INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY, HYDERABAD

A PROJECT REPORT ON

Emphasizing Seq2seq architecture using T5 and BART models in Question Answering

Submitted in partial fulfillment of the requirements of the degree of

Bachelor Of Technology

in

Modern Machine Learning

For the academic year 2024-2025

Submitted by :

Amit Das	UGMR20230001
Vibhor Joshi	UGMR20230003
Shilpa Tichkule	UGMR20230009
Medhavi Nasare	UGMR20230011
Sanika Nandurkar	UGMR20230036

Under the guidance of

Amit Pandey

Department Of Data Science, Iot , Cyber Security

G. H. RAISONI COLLEGE OF ENGINEERING, NAGPUR

Abstract : *Sequence-to-sequence (Seq2Seq) models have revolutionized the field of natural language processing (NLP) by enabling machines to handle complex tasks involving variable length input and output sequences. These models, introduced by Google in 2014, have found widespread applications in areas such as machine translation, text summarization, and question answering. This report provides an overview of the Seq2Seq architecture based Questioning and Answering implementation project, its core components, and its recent advancements, particularly the attention mechanism. We also discuss the various applications of Seq2Seq models in NLP and their potential for future developments for references.*

I. Introduction

The rapid advancements in deep learning have paved the way for more sophisticated and effective approaches to NLP tasks. Among these, the sequence-to-sequence (Seq2Seq) architecture has emerged as a powerful framework for handling complex tasks involving variable-length input and output sequences. Introduced by Google in 2014, Seq2Seq models have demonstrated remarkable performance in a wide range of applications, including machine translation, text summarization, and question answering.

A. Seq2Seq Architecture

At the center of the Seq2Seq model lies the encoder-decoder framework as shown in figure1. The encoder processes the input sequence and compresses its information into a fixed-size context vector, often utilizing recurrent neural networks (RNNs) or long short-term memory networks (LSTMs). This context vector encapsulates the essential features of the input sequence, which the decoder then uses to generate the output sequence step-by-step. The decoder, similar to the encoder, can also employ RNNs or LSTMs, and it generates each element of the output based on the context vector and the previously generated elements.

One of the significant advancements in Seq2Seq models is the introduction of the attention mechanism. This mechanism allows the model to focus on different parts of the input sequence when generating each element of the output, improving its ability to handle long sequences and enhancing its performance on tasks like question answering. In the context of question answering, for example, the model can learn to attend to specific segments of the input text that are most relevant to the question being asked, thereby improving the accuracy of the generated answers.

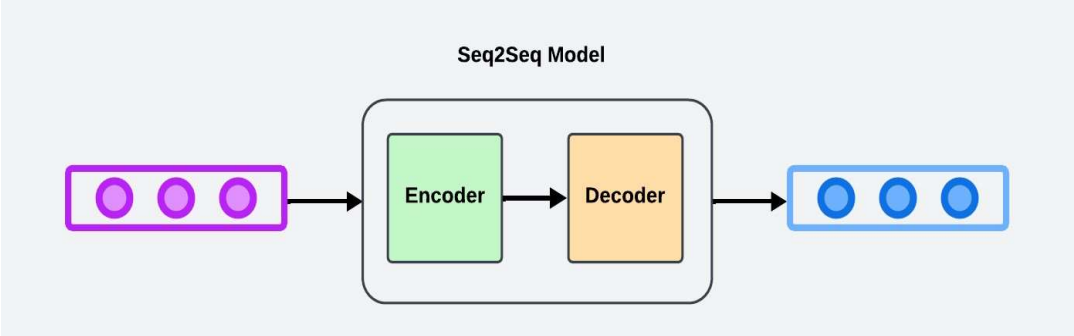

Figure 1: Seq2seq model with encoder- decoder framework.

- **Why is Seq2Seq Important ?**

Sequence-to-sequence (Seq2Seq) models have revolutionized the field of natural language processing (NLP) by enabling machines to handle complex tasks involving variable-length input and output sequences. These models have had a significant impact on various NLP tasks, making them an essential tool in the field. Seq2Seq models are highly flexible, capable of handling a wide range of tasks such as machine translation, text summarization, and image captioning, as well as variable-length input and output sequences. This flexibility makes them highly versatile and applicable to many real-world problems.

Another key aspect of Seq2Seq models is their ability to effectively handle sequential data such as natural language, speech, and time series data. They can capture the dependencies and patterns within sequences, making them powerful tools for processing and generating sequential information. The attention mechanism, a key component of Seq2Seq models, allows the model to focus on specific parts of the input sequence when generating the output. This helps improve performance, especially for long input sequences, by enabling the model to selectively attend to relevant information.

The economic impact of Seq2Seq models can be substantial. For instance:

- **Cost Savings:** Companies using automated translation and transcription services can save millions in labor costs. For example, businesses that previously relied on human translators can reduce costs by up to 80% by adopting automated solutions.
- **Increased Revenue:** Enhanced customer engagement through chatbots and improved user experiences can lead to higher conversion rates. Companies like eBay have reported improved sales through better product descriptions generated by Seq2Seq models, translating to increased revenue.
- **Market Expansion:** With improved translation capabilities, businesses can enter new markets more efficiently, potentially increasing their market share and overall revenue.
- **Efficiency Gains:** Organizations utilizing Seq2Seq models for summarization and content generation can significantly reduce the time spent on these tasks, translating into faster project completions and the ability to take on more work.

In short, Seq2Seq models not only enhance the performance of various NLP tasks but also provide significant economic benefits by reducing costs, increasing revenue, and improving efficiency across multiple industries. The ongoing advancements in this technology promise to further amplify these benefits in the future.

B. Applications

Seq2Seq models have found applications across various domains in NLP. In question answering systems, they are used to generate answers by locating relevant spans in context paragraphs, leveraging datasets like the Stanford Question Answering Dataset (SQuAD) for training and evaluation. These models not only facilitate direct question answering but also contribute to educational tools and conversational agents, where generating contextually relevant responses is crucial.

Seq2Seq models have a wide range of applications across various domains, showcasing their versatility and effectiveness in handling sequential data. Here are some of the prominent applications and their latest usage:

1. Machine Translation

Seq2Seq models have significantly transformed machine translation by enabling accurate translations between languages. They excel at capturing the nuances of language structure and context, allowing for coherent translations. Recent advancements include the development of transformer-based models that utilize self-attention mechanisms, improving translation quality and enabling multilingual capabilities, where a single model can translate between multiple language pairs simultaneously as shown in figure 2.

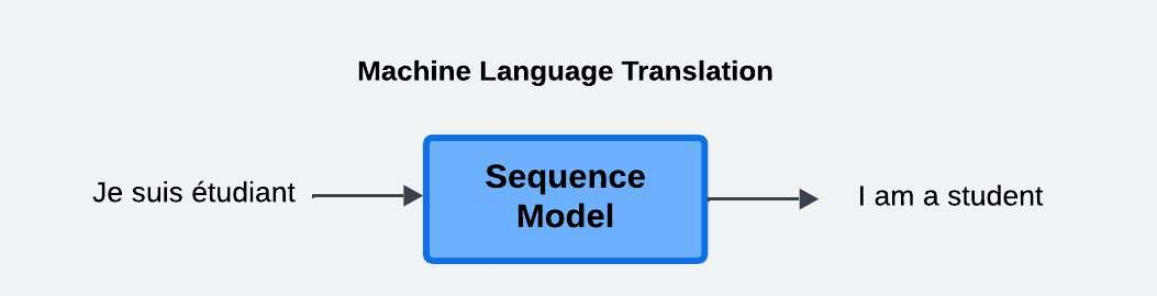

Figure 2: Sequence model for machine language translation.

2. Text Summarization

In text summarization, Seq2Seq models are employed to generate concise summaries of longer texts, making them highly effective for news articles, research papers, and content curation as shown in figure 3. They utilize advanced architectures, often incorporating attention mechanisms, to enhance the coherence and informativeness of the generated summaries. This has become increasingly important in industries that require quick content digestion.

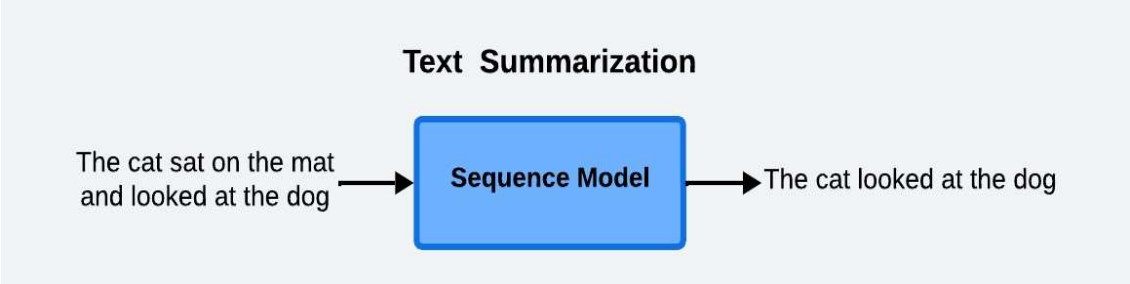

Figure 3: Sequence model for Text summarization.

3. Speech Recognition

Seq2Seq models play a crucial role in automatic speech recognition (ASR) by converting spoken language into written text. They have led to significant improvements in transcription accuracy, making them essential for applications like voice assistants and transcription services. Recent developments include adaptations for multilingual ASR, allowing for accurate transcription across different languages.

4. Image Captioning

In image captioning, Seq2Seq models generate natural language descriptions for images by combining features from convolutional neural networks (CNNs) with textual generation capabilities. This application is particularly valuable for accessibility tools and content generation, as it allows for creative and contextually relevant captions. Additionally, Seq2Seq models are utilized in visual question answering (VQA), where they generate answers based on image content.

5. Conversational Agents and Chatbots

Seq2Seq models are widely used in chatbots and conversational agents, enabling more interactive and natural conversations as shown in figure 4. By understanding context and generating appropriate responses, these models enhance user experience in customer service and personal assistant applications. The integration of attention mechanisms further improves the relevance and coherence of responses generated by these systems.

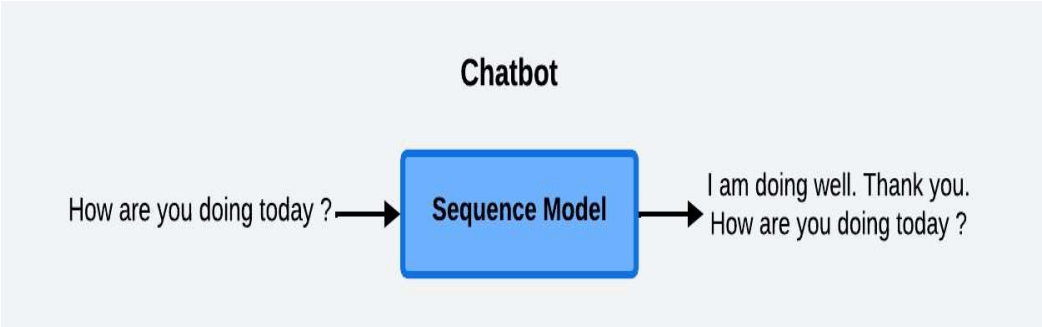

Figure 4: Sequence model for chatbots.

6. Recent Innovations

Recent research has focused on refining Seq2Seq models through various approaches, such as incorporating hierarchical structures for better handling of long sequences and exploring multimodal capabilities that integrate text with other data types like images and audio. These innovations aim to enhance the models' performance and expand their applicability across different domains.

C. QUESTIONING & ANSWERING

Questioning and answering (Q&A) systems are integral components of natural language processing (NLP) that enable machines to understand and respond to human queries. These systems are designed to interpret a question posed in natural language, retrieve relevant information from a knowledge base or context, and generate a coherent and contextually appropriate answer. The primary goal of Q&A systems is to facilitate human-computer interaction by providing accurate and relevant responses, thereby enhancing user experience in various applications, including customer support, educational tools, and information retrieval.

Implementation of Q&A Using Seq2Seq Models

The sequence-to-sequence (Seq2Seq) model architecture is particularly well-suited for implementing Q&A systems due to its ability to process input sequences and generate corresponding output sequences as shown in figure 5. At its core, the Seq2Seq model comprises two main components: an encoder and a decoder. The encoder processes the input question and encodes it into a fixed length context vector, which captures the essential features of the input sequence. This context vector serves as a condensed representation of the input question, allowing the model to retain crucial information while discarding irrelevant details.The decoder, on the other hand, takes this context vector and generates the output sequence, which in the case of a Q&A system, is the answer to the question. The decoder operates in an autoregressive manner, producing one token of the output sequence at a time while considering previously generated tokens. This step-by-step generation allows the model to create coherent and contextually relevant answers.

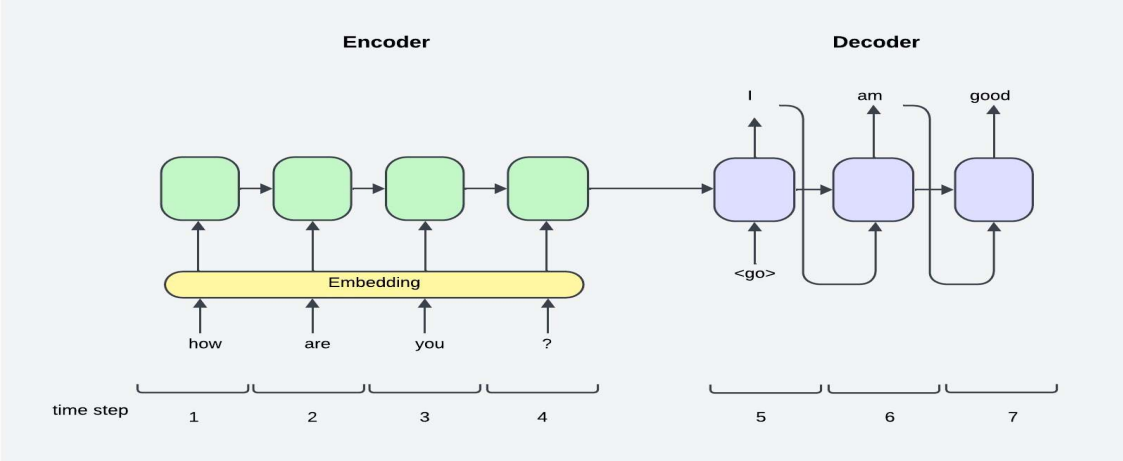

Figure 5: Seq2seq model architecture for question answering.

Attention Mechanism

A significant enhancement to the basic Seq2Seq architecture is the incorporation of the attention mechanism. This mechanism allows the model to dynamically focus on different parts of the input sequence when generating each token of the output sequence. In the context of Q&A, this means that when generating an answer, the model can weigh the importance of various segments of the input question and any accompanying context, improving its ability to produce accurate and relevant responses. The attention mechanism effectively addresses the limitations of standard Seq2Seq models, particularly when dealing with longer input sequences, where critical information may be lost in the fixed-size context vector.

Training and Optimization

Training a Seq2Seq model for Q&A tasks involves using a dataset comprising pairs of questions and answers. The model learns to maximize the likelihood of generating the correct answer given the input question. Techniques such as teacher forcing, where the decoder is provided with the ground truth answer during training, help stabilize the training process and improve convergence. Additionally, recent advancements in Seq2Seq models, such as the use of transformer architectures, have further enhanced their performance by allowing for better handling of long-range dependencies and parallel processing of input sequences.

Applications and Impact

Seq2Seq models have been successfully applied in various Q&A systems, including chatbots, virtual assistants, and automated customer support solutions. These applications not only streamline user interactions but also significantly reduce operational costs for businesses by automating responses to frequently asked questions. The ability of Seq2Seq models to generate contextually appropriate answers has transformed how information is accessed and retrieved, making it easier for users to obtain the information they need quickly and efficiently.In summary, questioning and answering systems leverage the capabilities of Seq2Seq models to process and generate natural language responses. By utilizing the encoder-decoder architecture and incorporating attention mechanisms, these models can effectively interpret user queries and produce accurate answers, enhancing the overall user experience in various applications. As research and development in this area continue to evolve, the potential for Seq2Seq models in Q&A systems remains vast, promising further advancements in human-computer interaction.

II. Literature Review

1. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding by Jacob Devlin et al.(2019)

This paper introduces BERT (Bidirectional Encoder Representations from Transformers), a transformer-based model that revolutionizes natural language processing (NLP) through its bidirectional training approach, which allows the model to consider context from both left and right sides of a word unlike traditional left-to-right or right-to-left language models.

BERT uses a multi-layer bidirectional Transformer encoder architecture, which is similar to the original Transformer model:The encoder has a stack of identical layers, with each layer having two sub-layers: a multi-head self-attention mechanism and a simple feed-forward neural network, The input to the first encoder layer is the sum of the token embeddings, segment embeddings (to differentiate between sentences), and position embeddings, The output of the final encoder layer is used for downstream tasks.

It employs two primary pre-training tasks: Masked Language Model (MLM), where 15% of the input tokens are masked at random, and the model is trained to predict these masked tokens based on their context. This allows BERT to learn deep contextual representations of words, and Next Sentence Prediction (NSP), which helps the model understand the relationship between sentence pairs by predicting whether a given sentence follows another in the training corpus. This helps the model understand the relationship between sentences.

BERT's architecture consists of multiple layers of self-attention, with two model sizes (BERT BASE: 12 layers, 768 hidden size, 12 attention heads, 110M parameters and BERT LARGE: 24 layers, 1024 hidden size, 16 attention heads, 340M parameters and it can be fine-tuned for various NLP tasks, achieving state-of-the-art results on benchmark datasets like SQuAD.

2. **T5: Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer"** by Colin Raffel et al. (2020)

This paper introduces the T5 (Text-to-Text generation)model, which reformulates all natural language processing (NLP) tasks as text-to-text tasks, enabling a unified approach to various applications such as question answering, translation, and summarization. The authors emphasize the effectiveness of transfer learning, where the model is pre-trained on a large dataset, specifically the "Colossal Clean Crawled Corpus," and then fine-tuned on task-specific datasets like SQuAD and Natural Questions. T5 employs a standard transformer architecture that includes both an encoder and a decoder. The encoder processes the input text, while the decoder generates the output text. This design allows T5 to effectively handle a wide range of NLP tasks. Text-to-Text Framework: T5 treats every NLP task as a text generation problem, allowing the model to use the same architecture, loss function, and hyperparameters across different tasks. Pre-training and Finetuning: The model is pre-trained using a combination of unsupervised and supervised tasks. Performance: T5 has demonstrated state-of-the-art results on various benchmarks, including GLUE, SuperGLUE, and SQuAD, showcasing its versatility and effectiveness in handling multiple NLP tasks. The paper reports significant improvements over previous models, establishing T5 as a leading framework in the field.

3. **ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators"** by Kevin Clark et al(2020)

Unlike traditional masked language modeling (MLM) methods like BERT, which mask a subset of input tokens and train the model to predict them, ELECTRA introduces a new task called replaced token detection. In this method, a small generator model replaces tokens in the input text with plausible alternatives. The main model, referred to as the discriminator, is trained to identify which tokens were replaced. This approach allows the model to learn from all input tokens rather than just a small masked subset, leading to more efficient training.

ELECTRA consists of two transformer models: a generator and a discriminator. Both models share the same architecture, which is based on the transformer encoder.The generator is trained as a masked language model to corrupt the input by replacing tokens, while the discriminator learns to identify these replacements. The architecture includes multiple layers of self-attention and feedforward networks, similar to BERT, allowing it to capture complex language patterns.. This focus enhances the contextual representations learned by the model, resulting in improved performance on downstream tasks.

ELECTRA models can outperform larger models like GPT by using far less compute, making it a more accessible option for researchers and practitioners.The paper reports extensive experiments showing that ELECTRA models, even at smaller scales, outperform BERT and other state-of-theart models on the GLUE benchmark.

4. **ALBERT: A Lite BERT for Self-supervised Learning of Language Representations**" by Zhenzhong Lan et al.(2019)

This paper presents a streamlined version of the BERT model, aimed at reducing the number of parameters while maintaining or improving performance on various natural language processing (NLP) tasks, including question answering. Parameter Reduction Techniques: Factorized Parameter Embedding: ALBERT decouples the size of the vocabulary embeddings from the hidden layer size by factorizing the embedding matrix into two smaller matrices. This allows for a larger hidden size without a proportional increase in the number of parameters, making it easier to scale the model while keeping memory consumption low. It also uses Cross-layer Parameter Sharing technique which allows the model to share parameters across different layers of the transformer architecture. Instead of having unique parameters for each layer, ALBERT shares weights among layers, which significantly reduces the total number of parameters. For example, the weights of the attention and feed-forward networks are shared across all layers, leading to a more compact model.Self-supervised Learning Objective: ALBERT replaces the Next Sentence Prediction (NSP) task used in BERT with a new task called Sentence-Order Prediction (SOP). SOP focuses on modeling inter-sentence coherence, which addresses some limitations observed with NSPand improves the model's performance on tasks that involve multiple sentences. It consists of multiple layers of self-attention and feed-forward networks, similar to BERT, but with fewer parameters due to the factorization and sharing techniques. ALBERT was pre-trained on large corpora, including English Wikipedia (2,500 million words) and BookCorpus (800 million words). It was then fine-tuned on various downstream tasks, including: SQuAD: For question answering. This makes ALBERT not only more efficient but also scalable for various applications in natural language processing, including question answering.

5. **BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Processing**" by Mike Lewis et al.

This paper introduces BART (Bidirectional and Auto-Regressive Transformers), a novel model designed for various natural language processing (NLP) tasks, including text generation,Question answering, and comprehension.BART combines the strengths of both BERT and GPT architectures: Bidirectional Encoder: Similar to BERT, BART uses a bidirectional encoder that processes the entire input sequence at once, allowing it to capture context from both directions and Auto-Regressive Decoder: Like GPT, BART employs a left-to-right decoder that generates text sequentially, predicting the next token based on previously generated tokens.This architecture allows BART to effectively handle a wide range of NLP tasks.

BART is trained as a denoising autoencoder. The training process involves two main steps: Corruption: The input text is corrupted using various noising functions, such as masking random tokens, deleting tokens, or permuting sentences, Reconstruction: The model learns to reconstruct the original text from the corrupted version. This denoising process helps the model learn robust representations of language. Pre-training Tasks: Token Masking: Random tokens in the input are masked, and the model learns to predict these masked tokens, Token Deletion: Random tokens are deleted from the input, and the model learns to reconstruct the original sequence, Sentence Permutation: The order of sentences is shuffled, and the model learns to reorder them correctly,

Span Masking: A span of tokens is replaced with a single mask token, requiring the model to learn to infer the missing information.
After pre-training, BART can be fine-tuned on specific tasks, such as summarization, translation, and question answering. This fine-tuning process allows the model to adapt its learned representations to the nuances of the target task.

6. **Question Answer System: A State-of-Art Representation of Quantitative and Qualitative Analysis**

In the research paper, a comprehensive bibliometric and literature analysis of Question Answering Systems (QAS) is presented, which sheds light on the diverse techniques and dimensions in the field. The study draws upon Scopus and Web of Science databases for systematic performance analysis and science mapping, helping to trace the evolution of QAS architectures. The paper identifies key trends, influential authors, and seminal works within the QAS domain, providing valuable insights into the development of Knowledge Base (KB)-based and Graph Neural Network (GNN)-based approaches. Through network analysis, the study also maps citation and co-citation patterns to understand the intellectual structure and progression of the field. This literature review underscores how QAS methodologies have evolved and emphasizes the importance of understanding these advancements to identify future research directions, especially the integration of KBs and GNNs to enhance system performance.

7. **Text-based Question Answering from Information Retrieval and Deep Neural Network Perspectives: A Survey**

This research paper provides an in-depth overview of the evolution and methodologies of textbased Question Answering (QA) systems, combining both traditional Information Retrieval (IR) techniques and modern Deep Neural Networks (DNNs). It begins by discussing the foundational challenges in QA, which include efficiently retrieving relevant information, understanding the user's query, and generating or extracting a suitable response. Initially, QA systems relied heavily on IR-based approaches, where models like word count and probabilistic classifiers matched question-answer pairs based on word co-occurrence and term relevance, often using Inverse Document Frequency (IDF) to refine the matching process. As the field advanced, deep learning based approaches gained prominence, shifting focus toward representation-based models, which generate fixed-dimensional vector representations for both questions and answers. These vectors are then compared in a latent space to determine the most relevant match. The survey also discusses interaction-based models that assess the relationship between individual terms in the question and potential answers, emphasizing the semantic or syntactic similarity between them. Hybrid models that combine both representation and interaction techniques were proposed for better performance. The paper highlights the significant improvements deep learning models, especially transformers like BERT, have brought to QA tasks, outperforming earlier IR-based methods. Finally, the paper addresses the evaluation metrics used to assess the performance of these systems and discusses the persistent challenges in QA, such as handling word mismatch, managing contextual understanding, and resolving ambiguities in the questions.

8.**Strong Baselines for Simple Question Answering over Knowledge Graphs with and without Neural Networks**

The research paper explores effective strategies for question answering (QA) over knowledge graphs (KGs), focusing on establishing strong baseline models. The authors argue that even simple, traditional methods can serve as competitive baselines compared to more complex neural network models. The study examines both neural and non-neural approaches for simple QA tasks, highlighting the effectiveness of rule-based methods, information retrieval (IR) techniques, and semantic parsing. It demonstrates that certain straightforward techniques, such as using entity linking and relation matching with manually crafted features, can achieve substantial performance. Additionally, the paper evaluates neural network-based models, showing their potential for further enhancing QA over KGs. Ultimately, the research emphasizes the importance of strong, wellconstructed baselines to better assess the advancements introduced by complex models in QA tasks.

9.**Question Answering Survey: Directions, Challenges, Datasets, Evaluation Matrices**

The paper provides a detailed survey of recent advancements in question answering (QA) systems. It examines various research areas, including reading comprehension, conversational QA, visual QA, and question generation. The paper highlights challenges in handling low-resource languages and notes that while deep learning techniques are gaining popularity in resource-rich languages, rule-based or machine learning approaches are still prevalent in low-resource languages. The study also reviews important QA datasets and evaluation measures used in current research.

10.**WebGPT: Browser-assisted question-answering with human feedback**

The paper presents a model called WebGPT that combines the strengths of language models with browsing capabilities to provide more accurate and dynamic question-answering. WebGPT is designed to browse the web to gather information in real-time, enabling it to answer questions with up-to-date and contextually relevant information. The model leverages human feedback to finetune its performance, ensuring better quality responses by incorporating input on how well the model's browsing and generated answers meet expectations. The paper highlights how WebGPT addresses the challenges of using the internet for question-answering, including dealing with web noise, citation accuracy, and relevance of the content retrieved. It discusses how the integration of web browsing enhances the model's ability to handle diverse and specific queries that require information beyond its training data. This approach improves performance on a variety of benchmarks and paves the way for future advancements in browser-assisted question answering.

III. Model Description

1.BERT

BERT (Bidirectional Encoder Representations from Transformers) is a groundbreaking language representation model developed by Google in 2018. It has significantly advanced the field of natural language processing (NLP) by providing a powerful framework for understanding the context of words in a sentence based on their surrounding words.
Unlike previous models like GPT (which are unidirectional), BERT is bidirectional, meaning it looks at the entire context of a word by considering both the words that come before and after it in a sentence. This allows BERT to understand the context more accurately, especially when the meaning of a word depends on its surroundings.

How BERT Works

BERT's success lies in its innovative architecture and training approach. It utilizes a multi-layer bidirectional Transformer encoder, which allows the model to consider the context from both the left and right sides of a word. This bidirectionality is a significant advancement over traditional left-to-right models, enabling BERT to capture more nuanced meanings in language.During pretraining, BERT is trained on a massive corpus of unlabeled text data using two main tasks:

1. Masked Language Model (MLM): 15% of the input tokens are randomly masked, and the model predicts these masked tokens based on the surrounding context. This process helps BERT learn deep contextual relationships.
2. Next Sentence Prediction (NSP): The model predicts whether one sentence logically follows another, aiding its understanding of sentence relationships.

After pre-training, BERT can be fine-tuned for specific NLP tasks by adding a taskspecific output layer. Fine-tuning is efficient, requiring only a small amount of labeled data due to the comprehensive language understanding acquired during pre-training.BERT has several advantages that make it a powerful tool in NLP tasks. Its bidirectional approach allows it to capture rich contextual information from both sides of a sentence.

Additionally, BERT is pre-trained on vast amounts of text, which provides a strong foundation for language comprehension before fine-tuning for specific tasks. This makes it highly versatile, as it can be adapted for a variety of tasks, such as question answering and sentiment analysis. However, BERT also faces challenges. Its high computational cost makes it resource-intensive to train and fine-tune, requiring significant hardware. Moreover, BERT is not designed for text generation, limiting its capabilities in tasks that involve generating coherent sequences, for which models like GPT are better suited.

2. ProphetNet

ProphetNet is a sequence-to-sequence pre-training model developed by Microsoft Research, introduced in the paper "ProphetNet: Predicting Future N-gram for Sequence-toSequence Pre-training" by Yu Yan and colleagues in January 2020. This model is specifically designed for natural language processing tasks, particularly focusing on language modeling and text generation.This model is designed to improve upon traditional sequence-to-sequence (seq2seq) models used in natural language generation tasks like summarization, translation, and text generation. While most models predict one token (word or sub-word) at a time, ProphetNet introduces a new n-gram prediction mechanism, which predicts multiple tokens simultaneously to generate more coherent and fluent sequences.

Traditional seq2seq models like GPT or BART predict the next token in the sequence based on the previously generated tokens. This is known as autoregressive generation. However, this approach has a limitation—it only focuses on the immediate next token, making it prone to errors when generating long sequences.ProphetNet changes this by using future n-gram prediction, where instead of predicting just the next token, it predicts multiple future tokens at once.It is built on the Transformer architecture, enhances the decoder to support this multi-token prediction.However, the n-gram prediction mechanism adds complexity, making ProphetNet computationally more intensive than traditional models. Despite this, its superior performance in generating natural, coherent text makes it a standout model for sequence-to-sequence tasks.

ProphetNet is pre-trained on large datasets (16GB and 160GB) using the future n-gram prediction objective along with a masked language modeling task. This pre-training stage allows the model to learn general language understanding.After pre-training, ProphetNet can be finetuned on specific sequence-to-sequence tasks.Experimental results show that ProphetNet achieves state-of-the-art performance on various benchmarks, including CNN/DailyMail, Gigaword, and SQuAD 1.1, outperforming other models using the same pre-training data .ProphetNet's ability to generate high-quality text makes it applicable to a wide range of sequence-to-sequence tasks.

3. T5

T5, or Text-to-Text Transfer Transformer, is a versatile language model developed by Google that reformulates all natural language processing (NLP) tasks into a unified text-to-text format. This means that both the input and output are treated as text strings, allowing the same model architecture to be applied across various tasks such as translation, summarization, question answering, and text classification.

Features:

- Text-to-Text Framework: T5 converts every NLP task into a text-to-text format. For example, for a translation task, the input might be "translate English to French: [text]," and the output would be the translated text. This uniform approach simplifies the model architecture and facilitates multitasking.
- Pre-training on C4 Dataset: T5 is pre-trained on the Colossal Clean Crawled Corpus (C4), a large dataset that consists of approximately 750GB of clean text data. This extensive pretraining enables T5 to learn a wide range of language patterns and structures.

- Transformer Architecture: T5 is built on the Transformer architecture, featuring both encoder and decoder components. The encoder processes the input text, while the decoder generates the output text, allowing the model to capture complex relationships in the data.
- Fine-tuning for Specific Tasks: After pre-training, T5 can be fine-tuned on specific tasks by adding a task-specific output layer. This fine-tuning process allows the model to adapt its learned representations to the nuances of different tasks, improving performance.

Variants and Model Sizes
T5 comes in various sizes, including T5-Small, T5-Base, T5-Large, T5-3B, and T5-11B, with the largest variant containing 11 billion parameters. Each size offers different levels of performance and computational requirements, allowing users to choose a model that fits their resource availability and task complexity.

T5's pre-training process differs from other models in a few key ways:

1. Multi-task pre-training: T5 is pre-trained on a mixture of unsupervised and supervised tasks, allowing it to learn from a diverse set of objectives simultaneously. This contrasts with models like BERT which are typically pre-trained on a single unsupervised task like masked language modeling.
2. Text-to-text format: All tasks are converted into a text-to-text format for T5, where both the input and output are text strings. This allows a single model to be used for various tasks like translation, summarization, classification etc. by prepending a task-specific prefix to the input. Other models may have separate heads for different tasks.
3. Larger pre-training dataset: T5 is pre-trained on the Colossal Clean Crawled Corpus (C4), which is a cleaned version of Common Crawl and is significantly larger than datasets used for pre-training other models like BERT which used Wikipedia and BookCorpus.
4. Denoising objective: During unsupervised pre-training, T5 uses a denoising objective where tokens are randomly removed from the input and the model has to predict the missing tokens. This is similar to BERT's masked language modeling but T5 also predicts the position of the masked tokens.
5. Relative position embeddings: T5 uses relative position embeddings instead of absolute position embeddings used in BERT. This allows the model to generalize better to input sequences longer than those seen during pre-training.

4. BART

BART (Bidirectional and Auto-Regressive Transformer) is a flexible and powerful model developed by Facebook AI for a range of natural language processing (NLP) tasks. It combines the strengths of BERT's bidirectional understanding and GPT's autoregressive text generation, making it well-suited for tasks like text summarization, translation, and question answering. BART's unique approach comes from its ability to both understand and generate text, leveraging a combination of an encoder that reads the input from both directions and a decoder that generates output one token at a time.

Features of BART:

1. Bidirectional Encoder: BART's encoder works similarly to BERT, processing input text in both directions to capture rich contextual information. This allows the model to understand the meaning of words in relation to their surrounding context more effectively.
2. Autoregressive Decoder: The decoder in BART works like GPT, generating output text by predicting tokens one by one in a left-to-right manner. This autoregressive nature makes BART effective for text generation, where it produces fluent and coherent outputs.
3. Denoising Autoencoder: BART is trained with a denoising objective, meaning it learns to reconstruct sentences that have been deliberately corrupted by processes like token masking, deletion, or sentence shuffling. This helps the model learn robust representations and improve its ability to generate coherent sequences even when the input is noisy or incomplete.

5.ALBERT

A Lite BERT for Self-Supervised Learning of Language Representations is a modified version of BERT that focuses on improving both efficiency and performance by introducing new design changes. The goal of ALBERT is to reduce the memory consumption and speed up training times, making it more suitable for large-scale natural language processing (NLP) tasks without sacrificing accuracy.

Applications of ALBERT

ALBERT is ideal for NLP tasks that benefit from both a high degree of language understanding and efficiency in terms of model size and computation, including:

- Text Classification: Classifying text into predefined categories.
- Named Entity Recognition (NER): Identifying and classifying entities like names, dates, and locations in text.
- Question Answering: Extracting answers to questions from passages of text.
- Sentence and Document Similarity: Measuring similarity for tasks like paraphrasing or duplicate detection.

Major Enhancements in ALBERT:

1. Cross-Layer Parameter Sharing: One of the primary differences between ALBERT and BERT is that ALBERT significantly reduces the number of parameters by sharing parameters across layers. In BERT, each layer has its own set of parameters, which can become resource-intensive as the model gets deeper. ALBERT eliminates redundancy by reusing the same parameters for different layers. This allows ALBERT to scale better, with far fewer parameters, while still maintaining performance.
2. Factorized Embedding Parameterization: Another key improvement is the use of factorized embedding parameterization. In BERT, the size of the word embeddings is tied to the hidden layers of the model. ALBERT decouples these by reducing the size of the embedding matrix and using two smaller matrices instead, which factorizes the large vocabulary space into a more compact representation. This significantly reduces the overall parameter count, leading to better memory efficiency.

3. Sentence Order Prediction (SOP): While BERT used the Next Sentence Prediction (NSP) task to help the model understand sentence relationships, ALBERT replaces this with Sentence Order Prediction (SOP). SOP improves inter-sentence coherence by training the model to identify whether two consecutive sentences have been swapped. This task proves to be more effective in capturing sentence-level context, enhancing ALBERT's performance in tasks involving sentence ordering and discourse-level understanding.

4. Smaller Yet Competitive: Despite being smaller in size, ALBERT achieves results comparable to or even exceeding BERT on several benchmark tasks like the GLUE, SQuAD, and RACE datasets. The smaller size and faster training speed make ALBERT more accessible for applications where computational resources are limited.

   ALBERT offers several advantages. First, it significantly reduces model size by using parameter sharing across layers, making it much more memory-efficient than BERT. Additionally, factorized embedding parameterization helps decrease the number of parameters without losing performance. ALBERT also replaces the Next Sentence Prediction task with Sentence Order Prediction, improving sentence-level understanding. Despite being smaller, ALBERT maintains strong performance on various NLP tasks, making it faster to train and ideal for use in resource-limited environments.

Table 1: Comparision table of models.

Model	**Key Feature**
BERT	Utilizes masked language modeling with a bidirectional context for NLP tasks
ProphetNet	Future n-gram prediction for improved text generation.
T5	Unified text-to-text framework for various NLP tasks.
BART	Combines bidirectional and autoregressive training for effective text reconstruction.
ALBERT	uses parameter sharing and factorized embeddings to reduce size while maintaining strong performance.

IV. Implementation

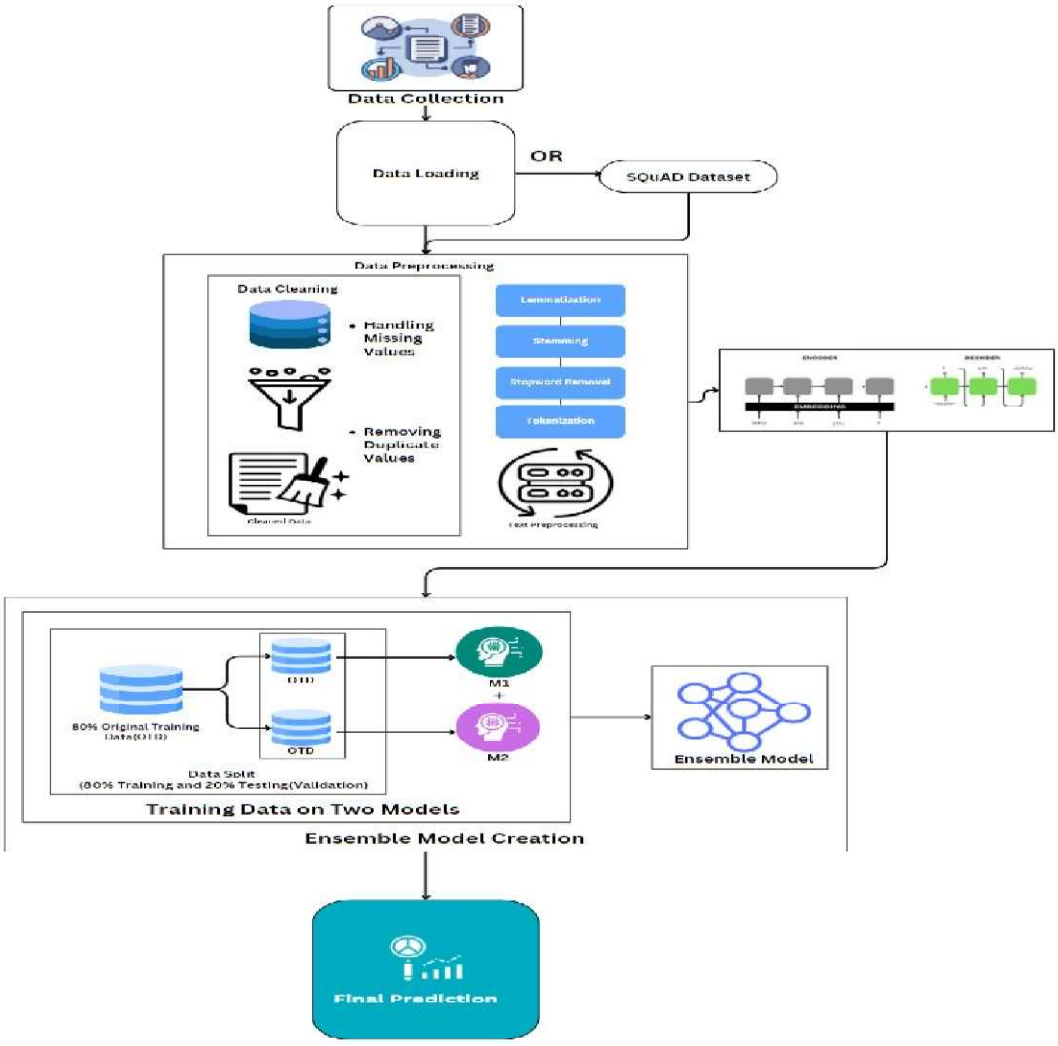

Figure 6: Framework of proposed model.

1.Data preparation

The Stanford Question Answering Dataset (SQuAD) was selected as the benchmark for this project due to its popularity in the question-answering domain and its structure, which is well-suited for training models that can answer questions based on a given context.

1.1 About the SQuAD Dataset

The SQuAD (Stanford Question Answering Dataset) is a large-scale dataset for question answering tasks. It contains a collection of question-answer pairs based on passages from a wide variety of Wikipedia articles. Each entry in the dataset includes a *context*, which is a paragraph from a

Wikipedia article; a *question*, which is derived from the context paragraph; and an *answer*, which is a specific span of text in the context that answers the question. Each question-answer pair is carefully crafted to ensure that the answer is located within the context. The SQuAD dataset provides two main files: a training set that contains context, question, and answer pairs for finetuning the model, and a validation set for evaluating model performance post-training. This dataset format supports supervised learning, where the model is trained to predict an answer based on the given context and question.

1.2 Data Preprocessing

To prepare the data for training, several preprocessing steps were applied as shown in figure 6. First, tokenizer setups specific to each model were implemented: T5Tokenizer for T5 small, which is designed for text generation tasks, and BartTokenizer for BART base, optimized for language generation and denoising. In the *input format*, the context and question were concatenated and tokenized together to form a single input sequence, while the *output format* tokenized the answer span separately as the label, allowing the model to learn the relationship between the question, context, and answer. The tokenized inputs were then transformed into PyTorch tensors to ensure compatibility with the training framework. These tensors included input_ids to encode the tokenized context and question, attention_mask to mask out padding tokens for more efficient processing, and labels to encode the answer text as the target output.
Edge cases were also handled in preprocessing. *Empty answers* were managed by filtering or ignoring during training to prevent issues with missing labels. *Multiple answers* were consolidated into a single answer or consistently formatted when multiple correct answers were available. Additionally, a token limit of 128 tokens was imposed on the label tensor to manage memory usage and avoid excessive padding.

2. Model Selection

For this project, two transformer models were selected: T5 small and BART base. Both models are well-suited for question-answering tasks, with each offering distinct advantages.

2.1 T5 Small

The T5 (Text-To-Text Transfer Transformer) model treats all NLP tasks as a text-generation problem, which makes it flexible for a variety of applications, including question-answering. T5 small, a smaller variant of the T5 model, was selected to reduce computational requirements while maintaining solid performance. The model receives the context and question as input and generates the answer as text, making it suitable for tasks requiring precise answer generation.

2.2 BART Base

BART (Bidirectional and Auto-Regressive Transformer) is a denoising autoencoder that excels in text generation tasks. It performs particularly well in text summarization, machine translation, and question-answering. BART base was chosen for its strong ability to generate answers based on context, as well as its architecture, which combines both bidirectional and autoregressive capabilities.

3. Model Training and Fine-Tuning

3.1 Training Configuration

Each model was fine-tuned on the SQuAD dataset using configurations optimized for performance. Key hyperparameters included the number of epochs (set to 3, based on best practices for fine-tuning transformer models to avoid overfitting) and a batch size of 8 for both training and evaluation, balancing memory efficiency and training speed. The learning rate was set to 3e-5, a standard rate for fine-tuning transformers without causing gradient instability. Additionally, gradient accumulation steps of 4 were used to accumulate gradients over multiple steps, effectively increasing the batch size for memory efficiency. FP16 precision was enabled for BART, allowing faster training with reduced memory usage.

3.2 Model Training Setup

The Hugging Face Trainer API was used to streamline the training pipeline, handling checkpointing, logging, and evaluation. This simplified the implementation and ensured reproducible results. Checkpoints were saved at each epoch, allowing for progress tracking and selection of the best-performing model, and metrics like accuracy, F1-score, and loss were recorded to monitor training convergence and assess overfitting.

3.3 Fine-Tuning T5 Small

The T5 small model was fine-tuned to generate answers based on concatenated context and question inputs. The input format was configured as "context: question" to help T5 understand the input structure. T5 was trained for 3 epochs, with intermediate evaluations at the end of each epoch. Checkpoints were saved at each epoch, and the evaluation metrics were used to select the best checkpoint.

3.4 Fine-Tuning BART Base

The BART base model was fine-tuned similarly, with adjustments to leverage its generative strengths. Labels were limited to 128 tokens to optimize memory usage, and BART was trained in mixed precision (FP16), which sped up training and reduced memory requirements. As with T5, checkpoints and evaluation metrics were recorded at each epoch.

4. Ensemble Learning on BART Base

To further improve the robustness and accuracy of predictions, ensemble learning was applied to the BART base model. Multiple BART models were independently fine-tuned to form an ensemble. During inference, predictions from each model were averaged, creating an ensemble output that mitigates variance and improves accuracy. The ensemble model showed improved metrics over individual models, highlighting the effectiveness of combining predictions.

V. Results and Evaluation

Upon completing training, the models were evaluated on the SQuAD validation set.The results of the project showed that the BART Base model outperformed T5 Small in the question-answering task, demonstrating strong performance in generating accurate and contextually relevant answers from the given passages. While T5 Small performed well, it was slightly less effective in handling complex patterns compared to BART. However, the best results were achieved using the BART Ensemble approach, where multiple fine-tuned BART models were combined. The context provided is: *"Peacock is the national bird of India and we are building a seq2seq model using BART for question answering while drinking coffee."* Based on this context, the user inputs the question: *"What are we drinking?"* The model processes the input and generates the answer: *"coffee."* This highlights the ability of the application to understand and extract relevant information from the given context to answer the question accurately. This ensemble method significantly improved accuracy by averaging the predictions, reducing variance, and enhancing overall robustness and consistency in answer generation. Thus, the BART Ensemble model provided the most reliable and accurate results, followed by BART Base and T5 Small.

VI. Conclusion

The project successfully implemented and fine-tuned the T5 small and BART base models for question answering on the SQuAD dataset. Among the two, the BART ensemble outperformed the individual models, offering the best results in terms of accuracy and robustness. The preprocessing techniques, hyperparameter tuning, and the ensemble learning approach were instrumental in achieving these results.

The project demonstrated that transformer-based models like T5 and BART are highly effective for question-answering tasks, particularly when fine-tuned on large, structured datasets like SQuAD. The use of ensemble learning further enhanced the overall model performance, making it a valuable strategy for improving accuracy and mitigating variance in predictions.

VII. Future scope

The future scope for this project includes:

1. Exploring Larger Models:
    - Experimenting with larger versions of the models (T5-large or BART-large) to see if performance can be further improved.
2. Domain-Specific Question Answering:
    - Fine-tuning the models on domain-specific datasets, such as medical or legal domains, to create models capable of answering specialized questions.
3. Multimodal Question Answering:
    - Integrating other data types (images, audio) alongside text to build multimodal question-answering systems that can handle more complex queries.

4. Real-World Applications:
    - Deploying the trained models in real-time applications such as customer service bots or virtual assistants, where fast and accurate question-answering is required.
5. Model Optimization:
    - Exploring model optimization techniques, such as quantization or pruning, to make the models more memory-efficient and suitable for deployment on mobile devices or edge computing environments.

VIII. References

1.Kenton, J. D. M. W. C., & Toutanova, L. K. (2019, June). Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of naacL-HLT* (Vol. 1, p. 2).

2. Colin, R. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, *21*, 140-1.

3. Clark, K. (2020). Electra: Pre-training text encoders as discriminators rather than generators. *arXiv preprint arXiv:2003.10555*.

4. Lan, Z. (2019). Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*.

5. Lewis, M. (2019). Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

6. Zope, B., Mishra, S., Shaw, K., Vora, D. R., Kotecha, K., & Bidwe, R. V. (2022). Question answer system: A state-of-art representation of quantitative and qualitative analysis. *Big Data and Cognitive Computing*, *6*(4), 109.

7. Abbasiantaeb, Z., & Momtazi, S. (2021). Text-based question answering from information retrieval and deep neural network perspectives: A survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *11*(6), e1412.

8. Mohammed, S., Shi, P., & Lin, J. (2017). Strong baselines for simple question answering over knowledge graphs with and without neural networks. *arXiv preprint arXiv:1712.01969*.

9.Pandya, H. A., & Bhatt, B. S. (2021). Question answering survey: Directions, challenges, datasets, evaluation matrices. *arXiv preprint arXiv:2112.03572*.

10.Nakano, R., Hilton, J., Balaji, S., Wu, J., Ouyang, L., Kim, C., ... & Schulman, J. (2021). Webgpt: Browser-assisted question-answering with human feedback. *arXiv preprint arXiv:2112.09332*.