of publication**: 2021

**Description:**

Our proposed idea of producing MCQs with software can help those types of organizations and institutions to evaluate in a simple and timely manner, compared to the traditional method where we produce MCQ's through brute force which is a tedious and time-consuming task.

In this paper we have 3 different fields to generate MCQ's (questions,key,false option).To make our MCQ's more tricky and comparable to human made MCQ's,they should be properly analyzed and accurate and with a well-defined structure. We train the model such that the model is able to improvise by itself and change according to the trends and patterns. The whole methodology can be grouped into 4 parts. The first part is interaction with data which is provided by the user, the second phase is processing, analyzing and extraction of key words from the data provided, then in the next phase sentences for the keywords are mapped accordingly and the last phase is where false option is produced which together creates a standard concise MCQ's.

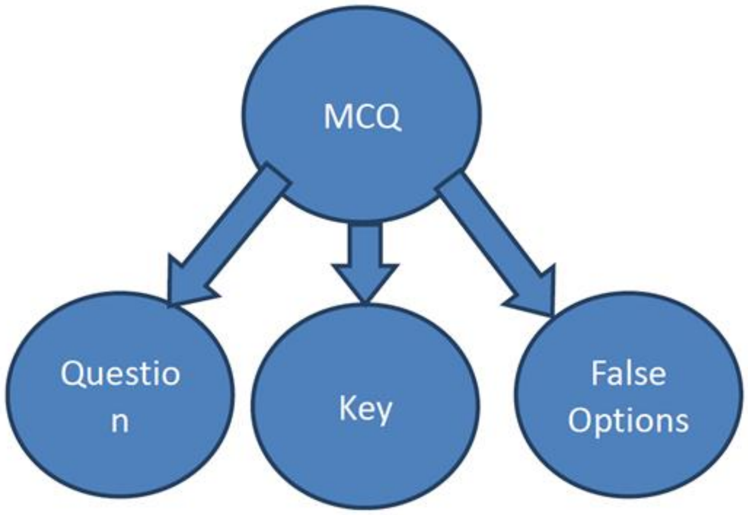


Figure 3.7: Composition of MCQ

**Advantages:**

- Reduce workload drastically.
- Saves time.
- Easy to interact with our software.

**Disadvantages:**

- The input data has to be accurate with no errors.
- The data should not have any duplications or similar MCQ's might get generated.

## 3.3 Fill in the Blanks

### **3.3.1** Learning to Automatically Generate Fill in the Blanks Quizzes[8]

-Edison Marrese-Taylor, Ai Nakajima, Yutaka Matuso

**Year of publication**: 2018

**Description:**

This Paper Fill in the Blanks is produced utilizing two distinct methods. Question Generation via Automation as "Sequence Labeling". We will create a single hot vector of size for the specified class for an embedded input. The vector we obtained is a list of binary classes; the positive class is represented by a single item.

In this instance, we employ bidirectional LSTM

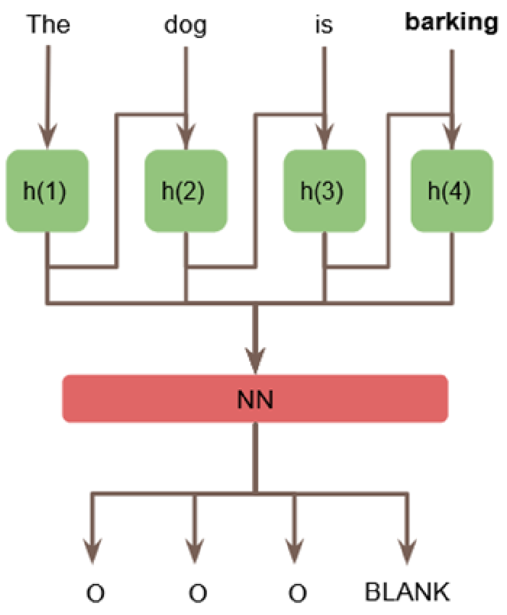


Figure 3.8: Our sequence labeling model based on an LSTM for AQG

Automatic Question Generation as Sequence Classification. We employed an attention-based methodology that produces dictaphonemes of varied sizes as a SoftMax outcome. After doing bidirectional to get dense representations for each input, using a pooling method like max and min to obtain the input sequence's outline or only the most recent hidden state. Drop and replace is adequate, after which we will add a global attention layer based on content so that we can distinguish between each concealed state by obtaining its vector outline.

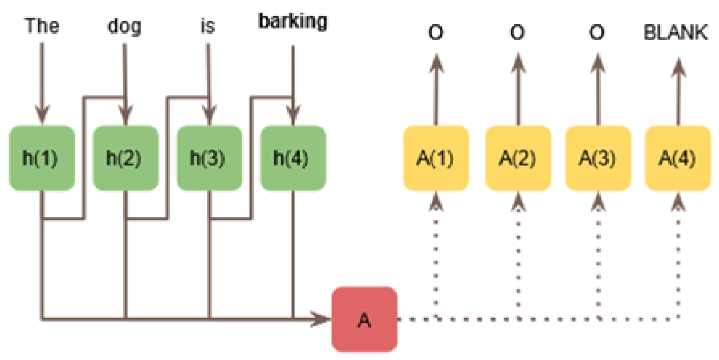


Figure 3.9: Our sequence classification model based on LSTM for AQG

**Advantages:**

It is a more effective method of creating fill-in-the-blanks than the conventional, labor-intensive human method.

**Disadvantages:**

Only 90% of the time is it correct, thus we cannot rely on it totally.

## 3.4 True or False Questions:

### **3.4.1** Generation of Multiple True or False Questions[9]

-Regina Kasakowskij , Thomas Kasakowskij, Niels Seidel

**Year of publication**: 2022

**Description:**

The True or False in this paper is created in two processes. First, we constructed a pipeline utilizing natural language processing to produce true assertions from text extractions. Next, we used created pre-trained transformer model to produce an infinite amount of false claims (GPT-2)

Summarizing the text:

We employ a summarizer to extract pertinent statements from the textbook material in order to generate true claims. The input text is first pre-processed by handling special characters, separators, and resolving abbreviations. The Summarizer then searches the text for significant part-of-speech tags (verbs, adjectives, and nouns) and eliminates stop words (such as "and," "in," and "about," as well as terms that depend on their context, such as "for example," "figure," and "section."

The word frequencies of the filtered part-of-speech tags are calculated in order to find the most significant words and, consequently, the text's topic. The importance of each statement is assessed using them as a weighted. When a phrase includes several terms that recur often in the text, it is considered as more relevant. For the summary, roughly 10% of the most relevant sentences are taken into consideration.

False statements are generated: They must be split to generate an alternation of the true and false assertions from the chosen statements. The GPT-2 Pipelines software was used to reduce the statements to two-thirds of their original length and finish them into a single new sentence. The GPT-2 algorithm can identify the subject area and create appropriate sentence ends more precisely the more of the phrase is used.

**Advantages:**

It is more effective method of creating the false statement from the true statements than the conventional labor-intensive method

**Disadvantages:**

Only 15% - 57% of all true-false statement combinations were appropriate. Therefore, it was possible to use 33% of all statement combinations on average without any hesitation.

# CHAPTER 4

# Methodology

This chapter contains an in-depth description of the approach used to solve the current issue.

## 1. Overview

The methodologies general framework is to produce pertinent questions from literature to assess the user's comprehension. We created a variety of questions using middle school literature as the input, including Multiple Choice Questions, False from True assertions, Match the Following and Fill in the Blanks.

Data Preprocessing is crucial since the correctness of the project, which determines the ultimate result, is directly reflected in it. Therefore, it is crucial to give data pre-processing some time. The preparation of the data includes formatting the necessary features and removing the unnecessary information.

## 2. MCQ Generation

There are two main steps that must be used in order to construct a MCQ question.

- Creation of Distractors
- Creation of Questions

Keyword extraction is done by FlashText which is a Python library designed primarily for word replacement and searching in documents. Now, for FlashText to function, it needs a word or list of words as well as a string. The string is then searched for or substituted with the terms that FlashText refers to as keywords.

Distractors Generation:

In a Multiple Choice Question, the incorrect responses are known as Distractors. The complexity of the question increases as the relevance of the distractors to the correct solution increases.

The distractors should have certain qualities, like as:

- They shouldn't be excessively alike or dissimilar. Distractors' content ought to be uniform.

  Distractors of Red are:Pink, Blue, Watch, Green.

- Distractors ought to be exclusive of one another. Therefore, avoid using synonyms.

We employ a variety of methods to produce the Distractors for the Multiple Choice Questions, and we select two effective algorithms.

(1) WordNet

This is a lexical database which contains the meaning relationships between words in more than 200 different languages. Synonyms, Hyponyms and Hypernyms are only a few of the semantic relationships that WordNet links word into. It labels the semantic relationship between the words and captures relationships between the words.

Specific distractors are produced by utilizing semantic relations, which are hypernyms and hyponyms found in the wordnet. A hypernym and its hyponym have a certain kind of relationship. A hypernym is an all-encompassing term.

**Disadvantage of WordNet:**

Pure simple words are difficult to distinguish from multi-word structures like collocations, verbal phrases, frozen expressions, idioms etc.

(2) ConceptNet

A free multilingual knowledge graph is ConceptNet. Both expert created materials and crowd sourced expertise are included. Word embeddings can also be made with ConceptNet, exactly like Word2Vec. ConceptNet marks the conceptual connections between words in a manner similar to WordNet. Compared to WordNet, it's more thorough. The idea includes relationships between terms such as "is a", "made of", "similar to", "is used for" and others.

Distractors for things like places, things etc. that have a "Part Of" relationship can be produced well using ConceptNet. A state like "California", for instance, might be a component of the US. There is no way to distinguish between various word senses in ConceptNet with a particular term, we must proceed with whatever sense it provides.

**Disadvantage of ConceptNet:**

Conceptual information is not captured.

(3) Sense2Vec

A Neural Network model called Sense2Vec creates vector space representations of words from significant corpora. This algorithm is an improvement on the infamous Word2Vec one. Instead of producing tokens of words, Sense2Vec produces embeddings for "senses". Information about the context is recorded. Since the outcomes were a lil better, we will use trained vectors from 2015 rather than those from 2019.

In contrast to some word vector algorithms, which are taught with only single words, noun phrases and named entities are annotated during training, allowing for the inclusion of multiword phrases like "Natural Language Processing" as well.

**Disadvantage of Sense2Vec:**

We'll come across a lot of almost identical terms and misspelled words that should be filtered out. We come upon further copies, but you just need to keep one of them as a diversion.

(4) Sentence Transformers

Sentence Transformers is a Python framework for cutting-edge embeddings of text, sentences and images. This framework may be used to construct sentence and text embeddings for more than 100 different languages.

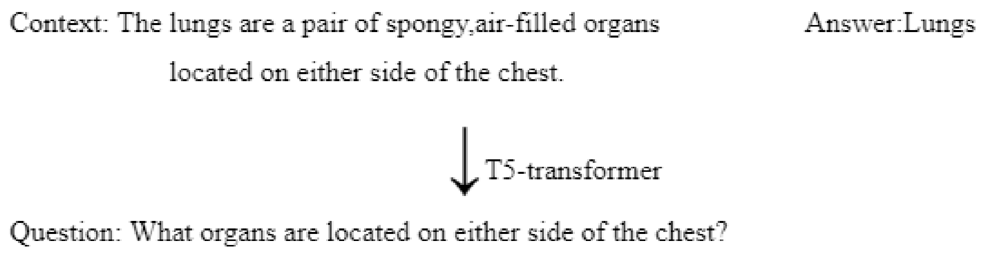
Along with the similarity of already chosen keywords and key phrases, MMR (Maximal Marginal Relevance) takes the document's keywords and key phrases into account. As a result, the keywords chosen are as diverse as possible within the context of the content.

## Question generation:

T5 Transfer Model

Text to Text Transfer Transformer, often known as T5, is a transformer-based architecture that employs this method. Every task is framed as feeding the model text as input and training it to output some goal text, including translation, question answering and classification.

The T5 model will be trained to produce formal queries. We provide a context which may be a sentence or a paragraph- as well as a response, and the T5 model uses this information as input to create a question.



Context: The lungs are a pair of spongy,air-filled organs located on either side of the chest.

Answer:Lungs

Question: What organs are located on either side of the chest?

Thus, given a situation in which a sentence or paragraph is provided along with its solution, our T5 model then generates the question/s you can see above. Transfer learning involves pre-training a model on a task with lots of data before fine-tuning it on a subsequent task, has become a potent method for Natural Language Processing.

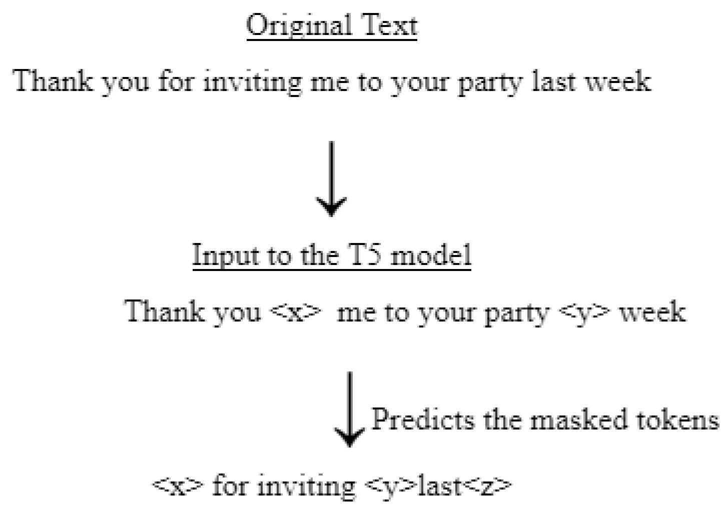
Models should be trained with general tasks first, then they should be fine-tuned with small amounts of text to perform particular tasks.

T5 views every task processing issue as a "text-to-text" issue, where the text is the input and new text is produced as the output.

A single T5 model is capable of carrying out numerous tasks, including summarization and translation. Simply alter the prefix and specify the action to take.

T5 introduces the "Colossal Clean Crawling Corpus", a dataset made up of a large amount of clean English text that was downloaded from the internet. Code, little bullet points, colons are removed. New models like transformers outperformed RNN. Self-attention, which analyzes a sequence of substituting each element with a weighted average of the remainder of the sequence, is the main component of the transformer.

Utilizing Cross-Entropy loss, T5 is trained. T5 is a masked language modeling algorithm, or ir is trained on a denoising objective, where the model is trained to anticipate corrupted or missing tokens in the input. Input is a statement with a few words masked out. T5 employs a greedy algorithm during testing (i.e, choosing the highest probability logit at every timestamp).

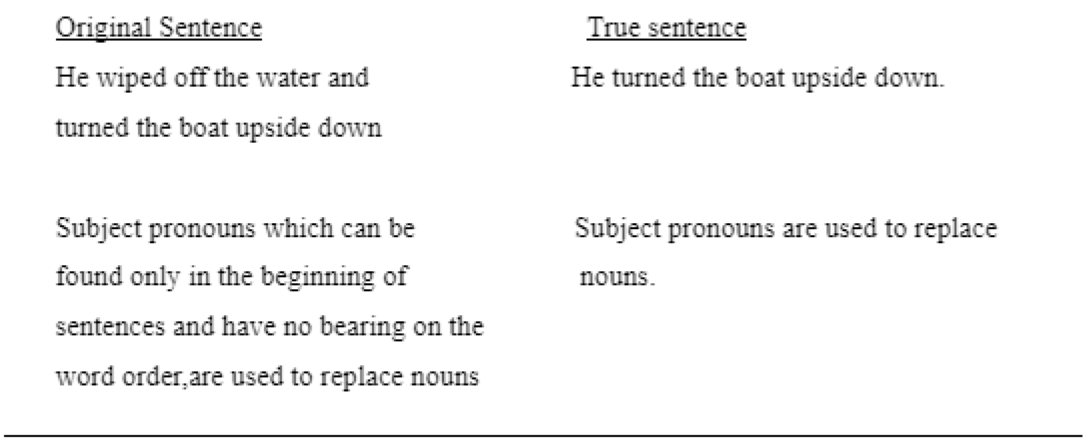

Original Text
Thank you for inviting me to your party last week
Input to the T5 model
Thank you <x> me to your party <y> week
Predicts the masked tokens
<x> for inviting <y>last<z>

## 3. Generate False statements from True statements

Using Constituency Parsing and OpenAI GPT2, we will create True or False questions based on the input text.

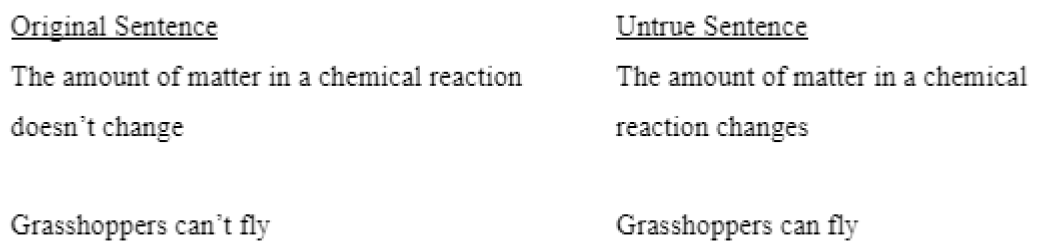
This includes following steps:

- Ways to Generate true sentence:
    - Split complex sentence into a simple sentence



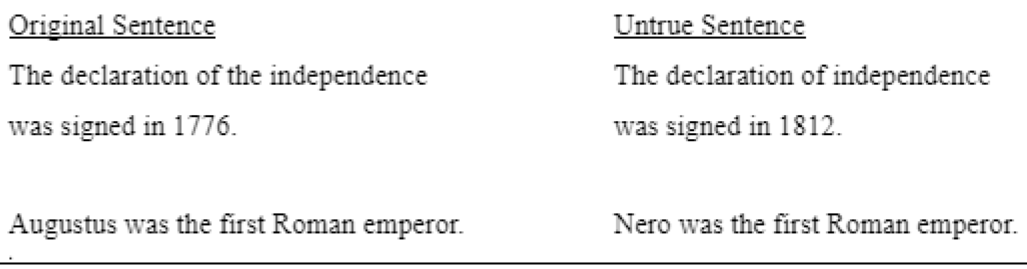
| Original Sentence | True sentence |
|---|---|
| He wiped off the water and turned the boat upside down | He turned the boat upside down. |
| Subject pronouns which can be found only in the beginning of sentences and have no bearing on the word order,are used to replace nouns | Subject pronouns are used to replace nouns. |

- Ways to Generate false sentence:
    - A verb or noun phrase's negations can be added or removed.



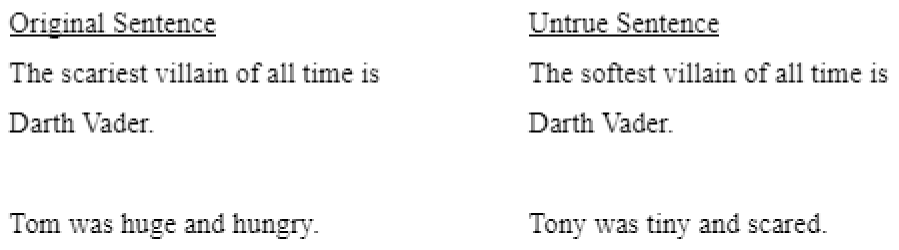
| Original Sentence | Untrue Sentence |
|---|---|
| The amount of matter in a chemical reaction doesn't change | The amount of matter in a chemical reaction changes |
| Grasshoppers can't fly | Grasshoppers can fly |

- A named entity is modified.



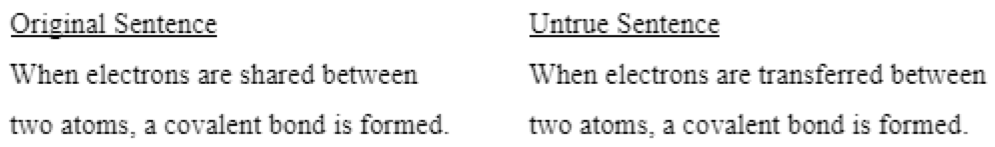
| Original Sentence | Untrue Sentence |
|---|---|
| The declaration of the independence was signed in 1776. | The declaration of independence was signed in 1812. |
| Augustus was the first Roman emperor. | Nero was the first Roman emperor. |

- Alter adjective.



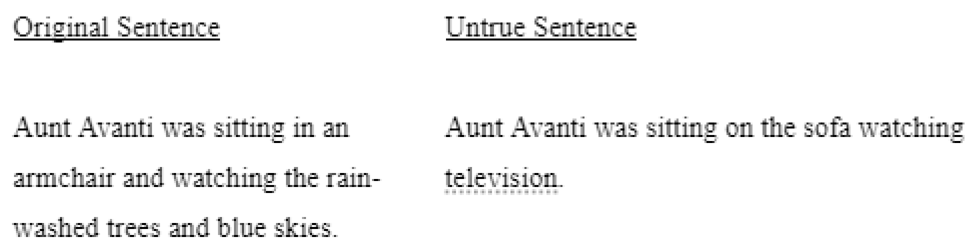
| Original Sentence | Untrue Sentence |
|---|---|
| The scariest villain of all time is Darth Vader. | The softest villain of all time is Darth Vader. |
| Tom was huge and hungry. | Tony was tiny and scared. |

- Alter the main verb.



| Original Sentence | Untrue Sentence |
|---|---|
| When electrons are shared between two atoms, a covalent bond is formed. | When electrons are transferred between two atoms, a covalent bond is formed. |

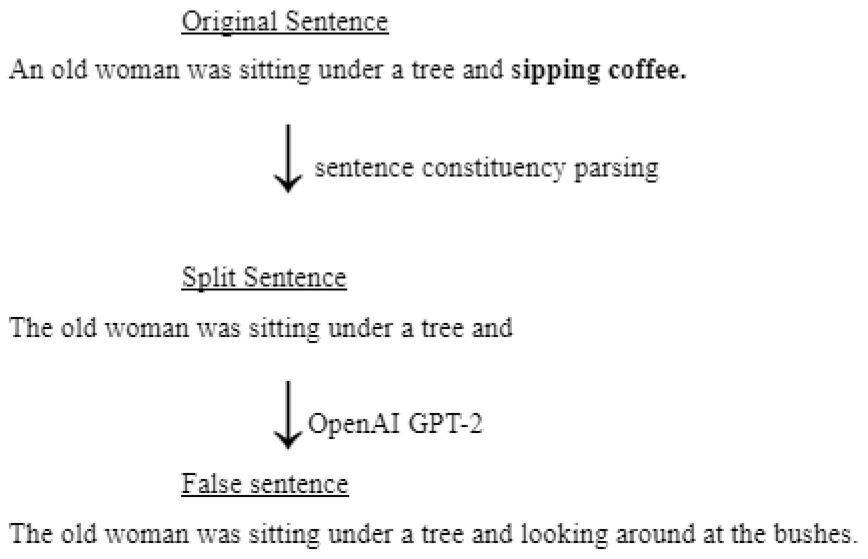
- Substitute verb or noun phrase



| Original Sentence | Untrue Sentence |
|---|---|
| Aunt Avanti was sitting in an armchair and watching the rain-washed trees and blue skies. | Aunt Avanti was sitting on the sofa watching television. |

Generate a False sentence using a different verb or noun phrase:

Using Natural Language Processing techniques, we will change the verb or noun phrase in a valid statement to create a false one. Suppose there is a text:



Original Sentence

An old woman was sitting under a tree and **sipping coffee.**

↓ sentence constituency parsing

Split Sentence

The old woman was sitting under a tree and

↓ OpenAI GPT-2

False sentence

The old woman was sitting under a tree and looking around at the bushes.

- Finding a noun phrase is the first step in using constituency parsing techniques to comprehend the sentence and identify the final noun or verb phrase.

- Split out the final word or verb to finish the sentence.

- Give the incomplete sentence to the OpenAI GPT-2 to finish it.

## Constituency Parsing of a sentence:

Constituency Parsing is the process of separating a text into its constituent words or phrases. Types of phrases make up the parse tree's non terminals, while the sentence's words make up its terminals. To get rid of the final verb or noun phrase, we'll use the allennlp parser.

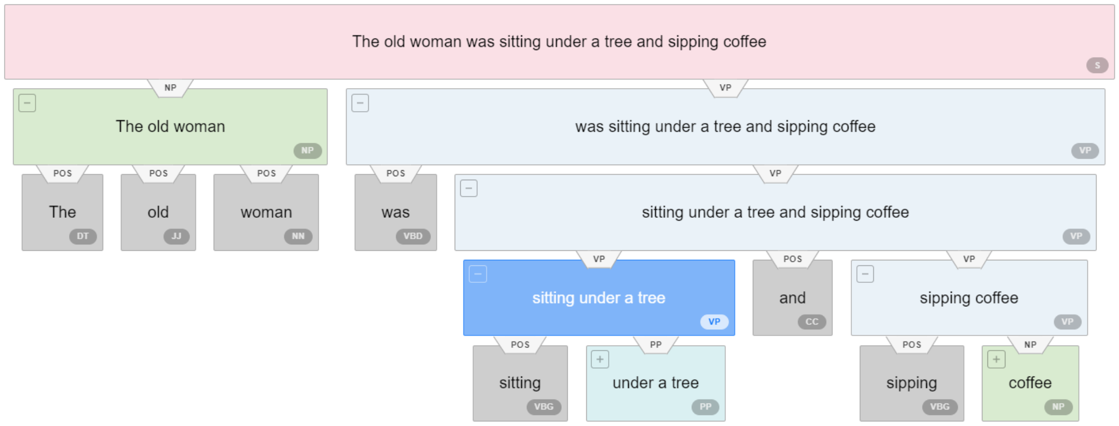
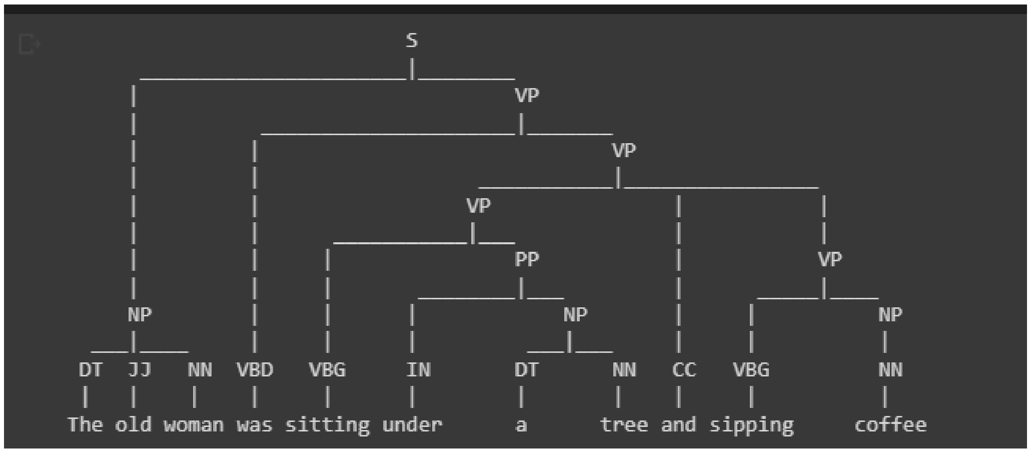

The old woman was sitting under a tree and sipping coffee
S
NP
The old woman
NP
VP
was sitting under a tree and sipping coffee
VP
POS
The
DT
old
JJ
woman
NN
was
VBD
sitting under a tree and sipping coffee
VP
sitting under a tree
VP
and
CC
sipping coffee
VP
sitting
VBG
PP
under a tree
PP
sipping
VBG
coffee
NP

Figure 4.1: AllenNlp model output

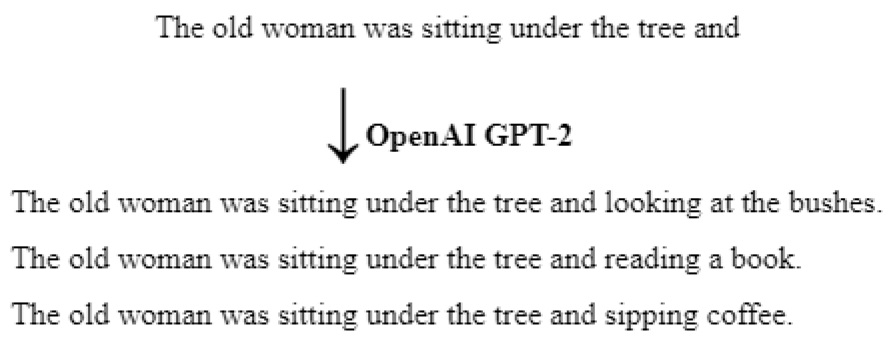


```
                             S
             ________________|__________
            |                           VP
            |         __________________|________
            |        |                           VP
            |        |                ___________|________________
            |        |               VP                |          |
            |        |         ______|___              |          |
            |        |        |          PP            |          VP
            |        |        |     _____|___          |     _____|____
            NP       |        |    |         NP        |    |          NP
         ___|____    |        |    |      ___|___      |    |          |
        DT  JJ   NN VBD      VBG   IN    DT      NN    CC  VBG         NN
        |   |    |   |        |    |     |       |     |    |          |
       The old woman was   sitting under a      tree  and sipping    coffee
```

Figure 4.2: Constituency Parser

OpenAI GPT-2 Text Completion:

A language model called GPT-2 that has been trained to anticipate words is transformer-based. The most well-known applications for transformer models are text production, summarization and language translation.



The old woman was sitting under the tree and

OpenAI GPT-2

The old woman was sitting under the tree and looking at the bushes.

The old woman was sitting under the tree and reading a book.

The old woman was sitting under the tree and sipping coffee.

We choose the option that will provide the false statement the best out of all those produced by the OpenAI GPT-2.

## 4. Fill in the Blanks

Given input text:

- Utilize the Python Keyword Extraction package to extract key phrases.

- Return to our content and collect every sentence containing a specific key phrase.

- We choose any sentence from the list of sentences, just change our keyword to “blank”, and then instruct the reader to fill in the blanks.

Keyword Extraction Algorithm:

With the use of the Keywords Extraction text analysis technique, you may quickly learn crucial insights about a text. It should make it easier to extract pertinent keywords from any text and spare you the time-consuming task of reading the entire thing.

We have numerous keyword extraction algorithms in the PKE (Python Keyword Extraction) library and using a multipartite graph technique, we will extract unsupervised key phrases from them.

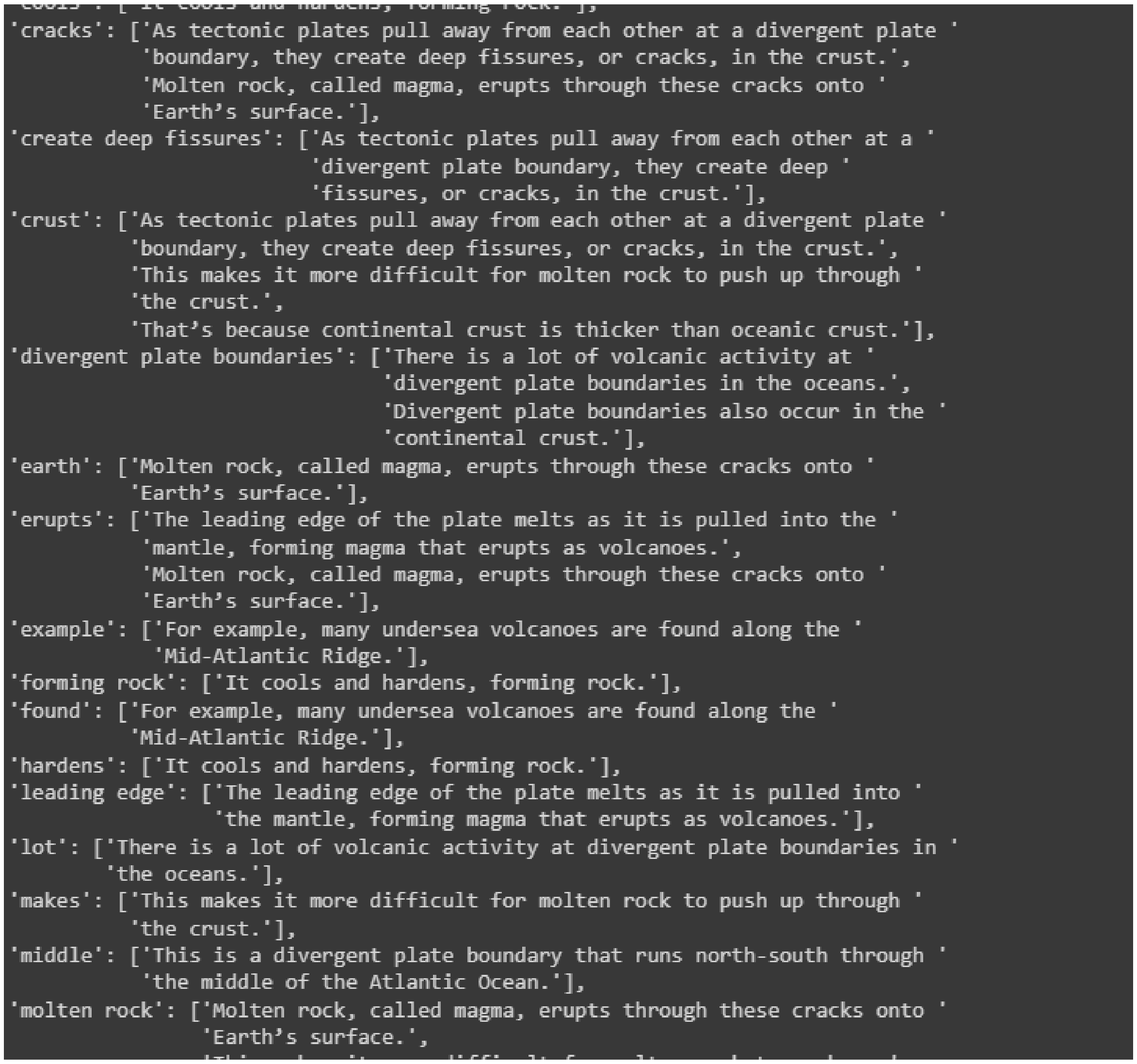
This unsupervised method allows it to work on any fresh document because there is no labeled data or prior training. We will extract elements from any supplied document, such as noun-phrases, verbs, or

adjectives. As a result, we can tell the algorithm what to use as a keyword and presume that we are only providing noun-phrases. Therefore, we categorize all of the collected nouns into subjects. To organize these keywords by subjects, they apply hierarchical clustering in the algorithm.

Finally, we have a single topic with three or four keywords underneath it, and we also have themes and keywords on high-level pages. There is a graph in front of us, and each point or node has a varied weight while connecting to the others. Weights are determined by the co-occurrence of the key phrases. We can claim that a keyword is important if there are many incoming nodes to a certain node that is to a particular keyword or key phrase.

Our objective is to identify the top-ranking nodes among all the nodes with associated edges and use them as the top 10 keywords or key phrases in the article.

The top n nodes in the graph are calculated using the text rank method to determine which nodes are the most significant, and those top n words are returned as key phrases. To evaluate the importance of a certain key phrase.



```
'cracks': ['As tectonic plates pull away from each other at a divergent plate '
           'boundary, they create deep fissures, or cracks, in the crust.',
           'Molten rock, called magma, erupts through these cracks onto '
           'Earth’s surface.'],
'create deep fissures': ['As tectonic plates pull away from each other at a '
                         'divergent plate boundary, they create deep '
                         'fissures, or cracks, in the crust.'],
'crust': ['As tectonic plates pull away from each other at a divergent plate '
          'boundary, they create deep fissures, or cracks, in the crust.',
          'This makes it more difficult for molten rock to push up through '
          'the crust.',
          'That’s because continental crust is thicker than oceanic crust.'],
'divergent plate boundaries': ['There is a lot of volcanic activity at '
                               'divergent plate boundaries in the oceans.',
                               'Divergent plate boundaries also occur in the '
                               'continental crust.'],
'earth': ['Molten rock, called magma, erupts through these cracks onto '
          'Earth’s surface.'],
'erupts': ['The leading edge of the plate melts as it is pulled into the '
           'mantle, forming magma that erupts as volcanoes.',
           'Molten rock, called magma, erupts through these cracks onto '
           'Earth’s surface.'],
'example': ['For example, many undersea volcanoes are found along the '
            'Mid-Atlantic Ridge.'],
'forming rock': ['It cools and hardens, forming rock.'],
'found': ['For example, many undersea volcanoes are found along the '
          'Mid-Atlantic Ridge.'],
'hardens': ['It cools and hardens, forming rock.'],
'leading edge': ['The leading edge of the plate melts as it is pulled into '
                 'the mantle, forming magma that erupts as volcanoes.'],
'lot': ['There is a lot of volcanic activity at divergent plate boundaries in '
        'the oceans.'],
'makes': ['This makes it more difficult for molten rock to push up through '
          'the crust.'],
'middle': ['This is a divergent plate boundary that runs north-south through '
           'the middle of the Atlantic Ocean.'],
'molten rock': ['Molten rock, called magma, erupts through these cracks onto '
                'Earth’s surface.',
```

Figure 4.3: Mapping of keywords with their sentences

Once the keywords have been obtained, we return to the context, map the phrases that include the keywords, and then store them in a dictionary.

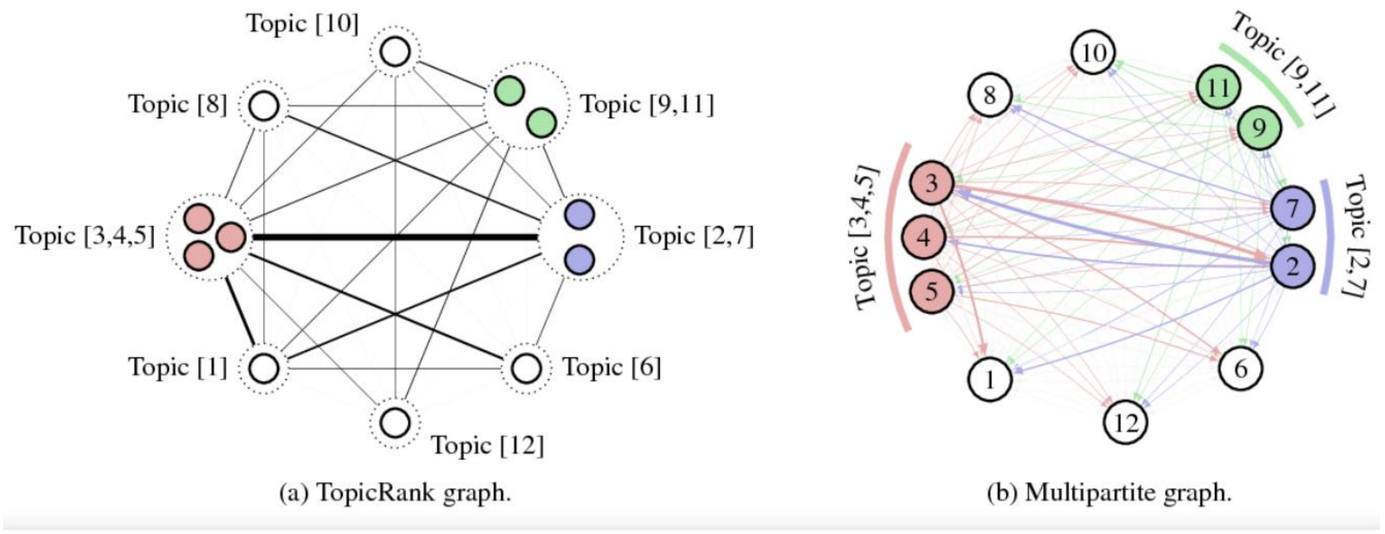


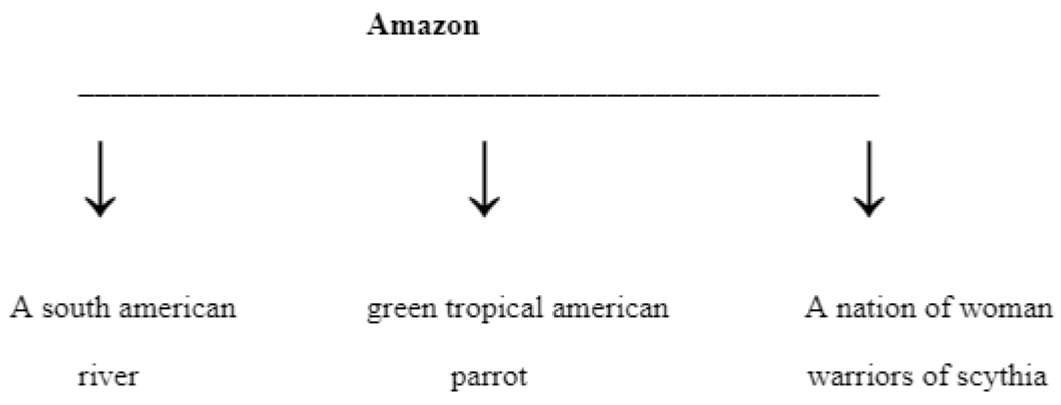
Figure 4.4: Unsupervised Key Phrase Extraction

## 5. MATCH THE FOLLOWING

Any middle school textbook will have the following assessment sorts of questions, where we must match words with their meanings on the right after a list of terms is presented on the left. Usually, the words and their meanings are mixed up to make things more difficult to understand.

We'll utilize the Python Keyword Extraction package to accomplish this and extract the keywords from the context. Therefore, once the keywords have been extracted, those keywords can be used to replace the words on the left of the following match, and their meanings are provided on the right side.

We've taken the keywords from a piece of material, but the task here is to take the sense of the word. We won't be able to discern the precise meaning of the word being used in our context without knowing the context's specifics. The term for this issue is Word Sense Disambiguation (WSD)


Amazon
A south american river
green tropical american parrot
A nation of woman warriors of scythia

Therefore, among all of these different contexts, we need to identify the proper context for the word "Amazon" that effectively clarifies and establishes the proper definition of the term.We gather all the sentences from the text that use the word "Amazon" as one of the methods for determining the context. To do this, we'll use the BERT algorithm, which is transfer based, much like the T5 transformer which is an encoder and decoder based technique. With BERT, any input or block of text is encoded into a fixed encoding vector because it is merely an encoder-based algorithm. Given that BERT is only an encoder-based model, it is mostly use for text classification. BERT accepts a sentence or context as input, and its output is a fixed vector.

Our primary goal is to demonstrate how, given a phrase, we can utilize a trained model to do word Disambiguation.

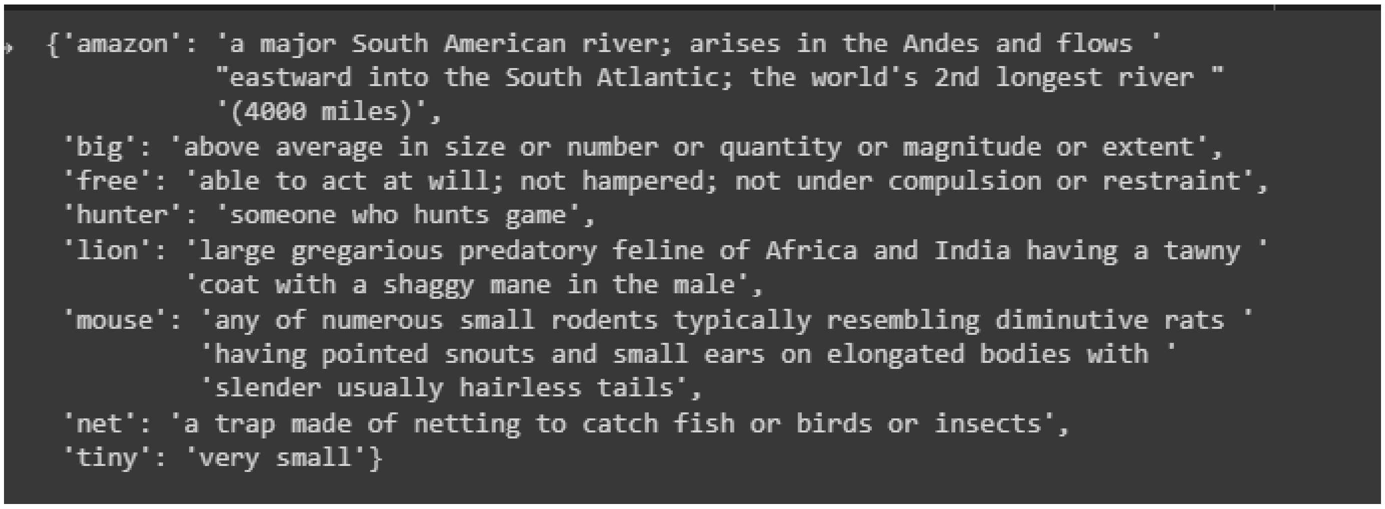
Input: [CLS]Once upon a time, there lived a lion in the dense [TGT]amazon[TGT] rainforest[SEP]A South American river[SEP].

First, we're going to put some tokens around the world, in this case, amazon, that we want to decipher. We'll do this by adding special tokens called TGTs to the area around the word. Now we go to WordNet and get every possible sense of the term "Amazon" as well as the definitions of those senses. We insert a separator [SEP] token into the input text to indicate to the BERT algorithm that there are two distinct contexts. The word "Amazon" has several definitions in WordNet, including "A South American river", "Green Tropical American Parrot" and "A Nation of women warriors of Scythia".

In order to create a score for us, we integrate the original sentence with potential definitions from WordNet and train the algorithm based on the BERT method. The context and sense pair with the highest relevance will produce the highest score for us. The term that earns us the highest score will be used in the following match.

Output: 0.95

Since the first sense, "A South American river", makes the most sense and provides the best score of all the senses (0.95), we will apply it for the following match.



```
{'amazon': 'a major South American river; arises in the Andes and flows '
           "eastward into the South Atlantic; the world's 2nd longest river "
           '(4000 miles)',
 'big': 'above average in size or number or quantity or magnitude or extent',
 'free': 'able to act at will; not hampered; not under compulsion or restraint',
 'hunter': 'someone who hunts game',
 'lion': 'large gregarious predatory feline of Africa and India having a tawny '
         'coat with a shaggy mane in the male',
 'mouse': 'any of numerous small rodents typically resembling diminutive rats '
          'having pointed snouts and small ears on elongated bodies with '
          'slender usually hairless tails',
 'net': 'a trap made of netting to catch fish or birds or insects',
 'tiny': 'very small'}
```

Figure 4.5: Mapping of Keywords and context

# CHAPTER 5

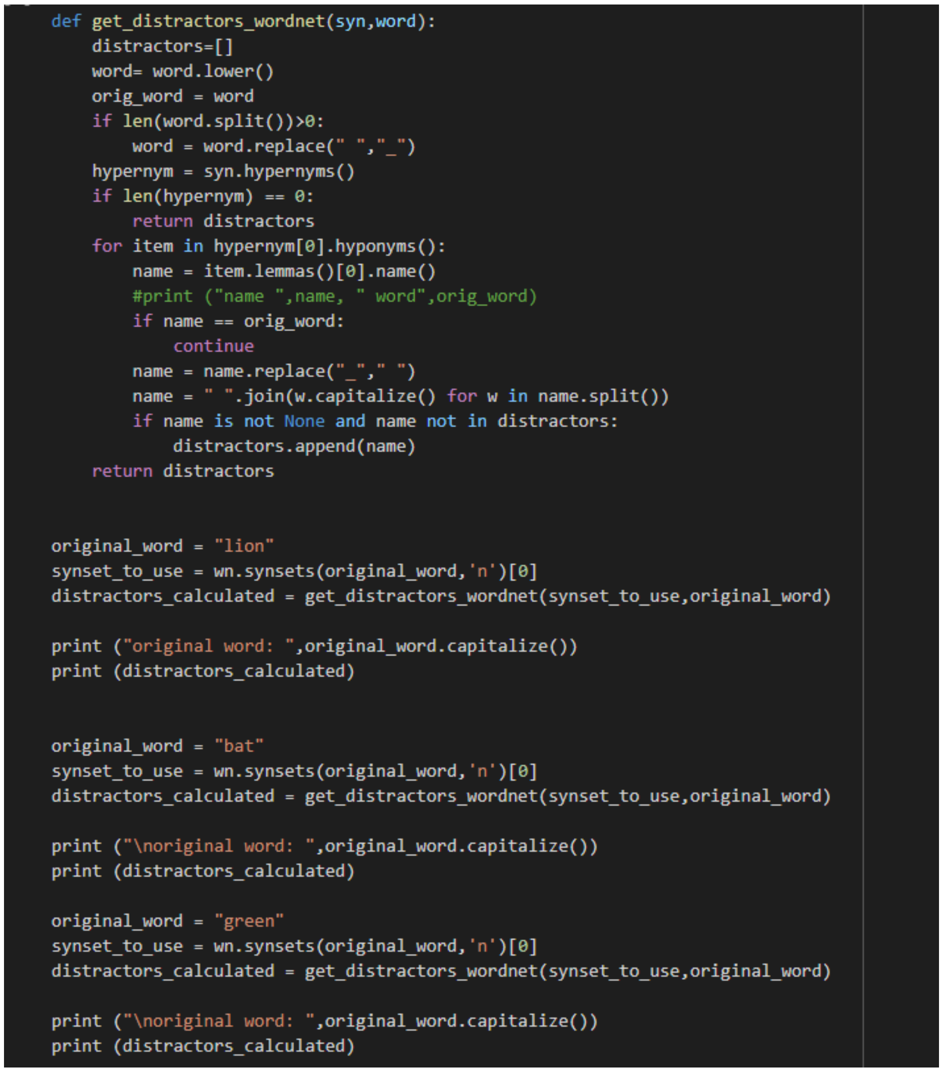
# RESULTS AND DISCUSSION

This chapter is a summary of all significant findings and the commentary that goes with them.

## 1. MCQ Generation

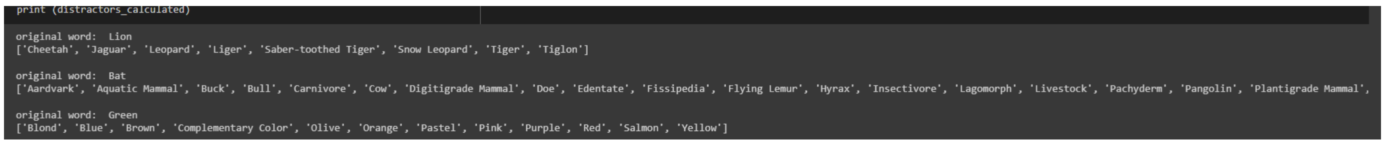
- Distractors

   (1) WordNet



```python
def get_distractors_wordnet(syn,word):
    distractors=[]
    word= word.lower()
    orig_word = word
    if len(word.split())>0:
        word = word.replace(" ","_")
    hypernym = syn.hypernyms()
    if len(hypernym) == 0:
        return distractors
    for item in hypernym[0].hyponyms():
        name = item.lemmas()[0].name()
        #print ("name ",name, " word",orig_word)
        if name == orig_word:
            continue
        name = name.replace("_"," ")
        name = " ".join(w.capitalize() for w in name.split())
        if name is not None and name not in distractors:
            distractors.append(name)
    return distractors


original_word = "lion"
synset_to_use = wn.synsets(original_word,'n')[0]
distractors_calculated = get_distractors_wordnet(synset_to_use,original_word)

print ("original word: ",original_word.capitalize())
print (distractors_calculated)


original_word = "bat"
synset_to_use = wn.synsets(original_word,'n')[0]
distractors_calculated = get_distractors_wordnet(synset_to_use,original_word)

print ("\noriginal word: ",original_word.capitalize())
print (distractors_calculated)

original_word = "green"
synset_to_use = wn.synsets(original_word,'n')[0]
distractors_calculated = get_distractors_wordnet(synset_to_use,original_word)

print ("\noriginal word: ",original_word.capitalize())
print (distractors_calculated)
```

Figure 5.1: WordNet

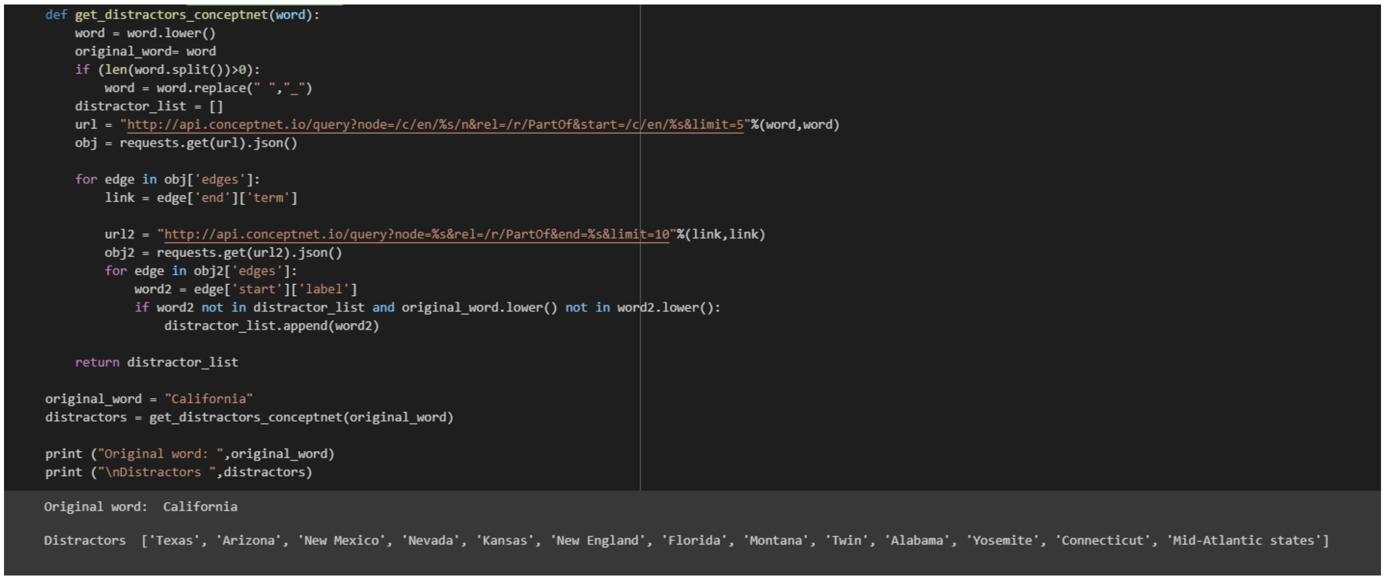


```
print (distractors_calculated)

original word:  Lion
['Cheetah', 'Jaguar', 'Leopard', 'Liger', 'Saber-toothed Tiger', 'Snow Leopard', 'Tiger', 'Tiglon']

original word:  Bat
['Aardvark', 'Aquatic Mammal', 'Buck', 'Bull', 'Carnivore', 'Cow', 'Digitigrade Mammal', 'Doe', 'Edentate', 'Fissipedia', 'Flying Lemur', 'Hyrax', 'Insectivore', 'Lagomorph', 'Livestock', 'Pachyderm', 'Pangolin', 'Plantigrade Mammal',

original word:  Green
['Blond', 'Blue', 'Brown', 'Complementary Color', 'Olive', 'Orange', 'Pastel', 'Pink', 'Purple', 'Red', 'Salmon', 'Yellow']
```

Figure 5.2: Output of WordNet

These are the distractors generated using WordNet. As we can see, a simple word like bat is hard to distinguish due to its Multiform structure.

(2) ConceptNet

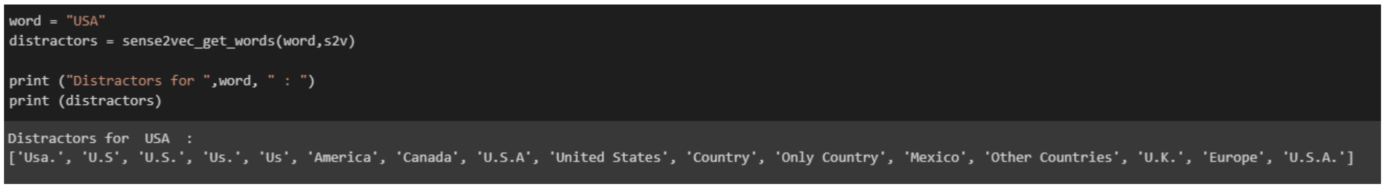


```
def get_distractors_conceptnet(word):
    word = word.lower()
    original_word= word
    if (len(word.split())>0):
        word = word.replace(" ","_")
    distractor_list = []
    url = "http://api.conceptnet.io/query?node=/c/en/%s/n&rel=/r/PartOf&start=/c/en/%s&limit=5"%(word,word)
    obj = requests.get(url).json()

    for edge in obj['edges']:
        link = edge['end']['term']

        url2 = "http://api.conceptnet.io/query?node=%s&rel=/r/PartOf&end=%s&limit=10"%(link,link)
        obj2 = requests.get(url2).json()
        for edge in obj2['edges']:
            word2 = edge['start']['label']
            if word2 not in distractor_list and original_word.lower() not in word2.lower():
                distractor_list.append(word2)

    return distractor_list

original_word = "California"
distractors = get_distractors_conceptnet(original_word)

print ("Original word: ",original_word)
print ("\nDistractors ",distractors)

Original word:  California

Distractors  ['Texas', 'Arizona', 'New Mexico', 'Nevada', 'Kansas', 'New England', 'Florida', 'Montana', 'Twin', 'Alabama', 'Yosemite', 'Connecticut', 'Mid-Atlantic states']
```

Figure 5.3: ConceptNet

As we can see, for California [“US” and “Southwest”] are hypernyms and without knowing the sentence, it’s hard to classify and generate distractors.

(3) Sense2Vec

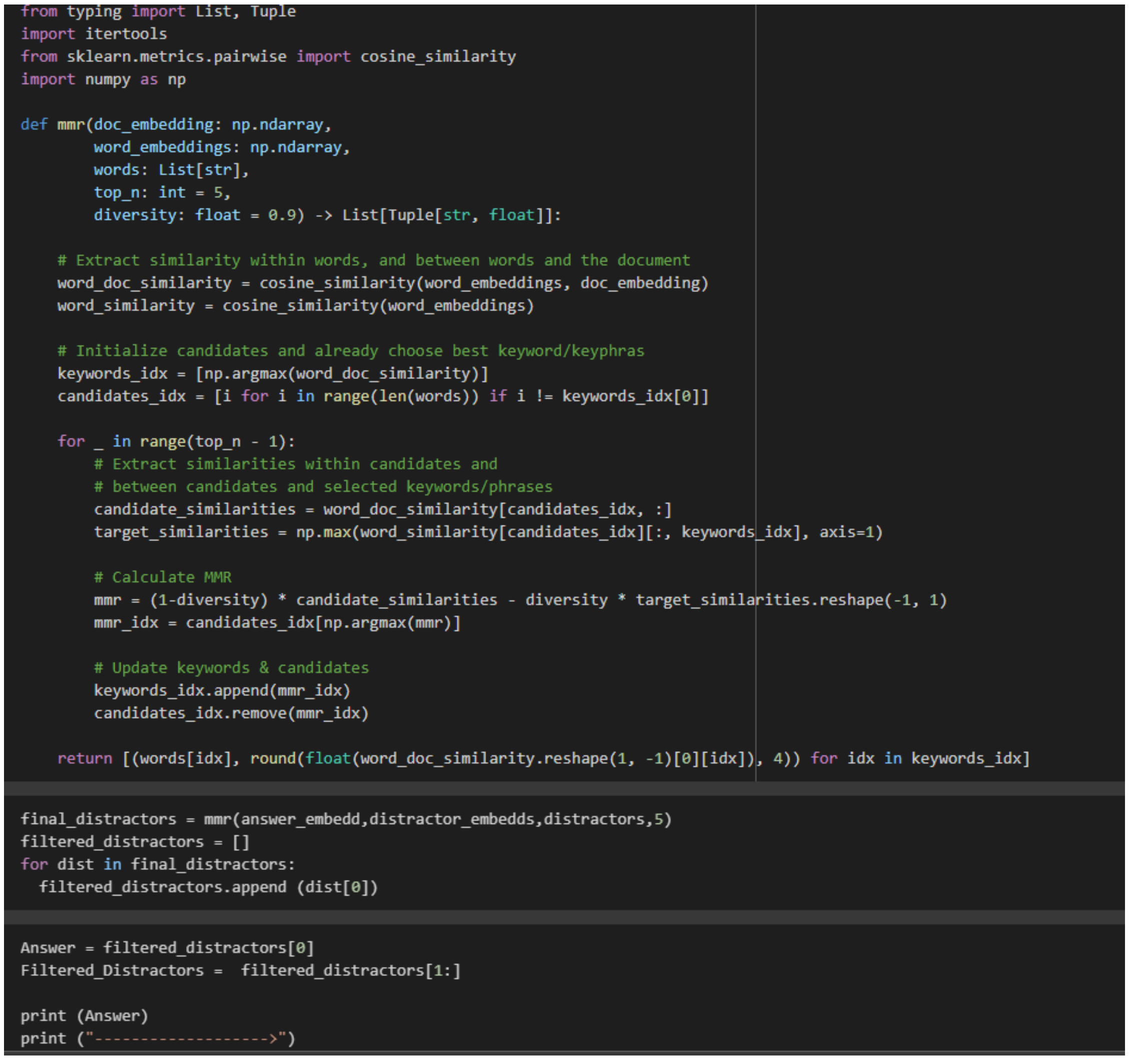


```
word = "USA"
distractors = sense2vec_get_words(word,s2v)

print ("Distractors for ",word, " : ")
print (distractors)

Distractors for  USA  :
['Usa.', 'U.S', 'U.S.', 'Us.', 'Us', 'America', 'Canada', 'U.S.A', 'United States', 'Country', 'Only Country', 'Mexico', 'Other Countries', 'U.K.', 'Europe', 'U.S.A.']
```

Figure 5.4: Sense2Vec

Although it takes the sense of words, the distractors are not relevant("U.S", "Usa" and "U.S.A" are ideal and therefore can't be used as distractors)

(4) Sentence Transformers

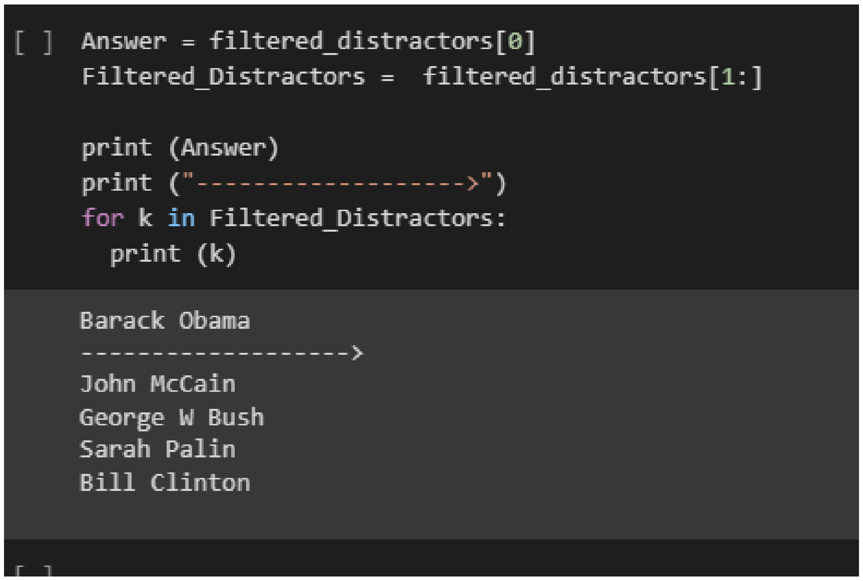


```
from typing import List, Tuple
import itertools
from sklearn.metrics.pairwise import cosine_similarity
import numpy as np

def mmr(doc_embedding: np.ndarray,
        word_embeddings: np.ndarray,
        words: List[str],
        top_n: int = 5,
        diversity: float = 0.9) -> List[Tuple[str, float]]:

    # Extract similarity within words, and between words and the document
    word_doc_similarity = cosine_similarity(word_embeddings, doc_embedding)
    word_similarity = cosine_similarity(word_embeddings)

    # Initialize candidates and already choose best keyword/keyphras
    keywords_idx = [np.argmax(word_doc_similarity)]
    candidates_idx = [i for i in range(len(words)) if i != keywords_idx[0]]

    for _ in range(top_n - 1):
        # Extract similarities within candidates and
        # between candidates and selected keywords/phrases
        candidate_similarities = word_doc_similarity[candidates_idx, :]
        target_similarities = np.max(word_similarity[candidates_idx][:, keywords_idx], axis=1)

        # Calculate MMR
        mmr = (1-diversity) * candidate_similarities - diversity * target_similarities.reshape(-1, 1)
        mmr_idx = candidates_idx[np.argmax(mmr)]

        # Update keywords & candidates
        keywords_idx.append(mmr_idx)
        candidates_idx.remove(mmr_idx)

    return [(words[idx], round(float(word_doc_similarity.reshape(1, -1)[0][idx]), 4)) for idx in keywords_idx]
```

```
final_distractors = mmr(answer_embedd,distractor_embedds,distractors,5)
filtered_distractors = []
for dist in final_distractors:
  filtered_distractors.append (dist[0])
```

```
Answer = filtered_distractors[0]
Filtered_Distractors =  filtered_distractors[1:]

print (Answer)
print ("------------------->")
```

Figure 5.5: Sentence Transformers



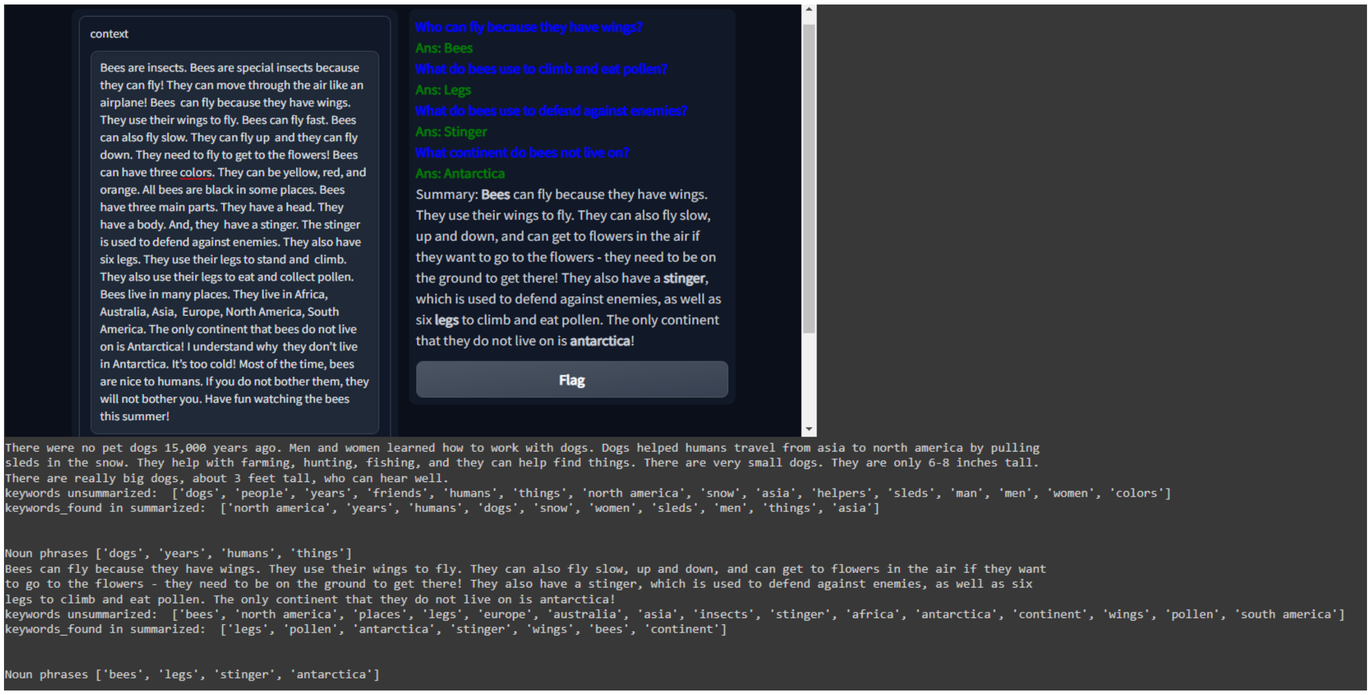
```
[ ]  Answer = filtered_distractors[0]
     Filtered_Distractors =  filtered_distractors[1:]

     print (Answer)
     print ("------------------->")
     for k in Filtered_Distractors:
       print (k)
```

```
Barack Obama
------------------->
John McCain
George W Bush
Sarah Palin
Bill Clinton
```

Figure 5.6: Output of Sentence Transformers

We used Maximum Marginal Relevance which chooses words that are more relevant and with same sense, therefore is more accurate than other algorithms.

- Question Generation



context

Bees are insects. Bees are special insects because they can fly! They can move through the air like an airplane! Bees can fly because they have wings. They use their wings to fly. Bees can fly fast. Bees can also fly slow. They can fly up and they can fly down. They need to fly to get to the flowers! Bees can have three colors. They can be yellow, red, and orange. All bees are black in some places. Bees have three main parts. They have a head. They have a body. And, they have a stinger. The stinger is used to defend against enemies. They also have six legs. They use their legs to stand and climb. They also use their legs to eat and collect pollen. Bees live in many places. They live in Africa, Australia, Asia, Europe, North America, South America. The only continent that bees do not live on is Antarctica! I understand why they don't live in Antarctica. It's too cold! Most of the time, bees are nice to humans. If you do not bother them, they will not bother you. Have fun watching the bees this summer!

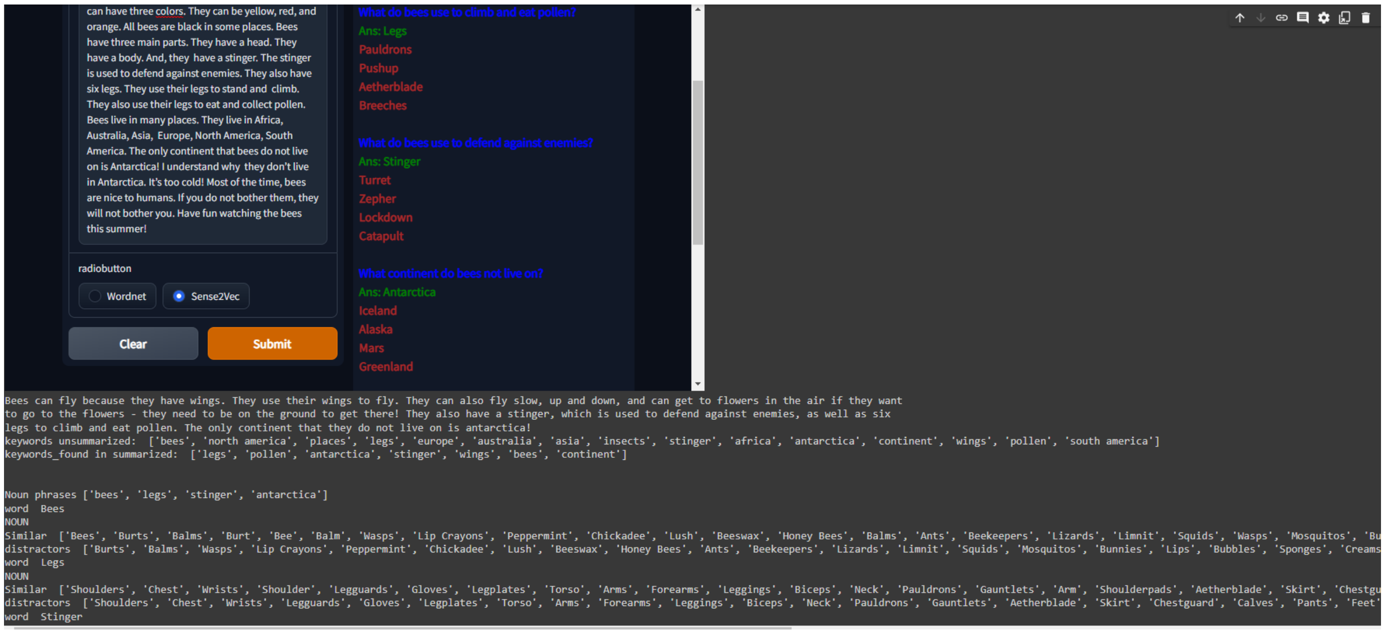
Who can fly because they have wings?
Ans: Bees
What do bees use to climb and eat pollen?
Ans: Legs
What do bees use to defend against enemies?
Ans: Stinger
What continent do bees not live on?
Ans: Antarctica

Summary: **Bees** can fly because they have wings. They use their wings to fly. They can also fly slow, up and down, and can get to flowers in the air if they want to go to the flowers - they need to be on the ground to get there! They also have a **stinger**, which is used to defend against enemies, as well as six **legs** to climb and eat pollen. The only continent that they do not live on is **antarctica**!

Flag

```
There were no pet dogs 15,000 years ago. Men and women learned how to work with dogs. Dogs helped humans travel from asia to north america by pulling
sleds in the snow. They help with farming, hunting, fishing, and they can help find things. There are very small dogs. They are only 6-8 inches tall.
There are really big dogs, about 3 feet tall, who can hear well.
keywords unsummarized:  ['dogs', 'people', 'years', 'friends', 'humans', 'things', 'north america', 'snow', 'asia', 'helpers', 'sleds', 'man', 'men', 'women', 'colors']
keywords_found in summarized:  ['north america', 'years', 'humans', 'dogs', 'snow', 'women', 'sleds', 'men', 'things', 'asia']


Noun phrases ['dogs', 'years', 'humans', 'things']
Bees can fly because they have wings. They use their wings to fly. They can also fly slow, up and down, and can get to flowers in the air if they want
to go to the flowers - they need to be on the ground to get there! They also have a stinger, which is used to defend against enemies, as well as six
legs to climb and eat pollen. The only continent that they do not live on is antarctica!
keywords unsummarized:  ['bees', 'north america', 'places', 'legs', 'europe', 'australia', 'asia', 'insects', 'stinger', 'africa', 'antarctica', 'continent', 'wings', 'pollen', 'south america']
keywords_found in summarized:  ['legs', 'pollen', 'antarctica', 'stinger', 'wings', 'bees', 'continent']


Noun phrases ['bees', 'legs', 'stinger', 'antarctica']
```

Figure 5.7: Question Generation



can have three colors. They can be yellow, red, and orange. All bees are black in some places. Bees have three main parts. They have a head. They have a body. And, they have a stinger. The stinger is used to defend against enemies. They also have six legs. They use their legs to stand and climb. They also use their legs to eat and collect pollen. Bees live in many places. They live in Africa, Australia, Asia, Europe, North America, South America. The only continent that bees do not live on is Antarctica! I understand why they don't live in Antarctica. It's too cold! Most of the time, bees are nice to humans. If you do not bother them, they will not bother you. Have fun watching the bees this summer!

radiobutton

Wordnet [✓ Sense2Vec]

Clear Submit

What do bees use to climb and eat pollen?
Ans: Legs
Pauldrons
Pushup
Aetherblade
Breeches

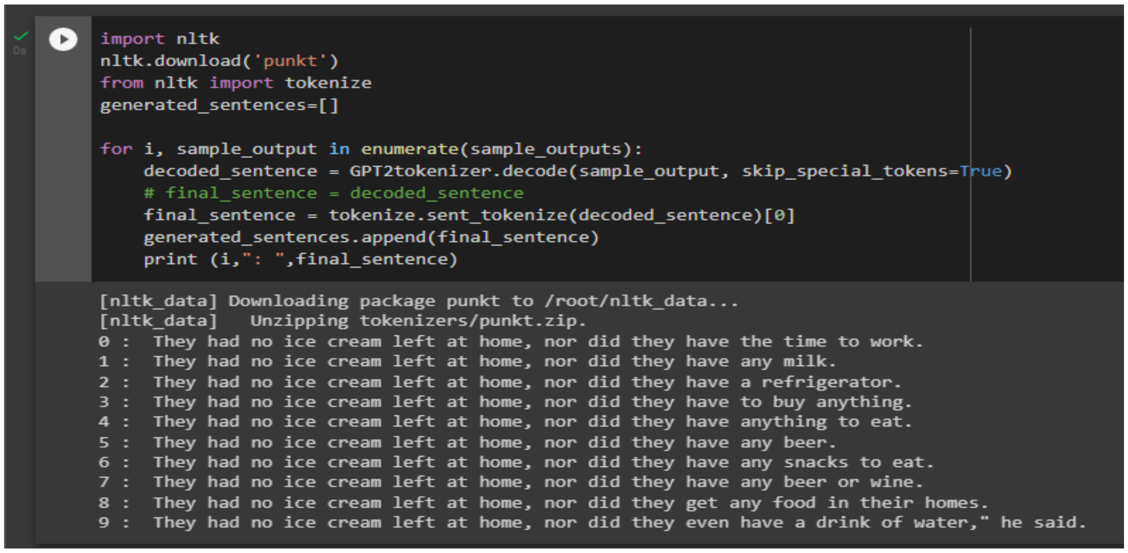
What do bees use to defend against enemies?
Ans: Stinger
Turret
Zepher
Lockdown
Catapult

What continent do bees not live on?
Ans: Antarctica
Iceland
Alaska
Mars
Greenland

```
Bees can fly because they have wings. They use their wings to fly. They can also fly slow, up and down, and can get to flowers in the air if they want
to go to the flowers - they need to be on the ground to get there! They also have a stinger, which is used to defend against enemies, as well as six
legs to climb and eat pollen. The only continent that they do not live on is antarctica!
keywords unsummarized:  ['bees', 'north america', 'places', 'legs', 'europe', 'australia', 'asia', 'insects', 'stinger', 'africa', 'antarctica', 'continent', 'wings', 'pollen', 'south america']
keywords_found in summarized:  ['legs', 'pollen', 'antarctica', 'stinger', 'wings', 'bees', 'continent']


Noun phrases ['bees', 'legs', 'stinger', 'antarctica']
word  Bees
NOUN
Similar  ['Bees', 'Burts', 'Balms', 'Burt', 'Bee', 'Balm', 'Wasps', 'Lip Crayons', 'Peppermint', 'Chickadee', 'Lush', 'Beeswax', 'Honey Bees', 'Balms', 'Ants', 'Beekeepers', 'Lizards', 'Limnit', 'Squids', 'Wasps', 'Mosquitos', 'Bu
distractors  ['Burts', 'Balms', 'Wasps', 'Lip Crayons', 'Peppermint', 'Chickadee', 'Lush', 'Beeswax', 'Honey Bees', 'Ants', 'Beekeepers', 'Lizards', 'Limnit', 'Squids', 'Mosquitos', 'Bunnies', 'Lips', 'Bubbles', 'Sponges', 'Creams
word  Legs
NOUN
Similar  ['Shoulders', 'Chest', 'Wrists', 'Shoulder', 'Legguards', 'Gloves', 'Legplates', 'Torso', 'Arms', 'Forearms', 'Leggings', 'Biceps', 'Neck', 'Pauldrons', 'Gauntlets', 'Arm', 'Shoulderpads', 'Aetherblade', 'Skirt', 'Chestgu
distractors  ['Shoulders', 'Chest', 'Wrists', 'Legguards', 'Gloves', 'Legplates', 'Torso', 'Arms', 'Forearms', 'Leggings', 'Biceps', 'Neck', 'Pauldrons', 'Gauntlets', 'Aetherblade', 'Skirt', 'Chestguard', 'Calves', 'Pants', 'Feet'
word  Stinger
```

Figure 5.9: MCQ Generation

## 2. True or False Questions



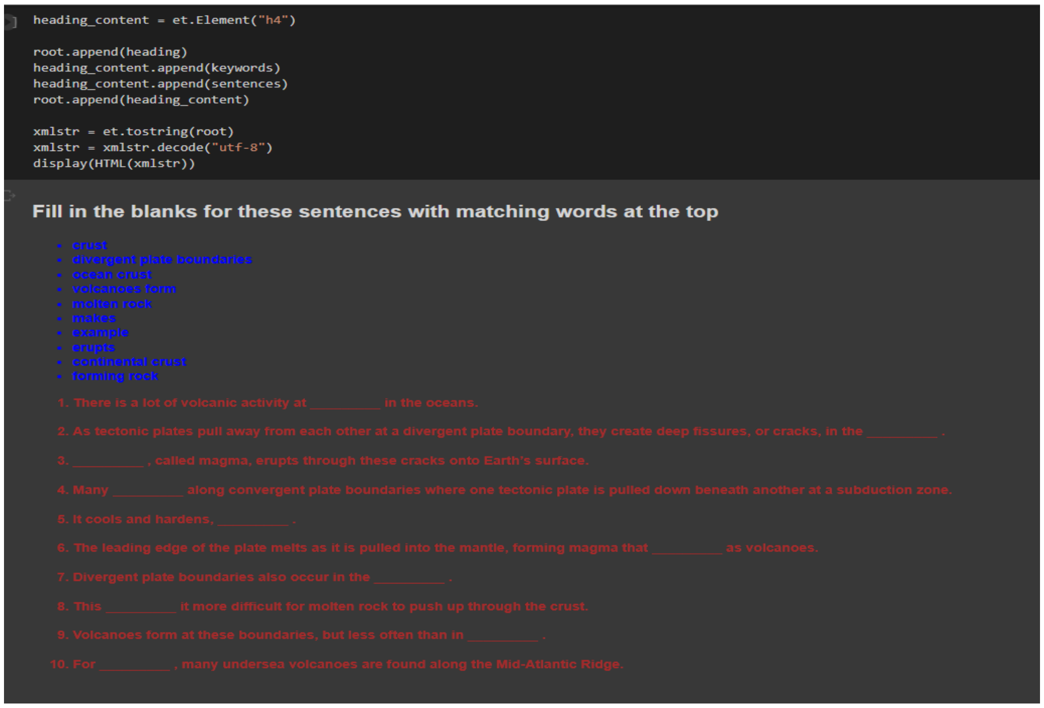
```
import nltk
nltk.download('punkt')
from nltk import tokenize
generated_sentences=[]

for i, sample_output in enumerate(sample_outputs):
    decoded_sentence = GPT2tokenizer.decode(sample_output, skip_special_tokens=True)
    # final_sentence = decoded_sentence
    final_sentence = tokenize.sent_tokenize(decoded_sentence)[0]
    generated_sentences.append(final_sentence)
    print (i,": ",final_sentence)
```

```
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Unzipping tokenizers/punkt.zip.
0 :  They had no ice cream left at home, nor did they have the time to work.
1 :  They had no ice cream left at home, nor did they have any milk.
2 :  They had no ice cream left at home, nor did they have a refrigerator.
3 :  They had no ice cream left at home, nor did they have to buy anything.
4 :  They had no ice cream left at home, nor did they have anything to eat.
5 :  They had no ice cream left at home, nor did they have any beer.
6 :  They had no ice cream left at home, nor did they have any snacks to eat.
7 :  They had no ice cream left at home, nor did they have any beer or wine.
8 :  They had no ice cream left at home, nor did they get any food in their homes.
9 :  They had no ice cream left at home, nor did they even have a drink of water," he said.
```

Figure 5.2: True or False Questions

These are the False statements that are generated from the given input sentence.

## 3. Fill in the Blanks



```
heading_content = et.Element("h4")

root.append(heading)
heading_content.append(keywords)
heading_content.append(sentences)
root.append(heading_content)

xmlstr = et.tostring(root)
xmlstr = xmlstr.decode("utf-8")
display(HTML(xmlstr))
```

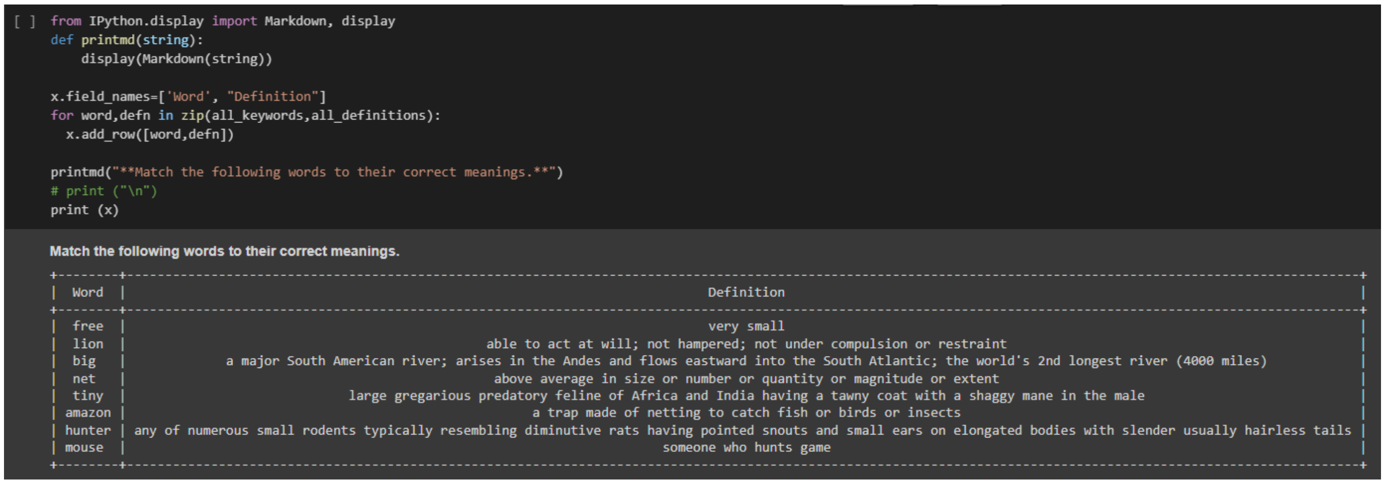
**Fill in the blanks for these sentences with matching words at the top**

- crust
- divergent plate boundaries
- ocean crust
- volcanoes form
- molten rock
- makes
- example
- erupts
- continental crust
- forming rock

1. There is a lot of volcanic activity at __________ in the oceans.
2. As tectonic plates pull away from each other at a divergent plate boundary, they create deep fissures, or cracks, in the __________ .
3. __________ , called magma, erupts through these cracks onto Earth's surface.
4. Many __________ along convergent plate boundaries where one tectonic plate is pulled down beneath another at a subduction zone.
5. It cools and hardens, __________ .
6. The leading edge of the plate melts as it is pulled into the mantle, forming magma that __________ as volcanoes.
7. Divergent plate boundaries also occur in the __________ .
8. This __________ it more difficult for molten rock to push up through the crust.
9. Volcanoes form at these boundaries, but less often than in __________ .
10. For __________ , many undersea volcanoes are found along the Mid-Atlantic Ridge.

Figure 5.10: Fill in the Blanks

## 4. Match the Following



```
[ ] from IPython.display import Markdown, display
    def printmd(string):
        display(Markdown(string))

    x.field_names=['Word', "Definition"]
    for word,defn in zip(all_keywords,all_definitions):
      x.add_row([word,defn])

    printmd("**Match the following words to their correct meanings.**")
    # print ("\n")
    print (x)
```

**Match the following words to their correct meanings.**

```
+--------+-----------------------------------------------------------------------------------------------------------------------------------------------------------------+
|  Word  |                                                                           Definition                                                                            |
+--------+-----------------------------------------------------------------------------------------------------------------------------------------------------------------+
|  free  |                                                                           very small                                                                            |
|  lion  |                                              able to act at will; not hampered; not under compulsion or restraint                                               |
|  big   |            a major South American river; arises in the Andes and flows eastward into the South Atlantic; the world's 2nd longest river (4000 miles)             |
|  net   |                                               above average in size or number or quantity or magnitude or extent                                                |
|  tiny  |                            large gregarious predatory feline of Africa and India having a tawny coat with a shaggy mane in the male                             |
| amazon |                                                    a trap made of netting to catch fish or birds or insects                                                     |
| hunter | any of numerous small rodents typically resembling diminutive rats having pointed snouts and small ears on elongated bodies with slender usually hairless tails |
| mouse  |                                                                     someone who hunts game                                                                      |
+--------+-----------------------------------------------------------------------------------------------------------------------------------------------------------------+
```

Figure 5.11: Match the Following

# CHAPTER 6

# CONCLUSION AND FUTURE WORK

The Extensive Literature Survey was the initial stage in figuring out the project's scope. The concepts can be made more specific in order to more effectively incorporate them. During this time, we looked at some publications that were related to our problem. Extensive literature searches on each of these modules were done in order to understand the models that are accessible and the implementation components. A few applications of already-used methods were also tested. We looked into a number of study articles and blogs to gain understanding about that. After reading, we received a synopsis of the model we intend to construct.

We come to the conclusion that creating a variety of questions not only acts as an assessment tool but also influences student learning. The maximum number of Multiple-Choice Questions (MCQs) we can produce from a text is 4, which is an area that needs improvement.

## BIBLIOGRAPHY

[1] Ahmed Emad, Fady Bassel, Mark Refaat, Mohamed Abdelhamed , Nada Shorim , Ashraf AbdelRaouf , “Automatic Video summarization with Timestamps using natural language processing text fusion”, Misr International University, Cairo,Egyptahmed1703326,fady1710742,mark1711712,mohamed1701989, nada.ayman,ashraf.raouff@miuegypt.edu.egg,2021

[2] Chi Zhang, Shagan Sah, Thang Nguyen, Dheeraj Peri, Alexander Louiy , Carl Salvaggio , Raymond Ptucha , “Semantic Sentence Embeddings for Paraphrasing and Text Summarization” Rochester Institute of Technology, Rochester, NY 14623, USA , Kodak Alaris Imaging Science R&D, Rochester, NY 14615, USA,2017

[3] Pritam Kumar Mehta, Prachi Jain, Chetan Makwana, M Raut, “Automated MCQ Generator using NLP”, Datta Meghe College of Engineering, Navi Mumbai, India,Dept. of Computer Engineering, Datta Meghe College of Engineering, Navi Mumbai, India ,2021

[4] Zygmunt Pizlo, Arnon Amir, Dulce B. Ponceleon, Edward J.Delp, “Automated Video Program Summarization using Speech transcripts”, Article in IEEE Transactions on Multimedia, September 2006

[5] Marco Furini, Vittorio Ghini, “An audio-video summarization scheme based on audio and video analysis”, Conference Paper · February 2006

[6] Pravin Khandare, Sanket Gaikwad, Aditya Kukade, Rohit Panicker, Swaraj Thamke, “AUDIO DATA SUMMARIZATION SYSTEM USING NATURAL LANGUAGE PROCESSING”, Assistant Professor, Dept. of Computer Engineering, Vishwakarma Institute of Technology, Pune, Maharashtra, 2,3,4,5UG Student, Vishwakarma Institute of Technology, Pune, Maharashtra (India)

[7] Dr. S. Muthusundari, Vishwa A, Srinivasa kiruthik K S, Sukesh Raj R, “Automatic MCQ Generator”, Associate Professor, Computer Science & Engineering Department R.M.D Engineering College Kavaraipettai, Chennai, India 2Zoho Corporation, Chennai

[8]Edison Marrese-Taylor, Ai Nakajima, Yutaka Matsuo,“Learning to Automatically Generate Fill-In-The-Blank Quizzes”,Graduate School of Engineering The University of Tokyo

[9]Regina Kasakowskij , Thomas Kasakowskij, Niels Seidel,”Generation of Multiple True False Questions”