Emphasizing Seq2Seq architecture using T5 and BART models in Question Answering

Amit Das
G H Raisoni College of Engineering
Nagpur, Maharashtra, India
amit.das.ds@ghrce.raisoni.net

Vibhor Joshi
G H Raisoni College of Engineering
Nagpur, Maharashtra, India
vibhor.joshi.ds@ghrce.raisoni.net

Shilpa Tichkule
G H Raisoni College of Engineering
Nagpur, Maharashtra, India
shilpa.tichkule.ds@ghrce.raisoni.net

Sanika Nandurkar
G H Raisoni College of Engineering
Nagpur, Maharashtra, India
sanika.nandurkar.ds@raisoni.net

Amit Pandey
International Institute of Information Technology, Hyderabad, Telangana, India
amitpandeyresearch@gmail.com

Medhavi Nasare
G H Raisoni College of Engineering
Nagpur, Maharashtra, India
medhavi.nasare.ds@raisoni.net

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 study provides an overview of the Seq2Seq architecture based Questioning and Answering implementation project, its core components, and its recent advancements, particularly the attention mechanism. Implementation of such model using BART as well as T5 pretrained models helped us analyze the complexities associated with this architecture. We also discuss the various applications of Seq2Seq models in NLP and their potential for future developments for references.

KEYWORDS

Seq2Seq architecture, Natural Language Processing, Attention Mechanism, Question Answering Task, Ensemble Learning, BART base, T5 small.

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.

At the center of the Seq2Seq model lies the encoder-decoder framework. 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. 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.

II. LITERATURE SURVEY

The paper[1] introduces a revolutionary model that significantly enhances natural language processing (NLP) by enabling a bidirectional understanding of context. Unlike earlier models that analyzed text in a linear fashion, BERT processes entire sequences simultaneously, which allows for a deeper grasp of linguistic subtleties. The training process consists of two key stages: ***pre-training****, where the model learns from extensive unlabeled text through tasks like Masked Language Modeling and Next Sentence Prediction, and* ***fine-tuning****, which tailors the model to specific tasks with minimal modifications. BERT has achieved remarkable results across various benchmarks, notably improving performance in areas such as question answering and sentiment analysis. By showcasing the advantages of bidirectional training for creating richer contextual representations, BERT establishes a new benchmark for NLP models and simplifies the architecture required for diverse applications.*

The paper[2] talks about T5, which stands for Text-to-Text Transfer Transformer. This model is pretty cool because it takes every natural language processing (NLP) task and turns it into a text-to-text problem, meaning both the input and output are just strings of text. So whether you're translating languages, summarizing articles, or answering questions, T5 uses the same setup for all these tasks, making it super flexible and easy to work with. It's built on a standard encoder-decoder structure and trained on a huge dataset called C4 that has a ton of clean English text. T5 uses techniques like teacher forcing during training and can be fine-tuned for specific jobs, which helps it achieve top-notch results across many different benchmarks. Overall, T5 shows how effective it is to have one model that can handle multiple tasks without needing separate setups for each one, making it a big step forward in transfer learning for NLP.

The paper[3] introduces a fresh way to train language models by focusing on a discriminator instead of a generator. In this setup, ELECTRA uses two transformer models: a small generator that swaps out some tokens in a sentence and a larger discriminator that figures out which tokens were changed. Instead of the usual method where the model predicts masked words, ELECTRA uses a task called replaced token detection, where the discriminator learns to tell apart original tokens from the replaced ones. This approach is way more efficient and allows ELECTRA to perform really well on various NLP tasks while using less computing power than models like BERT. By showing that being a discriminator can be more effective than traditional methods, ELECTRA raises the bar for language models in natural language processing.

The paper[4] talks about ALBERT, which is a lighter version of the original BERT model made to improve performance on various natural language processing (NLP) tasks while using fewer resources. ALBERT does this by using two main tricks: first, it breaks down the embedding parameters, which helps save memory by separating the input and hidden layer embeddings; second, it shares parameters across different layers to cut down on redundancy. Because of these changes, ALBERT can achieve similar or even better results on tasks like GLUE and SQuAD while having way fewer parameters than BERT. Overall, ALBERT shows that you can still get great performance in NLP with a smaller and more efficient model design.

The paper[5] introduces BART, which stands for Bidirectional and Auto-Regressive Transformers. This model is designed to tackle various tasks in natural language processing (NLP). BART works as a denoising autoencoder, meaning it first messes up some text using a noising function and then learns to fix it back to the original form. Its setup combines a bidirectional encoder like BERT and a left-to-right decoder similar to GPT, allowing it to understand context really well. This makes BART great for tasks such as generating text, summarizing information, and translating languages. By using both bidirectional and autoregressive training methods, BART achieves top results on many benchmarks, proving it's a versatile and effective tool for complex language tasks.

III. IMPLEMENTATION

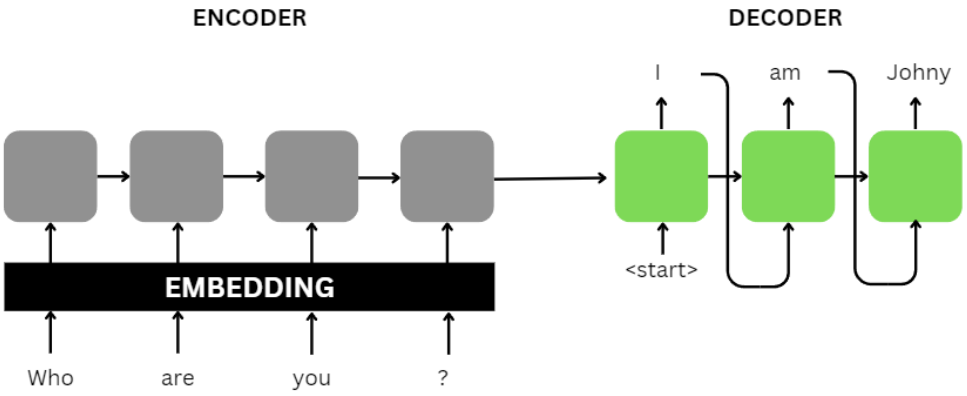

Fig 1. Seq2seq model workflow architecture for question answering

The implementation of the seq2seq model for question answering is structured to effectively utilize the "SQuAD" provided by Stanford. This section outlines the key components of the implementation, including dataset characteristics, model training, and evaluation processes.

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 fine-tuning 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.

Model Implementation:

1. Environment Setup: Necessary libraries such as Pytorch, transformers and others are imported to facilitate model training and evaluation.
2. Data Loading: The dataset is loaded from the datasets library, and are preprocessed into tokens to meet the input requirements of the architecture.
3. Model Selection: Two transformer models were selected: **T5 small** and **BART base**. Both models are well-suited for question-answering tasks, with each offering distinct advantages.
4. Preprocessing: To prepare the data for training, several preprocessing steps were applied. 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.
5. Model Configuration: Both the model is configured with appropriate hyperparameters, including Learning Rate, Batch sizes and epochs according the hardware specifications
6. Training and testing Process: The Hugging Face Trainer API was used to streamline the training pipeline, handling checkpointing, logging, and evaluation. Both models were fine-tuned to generate answers based on concatenated context and question inputs.
7. Experimentation with Ensemble learning was also performed on BART model for further improvement.
8. Results Analysis: Upon completing training, the models were evaluated on the SQuAD validation set.
9. Deployment: Apps are deployed for public availability of the question answering task using hugging face Spaces.

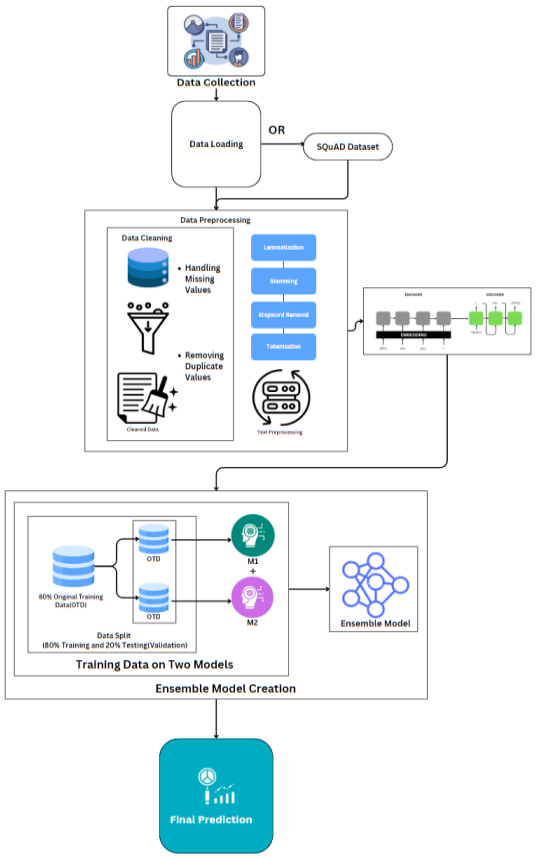

Fig 2. Flowchart of implementation

(i) Question & 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.

(ii) Implementing Seq2Seq for Q&A

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. 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.

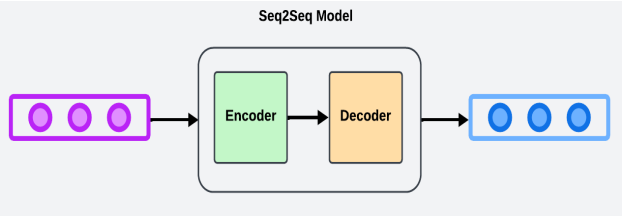

Fig 3. Seq2Seq Model

(iii) 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 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. As research and development in this area continue to evolve, the potential for Seq2Seq models in Q&A systems remains vast.

IV. RESULT ANALYSIS

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. This ensemble method significantly improved accuracy by averaging the predictions, reducing variance, and enhancing overall robustness and consistency in answer generation. Thus, the BART base Ensemble model provided the most reliable and accurate results, followed by BART Base and T5 Small.

V. 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 study 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.

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*.