News Article Based Question Answering System

An M. Tech Project Report Submitted

in Partial Fulfillment of the Requirements

for the Degree of

Master of Technology

by

Sushovan Saha

(214161011)

under the guidance of

Dr. Ashish Anand

Dr. Prithwijit Guha

to the

DATA SCIENCE PROGRAMME

INDIAN INSTITUTE OF TECHNOLOGY GUWAHATI

GUWAHATI - 781039, ASSAM

CERTIFICATE

*This is to certify that the work contained in this thesis entitled "**News Article Based Question Answering System**" is a bonafide work of **Sushovan Saha (Roll No. 214161011)**, carried out in the Data Science programme, Indian Institute of Technology Guwahati under my supervision and that it has not been submitted elsewhere for a degree.*

Supervisor: **Dr. Ashish Anand**
Prof.
May, 2023
Department of CSE
IIT Guwahati
Assam

Supervisor: **Dr. Prithwijit Guha**
Associate Prof.
May, 2023
Department of EEE
IIT Guwahati
Assam

Acknowledgements

First and foremost, I would like to express my heartfelt gratitude to my thesis supervisors, Dr. Ashish Anand, Department of Computer Science and Engineering, Indian Institute of Technology Guwahati, and Dr. Prithwijit Guha, Department of Electronics and Electrical Engineering, for allowing me to work on this project. Their in-depth knowledge of the subject area encouraged me to work harder on the assignment and keep studying. I want to express my gratitude to everyone who helped me with my efforts.

Sushovan Saha

IIT Guwahati,

November, 2022.

Abstract

Open-domain question answering (ODQA) has significantly improved in recent years due to the development of deep learning techniques and the availability of sizable QA data sets. Wikipedia and other synchronic writings are the sources of many data sets. The underuse of temporal news collections, which include valuable information spanning over decades, results in the loss of crucial information that cannot be found on Wikipedia. So, an approach is used to construct a question-answering dataset from news events in India that cannot be adequately addressed using Wikipedia only. The news event question-answering model is trained using the data gathered from this data collection. As a result, our work tries to determine the authenticity of the aforementioned dataset in effective but practical ways. Additionally, we propose strategies for creating a flexible Question-Answering (QA) model that can handle various issues laid out step-by-step in this study. We overcome the difficulties by preparing the data and utilizing advanced NLP methods. Furthermore, we investigate how to improve learning by combining a teacher-student method with a low-parameter model. The trials show our approach's effectiveness and adaptability across various domains. This study gives insights on effective model development for a variety of AI applications as well as advances QA systems.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The task of answering news questions (NQA) entails putting together semantically and syntactically sound natural language responses to questions that are related to recent or ongoing news events. The NQA assignment must be completed manually, which involves extensive surfing and reading of several news articles. Additionally, there are initiatives to memorise important subjects for quicker manual NQA. Automated systems are necessary for the NQA process due to these considerations. Example of such a question-answer (QA) pair is as follows.

Q: *Which company clocked* 2 *billion in GMV in calendar 2020?*

A: *Myntra*

1.1 Motivation

These days, we have a sizable archive of digital news pieces that date back decades. They do include useful information, but because of their size and complexity, it has become difficult to use these articles. This issue has been addressed by open-domain question answering (ODQA), which provides solutions to the queries based on substantial unstructured documents. Large-scale data sets currently in use are built using Wikipedia or other synchronic documents. Temporal news reports can aid journalists in relating their accounts of certain

historical occurrences. It can also be useful for historians who research the past and for those who gauge risk based on prior occurrences.

This piece takes a two-step approach to the NQA task. We first create a tool that can extract information and photos from news stories. We have extracted the news data from a variety of sources in order to maintain diversity. Once we get the data, we train it using a retriever-reader model. In this paradigm, the reader module is in charge of locating the answer span in the document, and the retriever is in charge of locating the pertinent document(s) that may contain the answer.

1.2 Background

Machine reading comprehension (MRC) [ZLL$^+$20] is the capacity to read and comprehend the material. Knowing the definitions of words, being able to deduce a word's meaning from context, being able to follow a passage's organization and spot antecedents and references, being able to conclude a passage's content, being able to pinpoint the passage's main idea, and being able to respond to questions within a passage are fundamental abilities needed in an efficient machine reading comprehension system. MRC's Question and Answer (QnA) program are one of its most well-liked ones. The effectiveness of machine learning methods, particularly Deep Neural Networks, in processing sequential data, such as texts, in recent years has made MRC an active topic in the NLP community.

1.3 Problem Definition

In this work, we make two separate attempts to assess the generated dataset. The first approach does a manual verification to verify the calibre of the QA pairs via a website where users may manually analyse the correctness of the answer against the passage. In the next part, we evaluated the consistency of the dataset using a well-known NLP-based model called BERTSerini[YXL$^+$19]. In the following phase of the project, we focus on

broadening dataset variety without compromising consistency.

1.4 Organization of The Report

The remainder of the report is structured as follows. The methods used to construct data sets for completing the ODQA assignment are discussed in Chapter 2, along with a review of the literature and earlier work on open domain question answering models. The methods used to construct the dataset in the Indian context is presented in Chapter 3. Additionally, a dataset analysis is presented. The manual verification of the produced dataset using the hosting website is described in Chapter 4. The dataset's effectiveness and consistency are assessed using the NLP-based method described in Chapter 5. Chapter 6 concludes by summarising the current work and outlining potential future expansions.

Chapter 2

Literature Review and Previous Work

The models pertinent to the NewsQA job are discussed in this chapter along with the current models on question-answer pair dataset generation and question answering.

2.1 Question-Answer Dataset Generation Model

A large amount of unstructured data is required for tasks like Open Domain Question Answering (ODQA). We also require supervised data, which contains a huge number of question-answer pairings, in order to train the models. Here, the question serves as the inquiry and the result, which we must forecast, is the answer. There are models that generate relevant data sets for our goal.

2.1.1 The NewsQA Dataset

The SQuAD[RZLL16] database is the most well-known data collection that is utilized for the Question Answering activity. This data set, which was gathered from the public, fixes issues with earlier data sets. However, Wikipedia articles served as the data set's primary source. The most relevant data collection for our work is thus NewsQA [TWY$^+$16]. It

has 12,744 CNN news stories and 119,633 natural language questions assembled by crowd workers.

Some challenging characteristics of NewsQA data-set which distinguish it from other data-sets are as follows.

1. Instead of using a single word or entity, the answer span varies within an article.
2. In the article, there are certain queries that go unanswered.
3. The answers to many queries go beyond simple word and context matches.

A 4-step approach of article curation, question sourcing, answer sourcing, and validation is used to get the data. To improve the data set's usefulness, a few pre-processing processes are applied after that.

2.1.2 The ArchivalQA Dataset

The absence of a suitable training data set is one of the primary problems hindering the development of open-domain question answering (ODQA). Some of them are too tiny to be utilized for training transformer-based models. Others, such as NewsQA, are adequate but their curation is done by crowdsourcing, which is a laborious procedure in and of itself. A framework for building data sets is offered in ArchivalQA[WJY21] and may be used to build a data set on a temporal collection of articles. 1,067,056 question-answer pairs with a temporal news question design make up this data collection. The pipeline of the framework consists of 4 phases.

1. **Article Selection:** The source for the raw data articles in this module is our temporal collection of data. The CNN/Daily Mail pieces are the ArchivalQA source.
2. **Question Generation Module:** As a model for generating questions, T5-base is a sizable encoder-decoder that has already been trained. The model was initially polished on SQuAD. After making final adjustments to the model, we begin providing

the named entity of each paragraph together with the paragraph itself as input to the model. The amount of tokens in the paragraphs and the responses to the associated questions were similarly limited.

3. **Syntactic & Temporal Filtering Module:** This module defines a few rules that assisted in preserving the dataset's quality. Rules include things like eliminating duplicate questions, eliminating questions with an insufficient or excessive number of specified entities, etc.

4. **General & Temporal Ambiguity Filtering:** To exclude question-answer combinations that are temporally ambiguous, 3 distinct BERT-based models were trained. The question "How many individuals participated in the COVID immunisation drive?" is an illustration of a temporal ambiguous question.

2.2 Question Answering Models

We require suitable models to train the data on once the question-answer dataset is prepared. The performance of ODQA tasks has significantly improved as a result of recent developments in *Retriever-Reader* models. To complete the work, we used a basic model called BERTserini[YXL+19].

2.2.1 BERTserini

The BERTserini[YXL+19] is a retriever reader ODQA model. It combines IR best practises with a BERT reader to find the range of responses to a given topic in the extensive Wikipedia corpus.

Anserini Retriever

It is a single-stage retriever that can choose out passages of Wikipedia material to send to the BERT reader. We extract k text segments using the question as a "bag of words"

query in order to draw conclusions. The model uses the post-v0.3.0 branch of Anserini with BM25 as the ranking function (Anserini's default settings).

BERT Reader

The BERT reader receives the text chunks from the retriever. The model was taken from [DCLT18]; however, it has undergone major changes. To allow for data aggregation and comparison across diverse segments, the final SoftMax layer that covered a range of response spans was deleted. The model implementation supplied by Google served as the foundation for the current BERT reader.

For training, the SQuAD training set is employed, and the BERT-Base model is used for fine-tuning (v1.1). The reader's inputs are padded to 384 tokens, and all other default settings are used.

We put each of the retrieved papers through the BERT reader individually at the time of inference. The reader selects the text passage that is the most persuasive and grades it. The reader score is then combined with the retriever score via linear interpolation as

$$S = (1 - \mu).S_{Anserini} + \mu.S_{BERT} \tag{2.1}$$

where $\mu \in (0, 1)$ is a hyper-parameter.

2.3 Evaluation Metric

2.3.1 BERTScore

In ODQA operations, exact matching is frequently employed as an evaluation metric. However, BERTScore[ZKW$^+$19] offers a more precise number as a measure of evaluation. By focusing on the semantics of the produced data and the real data, it does the comparison.

We consider both the reference (ground truth) and candidate when we run them through the pre-trained BERT[DCLT18] model, which generates contextual embedding for each

word (created one). After obtaining the final embeddings for each of these words, we do an n-squared computation to see how similar each word is to each word in the candidate. From the list of references, we identify and pick the phrase that most closely resembles Candidate One, after which we compute Precision, Recall, and F1-Score.

2.4 Conclusion

Here, we learnt about all the models that are applied to both the dataset's generation and its assessment.

2.5 Previous Work

Different ODQA tasks required the usage of various datasets. They could not obtain any relevant datasets that could serve as a base dataset for the NewsQA task because they are working on Indian News. They, therefore, produced their own dataset.

2.5.1 Dataset Generation Framework

The Indian news question-answer dataset was created in two main processes. We must first obtain the news items from various news sources before using the ArchivalQA [WJY21] model to create the question and answer pair.

2.5.2 Scrapping new articles

They had to crawl the web to obtain the news stories. To retain the diversity of the dataset, they chose five distinct sources from which to obtain the articles. They created a platform that allowed users to download news stories from a specified time period. Five separate scrapers were created to gather information from all five sources—**TOI, NewsByte, Print, Scroll, and NDTV**. The information is kept in a certain format. The article's core material is kept in a text file whose location is preserved in a JSON file,

together with all of the article's fundamental details, including weblinks, dates, titles, and relevant images.

2.5.3 Generation and Analysis

They constructed a raw pickle file (containing binary data) comprising a list of lists after gathering the news stories. Each inner list has three components: the text, the date, and the paragraph id. The data frame used as an input for the ArchivalQA [WJY21] model has this format. This model consists of the four processes that were previously covered in the literature review section. Data collection was the initial phase, which was carried out using our proprietary scraping technology for news articles. The second module, question generating module, was now fed the pickle file as an input. The final question-answer dataset is obtained when all of the modules of the ArchivalQA framework have sent this data on. Table 2.1 shows the number of questions created and filtered after each module.

Table 2.1 Stage-wise question-answer created for the year 2021

Stat/source	TOI	Economic Times	NDTV	Scroll	News Byte	Total
#ques(after module 2)	251,879	417,612	401,946	400,070	396,491	1,867,998
#ques(after module 3)	151,601	478,973	301,579	271,679	269,328	1,473,160
#ques(after module 4)	52,063	151,304	106,321	70,075	101,995	**481,758**

It has been noted that they were able to gather 500,000 question-answer pairs from 50,558 news stories in just one year. They separated the dataset into a train, development, and test set in a ratio of 90:5:5. Simple database extensions may be made with this framework.

Chapter 3

Contributions

3.1 Contribution I : Web App for Dataset Verification

3.1.1 Django Framework

Here, we discuss the **Model-View-Template(MVT)**-based Django framework. There are mostly three sections. This architecture is illustrated in Figure 3.1.

Model: The model will serve as the data's interface. It is in charge of data maintenance. It is represented by a database and serves as the logical data structure for the whole programme (here we used SQLite as database).

View: The View is the user interface, or what we see when we render a website in our browser. Jinja files, HTML, CSS, and Javascript files are used to represent it.

Template: The static components of the intended HTML output make up a template.

The controller's job is handled automatically by Django internally under the MVT pattern. The templates would need to be modified by us, the developers. Django searches up urls to reroute itself to the proper page when a user interacts with the application. HTML templates are files that use data to construct the front end.

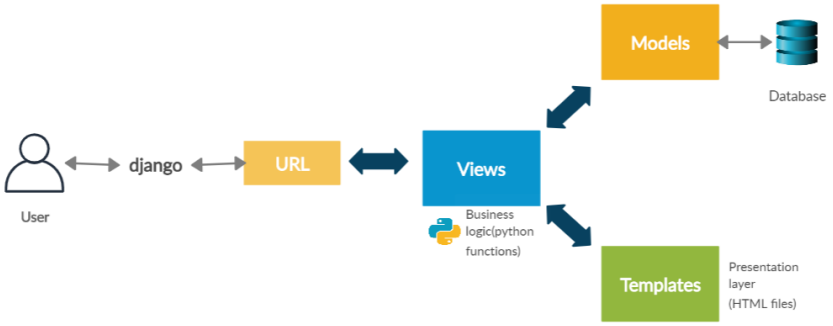

Fig. 3.1 MVT Architecture

Our web application, which is based on MVT architecture, chooses an example from the QA dataset that is created. The triplet containing a question, an answer, and a pertinent paragraph serves as the model for this QA pair. Now, a person with an account logs in to the website and manually verifies, by reading the paragraph, if the produced response is accurate for the related question. If the provided response is incorrect for the given question, the user can supply the appropriate response from the paragraph; otherwise, the response should be marked as satisfactory. This process is illustrated in Figure 3.2 Figure 3.3 Figure 3.4.

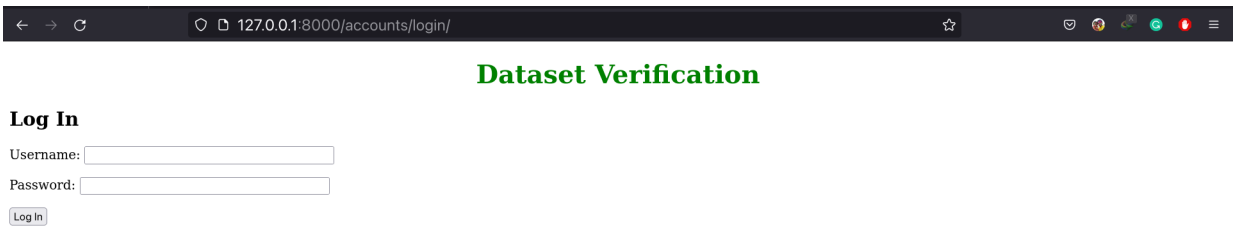

Fig. 3.2 Login Page

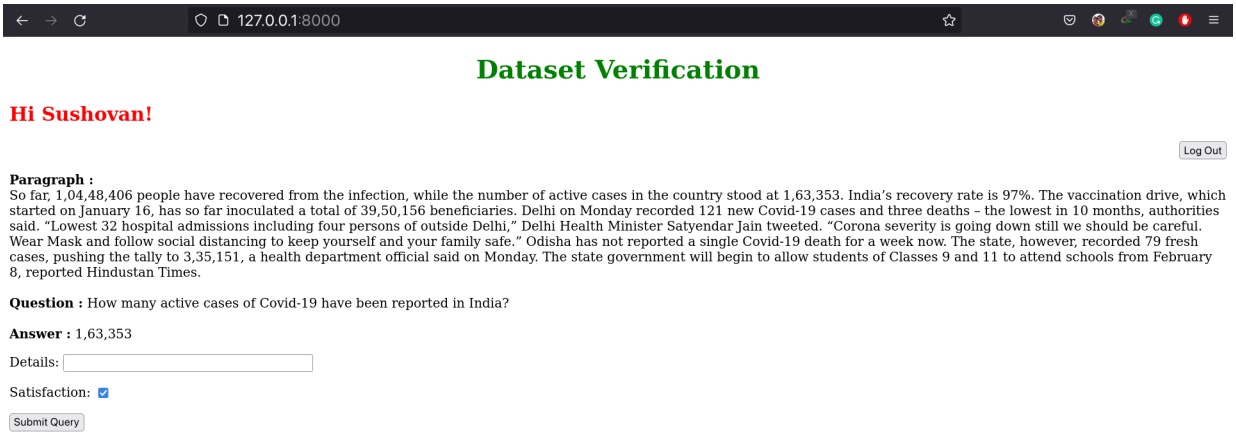

Fig. 3.3 Verification Page

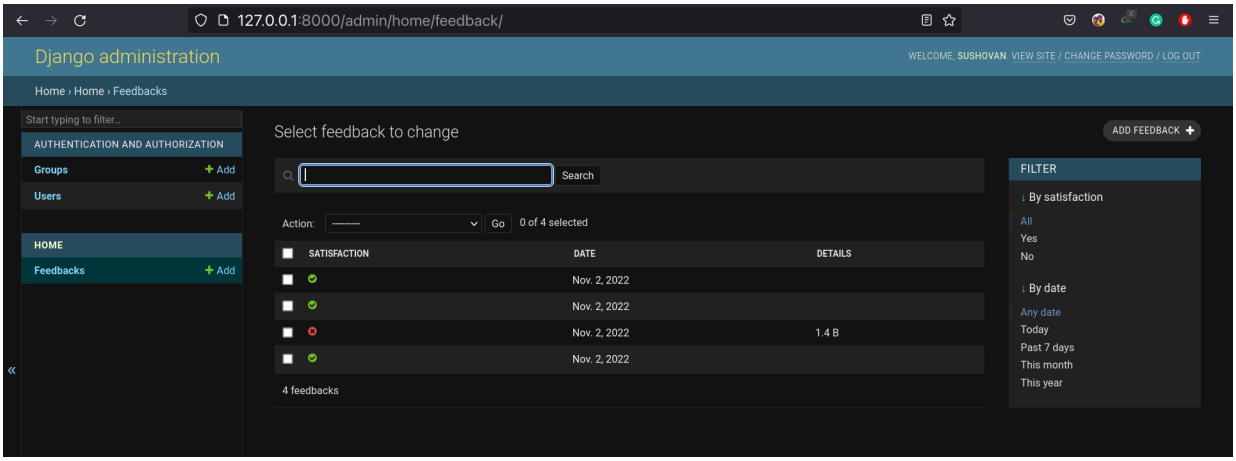

Fig. 3.4 Snapshot of Database Responses

3.1.2 Conclusion

Here We talked about the creation of the QA dataset. We will concentrate on evaluating the created dataset in the following chapter.

3.2 Contribution II : NLP based Evaluation of Dataset

3.2.1 Generate Answer using BERTSerini

An ODQA retriever-reader model is the BERTserini[YXL+19]. In order to determine the range of responses in the corpus, it combines the best principles of information retrieval (IR) with a BERT reader. It comprised of two main modules,

Anserini Retriever : selecting segments of text that contain the answer

BERT Reader : identify an answer span.

Now that BERTSerini gives users the option to select the corpus from which the response will be created, two options are available: the Wikipedia corpus and our own related texts. In our trial, we went with option 2.

3.2.2 Evaluation of Answer

The response that BERTSerini produces for a given question from the pertinent paragraph serves as our **reference answer**, while the response that is already included in the dataset for that question serves as our **candidate answer**. Now, BERTScore[ZKW+19] is used to compare the accuracy of our candidate response to the created reference answer.

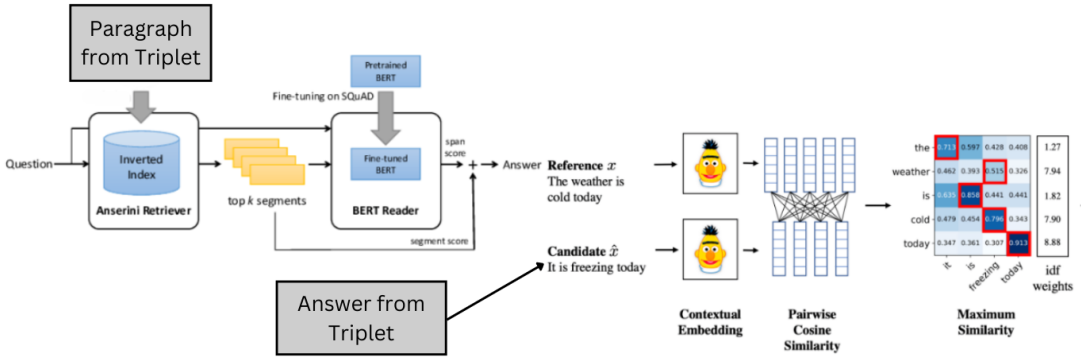

Fig. 3.5 BERTSerini Evaluation Model Architecture [YXL+19][ZKW+19]

$$R_{BERT} = \frac{1}{|x|} \sum_{x_i \in x} \max_{\hat{x}_j \in \hat{x}} \mathbf{x}_i^T \hat{\mathbf{x}}_\mathbf{j} \tag{3.1}$$

$$P_{BERT} = \frac{1}{|\hat{x}|} \sum_{\hat{x}_j \in \hat{x}} \max_{x_i \in x} \mathbf{x}_i^T \hat{\mathbf{x}}_\mathbf{j} \tag{3.2}$$

$$F_{BERT} = 2\frac{P_{BERT}.R_{BERT}}{P_{BERT} + R_{BERT}} \tag{3.3}$$

3.3 Contribution III : Model for Question Answering

3.3.1 Train Question Answering Model using Flan T5

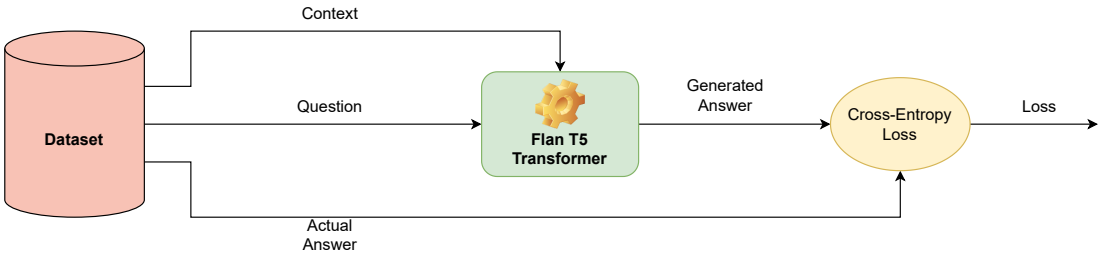

Fig. 3.6 Question Answering Model using Flan T5
[YXL+19][ZKW+19]

3.3.2 Train QA Model using Flan T5 and Context Fetching using BM25

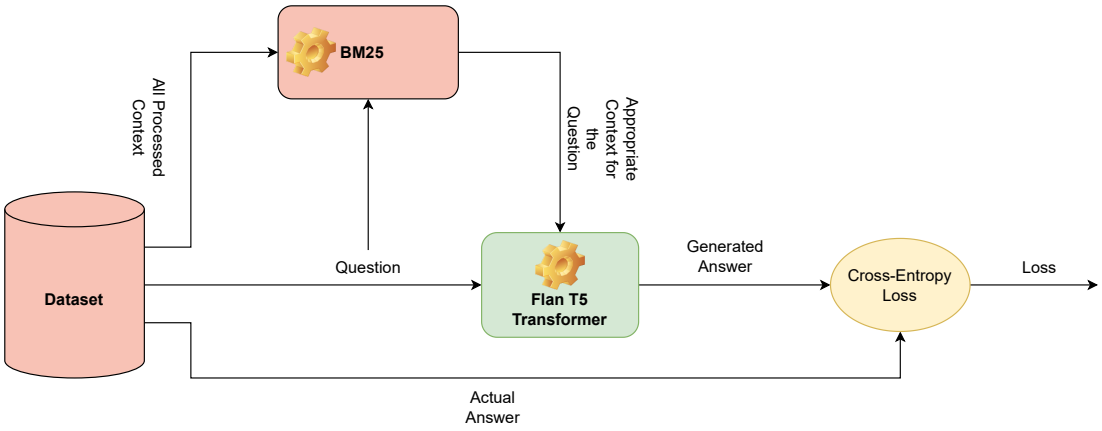

Fig. 3.7 QA Model using Flan T5 and Context Fetching using BM25
[YXL+19][ZKW+19]

3.3.3 Train QA Model using Flan T5 and Context Fetching using KNN

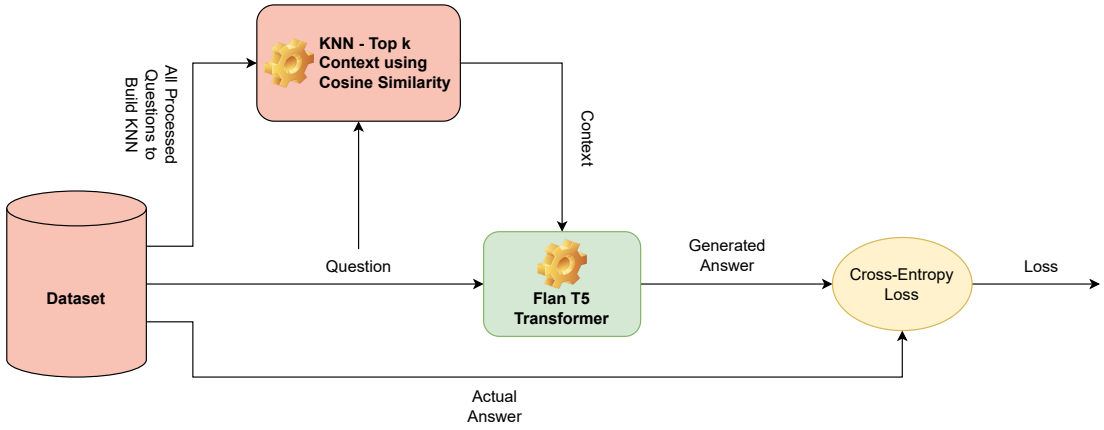

Fig. 3.8 QA Model using Flan T5 and Context Fetching using KNN [YXL+19][ZKW+19]

3.3.4 Train QA Model using Flan T5 and Context Fetching using SBERT

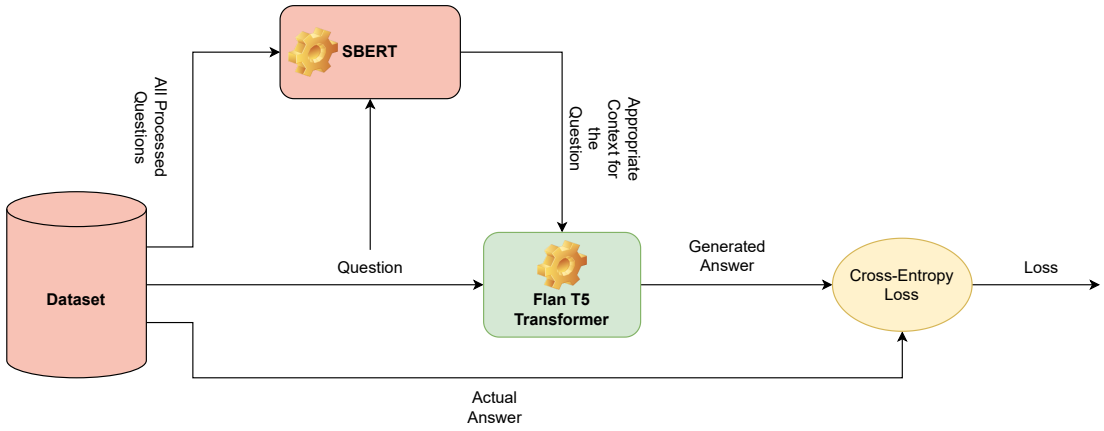

Fig. 3.9 QA Model using Flan T5 and Context Fetching using SBERT [YXL+19][ZKW+19]

3.4 Contribution IV : Teacher Student method for QA

3.4.1 Attention based Encoder-Decoder Model for QA

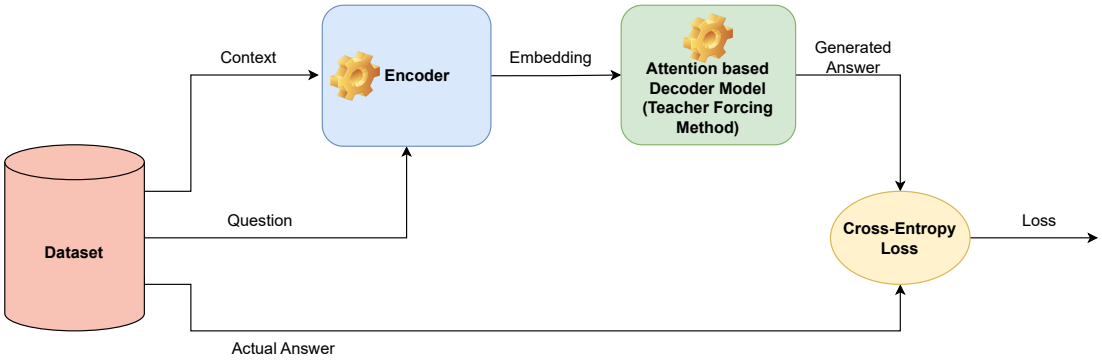

Fig. 3.10 Attention based Encoder-Decoder Model for QA [YXL+19][ZKW+19]

3.4.2 Improving Encoder-Decoder Model using Teacher Student Method

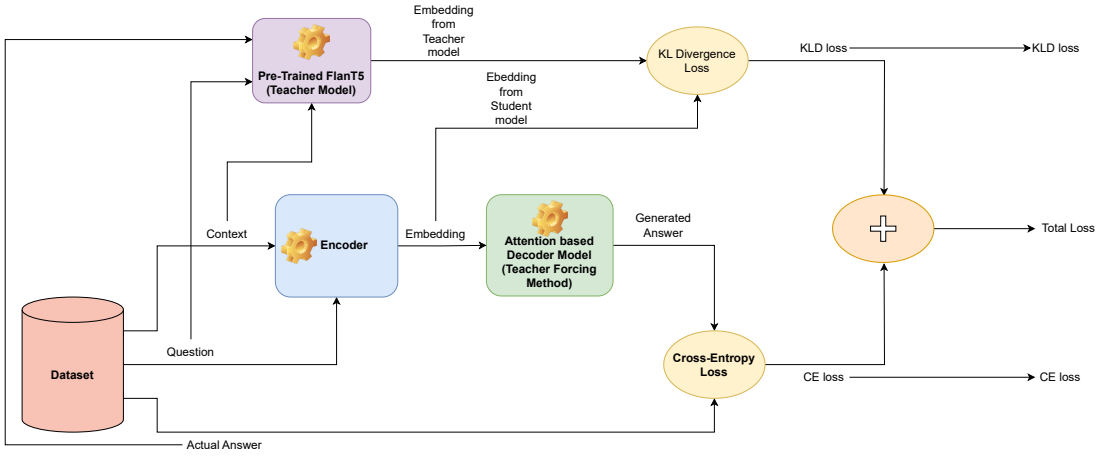

Fig. 3.11 Attention based Encoder-Decoder Model using Teacher Student Method [YXL+19][ZKW+19]

3.5 Contribution V : GUI for QA Model

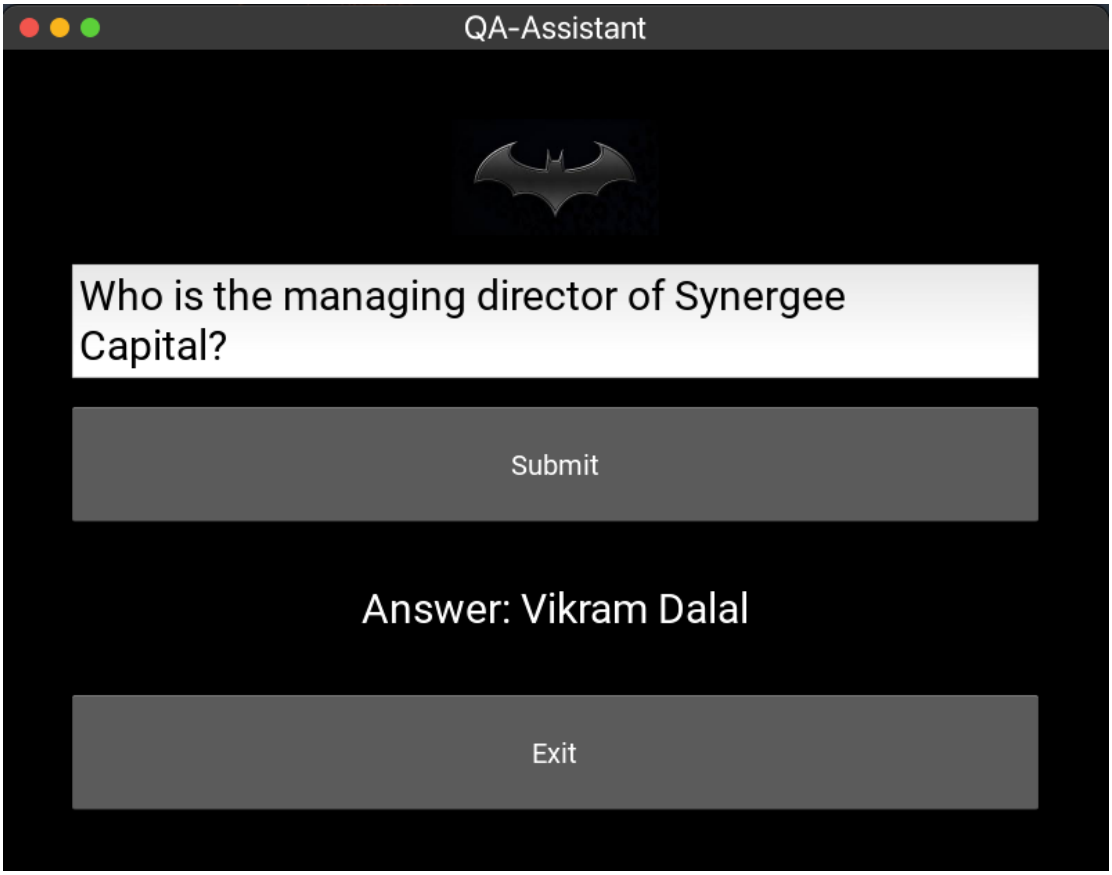

Fig. 3.12 GUI for QA Model
[YXL+19][ZKW+19]

Chapter 4

Experiment and Results

4.1 Results

4.1.1 Contribution II : NLP based Evaluation of Dataset

As a technique of evaluation, we employed BERTscore. Precision, recall, and F1 score were computed. We used the validation set to obtain the final results. We simply used the responses from the BERT model that had been pre-trained on the SQuAD dataset to get the scores.

Table 4.1 Score for the expermintal results

Model	Precision	Recall	F1
BERTserini(Our Dataset)	0.521	0.552	0.574

Chapter 5

Conclusion and Future Work

5.1 Conclusion

We assessed the constructed dataset in two distinct yet practical methods, making the dataset reliable for usage. We manually checked the QA pair in the first strategy. Even though it is a complicated process to review such a large dataset personally, we do this with the aid of a hosting website set up for this reason and spread among many users. The second section evaluates the dataset using an extremely well-known modern NLP model and evaluation matrices. The outcomes of our analysis demonstrate the consistency of the created QA pairings in the dataset. The dataset may be expanded using this manner for any future or prior time period.

5.2 Future Work

The dataset that we reviewed is fairly accurate and consistent, however there is still potential for growth in terms of pair quality and diversity. The questions are produced in accordance with the current dataset, which mostly emphasises replies of the designated entity type. Again, the answers in the dataset are brief. Therefore, we may add descriptive answers to this dataset to expand it.

We pick models that are extractive in nature when training them. The retriever reader model takes a set of articles and chooses the ones that are most pertinent before extracting the solutions. Generative models [IG21] are a brand-new kind of QA model. These models are trained on the dataset, however after becoming trained on the dataset, they produce the answers rather than extracting them. To train these models and compare their output, we may utilise our dataset.

References

[DCLT18] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018.

[IG21] Gautier Izacard and Edouard Grave. Leveraging passage retrieval with generative models for open domain question answering. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 874–880, Online, April 2021. Association for Computational Linguistics.

[RZLL16] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100, 000+ questions for machine comprehension of text. *CoRR*, abs/1606.05250, 2016.

[TWY+16] Adam Trischler, Tong Wang, Xingdi Yuan, Justin Harris, Alessandro Sordoni, Philip Bachman, and Kaheer Suleman. Newsqa: A machine comprehension dataset. *CoRR*, abs/1611.09830, 2016.

[WJY21] Jiexin Wang, Adam Jatowt, and Masatoshi Yoshikawa. Archivalqa: A large-scale benchmark dataset for open domain question answering over archival news collections. *CoRR*, abs/2109.03438, 2021.

[YXL+19] Wei Yang, Yuqing Xie, Aileen Lin, Xingyu Li, Luchen Tan, Kun Xiong, Ming Li,

and Jimmy Lin. End-to-end open-domain question answering with bertserini. *CoRR*, abs/1902.01718, 2019.

[ZKW+19] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with BERT. *CoRR*, abs/1904.09675, 2019.

[ZLL+20] Chengchang Zeng, Shaobo Li, Qin Li, Jie Hu, and Jianjun Hu. A survey on machine reading comprehension: Tasks, evaluation metrics, and benchmark datasets. *CoRR*, abs/2006.11880, 2020.