MCQ GENERATION USING NLP

Mini Project Report submitted by

ALDEN OLIVERO
4NM21AI010

ANKITH V POOJARY
4NM21AI014

CHIRAG H DEVADIGA
4NM21AI020

UNDER THE GUIDANCE OF
Mrs. Rakshitha
Assistant Professor
Department of Artificial Intelligence and Machine Learning Engineering

In partial fulfillment of the requirements for the

NATURAL LANGUAGE PROCESSING LAB-21AM604

NITTE EDUCATION TRUST | N.M.A.M. INSTITUTE OF TECHNOLOGY
(An Autonomous Institution affiliated to Visvesvaraya Technological University, Belagavi)
Nitte – 574 110, Karnataka, India

(ISO 9001:2015 Certified) | Accredited with ‘A’ Grade by NAAC

08258 - 281039 – 281263, Fax: 08258 – 281265

AY 2023 - 2024

NITTE
EDUCATION TRUST
N.M.A.M. INSTITUTE OF TECHNOLOGY
(An Autonomous Institution affiliated to Visvesvaraya Technological University, Belagavi)
Nitte – 574 110, Karnataka, India
(ISO 9001:2015 Certified), Accredited with 'A' Grade by NAAC)
☎ : 08258 - 281039 - 281263, Fax: 08258 - 281265

Department of Artificial Intelligence and Machine Learning Engineering

CERTIFICATE

Certified that the mini project work entitled

"MCQ Generation using NLP"

is a bonafide work carried out by as a component of

Natural Language Processing Lab (21AM604)

Alden Olivero
(4NM21AI010)

Ankith V Poojary
(4NM21AI014)

Chirag H Devadiga
(4NM21AI020)

in partial fulfilment of the requirements for the award of

Bachelor of Engineering Degree in Artificial intelligence and Machine Learning Engineering

prescribed by Visvesvaraya Technological University, Belgaum

during the year 2023-2024.

It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the report deposited in the departmental library.

The mini project report has been approved as it satisfies the academic requirements in respect of the mini project work prescribed for the Bachelor of Engineering Degree.

8/5/24
Signature of Guide

Signature of HOD

Evaluation

Name of the Examiners	Signature with Date
1. MAHESH B.L.	
2. Rakshitha	8/5/24

ACKNOWLEDGEMENT

We believe that our mini project will be complete only after we thank the people who have contributed to make this mini project successful.

First and foremost, our sincere thanks to our beloved principal, **Dr. Niranjan N. Chiplunkar** for giving us an opportunity to carry out our mini project work at our college and providing us with all the needed facilities.

I acknowledge the support and valuable inputs given by**, Dr. Sharada U Shenoy** the Head of the Department, Artificial Intelligence and Machine Learning Engineering, NMAMIT, Nitte

We express our deep sense of gratitude and indebtedness to our guide **Mrs. Rakshitha** Assistant Professor Artificial Intelligence and Machine Learning Engineering, for his inspiring guidance, constant encouragement, support and suggestions for improvement during the course for our mini project.

We also thank all those who have supported us throughout the entire duration of our mini project.

Finally, we thank the staff members of the Department of Artificial Intelligence and Machine Learning Engineering and all our friends for their honest opinions and suggestions throughout the course of our mini project.

ALDEN OLIVERO

ANKITH V POOJARY

CHIRAG H DEVADIGA

Table of Contents

Project Report: MCQ Generation Using NLP

Chapter 1: Abstract

Generating multiple-choice questions (MCQs) is a common task in educational and training contexts, yet manual question creation can be time-consuming and labor-intensive. This project explores an automated approach to MCQ generation using natural language processing (NLP), aiming to streamline the process and provide a scalable solution.

The system uses various NLP techniques to generate MCQs from textual content. It begins with text summarization using the T5 model from Hugging Face to condense lengthy texts while preserving essential information. Following summarization, keywords are extracted with the PKE (Pattern-based Keyword Extraction) library, identifying the core concepts that form the basis of MCQs. The summarized text and extracted keywords guide the generation of questions with the help of Hugging Face's T5-based question generation model.

One of the core components of this project is the keyword extraction phase, where the system identifies significant words and phrases within the summarized text. By utilizing the PKE (Pattern-based Keyword Extraction) library, the project ensures that these keywords are both contextually relevant and representative of the text's content. This step is crucial for generating meaningful questions that are closely tied to the source material. Additionally, the keyword extraction process serves as a foundation for creating both the questions and their corresponding distractors, linking various elements of the system.

The distractor generation process is equally important, as it directly affects the quality and effectiveness of the MCQs. Using Sense2Vec, WordNet, and ConceptNet, the project generates plausible distractors by exploring semantic relationships and identifying words with similar senses. This approach allows the system to produce incorrect answers that are not only challenging but also contextually appropriate. The flexibility of this method enables the system to adapt to different types of text, providing educators and trainers with a versatile tool for creating quizzes and assessments.

The project demonstrates how a combination of NLP techniques can automate the MCQ generation process, potentially reducing the time and effort required for educators and content creators. The system is designed to be adaptable to various text sources, offering a flexible solution for generating educational content.

Chapter 2: Introduction

Multiple-choice questions (MCQs) are a cornerstone of educational assessments, widely used in schools, universities, and training programs. They offer a straightforward format for evaluating knowledge and understanding, allowing educators to quickly gauge students' comprehension of a topic. However, creating high-quality MCQs manually is a time-consuming and labor-intensive task. Educators must devise questions, identify correct answers, and construct plausible distractors—all of which require careful thought and subject matter expertise.

The rise of Natural Language Processing (NLP) has opened new possibilities for automating the generation of MCQs. NLP provides tools and techniques for analyzing and processing text, enabling systems to extract key information, generate questions, and create suitable distractors. This project explores an automated approach to MCQ generation, leveraging NLP to reduce the workload for educators and streamline the process of creating educational content.

By focusing on text summarization, keyword extraction, question generation, and distractor creation, the project aims to demonstrate how NLP can automate MCQ generation from a variety of textual sources. The ultimate goal is to develop a scalable and adaptable solution that can be used in educational and training settings, providing a valuable resource for teachers, trainers, and content creators.

This report outlines the project's objectives, methodology, implementation details, and architecture. It discusses the key components of the MCQ generation system, the challenges encountered, and the proposed future work to enhance the system's functionality. Through this exploration, the project seeks to contribute to the ongoing development of automated educational tools and resources.

1.1 Objective:

The primary objective of this project is to develop an automated system for generating multiple-choice questions (MCQs) from textual sources using Natural Language Processing (NLP). The system should be capable of creating high-quality MCQs that include a question, a correct answer, and several plausible distractors. By automating this process, the project aims to reduce the time and effort required for educators and content creators to develop educational assessments, quizzes, and training materials.

To achieve this objective, the project focuses on several key tasks:

- **Text Summarization:** Summarize the input text to extract the most relevant information, providing a foundation for keyword extraction and question generation.
- **Keyword Extraction:** Identify significant keywords and phrases that represent the core concepts within the text.
- **Question Generation:** Generate questions from the summarized text and extracted keywords, ensuring clarity and relevance.
- **Distractor Generation:** Create plausible distractors (incorrect answers) using NLP techniques, such as Sense2Vec, WordNet, and ConceptNet.
- **User Interface:** Implement a user-friendly interface for generating MCQs, allowing educators and content creators to interact with the system and customize question generation.

By accomplishing these tasks, the project aims to demonstrate a scalable approach to MCQ generation that can be applied to a variety of educational and training contexts.

1.2 Motivation:

The motivation for this project arises from the challenges faced by educators, trainers, and content creators when generating MCQs manually. Creating high-quality questions requires significant time, effort, and expertise. The process involves not only crafting the questions but also ensuring the correctness of answers and the plausibility of distractors. This manual approach is prone to human error and often lacks scalability, especially when large numbers of questions are needed for assessments or training materials.

Advancements in NLP have provided an opportunity to automate this process. By leveraging NLP techniques, it is possible to analyse text, extract key concepts, and generate questions and distractors in a more efficient manner. The motivation behind this project is to explore these techniques and develop a system that can help educators and trainers save time, reduce errors, and focus on the pedagogical aspects of their work.

Automating MCQ generation has broader implications for education and training. It can facilitate personalized learning experiences, enable rapid assessment creation, and support adaptive learning platforms. Ultimately, the project seeks to contribute to the ongoing development of educational technologies that improve learning outcomes and make educational content creation more accessible.

Chapter 3: Methodology

This project employs a multi-step approach to generate multiple-choice questions (MCQs) from textual sources. The methodology involves several distinct phases, each focusing on a specific aspect of MCQ generation: text summarization, keyword extraction, question generation, distractor creation, and user interface. This section describes the detailed approach for each phase and explains the underlying NLP techniques and models.

3.1 Text Summarization

The first step in the methodology is text summarization, where the input text is condensed to focus on the most relevant information. This is done to reduce the amount of text and extract the essential concepts for MCQ generation. The summarization is achieved using the T5 model from Hugging Face. The T5 model is a transformer-based model that can be fine-tuned for various tasks, including text summarization.

To summarize the text, the following steps are taken:

1. Preprocessing: The input text is pre-processed to remove extra whitespace and ensure a consistent format.

2. Model Initialization: The T5 model and tokenizer are initialized from pre-trained weights. This ensures that the model is ready to process the text.

3. Summarization: The text is summarized by encoding it with the T5 tokenizer and generating a summary using the model. The generation parameters are set to achieve a balance between brevity and information retention.

The summarized text is then used for further processing, providing a foundation for keyword extraction and question generation.

3.2 Keyword Extraction

After summarization, the next step is to extract keywords from the summarized text. Keywords represent the core concepts and are used to generate questions and identify correct answers. The PKE (Pattern-based Keyword Extraction) library is used for this purpose. PKE allows for various extraction strategies, and this project uses the Multipartite Rank algorithm to extract significant keywords.

The keyword extraction process involves the following steps:

1. Initialization: The PKE extractor is initialized with the summarized text, specifying the language and other extraction parameters.

2. Candidate Selection: Keywords are selected based on part-of-speech tags, focusing

on nouns and proper nouns. This ensures that the keywords represent important concepts.

3. Keyword Extraction: The extractor identifies the most significant keywords using the MultipartiteRank algorithm, which considers the graph structure of the text and ranks the candidates.

The extracted keywords are then used to guide the generation of MCQs, serving as a basis for question framing and distractor creation.

3.3 Question Generation

With the summarized text and extracted keywords, the system proceeds to generate questions. The T5 model is used again, but this time for question generation instead of summarization. The model is fine-tuned to generate questions from a given context and answer, making it ideal for creating questions based on the summarized text and extracted keywords.

The question generation process includes the following steps:

1. Context and Answer Definition: A context is defined based on the summarized text, and a correct answer is chosen from the extracted keywords.

2. Question Generation: The T5 model generates a question based on the context and the specified answer. The generation parameters are adjusted to ensure question clarity and coherence.

3. Post-Processing: The generated question is post-processed to remove extraneous tokens and ensure grammatical correctness.

The generated questions are then combined with their corresponding correct answers to create the basic structure of the MCQs.

3.4 Distractor Creation

A critical aspect of MCQ generation is the creation of plausible distractors (incorrect answers). The methodology uses multiple NLP techniques to generate these distractors, focusing on semantic similarity and word sense analysis. The goal is to create distractors that are challenging yet appropriate in the context of the question.

The distractor creation process involves the following steps:

1. Sense2Vec for Similarity: The Sense2Vec library is used to find words with similar meanings based on a pre-trained model. This provides a list of potential distractors with a high degree of semantic similarity to the correct answer.

2. WordNet for Semantic Analysis: WordNet, a lexical database, is used to identify words with similar senses, allowing for a more comprehensive set of potential distractors.

3. ConceptNet for Relationships: ConceptNet is used to explore the relationships between concepts, providing additional context for generating plausible distractors.

To ensure that distractors are sufficiently different from the correct answer, the Maximum Marginal Relevance (MMR) technique is used. This method balances the similarity to the correct answer with the diversity among distractors, preventing them from being too similar.

3.5 User Interface

The final step in the methodology is to create a user-friendly interface for generating MCQs. Gradio, a Python library for building interactive user interfaces, is used to create a simple and accessible front-end. This interface allows users to input text, generate questions, and view the resulting MCQs with correct answers and distractors.

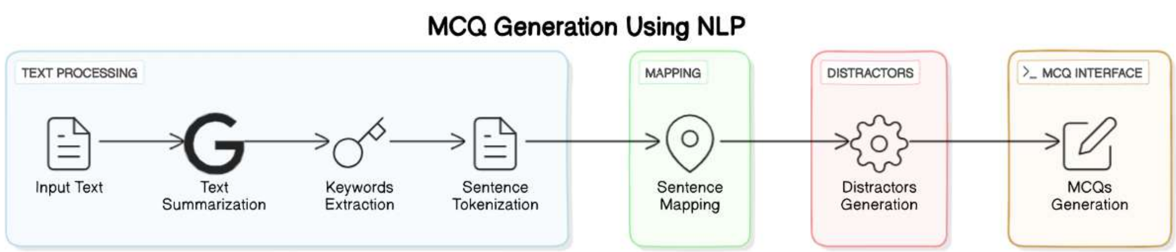
The user interface includes the following features:

1. Text Input: A textbox for users to input the text from which MCQs will be generated.

2. Radio Button for Distractor Method: A choice between WordNet and Sense2Vec for distractor creation.

3. Output Display: A display area to show the generated questions, correct answers, and distractors.

By providing this interface, the project aims to make the MCQ generation system accessible to a broader audience, allowing educators and content creators to generate questions without extensive technical knowledge.



Chapter 4: Implementation

The implementation of this project focuses on generating multiple-choice questions (MCQs) using a combination of natural language processing (NLP) techniques and machine learning models. The code is written in Python and uses a variety of NLP libraries, including Hugging Face Transformers, NLTK, PKE, and Gradio. This section provides an overview of the code structure, key functions, and the step-by-step process of generating MCQs.

3.4 Code Structure

The code is organized into several sections, each responsible for a specific aspect of the MCQ generation process. The primary components of the code structure are as follows:

1. **Library Installation and Setup**: This section installs the necessary Python libraries and prepares the environment for running the code. Libraries such as Hugging Face Transformers, NLTK, PKE, and Gradio are installed to support text summarization, keyword extraction, question generation, and distractor creation.

2. **Text Summarization**: The code uses the T5 model from Hugging Face to summarize input text. This section initializes the model, preprocesses the text, and generates a summary. The summarization step is crucial for reducing the length of the text and focusing on key information.

3. **Keyword Extraction**: The PKE library is used to extract keywords from the summarized text. The code initializes the keyword extractor, selects candidates based on part-of-speech tags, and identifies the most significant keywords using the MultipartiteRank algorithm. These keywords serve as the basis for question generation.

4. **Question Generation**: The T5 model is used to generate questions from the summarized text and extracted keywords. This section defines the context and correct answer, generates the question using the model, and post-processes the output to ensure clarity and coherence.

5. **Distractor Creation**: The code generates plausible distractors (incorrect answers) using Sense2Vec, WordNet, and ConceptNet. This section finds words with similar meanings, applies the Maximum Marginal Relevance (MMR) technique to ensure diversity among distractors, and refines the distractors based on semantic relationships.

6. **Gradio User Interface**: The Gradio library is used to create a user-friendly interface for generating MCQs. This section sets up a text input for users, a radio

button to choose the distractor generation method, and an output display to show the generated questions, correct answers, and distractors.

3.5 Key Functions and Classes

The code includes several key functions and classes that perform specific tasks in the MCQ generation process:

1. **Summarizer**: This function takes the input text and uses the T5 model to generate a summarized version. It processes the text, encodes it with the tokenizer, and generates a summary with specified parameters.

2. **Get Nouns Multipartite**: This function uses the PKE library to extract keywords from the summarized text. It selects candidates based on part-of-speech tags and ranks them to identify the most significant keywords.

3. **Get Question**: This function generates a question from a given context and correct answer using the T5 model. It encodes the context and answer, generates the question, and post-processes the output.

4. **Get Distractors**: This function generates distractors using Sense2Vec, WordNet, and ConceptNet. It finds words with similar meanings and applies the MMR technique to ensure diversity among the distractors.

5. **Generate Question (Gradio Interface)**: This function creates the Gradio interface and defines the interaction logic. It takes user input, summarizes the text, generates questions, creates distractors, and displays the output in a user-friendly format.

3.6 Example Workflow

The following is a step-by-step workflow illustrating how the code generates MCQs from input text:

1. **Input Text**: Users provide a paragraph or a larger text as input to the system.

2. **Text Summarization**: The input text is summarized using the T5 model, reducing its length and focusing on key information.

3. **Keyword Extraction**: The summarized text is processed to extract keywords that represent the core concepts.

4. **Question Generation**: Using the T5 model, questions are generated from the summarized text and extracted keywords.

5. **Distractor Creation**: Sense2Vec, WordNet, and ConceptNet are used to generate plausible distractors for the questions.

6. **User Interface**: The Gradio interface allows users to interact with the system, input text, generate MCQs, and view the output with questions, correct answers, and distractors.

This comprehensive workflow demonstrates how the code components interact to generate multiple-choice questions. The implementation is designed to be flexible and scalable, allowing educators and content creators to use the system for a variety of educational and training purposes.

Chapter 5: System Requirements

5.1 Software Requirements:

→Jupyter Notebook or Visual Studio Code

→Language: Python

5.2 OS Requirements:

→Windows (7-11), Ubuntu, Mac

→Used: Windows 11

5.3 Hardware Requirements:

→Processor: (Intel Core i3 or AMD Ryzen 3250u CPU or Higher)

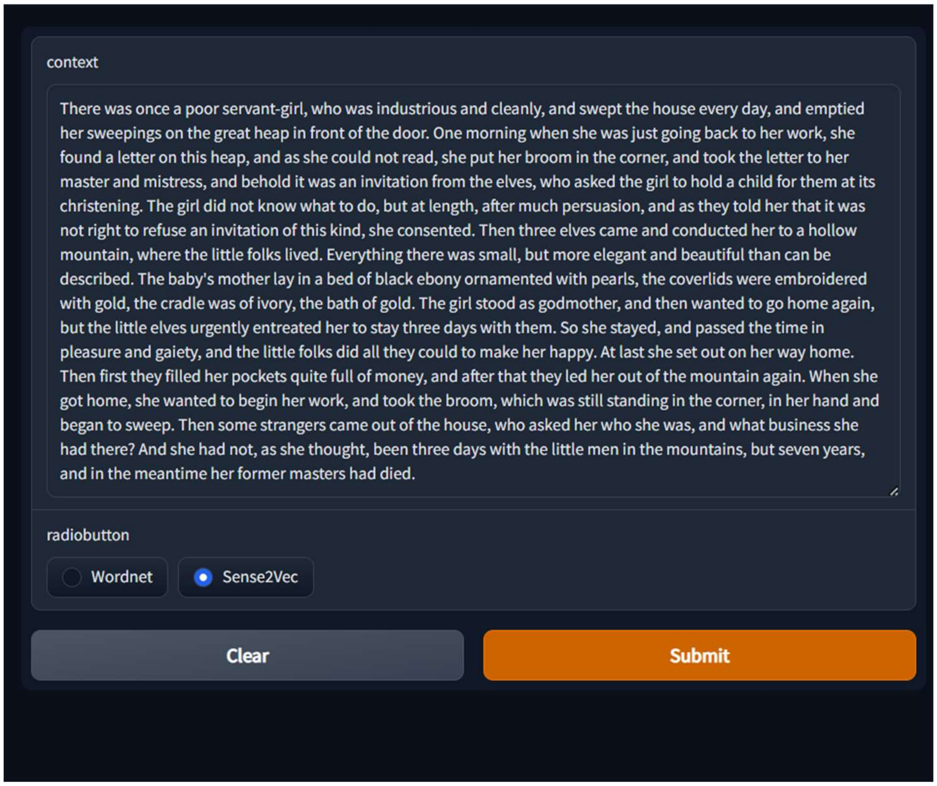
→Used: Intel Core i5 11^{th} Gen

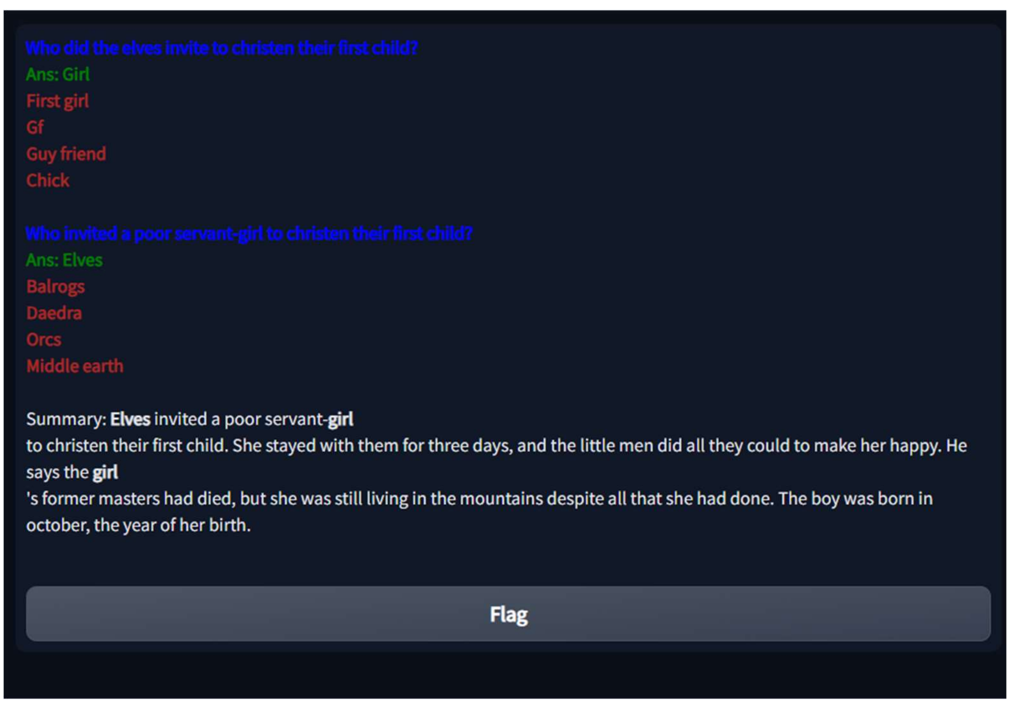
→RAM: (1 GB or more on-board system memory)

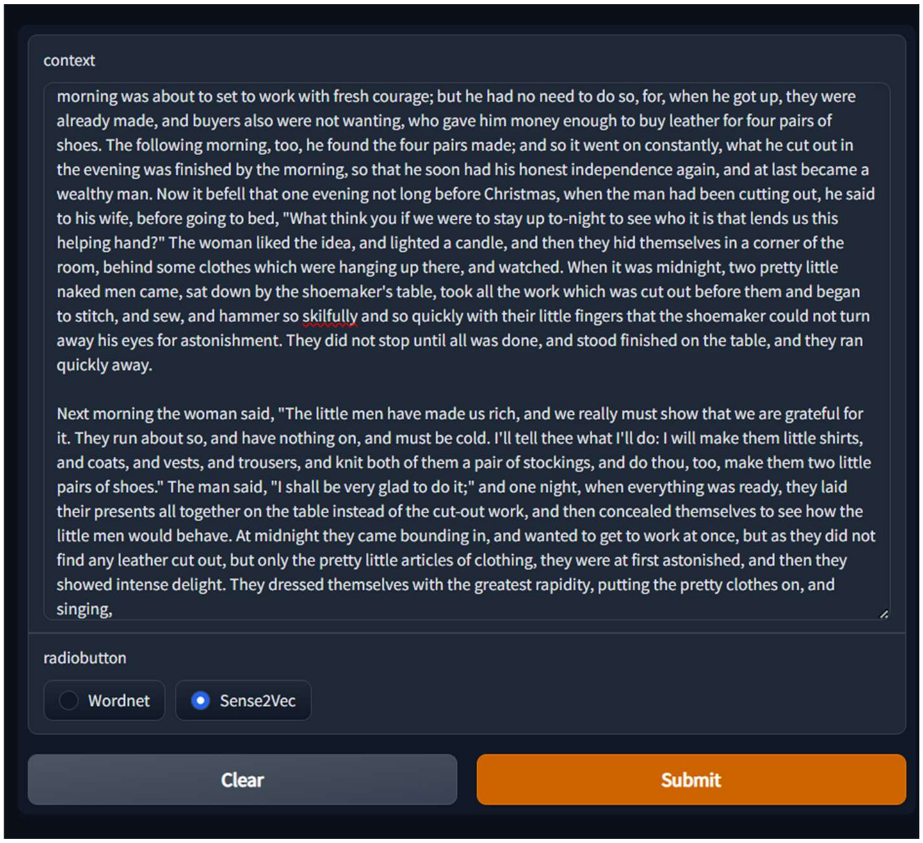
→RAM in the system used: 8GB

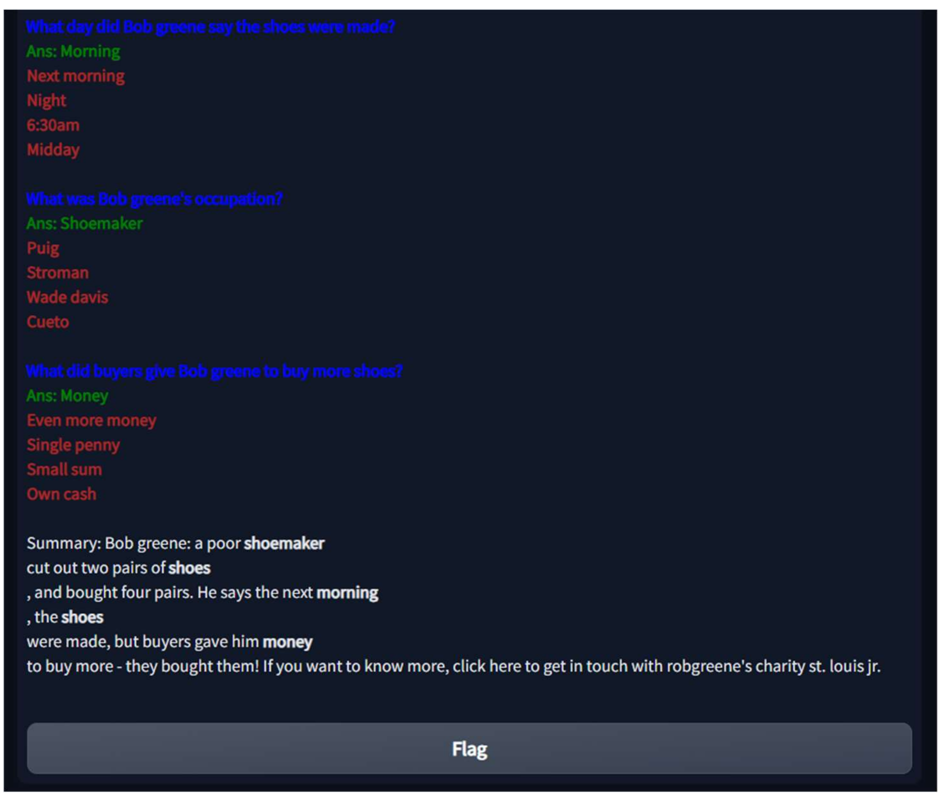
→Disk Space :(0.5-2GB of hard drive space)

Chapter 6: Results






context
morning was about to set to work with fresh courage; but he had no need to do so, for, when he got up, they were already made, and buyers also were not wanting, who gave him money enough to buy leather for four pairs of shoes. The following morning, too, he found the four pairs made; and so it went on constantly, what he cut out in the evening was finished by the morning, so that he soon had his honest independence again, and at last became a wealthy man. Now it befell that one evening not long before Christmas, when the man had been cutting out, he said to his wife, before going to bed, "What think you if we were to stay up to-night to see who it is that lends us this helping hand?" The woman liked the idea, and lighted a candle, and then they hid themselves in a corner of the room, behind some clothes which were hanging up there, and watched. When it was midnight, two pretty little naked men came, sat down by the shoemaker's table, took all the work which was cut out before them and began to stitch, and sew, and hammer so skilfully and so quickly with their little fingers that the shoemaker could not turn away his eyes for astonishment. They did not stop until all was done, and stood finished on the table, and they ran quickly away.
Next morning the woman said, "The little men have made us rich, and we really must show that we are grateful for it. They run about so, and have nothing on, and must be cold. I'll tell thee what I'll do: I will make them little shirts, and coats, and vests, and trousers, and knit both of them a pair of stockings, and do thou, too, make them two little pairs of shoes." The man said, "I shall be very glad to do it;" and one night, when everything was ready, they laid their presents all together on the table instead of the cut-out work, and then concealed themselves to see how the little men would behave. At midnight they came bounding in, and wanted to get to work at once, but as they did not find any leather cut out, but only the pretty little articles of clothing, they were at first astonished, and then they showed intense delight. They dressed themselves with the greatest rapidity, putting the pretty clothes on, and singing,
radiobutton
Wordnet
Sense2Vec
Clear
Submit


What day did Bob greene say the shoes were made?
Ans: Morning
Next morning
Night
6:30am
Midday
What was Bob greene's occupation?
Ans: Shoemaker
Puig
Stroman
Wade davis
Cueto
What did buyers give Bob greene to buy more shoes?
Ans: Money
Even more money
Single penny
Small sum
Own cash
Summary: Bob greene: a poor shoemaker
cut out two pairs of shoes
, and bought four pairs. He says the next morning
, the shoes
were made, but buyers gave him money
to buy more - they bought them! If you want to know more, click here to get in touch with robgreene's charity st. louis jr.
Flag

Chapter 7: Discussions

The automated generation of multiple-choice questions (MCQs) using natural language processing (NLP) represents a significant step forward in educational technology. This project explored a method to automate the creation of MCQs, focusing on text summarization, keyword extraction, question generation, and distractor creation. The results demonstrate that NLP can effectively generate MCQs with a high degree of accuracy and relevance, but several important points arise from this exploration.

7.1 Strengths of the Approach

One of the main strengths of this project is its comprehensive approach to MCQ generation. By leveraging text summarization with the T5 model, the system reduces the input text to its essential components, making it easier to extract keywords and generate meaningful questions. The use of PKE for keyword extraction ensures that the core concepts are identified, providing a solid foundation for question generation.

The question generation component, powered by the T5 model, is a key success factor. It allows for the creation of questions from a given context and correct answer, producing grammatically correct and relevant questions. The flexibility of the T5 model to handle various text types contributes to the system's adaptability.

The distractor creation process, which employs Sense2Vec, WordNet, and ConceptNet, adds to the robustness of the system. By using a combination of these tools, the project generates plausible distractors that make the MCQs challenging yet fair. The Maximum Marginal Relevance (MMR) technique helps maintain diversity among distractors, reducing the chances of overly similar incorrect answers.

7.2Challenges and Limitations

Despite its strengths, the project encountered several challenges and limitations. The first challenge is ensuring that the generated MCQs are of high quality and relevant to the original text. While the system generally performs well, there are instances where the summarized text might omit critical information, impacting the quality of the questions.

Another limitation is the generation of distractors. Although Sense2Vec and WordNet are effective in finding words with similar senses, there is a risk of generating distractors that are too similar to the correct answer or not relevant to the context. The MMR technique helps address this issue, but further refinement is needed to ensure consistent distractor quality.

Additionally, the system's adaptability to different text types and subjects poses a challenge. While the approach is designed to be flexible, certain specialized or technical texts may require additional processing or customization to generate accurate MCQs. This limitation indicates that further work is needed to enhance the system's versatility.

7.3 Implications for Education and Training

The ability to automate MCQ generation has significant implications for education and training. By reducing the time and effort required to create questions, educators can focus more on pedagogy and student engagement. Automated MCQ generation can also facilitate personalized learning experiences and adaptive assessments, allowing for more dynamic educational environments.

Furthermore, the project demonstrates the potential for integrating NLP-based systems into educational platforms. As technology continues to evolve, such systems could play a crucial role in supporting teachers, trainers, and content creators in developing high-quality educational materials.

Chapter 8: Conclusion

This project explored the automated generation of multiple-choice questions (MCQs) using natural language processing (NLP), aiming to reduce the time and effort required for creating educational assessments. By leveraging a combination of text summarization, keyword extraction, question generation, and distractor creation, the project demonstrated that it is possible to generate high-quality MCQs from various textual sources.

One of the key achievements of this project is the successful implementation of an NLP-based system for MCQ generation. The system can summarize text, extract significant keywords, generate relevant questions, and create plausible distractors. The use of advanced NLP models like T5 and tools such as Sense2Vec, WordNet, and ConceptNet contributed to the system's robustness and versatility.

The project identified several strengths, including the comprehensive approach to MCQ generation, the flexibility of the question generation model, and the effective distractor creation process. These strengths suggest that the system can be a valuable tool for educators and content creators, enabling them to produce MCQs with greater efficiency.
However, the project also revealed some challenges and limitations. Ensuring consistent quality in the generated MCQs, particularly in terms of question clarity and distractor plausibility, remains an ongoing concern. Additionally, the system's adaptability to different text types and subject areas requires further exploration.

Despite these challenges, the project's results demonstrate the potential for automated MCQ generation to play a significant role in education and training. The ability to automate the creation of educational materials can have a positive impact on teaching and learning, offering personalized and adaptive experiences for students.

Recommendations for Future Work

Based on the findings of this project, several recommendations for future work can be made:

1. **Improving Distractor Quality**: Further research into advanced techniques for distractor generation is needed to ensure distractors are challenging but not misleading.
2. **Expanding Adaptability**: The system's adaptability to a wider range of text sources and subject areas should be explored to make it more versatile for different educational contexts.
3. **Integrating with Educational Platforms**: Integrating the MCQ generation system with existing educational platforms could enhance its usability and reach.

Overall, this project contributes to the field of educational technology, demonstrating that NLP can automate MCQ generation and potentially transform how educational assessments are created. By addressing the identified challenges and pursuing the recommended future work, the system can become an even more effective tool for educators and content creators.

REFERENCES

- Bird, S., Klein, E., & Loper, E. (2009). "Natural Language Processing with Python." O'Reilly Media, Sebastopol, CA.
- Jurafsky, D., & Martin, J. H. (2008). "Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition." Prentice Hall, Upper Saddle River, NJ.
- Heilman, M., & Smith, N. A. (2010). "Good question! Statistical ranking for question generation." Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics, pp. 609-617.
- Mitkov, R., Ha, L. A., & Karamanis, N. (2006). "A computer-aided environment for generating multiple-choice test items." Natural Language Engineering, 12(2), 177-194.
- Manning, C. D., & Schütze, H. (1999). "Foundations of Statistical Natural Language Processing." In P. Cook (Ed.), MIT Press, Cambridge, MA, pp. 1-80.
- Paroubek, P. (Ed.). (2008). "Proceedings of the 6th International Natural Language Generation Conference." Association for Computational Linguistics.
- Gradio Documentation: Resources for building interactive user interfaces for machine learning applications.
  Website: https://www.gradio.app/
- Hugging Face Transformers Documentation: Provides information on various transformer-based models, including BERT and T5.
  Website: https://huggingface.co/transformers/
- NLTK Documentation: A guide to the Natural Language Toolkit library, covering tokenization, WordNet, and other NLP tasks.
  Website: https://www.nltk.org/
- Sentence Transformers Documentation: Provides information on using sentence-based transformer models for similarity and semantic embeddings.
  Website: https://www.sbert.net/EMNLP 2006 Proceedings: Proceedings from the 2006 Conference on Empirical Methods in Natural Language Processing.
  Website: https://aclanthology.org/venues/emnlp/2006/
- NAACL-HLT 2010 Proceedings: Proceedings from the 2010 North American Chapter of the Association for Computational Linguistics: Human Language Technologies.
  Website: https://aclanthology.org/venues/naacl-hlt/2010/