Attention-Based Neural Question Generation for Enhanced Reading Comprehension

[1]R. Deepthi, [2]L.Prasanna, [3]M.Usha Sri, [4]A. Jwala Sri Priya, [5]K. Bhavana Sudha
[1]Assistant Professor , [2,3,4,5]UG Students, [1,2,3,4,5]Department of Computer Science & Engineering, Geethanjali Institute Of Science And Technology, Nellore, India

Abstract

This project addresses the task of Automatic Question Generation (AQG) for reading comprehension by proposing a trainable, end-to-end neural architecture based on attention-driven sequence-to-sequence learning. In contrast to conventional AQG approaches that rely on rule-based systems or complex natural language processing (NLP) pipelines, our model offers a streamlined and flexible solution that eliminates the need for handcrafted features. We examine the influence of input granularity comparing sentence-level versus paragraph-level encoding on the quality and complexity of generated questions. Experimental results show that our model outperforms state-of-the-art rule-based systems across automatic evaluation metrics, producing questions with superior lexical diversity and syntactic structure. Furthermore, human evaluations confirm that the generated questions are more grammatically natural, fluently phrased, and cognitively demanding, often requiring deeper reasoning beyond surface-level information. These findings highlight the effectiveness of our attention-based model in generating contextually rich and educationally valuable questions for reading comprehension tasks.

Keywords: NLP, AQG, AQuAD dataset

Introduction

Generating meaningful questions from a paragraph is a crucial task in education, intelligent tutoring systems, and reading comprehension. This project focuses on building a neural question generation system using the T5 Transformer model. It automatically generates natural and context-aware questions from a given passage, helping users assess understanding and improve learning outcomes.

Traditional question generation methods are rule-based and limited in flexibility. This project replaces them with a deep learning-based approach that uses sequence-to-sequence learning with attention to generate fluent and relevant questions. The model is trained using the SQuAD dataset, which contains thousands of real-world question-answer pairs from Wikipedia articles.

A user-friendly Streamlit web app has been developed for interactive use. Users can input any paragraph and receive multiple AI-generated questions instantly. This can be used by educators, students, or developers to evaluate text comprehension and build advanced NLP applications.

In the following sections, you'll learn about the data preprocessing pipeline, the neural architecture used, and how questions are evaluated for quality. This project moves us closer to smarter reading tools and AI-powered educational platforms.

Motivation

The inspiration for this project arises from the growing need for intelligent educational tools that can support and enhance reading comprehension. In today’s information-rich world, students and learners are often overwhelmed by vast amounts of text. One proven way to improve comprehension and retention is through asking questionsa skill traditionally dependent on teachers or human-crafted assessmentsHowever, manually generating high-quality, relevant, and answer-focused questions from text is time-consuming and labor-intensive. This creates a major bottleneck in scalable learning environments, online tutoring systems, and automated reading comprehension platforms.

Imagine a world where a student can paste any paragraph from their textbook into a system, and within seconds, receive a set of insightful questions to test their understanding. Or where educators can auto-generate practice assessments tailored to the exact content their students are studying. This is the future we envision and Neural Question Generation (NQG) models make it possible.

Recent advances in natural language processing, particularly sequence-to-sequence architectures and attention mechanisms, have opened new doors for machines to not just answer questions, but also ask them. Unlike traditional rule-based systems, neural models can learn context, semantics, and phrasing patterns — resulting in more natural, varied, and meaningful questions.

By leveraging models like T5 and attention-based encoders, this project explores how deep learning can be used to transform passive reading into an active, inquiry-driven experience.

Objective

The objective of the "Neural Question Generation for Reading Comprehension Application" project is to design and implement an intelligent system capable of automatically generating meaningful and contextually relevant questions from textual passages. This addresses the growing demand for scalable solutions in education, intelligent tutoring systems, and machine comprehension platforms.

To achieve this, the project leverages the T5 (Text-to-Text Transfer Transformer) architecture a powerful sequence-to-sequence model pre-trained on large-scale language tasks. The model is fine-tuned on question-answer datasets like SQuAD, enabling it to understand the semantics of a given paragraph and generate grammatically correct, answer-focused questions. Unlike rule-based approaches, this neural method dynamically learns language patterns and contextual cues, enhancing the naturalness and diversity of the generated questions.

The primary goal is to develop a web-based interface where users can input a paragraph and receive high-quality questions generated in real-time. This involves implementing a clean preprocessing pipeline, fine-tuning the transformer model, and evaluating its performance through metrics like BLEU, METEOR, and human judgment. Ultimately, this system aims to support educational content creation, test generation, and automated reading comprehension tools by bridging the gap between text understanding and intelligent questioning.

Literature Review

It review several research studies to gain insight and understanding of the techniques proposed for automatic question generation (QG), especially in the context of reading comprehension. Question Generation is an important area of Natural Language Processing (NLP) that aims to convert input text into natural questions. This literature review covers the evolution from rule-based systems to advanced deep learning models, highlighting key contributions that paved the way for end-to-end question generation using neural networks

Few-shot Question Generation for Reading Comprehension

“Yin Poon, John S. Y. Lee, Yu Yan Lam”, 2024

The researchers compared four different models, including a pipeline model based on DuReader integrated with UNIMO and a traditional sequence-to-sequence model. Their approach focused on the effectiveness of prompt design and its impact on question quality. The results showed that carefully crafted prompts led to superior question generation, especially in low-resource settings. However, the study relied heavily on manual evaluation, and the broader impact of these improvements was not quantitatively measured.

2.2.2 Improving Question Generation with Augmentation and Ranking

“Nischal Ashok Kumar, Nigel Fernandez, Zichao Wang, Andrew Lan “,2023

Using Codex for diverse question generation, the study produced a set of candidate questions and applied dual ranking to select the most relevant ones. This hybrid method led to a +5% improvement in ROUGE-L over BART and showed the ability to generate more implicit and nuanced questions. Despite these advancements, the combined approach yielded only limited performance gains when integrating augmentation and ranking strategies.

QG-RCA: Modeling How and What to Ask

“Bilal Ghanem, Lauren Lutz Coleman, Julia Rivard Dexter “,2022

QG-RCA introduced a novel T5-based dual-component model focused on improving both the structure and content of generated questions. The HTA (How To Ask) component was trained on general question patterns from datasets like SQuAD and CosmosQA, while the WTA (What To Ask) module was fine-tuned on skill-specific data from the SBRCS dataset. This approach excelled at generating inferential questions and performed well even with limited training data. However, the system faced issues with semantic consistency (SCM errors) and the subjectivity involved in evaluating skill-based question quality (SVQ).

Proposed Model

The proposed system aims to enhance the process of automatic question generation using advanced neural network architectures and pre-trained language models. Unlike traditional rule-based or simple Seq2Seq models, this system leverages transformer-based models such as T5 (Text-to-Text Transfer Transformer) or BART (Bidirectional and Auto-Regressive Transformers), which are fine-tuned specifically for the task of question generation

Key Features of the Proposed System:

1. **Context-Aware Generation:** The system takes a passage or a paragraph along with a highlighted answer span (optional) as input and generates a meaningful and contextually relevant question. It understands the context and forms grammatically correct and semantically accurate questions.
2. **Use of Pre-trained Language Models:** By utilizing state-of-the-art models like T5 or BART, the system benefits from vast linguistic knowledge encoded in these models, resulting in better question formation and improved fluency.
3. **Question Diversity:** The system is designed to generate various types of questions, including factual (who, what, when), descriptive (explain, describe), and inferential questions, depending on the nature of the input passage.
4. **Minimal Need for Hand-crafted Rules:** The reliance on neural models reduces the need for manually defined rules or templates, thus improving scalability and adaptability to different domains (e.g., education, healthcare, legal).
5. **Training on QA Datasets:** The system is fine-tuned on benchmark datasets such as SQuAD (Stanford Question Answering Dataset), NewsQA, or Natural Questions to ensure high-quality question generation based on real-world data.
6. **Post-processing and Evaluation:** To enhance output quality, generated questions are passed through grammar checkers and ranked using BLEU or ROUGE scores to evaluate relevance and fluency.

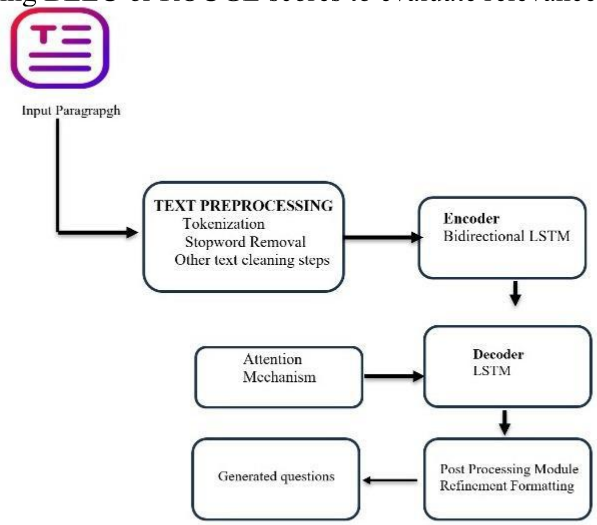


Fig.1. Proposed system Model

User Interface (Streamlit Web App):

A web-based interface built using Streamlit. Allows users to input a paragraph or sentence. Easy to use, no programming skills required. Displays the generated questions directly on the page.

Preprocessing Module:
Cleans and prepares the input text. Splits long paragraphs into smaller chunks (to fit model limits). Ensures the input is in the proper format before sending to the model.

Neural Question Generation Model :
Uses a fine-tuned T5 Transformer model (valhalla/t5-small-e2e-qg). Based on the encoder-decoder architecture. Takes cleaned text and generates questions using learned patterns. Trained on datasets like SQuAD to understand how questions relate to content.

Post processing Module:
Cleans up the model's output. Removes duplicate or low-quality questions. Formats the results clearly for user readability.

Output Display:
Shows the final list of generated questions. Ensures that users can easily view and analyse them. Useful for educational apps, quiz creation, and reading comprehension.
A Activity diagram (also known as a workflow) provides a graphic overview of the business process. Using standardized symbols and shapes, the workflow shows step by step how your work is completed from start to finish. It also shows who is responsible for work at what point in the process. Designing a workflow involves first conducting a thorough workflow analysis, which can expose potential weaknesses. A workflow analysis can help you define, standardize and identify critical areas of your process. An event is created as an activity diagram encompassing a group of nodes associated with edges. To model the behavior of activities, they can be attached to any modeling element. It can model use cases, classes, interfaces, components, and collaborations. It mainly models processes and workflows.

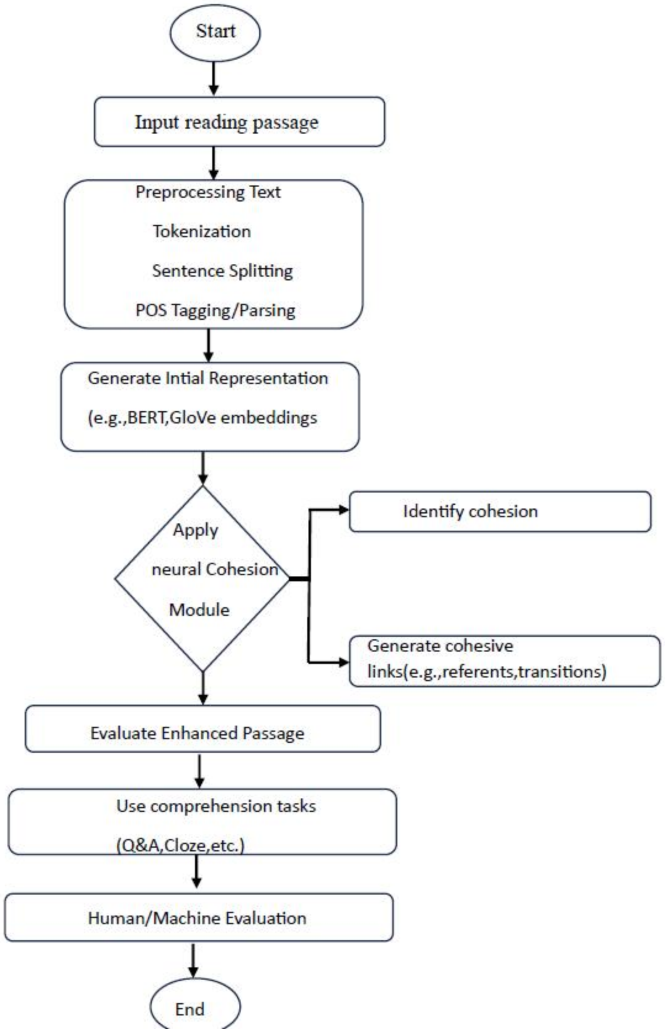


Fig.2. Activity Diagram

SYSTEM IMPLEMENTATION

SYSTEM MODULES

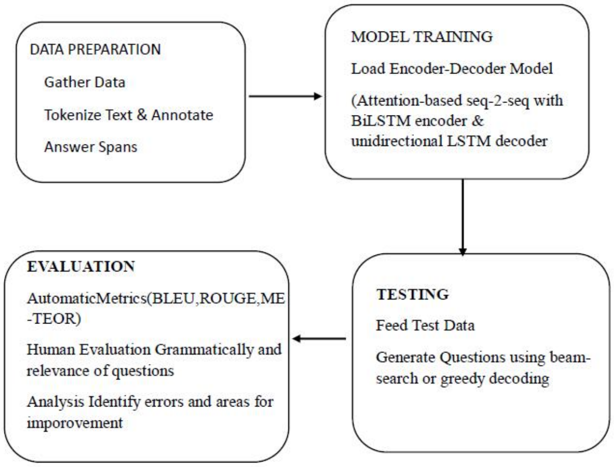

DATA PREPARATION
Gather Data
Tokenize Text & Annotate
Answer Spans
MODEL TRAINING
Load Encoder-Decoder Model
(Attention-based seq-2-seq with BiLSTM encoder & unidirectional LSTM decoder
TESTING
Feed Test Data
Generate Questions using beam-search or greedy decoding
EVALUATION
AutomaticMetrics(BLEU,ROUGE,ME-TEOR)
Human Evaluation Grammatically and relevance of questions
Analysis Identify errors and areas for imporovement

Fig.3. System modules

Data Preparation:

Gather Data: Collect question-answer pairs along with source passages from reading comprehension datasets such as SQuAD v1.1. Each entry includes a passage, an answer span, and a corresponding human-generated question.

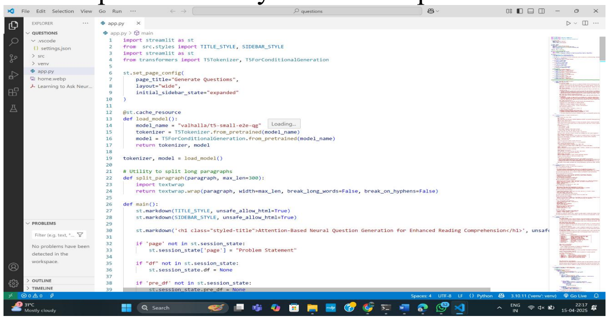
Preprocess Data: Tokenize passages and questions using tools like spaCy or NLTK. Normalize text (e.g., lowercasing, removing punctuation), and annotate answer spans using a BIO tagging scheme. Convert the data into suitable input-output pairs for training a sequence-to-sequence model.
Load Encoder-Decoder Model: Utilize an attention-based Seq2Seq model with a bidirectional LSTM encoder and a unidirectional LSTM decoder. The encoder processes the context (passage), while the decoder generates the corresponding question.
Answer Encoding: The location of the answer in the passage is marked using special tags. These are embedded and concatenated with word embeddings to provide answer-aware input to the encoder.
Optimization: Train the model using the optimizer with a learning rate (e.g., 0.001). Apply dropout to prevent overfitting and teacher forcing to guide learning during training

Testing:

Feed Test Data: Input unseen passages and target answer spans into the trained model.
Generate questions: The model outputs one or more questions using greedy decoding or beam search (typically with beam size = 5).
Pre Processing : Clean up generated text for punctuation and grammar, and filter out repetitive or degenerate questions.
Evaluation:
Automatic Metrics: Evaluate generated questions using BLEU, ROUGE, and METEOR scores, comparing them to reference (human-written) questions.
Human Evaluation: Assess the grammaticality, fluency, relevance, and difficulty level of the generated questions through manual review.
Analysis: Review error cases (e.g., ungrammatical questions or irrelevant outputs) to identify areas for model refinement, such as improving answer encoding or training on more diverse datasets.

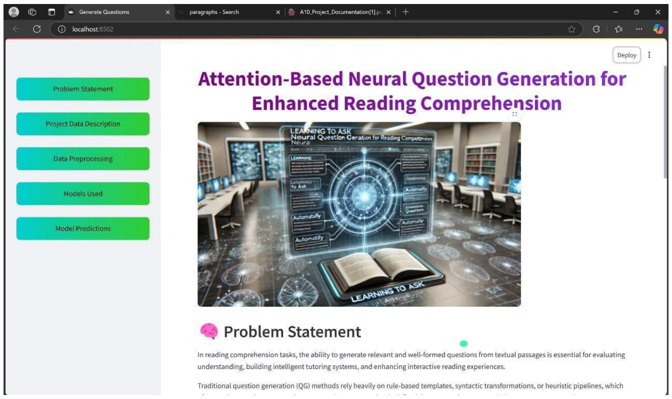
Results

The execution of the process will be explained clearly with the help of continuous screenshots.



```python
import streamlit as st
from  src.styles import TITLE_STYLE, SIDEBAR_STYLE
import streamlit as st
from transformers import T5Tokenizer, T5ForConditionalGeneration

st.set_page_config(
    page_title="Generate Questioms",
    layout="wide",
    initial_sidebar_state="expanded"
)

@st.cache_resource
def load_model():
    model_name = "valhalla/t5-small-e2e-qg"
    tokenizer = T5Tokenizer.from_pretrained(model_name)
    model = T5ForConditionalGeneration.from_pretrained(model_name)
    return tokenizer, model

tokenizer, model = load_model()

# Utility to split long paragraphs
def split_paragraph(paragraph, max_len=300):
    import textwrap
    return textwrap.wrap(paragraph, width=max_len, break_long_words=False, break_on_hyphens=False)

def main():
    st.markdown(TITLE_STYLE, unsafe_allow_html=True)
    st.markdown(SIDEBAR_STYLE, unsafe_allow_html=True)

    st.markdown('<h1 class="styled-title">Attention-Based Neural Question Generation for Enhanced Reading Comprehension</h1>', unsafe

    if 'page' not in st.session_state:
        st.session_state['page'] = "Problem Statement"

    if "df" not in st.session_state:
        st.session_state.df = None

    if 'pre_df' not in st.session_state:
        st.session_state.pre_df = None
```

Fig.4. Launching Streamlit Application.



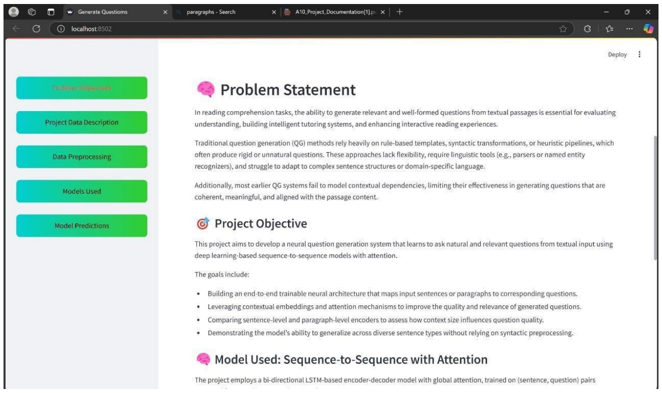
Attention-Based Neural Question Generation for Enhanced Reading Comprehension

🧠 Problem Statement

In reading comprehension tasks, the ability to generate relevant and well-formed questions from textual passages is essential for evaluating understanding, building intelligent tutoring systems, and enhancing interactive reading experiences.

Traditional question generation (QG) methods rely heavily on rule-based templates, syntactic transformations, or heuristic pipelines, which

Fig.5. A Streamlit UI showcasing an attention-based neural question generator for reading Comprehension



Problem Statement

In reading comprehension tasks, the ability to generate relevant and well-formed questions from textual passages is essential for evaluating understanding, building intelligent tutoring systems, and enhancing interactive reading experiences.

Traditional question generation (QG) methods rely heavily on rule-based templates, syntactic transformations, or heuristic pipelines, which often produce rigid or unnatural questions. These approaches lack flexibility, require linguistic tools (e.g., parsers or named entity recognizers), and struggle to adapt to complex sentence structures or domain-specific language.

Additionally, most earlier QG systems fail to model contextual dependencies, limiting their effectiveness in generating questions that are coherent, meaningful, and aligned with the passage content.

Project Objective

This project aims to develop a neural question generation system that learns to ask natural and relevant questions from textual input using deep learning-based sequence-to-sequence models with attention.

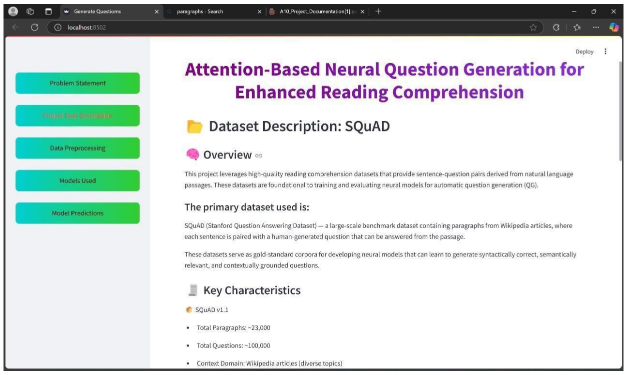
The goals include:

- Building an end-to-end trainable neural architecture that maps input sentences or paragraphs to corresponding questions.
- Leveraging contextual embeddings and attention mechanisms to improve the quality and relevance of generated questions.
- Comparing sentence-level and paragraph-level encoders to assess how context size influences question quality.
- Demonstrating the model's ability to generalize across diverse sentence types without relying on syntactic preprocessing.

Model Used: Sequence-to-Sequence with Attention

The project employs a bi-directional LSTM-based encoder-decoder model with global attention, trained on (sentence, question) pairs

Fig.6. Streamlit app interface highlighting the problem statement, objectives, and model Architecture for the question generation system.



Attention-Based Neural Question Generation for Enhanced Reading Comprehension

Dataset Description: SQuAD

Overview

This project leverages high-quality reading comprehension datasets that provide sentence-question pairs derived from natural language passages. These datasets are foundational to training and evaluating neural models for automatic question generation (QG).

The primary dataset used is:

SQuAD (Stanford Question Answering Dataset) — a large-scale benchmark dataset containing paragraphs from Wikipedia articles, where each sentence is paired with a human-generated question that can be answered from the passage.

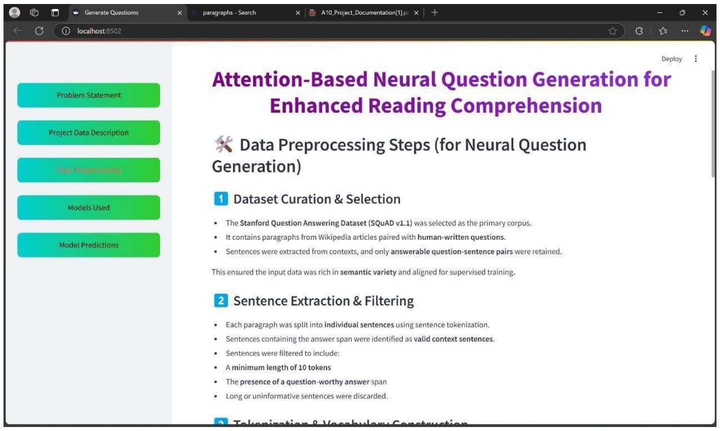
These datasets serve as gold-standard corpora for developing neural models that can learn to generate syntactically correct, semantically relevant, and contextually grounded questions.

Key Characteristics

SQuAD v1.1

- Total Paragraphs: ~23,000
- Total Questions: ~100,000
- Context Domain: Wikipedia articles (diverse topics)

Fig.7. Streamlit interface presenting the SQuAD dataset overview and key characteristics for neural question generation.



Attention-Based Neural Question Generation for Enhanced Reading Comprehension

Data Preprocessing Steps (for Neural Question Generation)

1 Dataset Curation & Selection

- The **Stanford Question Answering Dataset (SQuAD v1.1)** was selected as the primary corpus.
- It contains paragraphs from Wikipedia articles paired with **human-written questions.**
- Sentences were extracted from contexts, and only **answerable question-sentence pairs** were retained.

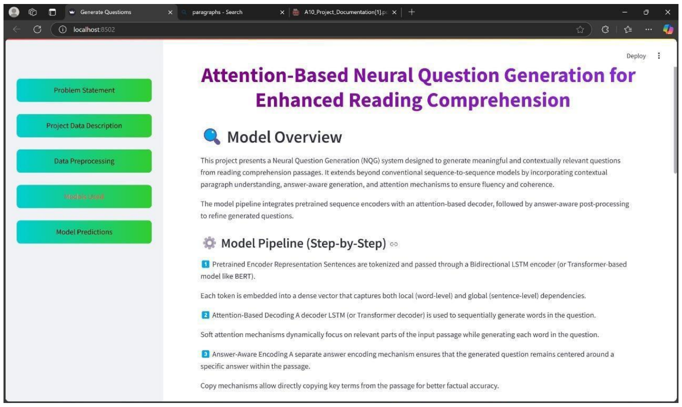
This ensured the input data was rich in **semantic variety** and aligned for supervised training.

2 Sentence Extraction & Filtering

- Each paragraph was split into **individual sentences** using sentence tokenization.
- Sentences containing the answer span were identified as **valid context sentences.**
- Sentences were filtered to include:
- A **minimum length of 10 tokens**
- The **presence of a question-worthy answer** span
- Long or uninformative sentences were discarded.

3 Tokenization & Vocabulary Construction

Fig.8. Streamlit interface outlining data preprocessing steps for neural question generation, including curation, extraction, and tokenization,etc.



Attention-Based Neural Question Generation for Enhanced Reading Comprehension

🔍 Model Overview

This project presents a Neural Question Generation (NQG) system designed to generate meaningful and contextually relevant questions from reading comprehension passages. It extends beyond conventional sequence-to-sequence models by incorporating contextual paragraph understanding, answer-aware generation, and attention mechanisms to ensure fluency and coherence.

The model pipeline integrates pretrained sequence encoders with an attention-based decoder, followed by answer-aware post-processing to refine generated questions.

⚙️ Model Pipeline (Step-by-Step)

1️⃣ Pretrained Encoder Representation Sentences are tokenized and passed through a Bidirectional LSTM encoder (or Transformer-based model like BERT).

Each token is embedded into a dense vector that captures both local (word-level) and global (sentence-level) dependencies.

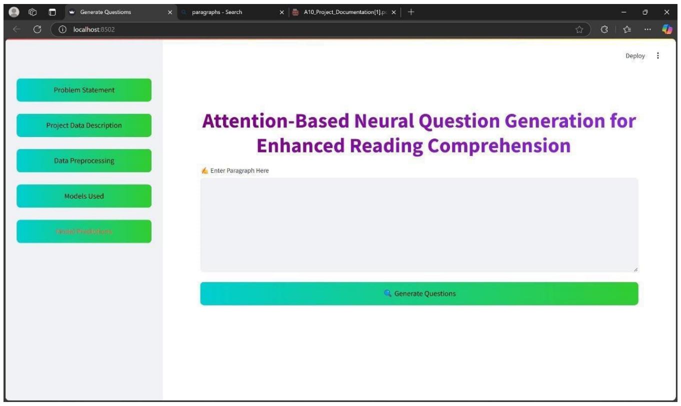
2️⃣ Attention-Based Decoding A decoder LSTM (or Transformer decoder) is used to sequentially generate words in the question.

Soft attention mechanisms dynamically focus on relevant parts of the input passage while generating each word in the question.

3️⃣ Answer-Aware Encoding A separate answer encoding mechanism ensures that the generated question remains centered around a specific answer within the passage.

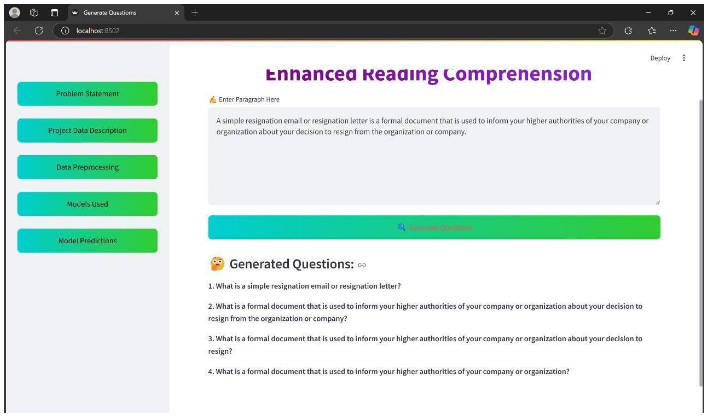
Copy mechanisms allow directly copying key terms from the passage for better factual accuracy.

Fig.9. Streamlit interface displaying the model overview and pipeline for neural question generation.



Attention-Based Neural Question Generation for Enhanced Reading Comprehension

Fig.10. Streamlit interface for neural question generation with input text box and navigation for data and model insights.



Enhanced Reading Comprehension

A simple resignation email or resignation letter is a formal document that is used to inform your higher authorities of your company or organization about your decision to resign from the organization or company.

😳 Generated Questions:

1. What is a simple resignation email or resignation letter?
2. What is a formal document that is used to inform your higher authorities of your company or organization about your decision to resign from the organization or company?
3. What is a formal document that is used to inform your higher authorities of your company or organization about your decision to resign?
4. What is a formal document that is used to inform your higher authorities of your company or organization?

Fig.11. Streamlit interface displaying generated questions from an input paragraph using an attention-based question generation model.

Conclusion

This project explores the use of deep learning for neural question generation (QG) using the SQuAD dataset and a sequence-to-sequence model with attention. Leveraging the pretrained T5-small-e2e-qg model, we developed a system that generates coherent, contextually relevant questions from natural language paragraphs. A Streamlit-based web interface allows users to input text and receive multiple generated questions, with the system reprocessing the input before applying the T5 model to produce diverse queries. This implementation demonstrates how transformer-based models can automate the creation of educational and evaluative content, enhancing intelligent tutoring systems, reading comprehension tools, and interactive learning platforms. The project's success highlights the effectiveness of attention-based neural architectures in bridging the gap between text understanding and question formation, paving the way for smarter, language-aware educational tools.

Future Scope

The field of Neural Question Generation (NQG) offers substantial potential for advancing automated reading comprehension systems. Future research can focus on integrating more sophisticated language models, such as large-scale transformers like GPT and T5, to enhance the fluency, relevance, and diversity of generated questions. Incorporating context-aware and multi-hop reasoning can enable the generation of deeper, more analytical questions. Personalized question generation, tailored to individual learner proficiency, can further support adaptive learning platforms. The integration of NQG with educational technologies, chatbots, and intelligent tutoring systems could significantly impact personalized education and assessment. Additionally, advancements in multilingual question generation and domain adaptation will extend the applicability of these models across various languages and specialized fields such as medicine and law. Ensuring robustness through bias reduction, factual accuracy, and human-in-the-loop evaluation will be critical to the future development of effective NQG systems.

References

1. Du, X., Shao, J., & Cardie, C. (2017). Learning to Ask: Neural Question Generation for Reading Comprehension. Proceedings of the 55th Annual Meeting of the Association for ComputationalLinguistics 1342–1352.
2. Zhou, Q., Yang, N., Wei, F., Tan, C., Bao, H., & Zhou, M. (2017). Neural Question Generation from Text: A Preliminary Study. National CCF Conference on Natural Language Processing and Chinese Computing, 662–671.
3. Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., ... & Zettlemoyer, L. (2020). BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. Proceedings of ACL, 7871–7880.
4. Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural Computation, 9(8), 1735–1780.
5. Chen, D., Bolton, J., & Manning, C. D. (2016). A thorough examination of the CNN/Daily Mail reading comprehension task. ACL, 2358–2367.
6. Rajpurkar, P., Zhang, J., Lopyrev, K., & Liang, P. (2016). SQuAD: 100,000+ questions for machine comprehension of text. EMNLP, 2383–2392.
7. Hermann, K. M., Kocisky, T., Grefenstette, E., Espeholt, L., Kay, W., Suleyman, M., & Blunsom, P. (2015). Teaching machines to read and comprehend. Advances in Neural Information Processing Systems (NIPS), 1693–1701.
8. Serban, I. V., García-Durán, A., Gulcehre, C., Ahn, S., Chandar, S., Courville, A., & Bengio, Y. (2016). Generating questions with recurrent neural networks. ACL, 588– 598.
9. Iyer, S., Konstas, I., Cheung, A., & Zettlemoyer, L. (2016). Summarizing source code using a neural attention model. ACL, 2073–2083.
10. Heilman, M., & Smith, N. A. (2010). Good question! Statistical ranking for question generation. NAACL-HLT, 609–617.
11. Mazidi, K., & Nielsen, R. D. (2014). Linguistic considerations in automatic question generation. ACL, 321–326