

State-of-the-Art Approach to e-Learning with Cutting Edge NLP Transformers: Implementing Text Summarization, Question and Distractor Generation, Question Answering

Spandan Patil¹, Lokshana Chavan², Janhvi Mukane³, Dr. Deepali Vora⁴, Prof. Vidya Chitre⁵
Department of Information Technology, Vidyalankar Institute of Technology, Mumbai, India^{1,2,3,5}
Department of Information Technology, Symbiosis Institute of Technology, Pune, India⁴

Abstract—Amid the worldwide wave of pandemic lockdowns, there has been a remarkable growth in E-learning. Online learning has become a challenge for students. It has become difficult for students to find the content they need. The mounting accessibility of textual content has necessitated comprehensive study in the areas of automatic text summarization and question generation. Multiple Choice Questions is very smooth for evaluations, and its assessment is implemented through computerized applications in order that results may be declared within some hours, and the evaluation system is 100% pure. The system proposes an interactive reading platform where the user can upload an E-Book and get textual summary and generates questions like MCQs, fill in the blanks and one word. The user can also evaluate the questions answered. The proposed system is an all-in-one interactive reading platform.

Keywords—Machine intelligence; natural language processing; neural networks; predictive models; text processing

I. INTRODUCTION

The “Live with Covid” era has notably modified the manner we live. The field of education can't isolate itself from the drastic adjustments, resulting in the near-total closures of schools, early childhood education and care (ECEC) services, colleges, and universities [1]. It has compelled us to follow the online mode of learning. As transferring to online learning has introduced us to flexibility and self-paced mastering. But there are few cons to this. The new format of classes for students has left them with a lack of motivation and creates a sense of isolation from the classrooms which may affect their academic performance throughout the term. Also, with the development of information technology, more and more information appears on the internet, retrieving the needed information and making sense out of huge data becomes difficult for users. Hence, there comes a need for a system which can provide us with summarization and question generation for easier and quicker retrieving of relevant information from huge chunks of data and to test the understanding of the subject through assessments.

In Section II, Review and Planning of the paper is discussed. In Section III, various text processing algorithms like LSTMs, T5, BERT, WordNet, ConceptNet, Sense2Vec, etc. are discussed and the best suited ones are elaborated. In

Section IV, Literature Survey was carried out wherein, technical research papers and some existing systems were studied. The gaps in the theory and applications are also addressed. In Section V, Inferences from the literature survey are mentioned. In Section VI, the Proposed System is described in detail along with its workflow and features. In Section VII, the implementation part of the system is discussed in detail. In Section VIII, the results are presented along with various comparisons between the findings. In Section IX, conclusions are stated and possible topics for future research are mentioned.

II. REVIEW AND PLANNING

Text Processing is one of the most common tasks in many ML applications. The review considered following queries.

Q1- What are the different techniques for performing the NLP tasks like Text Summarization, Question Generation and Question Answering?

Q2- What are the different distractor generator algorithms used to generate three distractor options in MCQs?

Q3- What challenges were faced while using these algorithms? And which one was the best suited for the use case?

For the survey, databases of IEEEEXplore, Google Scholar, and Articles were searched manually, by using various keywords like “Text Summarization”, “Question Generation”, “Question Answering”, “Distractor Generation”, etc. The search was narrowed down to research that only perform Abstractive Text Processing [2] which includes the tasks like text summarization, Question generation and question answering. The approaches to generate distractors for incorrect options of MCQs were also studied. Research papers of various Text Processing Approaches were studied from the mentioned repositories like Journals, Conferences and Articles. While trying to select the literature for the system, time duration was limited from 1997 to 2021. Along with Research Papers certain existing system were also reviewed. The research papers that satisfy the above conditions were studied and a few comparisons were made based on certain parameters. The notable points are highlighted in this paper. These remarks helped in identifying solutions to the review questions.

III. TEXT PROCESSING ALGORITHMS

Text Processing is one of the most common tasks in many NLP applications [3]. These algorithms help the computers to analyze, understand and derive meaning from human language in a smart and useful way. For this system, out of all the Text Processing Algorithms, RNN based Sequence model like LSTM and Transformer based models like T5 and BERT were studied. The highlights from research papers are mentioned below.

A. LSTM versus Transformers

LSTM were one of the most popular choices for performing Natural Language processing tasks [4], but were replaced after the introduction of current state of the art transformer which surpass LSTM over the accuracy and convenience of performing the NLP tasks [5]. Limitations of LSTM are- it is difficult to train (takes very long time), transfer learning never really worked, and it must be computed in serial per token [6].

B. T5 Transformer

The text-to-text framework introduced in the paper [7], allows NLP tasks, like document summarization, machine translation, question answering regression tasks to be trained to predict the string representation of a number instead of the number itself using the same model for loss function, and hyperparameters. The input and output of the T5 model is always purely text to text format i.e., text string as shown in Fig. 1.

C. BERT Transformer

The BERT model in [8], being inspired by the Cloze task (Taylor, 1953) alleviates the unidirectionality constraint by using a “Masked Language Model” (MLM) pre-training objective. Few of the input tokens are arbitrarily masked by the MLM and their original vocabulary id is predicted solely based on the context. It also makes use of “next sentence prediction” task in addition to mask language model to jointly pretrain text-pair representations as shown in Fig. 2.

D. BART Model

BART is a denoising autoencoder used for pre-training sequence-to-sequence models. It is trained by corrupting text with random noise function thus, model learns to restructure the original text as shown in Fig. 3. A standard Transformer with simple neural machine translation architecture is used in BART. It evaluates several noising approaches. It finds the optimum performance by arbitrarily shuffling the sequence of original sentences and thereby uses a novel in-filling scheme, where a single mask token is placed in spans of text.

E. Distractor Generator for Incorrect Options

For distractor generation mainly WordNet, ConceptNet, Sense2Vec were studied.

WordNet is a large lexical knowledgebase of English. Every word (i.e., Adjectives, nouns, verbs) is grouped into sets of logical synonyms (synsets), expressing a unique concept. Every Synset is interlinked to another by lexical relations and conceptual-semantic [10]. A hypernym is a higher-level category for a given word. Considering an example as shown

in Fig. 4, color is hypernym for red. Hyponyms are the sub-categories of an entity. A hyponym is a type-of relationship with its hypernyms [11]. A Co-Hyponym are words that shares the same hypernym as another word. To generate distractors the main goal is to extract co-hyponyms [12].

ConceptNet is a semantic network that is used to help computers understand the meaning of words that people use. It generates distractors for locations, items, etc. which have a “Part of” relationship [13]. ConceptNet Number batch is a set of semantic vectors, also known as word embeddings which can be used as direct representation of word definitions or an initial state for further machine learning [14]. Fig. 5 depicts an example of Generation of distractors using ConceptNet.

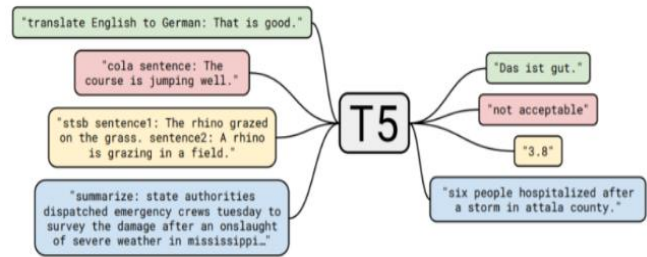


Fig. 1. T5 Text-to-Text Framework. [7].

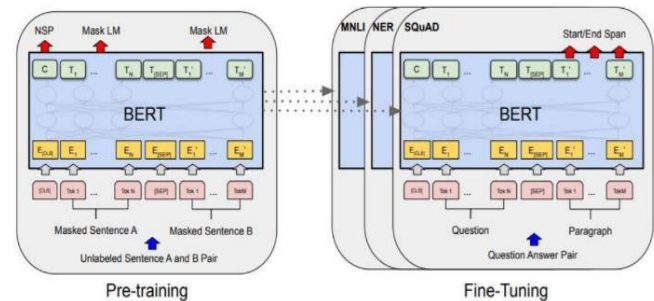


Fig. 2. BERT Fine-Tuning Procedure. [8].

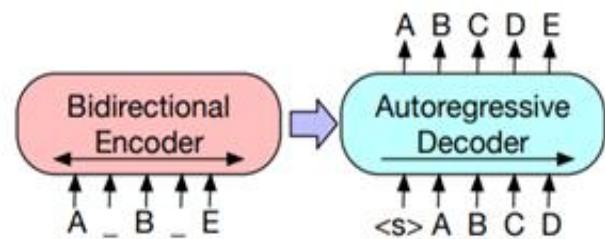


Fig. 3. A Schematic Diagram of BART. [9].

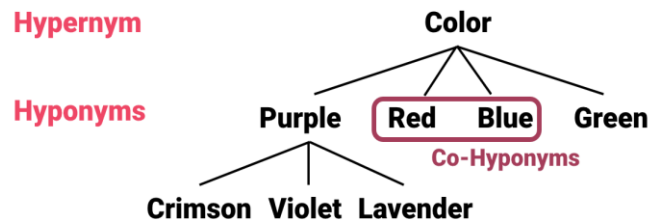


Fig. 4. An Example of Relationship between Hyponyms and Hypernyms.

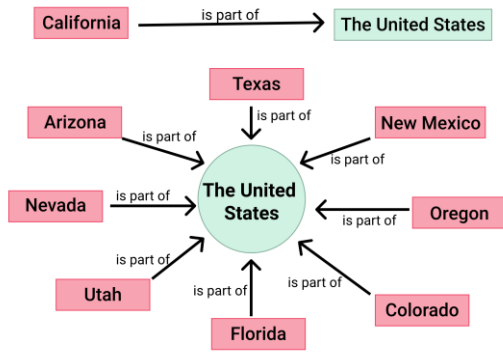


Fig. 5. An Example of Generating Distractors using ConceptNet.

Sense2Vec automatically generates relations among words from a text corpus in contrast to being human-curated. To predict a focus word given other words or to predict surrounding words of a given focus word a neural network algorithm is trained with millions of sentences as its dataset. Thus, resulting word vectors which are fixed size vectors or array representation of every word. The associations between different kind of words are represented by these word vectors, thus preserving the relationship among various words. The system uses 2015 Reddit vectors instead of the 2019 as the output obtained was slightly better.

IV. LITERATURE SURVEY

For the system, literature survey was conducted into two parts. For the first part of literature survey, several research papers related to text summarization, question generation and question answering were studied. The inferences from paper reading are tabulated in Table III. Later, study of existing systems was carried out that aid in the process of interactive reading experience and self-assessment. The observations from existing systems' review are tabulated in Table IV.

V. SURVEY INFERENCE

In the mentioned reviews, out of all the Text Processing Algorithms, RNN based Sequence model like LSTM and Transformer based models like T5 and BERT were studied. T5 is an integrated text-to-text model with text strings as its input and output, whereas BERT-style models generate outputs as a class label or a span of the input.

The issue of understanding each word based on the understanding of previous words couldn't be handled by traditional neural networks. Thus, Recurrent Neural Networks were introduced to handle the same. These networks have loops in them, allowing information to persist. The limitations of traditional RNN are that computation is slow because of the concurrent nature. If relu or tanh are used as activation functions, it becomes very difficult to process longer sequences. It is vulnerable to issues such as exploding and gradient vanishing. Further LSTMs came into picture. LSTMs are a special kind of RNN, capable of learning long-term dependencies and work well on a large variety of problems. Transformer became a huge achievement over the RNN based seq2seq models. Using transformer All to All comparison can be done fully parallel, it has multi-headed attention and

positional encoding and Transfer Learning worked well on it. But its limitations include attention can solely deal with text sequences whose size is pre-defined. The sequence must be split into fixed-sized segments or chunks before being given as input into the system. Fig. 7 depicts evolution of text processing algorithms.

Hence, due to its advantages in terms of speed and compatibility to the task, the different transformer models like T5, Distil BERT, Distil BART were decided to be used in the system for performing text summarization, question answering and question generation tasks as shown in Fig. 6.

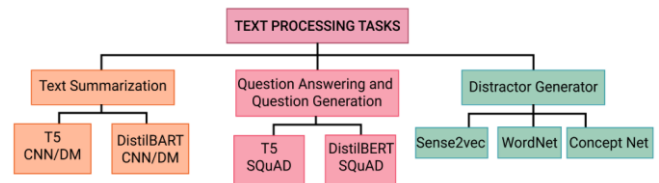


Fig. 6. Categories of Text Processing Tasks.

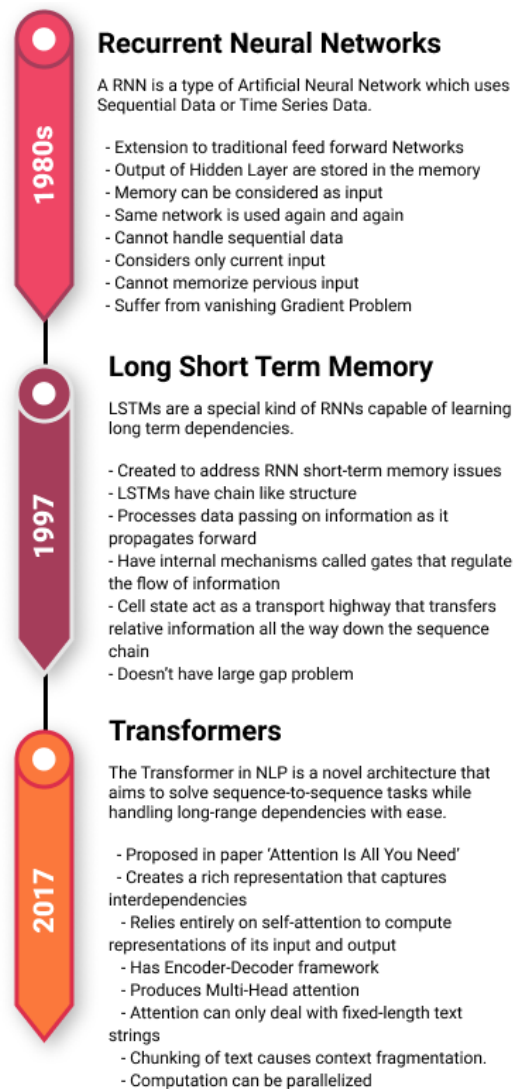


Fig. 7. Evolution of Text Processing Algorithms.

VI. PROPOSED SYSTEM

Firstly, the user needs to sign up to the system and using credentials, log in to the home page. They can reset their passwords and edit their profiles as per their wish. The users will be categorized into guests and authors.

When a user (guest) authenticates into the system, an application tutorial will be displayed to make them familiar on how to use various features of the system. The user will either choose an E-Book (pdf) from the system library or upload one of their own. Then, the user will be able to read the E-Book in an E-reader, also have access to page wise summary and a self-assessment of the E-Book. The assessment will have three categories of questions – MCQ's, Fill in the blanks and One-word type questions. Additionally, the user can generate summarization from series of pages and get solution to a question they ask in context to a specific page. The system will display a collection of top-rated E-Books scraped from an E-Book rating and cataloguing website.

When authenticated authors logs into the system, they will be allowed to publish their E-Book (pdf), auto-generated page wise summary and a self-assessment of the E-Book to the library. The author will be able to search for their published E-Book from the library and get a preview of the generated summary and self-assessment. Furthermore, if they wish they can modify the same. Fig. 12 depicts proposed system.

VII. IMPLEMENTATION

The systems' frontend designing is done using React.JS and in the backend, node.js express is used to host the web server and mongo DB Atlas is used for the database [15].

In the Mongo DB Atlas, a FLIP database is created in which there are three collections.

- User's E-Books: It stores all the data of the E-Book uploaded by the user.
- FLIP library: It stores all the data of the E-Books available in system's library.
- Authorized Authors: It stores the user's IDs of all the users that are categorized as authors.

MongoDB was chosen for its flexibility, scalability and cloud storage services. Also, it eases the restriction of a schema for of DB.

In the node server, APIs are available for uploading the E-Book to the server, performing CURD operations on the Mongo DB. It also runs three Python Scripts to perform mainly three tasks which are-

A. Summary Generation and Self-Assesment Script

Firstly, for summary generation, T5 for Conditional Generation and T5Tokenizer from T5 base is used. This model was chosen over its other competitors like BERT, because it's pretrained on the much large and cleaner C4 dataset and in comparison, (base version), it contains nearly twice the number of parameters as BERT (T5: 220M & BERT: 110M) [7]. Additionally, it was also pre-trained specific for the text summarization task so no further fine tuning was required [8].

The content is then Pre-processed for removing all white spaces and is passed through the T5Tokenizer to get it tokenized. Its limitation is that there is a maximum limit of 509 tokens (excluding special tokens) for generation of summary. It was overcome by extracting the first chunk of 509 tokens from the tokenized content and then special tokens were added to them. Further, they are passed to the T5ForConditionalGeneration model to generate the summary whose length must be between 100 to 508 tokens. This generated summary is then added back to the front of the tokenized content. Now, the above process is repeated till tokenized content has less than 509 tokens. What this essentially does is retains the context of the previously generated summary with addition to the context provided by the additional tokens added in current iteration. Repeating this process will shorten down the size of the tokenized content to 509 tokens or less so that they can be fed to the T5 model at once without losing much context of the previous iteration.

An alternative approach to this, would be using the pre-trained DistilBART model. The process would majorly remain unchanged except for the limit for the length of the token chunk would be increased from 509 to 1022.

Next, Extraction of keywords is done using NER (Name Entity Recognition). Basic advantage of this method over other keyword extraction algorithms like Multipartite Rank, Tfidf, TextRank, etc is that it is able to identify Named Entities (NEs) which are real-life objects that are proper names and quantities of interest. And heuristically, when selecting answers for MCQ or other type of question in non-language related subjects these Named Entities have shown to provide more relevant and correct questions.

The name entities are extracted using SpaCy library from the content to get a list of keywords to be used as answers to the questions generated [16]. Using Maximal Marginal Relevance (MMR), the top five most relevant name entities can be procured out of the extracted ones. In MMR, the keywords that are most analogous to the text are selected. Then, iteratively new candidates are selected such that they both are analogous to the text and not analogous to the previously selected keywords. The similarities are measures based on Cosine similarity [17].

An alternative to MMR is Max Sum Similarity (MSS), The maximum sum distance of a pairs of data is calculated as the maximized distance which exists in between the two data points. In this case, candidate similarity was expected to be maximum to the document while minimizing the similarity between candidates. But a drawback in this is that, to get more diverse options there is a need to provide larger number of keywords to filter from, which is not possible for this system every time.

Then, for Generation of the question pre-trained T5ForConditionalGeneration base model is taken and it is fine-tuned on the question generation task using the SQuAD-The Stanford Question Answering Dataset [18]. The construction of fine-tuning dataset is done in the form of 3 columns the context, the answer and the question. The input is provided in the format of


```
context : (-context-) answer : (-answer-) </s>  
question : (-question-) </s>
```

and provide the target in the following format-

In this way, nearly 80,000 rows are trained as a part of training dataset and 10,000 rows are used for validation of the trained model. For training the model batch size of 4 is used and for 1 epoch (which nearly takes about 4 hours to complete on the google collab notebook). Now, iteration over the keyword list is carried out and this fine-tuned model is used to generate questions for each keyword as output given the input as “context: (--summary--) answer: (--keyword--) </s>”.

To generate the distractors for the keywords the Sense2Vec Reddit 2015 is used. As Sense2Vec performs better with Named Entities compared to other algorithms like Wordnet, ConceptNet, etc.

The list of keywords is iterated and then- the best sense for selected keyword is generated and then the top thirty most similar words to it are found based on the best sense it gets. Then, the list of these words is filtered using Normalized Levenshtein Distance with a threshold of 0.7. Further, to select the top three distractors the MMR is used, by comparing the selected keywords and the distractors that are found. If any distractors for the keywords are not found, then those question-Keywords pair are used for one-word type questions.

Now, for generating fill in the blank’s questions, the top three keywords to be used as answers to the questions are found using the Multipartite Rank algorithm in the Python keyword extraction library as it seems to work best for these types of tasks [19]. Using the keyword processor in the FlashText library, the keywords are mapped with the sentences of which they’re a part of and then this Sentence-Keyword pair are used as Fill in the Blanks questions.

B. Summary Generation for Collection of Pages Script

This script takes concatenated content of series of pages as input and then summarizes them using procedure same as that of the Summary Generation part mentioned in the Summary generation and self-assessment script.

C. Question Answering for a Question asked in Context of a Specific Page Script

In this, pre-trained T5ForConditionalGeneration base model is taken and fine-tuned on the question answering task using the SQuAD- The Stanford Question Answering Dataset.[18] The fine-tuning dataset is constructed in the form of 3 columns- the context, the answer and the question. The input is provided in the format of “context: (--context--) question: (--question--) </s>” and provide the target in the following format of “answer: (--answer--) </s>” In this way nearly 80,000 rows are trained as the part of training dataset and 10,000 rows are used for validation of the trained model. For training the model batch size of 4 is used and for 1 epoch (which nearly takes about 4 hours to complete on the Google collab notebook). Now, this fine-tuned model is used to generate answers for the input questions, context which are fed to the model in the following format of “context: (--page_content--) question (--input_question--) </s>”.

An alternative approach to this would be using the pre-trained DistilBERT model. The process of fine tuning would majorly remain unchanged except for the input format would become “CLS (--question--) SEP (--context--) SEP” and for target is “CLS (--answer--) SEP”.

In the frontend, for sign up and sign in, Google Firebase Authentication Services are used. Thus, enabling the users to register using an email ID, password. And then signing in using the credentials provided to them. Further they are also provided with user support like resetting the password and updating the profile [20].

To upload an E-Book to the server, an upload API is used which in turn uses the node.js express- fileupload package and triggers the Summary generation and self-assessment Script where the summary and self-assessment for each page of the E-Book is generated and then the Inserting API of the CURD APIs is used to upload the generated content to the Mongo DB’s user’s E-Book collection. To retrieve the generated content of the E-Book uploaded by the user, the Retrieving API of the CURD APIs is used which uses E-Book name and user ID as a query in the Mongo DB user E-Book collection. To generate the summary for series of pages, the Summary Generation API is used which requires the content (concatenation of the content of series of pages) as request parameters. The API triggers Summary generation for Collection of pages Script and gives the script received content as an input. To generate answers to the questions in context to a specific page, the Question Answering API is used which requires the content (content of the specific page), question as request parameters. The API triggers Question Answering Script and provides the script with received content and question as input. For authors to upload an E-Book to the FLIP library, the library upload API is used which is same as upload API but the difference being that it uploads the E-Book to the FLIP library collection instead of the user’s E-Book collection. For authors to modify the content of the selected E-Book, the Modification API is used which uses the update API of the CURD APIs to update the data of the E-Book in the FLIP library collection.

VIII. RESULTS

A. Text Summarization

This task was performed using two approaches namely Pre-trained T5 model and Pre-trained DistilBART model. The performance of the two was evaluated based on the evaluation benchmarks like CNN/DM-ROUGE-1, CNN/DM-ROUGE-2 and CNN/DM-ROUGE-L which are recorded in the Table I.

1) *WMT 2016*: It is a group of datasets which can be used in the shared tasks - IT domain translation, an automatic post-editing, news translation, biomedical translation, etc. [21]. The score for WMT English to Romanian and Romanian to English are referred to as En-Ro and RO-EN [22].

2) *CNN/ daily mail*: It is a text summarization dataset. From CNN and Daily Mail Websites news stories, human generated abstractive summary bullets were generated as questions (with one of the entities hidden), and news stories as the corresponding passages which are used by the system to

answer the fill in the blank question as expected. The websites were crawled, using scripts released by authors and these scripts also extracted and generated pairs of passages and questions [23].

3) *ROUGE*: Recall-Oriented Understudy for Gisting Evaluation (ROUGE) is a collection of metrics used for evaluating natural language processing tasks like text summarization, machine translation software. An auto-produced summary or translation is compared against a reference or set of references (human-produced) summary or translation by the metrics.

- Rouge-N: Overlying of N-grams among the reference summaries and the system.
- Rouge-L: Longest Common Subsequence primarily totally based statistics. It takes into consideration sentence level structure similarity clearly and identifies longest co-occurring in collection N-grams automatically [24].

B. Question Answering

This task was performed using two approaches namely Pre-trained T5 model and Pre-trained DistilBERT model. The performance of the two was evaluated based on the evaluation benchmarks like GLUE and SQuAD which are recorded in the Table II.

1) *GLUE*: The General Language Understanding Evaluation (GLUE) benchmark is a set of resources used to train, evaluate, and analyze Natural Language Understanding systems. It is of the model-agonistic format and hence, any system that is capable to process sentence and sentence pairs or outputting corresponding predictions is eligible to participate. Its' final goal is to drive analysis within the development of general and robust NLU systems [25].

2) *SQuAD*: The Question Answering Datasets use two major metrics called SQuAD Exact Match (EM) and SQuAD

F1 Scores. SQuAD EM is a simple yet strict all-or-nothing metric wherein every question-answer pair, if the characters of the model's prediction exactly match the characters of (one of) the True Answer(s), EM = 1, otherwise EM = 0. SQuAD F1 score is metric used for classification problems, and QA. It is ideally used when precision and recall are of equal importance It is calculated over individual words within the prediction against those in the True Answer. The premise of F1 score is that the number of shared words between the truth and prediction: precision is the ratio of the quantity of shared words to the overall number of words in the prediction, and recall is the ratio of the count of shared words to the total count of words in ground truth [26].

$$F_1 = \frac{2}{\text{recall}^{-1} + \text{precision}^{-1}} = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \quad (1)$$

C. Question Generation

For gaining knowledge about the accuracy of the system trials were carried out on 5, 10, 15 and 20 samples of E-Books each five pages long. The results of this approach on the system are recorded in the Table V and Table VI.

TABLE I. PERFORMANCE BASED ON EVALUATION BENCHMARKS FOR TEXT SUMMARIZATION

	WMT EnRo / RO-EN	CNN/DM-ROUGE-1	CNN/DM-ROUGE-2	CNN/DM-ROUGE-L
T5	28.0	42.05	20.34	39.40
DistilBERT	37.96	44.16	21.28	40.90

TABLE II. PERFORMANCE BASED ON EVALUATION BENCHMARKS FOR QUESTION ANSWERING

	GLUE	SQuAD-EM	SQuAD-F1
T5	82.7	85.44	92.08
DistilBERT	77.0	77.7	85.8

TABLE III. STUDY OF RESEARCH PAPERS

Paper Title	Key Points	Drawbacks
Sepp Hochreiter, et al. [4]	<ul style="list-style-type: none">• LSTM being local in time and space having a complexity of O(1) per time weight and step.• LSTM performs higher number of successful runs, learns faster• It is efficient in solving complex, artificial tasks with long time lags	<ul style="list-style-type: none">• It is difficult to train (takes very long time).• Transfer learning never really worked, and it must be computed in serial per token.
Colin Raffel, et al. [7]	<ul style="list-style-type: none">• Explore transfer learning to introduce a unified text-to-text framework.• Comparison between various aspects on masses of NLU tasks.• Inference from C4 exploration and scale, advanced results on NLP tasks were recorded.	<ul style="list-style-type: none">• Size of T5 model is 30 times more than the general NLP models• It is expensive to use on commodity GPU hardware.
Jacob Devlin, et al. [8]	<ul style="list-style-type: none">• BERT a pre-trained deep bidirectional representation transformer model• Performs pre-training on unlabeled text by jointly conditioning on right direction and left direction of context in all layers.• Presents a MLM technique, for carrying out bidirectional training of models	<ul style="list-style-type: none">• The fine-tuning and pre-training are inconsistent.• The model file is too large, and the training time is too long.
Mike Lewis, et al. [9]	<ul style="list-style-type: none">• Proposes pre-training objective for sequence-to-sequence models as denoising autoencoder and uses Transformer architecture.• Training done by corrupting textual content along with arbitrary noise function and the Language Model denoises it.	<ul style="list-style-type: none">• Outputs are highly abstractive with few copied phrases.• Model has tendency to hallucinate unsupported information.

TABLE IV. STUDY OF EXISTING SYSTEM

Product	Key Points	Drawbacks
Text Summarization [27]	<ul style="list-style-type: none"> Based on advanced NLP and ML technologies. Summarizes text from the URL or provided document Can be easily used in any environment Capable of making HTTP requests 	<ul style="list-style-type: none"> The system does not allow to upload whole E-Book. It does not have question generation functionality.
Automatic Text Summarizer [28]	<ul style="list-style-type: none"> A multilanguage Text Summarizing and Paraphrasing AI Tool Uses specific algorithm for extracting key points and uses extraction-based summarization Accessible by an API and is still in its development phase 	<ul style="list-style-type: none"> Upload of whole E-Book is not possible No option of question generation
Quillionz [29]	<ul style="list-style-type: none"> AI-powered platform for creating questions, quizzes and notes, allows to edit those questions and notes. Highlights important parts, summarized points, reinforce key concepts using notes features. 	<ul style="list-style-type: none"> It has access to only limited number of E-Books. Impossible to upload whole E-Book.
Lumos Comprehend [30]	<ul style="list-style-type: none"> An automated solution that helps to build quality questions and answers for textual content powered by advanced AI and ML algorithms. User can use this application to convert long articles into meaningful questions and answers. Once the questions and answers are generated, can view them on the screen or export them in CSV file format. 	<ul style="list-style-type: none"> The system does not have text summarization functionality.

TABLE V. PERFORMANCE OF SYSTEM FOR QUESTION GENERATION

No. of E-Books	Total Questions	Ideal Total Questions	Time required (minutes)	Relevant Questions	Irrelevant Questions	Percentage of Correct Questions	Percentage of Incorrect Questions
5	159	200	37.5	126	33	79.25	20.75
10	332	400	77.5	261	71	78.61	21.397
15	469	600	119	372	97	79.32	20.68
20	637	800	160.5	506	131	79.43	20.57

TABLE VI. PERFORMANCE OF SYSTEM FOR DISTRACTOR GENERATION

Num. of E-Books	Total MCQ Options Generated	Correct Options for MCQs	Incorrect Options for MCQs	Percentage of Relevant Options	Percentage of Irrelevant Options
5	111	82	21	73.87	26.13
10	246	187	39	76.02	23.98
15	408	319	65	78.19	21.81
20	537	436	77	81.20	18.81

A collection of 20 sample E-Books were tested on the basis of above-mentioned sampling procedure on the system. The algorithms were trained to generate 8 questions per page. The Fig. 8 compares number of E-Books with time required to generate total questions. Questions generated by the system were considered to be relevant or irrelevant in accordance to grammatical and logical correctness in English language. For evaluation of the questions generated, it was found out that when 20 samples were tested, maximum accuracy of 79.435% is achieved. Otherwise for 5, 10 and 15 samples, accuracy of 79.245%, 78.614% and 79.317% is recorded respectively. The Fig. 9 compares ideal number of questions to be generated with the actual number of questions generated. The Fig. 10 depicts percentage relevancy of questions generated. For evaluation of the MCQs' options generated, it was observed that when 20 samples were tested, maximum accuracy of 81.199% is achieved. Otherwise for 5, 10 and 15 samples, accuracy of 73.873%, 76.016% and 78.186% is recorded, respectively. The Fig. 11 shows percentage relevancy of accurate options generated.

From the above data it can be summarized that with increased number of pages and question the accuracy of the

system went on increasing slightly. Also, the lower bound of the accuracy for relevant question generation was 78.614% and the same for relevant MCQ's option generation was 73.873%. Hence, we can conclude that the performance of the T5 transformer for question generation and question answering task, Distil BART for text summarization task and Sense2Vec for the distractor generation task was optimal and much better than the realm of mere guessing (50%).

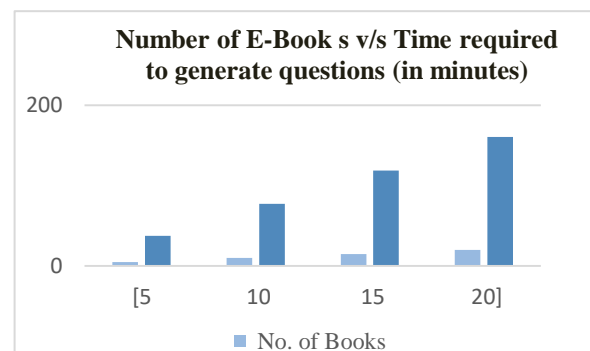


Fig. 8. Comparison of Number of E-Books and Time Required to generate Total Questions.

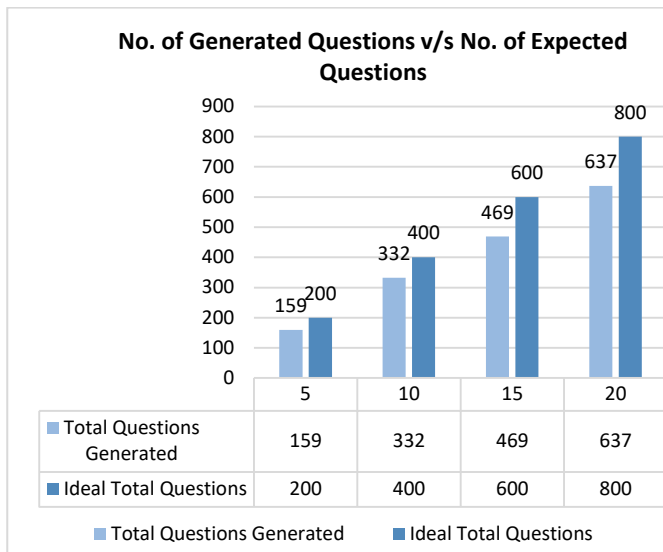


Fig. 9. Comparison of Ideal Number of Questions to be generated to Actual Number of Questions Generated.

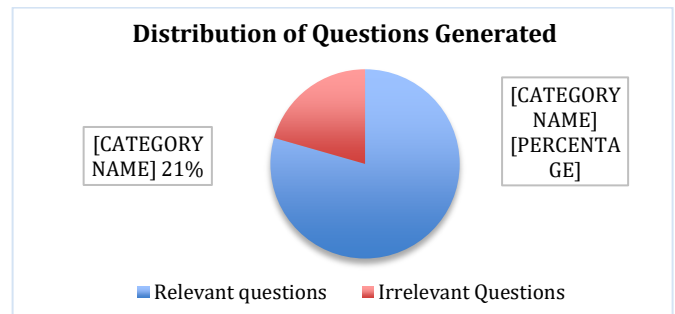


Fig. 10. Distribution of Questions Generated.

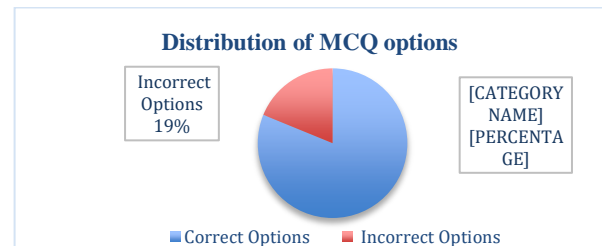


Fig. 11. Distribution of Multiple-Choice Questions.

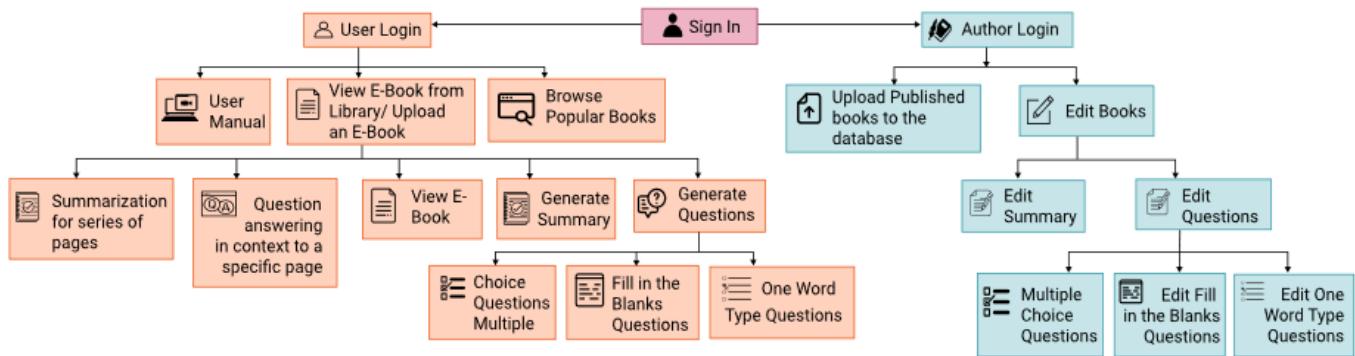


Fig. 12. Proposed System.

IX. CONCLUSION

The boost in the amount of text data generated with time and development of technology has demanded research in automatic text summarization and question generation. Because of lockdown, E-reading and online examinations have turn out to be very popular, which includes many important examinations. No all-in-one system existed which provided both text summarization, question generation and question answering all at once. Hence, using NLP Transformers models like T5, Distil BERT, Distil BART this project creates such system resulting in reduced reading time and providing concise summary along with a questionnaire by implementing the existing algorithms with optimal accuracy. On implementing the fine-tuned transformers, efficient results are found out. Thus, the objectives for creating a system that provides a one stop destination solution to text summarization and question generation tasks were achieved.

For future work, the system can be elevated by scoring the readers on the basis of number of correct questions answered. This system can be used to evaluate the student’s capability and skills efficiently. Also, for upgradation of the system, focus will be on creating challenging questions for better learning

process. The system can be integrated with educational platforms like Moodle. A subscription model can also be created along with basic one. The paid version will have a feature that will enable users to communicate with the authors to solve queries. And authors will be updating the auto generated questions that they feel are semantically incorrect.

REFERENCES

- [1] Edeh Michael Onyema, Chika Nwafor, et, al, “Impact of Coronavirus Pandemic on Education”, Journal of Education and Practice, Vol.11, No.13, 2020, pp 108-121, DOI: 10.7176/JEP/11-13-12.
- [2] N. Moratanch, S. Chitrakala, et al, "A survey on abstractive text summarization", 2016 International Conference on Circuit, Power and Computing Technologies (ICCPCT), March 2016, pp. 1-7, DOI: 10.1109/ICCPCT.2016.7530193.
- [3] Steven Bird, Ewan Klein, et al. O'Reilly Media, "Natural Language Processing with Python", Incorporated, 2009.
- [4] Sepp Hochreiter, Jurgen Schmidhuber, “Long Short-Term Memory”, PubMed, Neural Computation, 1997 Nov 15, pp. 1735-1780, DOI: 10.1162/neco.1997.9.8.1735.
- [5] Ashish Vaswani, Noam Shazeer, et, al, “Attention is All You Need”, 31st conference on Neural Information Processing Systems, NIPS 2017, Long Beach, CA, USA, pp. 6000-6010, arXiv:1706.03762.
- [6] Albert Zeyer, Parnia Bahar, et, al, "A Comparison of Transformer and LSTM Encoder Decoder Models for ASR", 2019 IEEE Automatic

- Speech Recognition and Understanding Workshop (ASRU), Singapore, December 2019, pp. 8-15, DOI: 10.1109/ASRU46091.2019.9004025.
- [7] Colin Raffel, Noam Shazeer, et, al, "Exploring the Limits of Transfer Learning with Unified Text-to-Text Transformer", Journal of Machine Learning Research 21(2020), June 2020, arXiv:1910.10683.
- [8] Jacob Devlin, Ming-Wei Chang, et, al, "BERT: Pre-Training of Deep Bidirectional Transformer for Language Understanding", Google AI Language, May 2019, arXiv:1810.04805.
- [9] Mike Lewis, Yinhan Liu, et, al, "BART: Denoising Sequence-to-Sequence Pre-Training for Natural Language Generation, Translation and Comprehension", Facebook AI, Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, July 2021, arXiv:1910.13461.
- [10] Princeton University, "WordNet-A Lexical Database for English", Accessed 01 August 2021, <https://wordnet.princeton.edu/>.
- [11] Julie Weeds, Daoud Clarke, et. al, "Learning to Distinguish Hypernyms and Co-Hyponyms", Proceeding of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers, pp. 2249-2259, August 2014.
- [12] Yuni Susanti, Takenobu Tokunaga, et.al, "Automatic Distractor generation For Multiple Choice English Vocabulary Question", Research and Practice in Technology Enhanced Learning 13, Article Number: 15, October 2018, DOI: 10.1186/s41039-018-0082-z.
- [13] Hugo Liu, Push Singh, et. Al, "The ConceptNet Project V2.1", Accessed July 2021, <http://alumni.media.mit.edu/~hugo/conceptnest/#papers>.
- [14] Ke Shen, Mayank Kejriwal, "A Data-Driven Study of Common-sense Knowledge Using the ConceptNet Knowledge Base", Jan 2021, arXiv:2011.14084.
- [15] MongoDB, "Welcome to the MongoDB Documentation", Accessed July 2021, <https://docs.mongodb.com/>.
- [16] SpaCy, "Library Architecture", Accessed July 2021, <https://spacy.io/api>.
- [17] Jaime Carbonell, Jade Goldstein, "The Use of MMR, Diversity-Based Reranking for Reordering Documents and Producing Summaries", SIGIR '98: Proceedings of the 21st Annual International ACM SIGIR conference on Research and Development in Information retrieval, pp. 335-336, Issue August 1998, DOI: 10.1145/290941.291025.
- [18] Mengyu Li, Boyao Sun, et, al, "Question Answering on SQuAD2.0", in Stanford CS224N Natural Language Processing with Deep Learning.
- [19] Stuart rose, Dave Engel, et, al, "Automatic Keyword Extraction from Individual Documents", in Text Mining: Application and Theory, pp. 1-20, Issue March 2010, DOI: 10.1002/9780470689646.ch1.
- [20] Google Firebase. "Google Firebase", Accessed July 2021, https://firebase.google.com/?gclid=Cj0kCQjw24qHBhCnARIsAPbdtl3nmCf5-9fIE3SiyvTqT3cjxcQ7piDulL_j24QDLarNu5YgUeJLkaAiJREALw_wcB&gclsrc=aw.ds.
- [21] Bojar et al. "WMT 2016", Findings of the 2016 Conference on Machine Translation. Accessed September 2021, <https://paperswithcode.com/dataset/wmt-2016>.
- [22] Bojar, Rajen Chatterjee, et, al, "Findings of the 2016 Conference on Machine Translation (WMT-2016)", Proceedings of the First Conference on Machine Translation, Volume 2: Shared Task Papers, Berlin, Germany, pp. 131-198, Issue August 2016, DOI: 10.18653/v1/W16-2301.
- [23] Ramesh Nallapati, Bowen Zhou et, al, "Abstractive text Summarization Using Sequence-to-Sequence RNNs and Beyond", CNN/Daily Mail. Accessed September 2021, <https://paperswithcode.com/dataset/cnn-daily-mail-1>.
- [24] Chin-Yew Lin. "Looking for a few Good Metrics: ROUGE and its Evaluation", Working Notes of NTCIR-4, Tokyo, Issue June 2004, Corpus ID: 55156862.
- [25] Alex Wang, Amanpreet Singh, et. Al, "GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding", Issue April 2018, arXiv:1804.07461.
- [26] Perer Flach, Meelis Kull, "Precision-Recall-Gain Curves: PR Analysis Done Right" In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, & R. Garnett (Eds.), Advances in Neural Information Processing Systems 28, Vol. 1, pp. 838-846, DOI: 10.1.1.1038.8511.
- [27] Text Summarization | Text Summarizer, "Text Summarization", Accessed July 2021, <http://textsummarization.net/>.
- [28] AutoSummarizer.com, "Automatic Text Summarizer", Accessed July 2021, <https://autosummarizer.com/>.
- [29] Harbinger AI Inc, "Quillionz", Accessed July 2021, <https://www.quillionz.com>.
- [30] Lumos Learning, "Lumos Comprehend", Accessed July 2021, <https://www.lumoslearning.com/llwp/free-question-answer-generator-online.html>.