Automated Context-based Question-Distractor Generation using Extractive Summarization

Akshay Khanolkar[1], Manasi Kulkarni[2]

[1]*Department of Computer Engineering and Information Technology, Veermata Jijabai Technological Institute, Mumbai, India*
[2] *Department of Computer Engineering and Information Technology, Veermata Jijabai Technological Institute, Mumbai, India*

---***---

Abstract - *Multiple Choice Question (MCQ) is a versatile question type where respondents select one correct answer from the other choices. Generation of such MCQs along with different implausible options is tedious and time-consuming work. Existing automated question generation systems reduce manual efforts to some extent as they were at only sentence level information. By looking widespread usage of MCQs from school exams to competitive exams, automated generation of more reliable questions is required. The proposed system generated questions based on an Extractive summary using the BERT model which holds paragraph-level context. BLEU score is used as an evaluation matrix to demonstrate the quality of model-generated questions across the human generated questions. The achieved BLEU score is 0.66. Though the proposed system focuses on Fill-in-the-blank questions scope can be extended to various different types of objective questions.*

Key Words: **Bidirectional Encoder Representation from Transformers, Wordnet, Bilingual Evaluation Understudy, Constituent Likelihood Automatic Word-tagging System, Python Keyword Extraction, Recurrent Neural Network, Recall-Oriented Understudy for Gisting Evaluation**

1. INTRODUCTION

In today's world of online education, Multiple Choice Question (MCQ) tests play an important role. These questions help to assess students' basic to complex level knowledge. Generation of such questions on specific topics is a time-consuming and hectic task. It received tremendous interest in recent years from both industrial and academic communities. These MCQ evaluation tests are focused on research perusing, have functional significance for understudy situations, and furthermore empower instructors to follow enhancements all through the scholarly year. Question generation task which takes a specific situation and replies as info and produces an inquiry that objectives the offered response. MCQ generator System helps to generate these questions. The existing question generation models mainly rely on recurrent neural networks (RNNs) augmented by attention mechanisms. The inherent sequential nature of RNN models suffers from the problem of handling long sequences. Existing systems mainly use only sentence-level information as context. When applied to a paragraph-level context, the existing models show significant performance degradation. The latest development is BERT, which has shown significant performance improvement over various NLP tasks. The power of BERT is able to simplify neural architecture design for natural language processing tasks.

2. RELATED WORK

For generating questions many systems are implemented. The vocabulary evaluation question generator system [1] creates questions in six different categories, including definition, synonym, antonym, hypernym, hyponym, and cloze questions. Also makes different decisions and word bank question designs. English nouns, verbs, adjectives, and adverbs are categorized into sets in Wordnet, a lexical knowledge base (synsets). Synsets from wordnet are extracted in order to build the questions. The criterion for the selection of distractors and the phrasing or presentation of the questions are the two main problems encountered while producing multiple choice questions. The algorithm selects distractions that have the same part of speech (POS) and same frequency as the correct response. The Kilgarriff word recurrence data set, which depends on the English Public Corpus, is utilized by this technique to choose distractor words (BNC) [2]. The system randomly selects the distractors from a set of 20 words from this database that have the same POS and the same or similar frequency as the right response. The CLAWS tagger applies POS labels to the words in the BNC and Word Recurrence data sets. For 75 low-frequency English words, the validity of the machine-generated vocabulary questions was evaluated in comparison to questions created by humans. This study compares students' performance on questions created by computers and by humans in terms of accuracy and reaction speed. As per trial discoveries, these consequently created questions give a proportion of jargon expertise that is profoundly related with subject execution on questions that were freely evolved by people. As per the resulting study, question creation errands are dealt with by the pre-prepared BERT language model [3]. The paper presents neuronal architecture created with BERT. The first one is a straightforward BERT employment which reveals the defects of directly using BERT for text generation and another is a remedy for the first one by restructuring the model into a sequential manner for taking input from previous results. Utilizing the SQuAD question answering dataset, models are trained and evaluated [4]. The SQuAD dataset contains 536 Wikipedia articles and around 100K reading comprehension questions. Two data split settings used. The first one is SQuAD73K were training (80%), development set (10%), test set (10%) and the second is SQuAD 81K where development data set is divided into development set (50%) and test set (50%). For evaluation results are compared with RNN models i.e. NQG [5] and PLQG [6] models based on the sentence level and paragraph level context using standard metrics BLEU and ROUGE-L. Results show that the model improves the BLEU 4 score from 16.85 to 21.04 in comparison with existing systems.

3. PROPOSED SYSTEM

The goal of this research is to implement a system that automatically generates Multiple choice questions, targeting the below objectives:

- Use of BERT model for question generation.
- Use of Wordnet and Conceptnet as knowledge bases for distractor generation tasks.
- Generation of MCQ by considering the paragraph level context.

BLEU score is used as an evaluation metric for evaluating a proposed model.

3.1. General System Architecture

The MCQ generator system takes input as a passage and generates questions based on that. Text passage is given as the main input to the system. From which stem is selected (Stem means sentence from which question is to be formed). A question and its answer are are framed utilizing this stem. For distractors, wordnet and concept knowledge bases are used. Words present in the synsets of the correct answer are extracted and used as distractors. The general system architecture is given as shown in Fig.1.

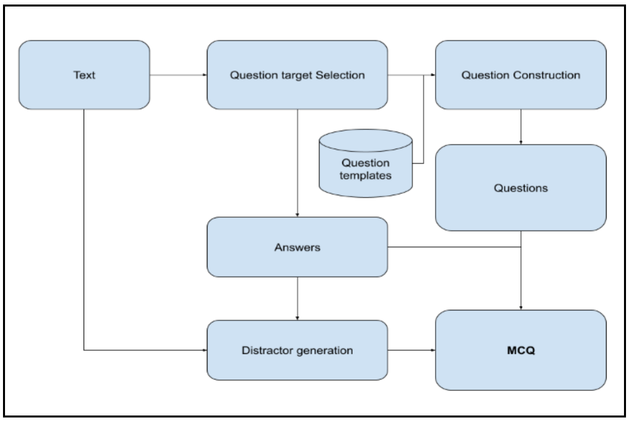


Fig-1: General System Architecture

3.2. Proposed System Architecture

The proposed system generates the Fill-in-the-blank questions which are knowns as cloze questions. A paragraph is provided as input to our system, from which we must produce questions. The proposed system architecture is given as shown in Fig.2.

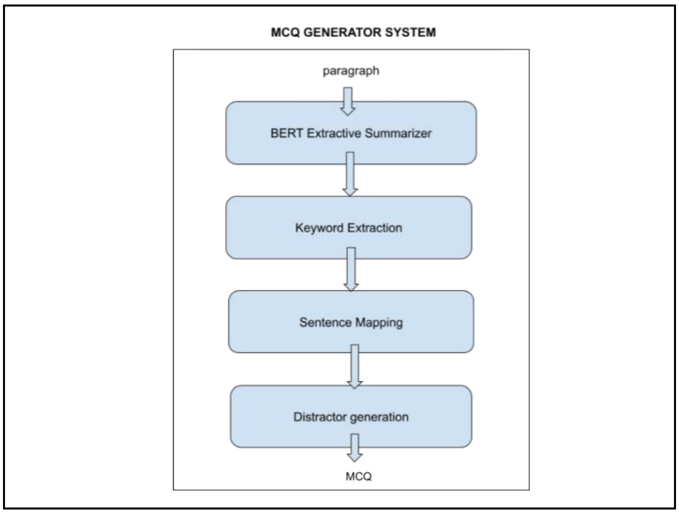


Fig-2: Proposed System Architecture

1. BERT Extractive Summarizer [7]

   To achieve our objective to generate questions that hold paragraph-level context we need to first generate a summary for a given passage. So for this purpose, an extractive summarizer is used. Abstractive and Extractive are two summarization strategies. The abstractive summarization technique closely emulates human summarization. Extractive summarization aims at identifying the main information that is then extracted and grouped together to form a proper summary. The extractive technique selects the top N sentences which represent the main points of the article. The Outline acquired contains accurate sentences from the first message. Different modules in BERT Extractive Summarizer are as shown in Fig.3.

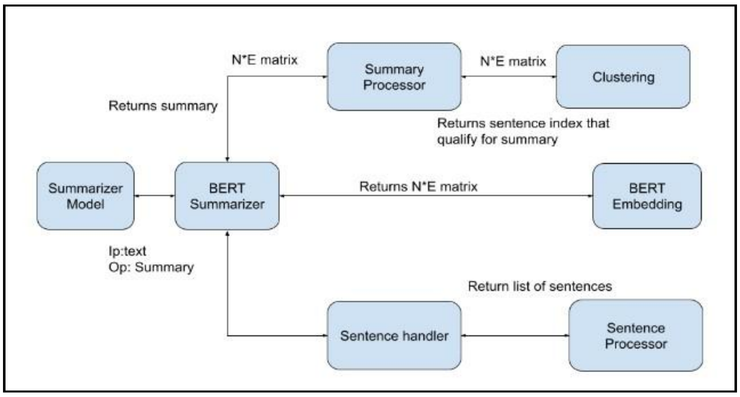


Fig-3: BERT Extractive Summarizer

- Extractive text summarization using BERT and kmeans:

  For creating summaries from input passages an extractive summarizer is used. Input entries are tokenized into sentences. After tokenization, these sentences are fed to the BERT model for extracting embedding then clustered the embedding with Kmeans. Sentences that are close to centroid are selected for a summary.

- BERT for text embedding:

  Due to superior performance over other models in NLP, the BERT model (BERT-LARGE) is used. Using the default pre-trained BERT model one can select multiple layers for embeddings. BERT model produces N*E matrix where N is No. of sentences in input passage and E is embedding dimension (For BERT-BASE=768 and BERTLARGE=1024)

- Clustering embeddings:

  One generation of the embedding matrix is completed then it is forwarded to the clustering module. During experimentation, both k-means and Gaussian Mixture Models are used. But due to very similar performance, Kmeans was finally selected for clustering. For the clusters, the sentences which are closest to the centroid were selected for a final summary generation.

2. Keyword Extraction

In this stage, we extract all the important keywords from the original text. Then we check whether those keywords are present in the summary. Then, at that point, keep just those catchphrases that are available in the summed up text. For extracting keywords we are using PKE i.e. Python Keyword Extraction Toolkit. PKE is an open-source python-based keyphrase extraction toolkit. In our system Multipartite rank, an unsupervised graph model [8] is used. Inside the setting of a multipartite chart structure, it encodes effective information. In order to increase candidate ranking, the model combines keyphrase candidates and subjects into a single graph and makes use of their mutually reinforcing interaction. The keyphrase candidates are represented by nodes in a fully directed multipartite graph, and only those nodes that belong to separate subjects are connected. Weight edges as per the distance between two competitors in the record. More formally, the weight w_{ij} from node i to node j is computed as the sum of the inverse distances between the occurrences of candidates c_i and c_j:

$$w_{i,j} = \sum_{p_i \epsilon P(c_i)} \sum_{p_j \epsilon P(c_j)} \frac{1}{|p_i - p_j|}$$

where $P(c_i)$ is the set of the word offset positions of candidate c_i

This weighting scheme gives comparable results to window-based co-occurrence counts. The resulting graph is a complete k-partite graph in which nodes are partitioned into k different independent sets where k represents the number of topics. After the chart is built, a diagram based positioning calculation is utilized to rank keyphrase competitors, and the top N are then picked as key expressions.

3. Sentence Mapping

In this stage for each of the keywords, the system will extract corresponding sentences that have the word from the summarized text. This text is called Stem. From this stem, a question is generated.

4. Distractor generation

The system will extract distractors (options for MCQ) from Wordnet and Conceptnet to generate final MCQ Questions. Wordnet is a lexical database of semantic relations between words in more than 200 languages [9]. It is freely available.Wordnet joins words into semantic relations including equivalents, hyponyms, and meronyms. Synonyms are grouped into Synonym Sets (Synsets) with short definitions and usage examples as shown in Fig.4. It is used for Word Sense Disambiguation, Information retrieval, Automatic text classification, and Text summarization. Synsets are interlinked through conceptual relations. For given answer from corresponding synsets synonyms are extracted which is used as options(i.e. Distractors) for MCQ. In the extraction of synsets wordsense is needed. To find wordsense Wu Palmer similarity and adapted lesk is used. Wu Similiarity basically calculates relatedness between two synsets considering depth of LCS. And adapted lesk is used to find similarity between context in sentence and its dictionary meaning. At the end lowest index is calculated between Wup similarity index and adapted lesk output. Lowest index is more closer to its exact meaning. These synonyms are related to the original word so we will get all relevant distractors at the end which hold similar semantic meanings.

4. EXPERIMENTAL SETUP

4.1. Data set

Our proposed system focuses on generating Fill-in-the-blank type of questions. For evaluating the questions generated by our proposed system some set of standard questions is needed. But there is no such standard question dataset is available. So for this purpose, we have to make our own dataset by collecting questions through the survey. We have conducted a survey based on 5 passages from SQuAD (Standford Question answering dataset) dataset. This survey was conducted using Google Forms. This google form was circulated among undergraduate and postgraduate students.

A total of 50 respondents responded to this form. Statistics of this survey are mentioned in Table 1.

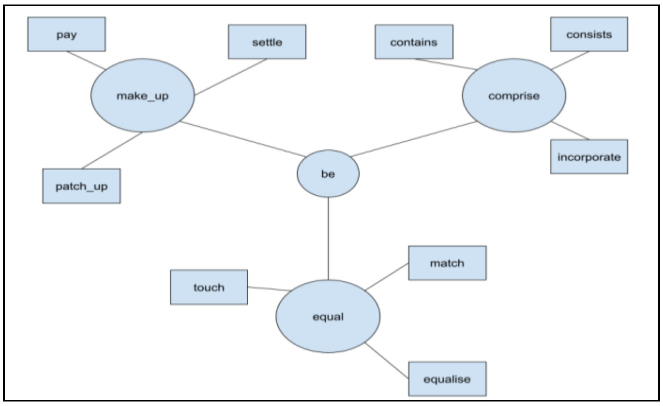


Fig-4: Wordnet Synset Structure

Table-1: Statistics of conducted survey which is used for making manual dataset

SQuAD Psgs	**(Sents, words)**	**#ques formed by 50 respon -dents**	**#unique and #repeated ques**	**#types of ques generated**
Chicago	(4,115)	156	113,43	6
Kenya	(5,112)	153	125,28	8
Oxygen	(7,136)	162	129,33	9
Immune	(4,101)	152	124,28	4
Construction	(4,74)	152	105,47	5
Total ques		**775**		

4.2. Experiments

1. ***Set of experiments on summary***

To achieve the objective of generating the questions which hold paragraph context summary is generated on the input passage. BERT extractive summarizer is used for generating the summary. Later this summary is compared with the summary generated by another online summarizer to see the efficiency of our summarizer. Results of this comparison are given in the results section.

2. ***Set of experiments on questions***

After generating the questions in summary we compared these questions with our dataset which was gathered using a survey. For Quantitative analysis, we found out how many questions were generated by our model compared to humans. Also individual analysis for coverage of questions on input passage and summary to see whether summary limits the no. of questions. Different size inputs such as single passage, multi passage, multi-page, and book are given to the system to find coverage. For Qualitative analysis, we compare the system generated questions with human-generated questions gathered through a survey using the BLEU score to see the quality of the questions. Also, we conduct one more survey in which respondents rate each question generated by the system on a scale of 1-5 to see the quality of the question from the human perspective. This survey was taken on two sets of questions one set is questions generated by the system on technical passage and another question on literature passage. These two sets are used to see if the system is useful for which type of passage and how many questions are formed. For distractors analysis, we found out coverage on no. of unique distractors generated using Wordnet and Conceptnet for passages which we are using for our analysis.

4.3. Evaluation Metric

1. ***BLEU Score***

BLEU score stands for Bilingual evaluation understudy. It is a calculation for assessing the nature of text which has been machine-interpreted starting with one normal language then onto the next. Quality is viewed as the correspondence between a machine's result and that of a human. BLEU's result is dependably a number somewhere in the range of 0 and 1. This worth shows how comparative the applicant message is to the reference messages, with values more like 1 addressing more comparative messages. Those scores are then arrived at the midpoint of over the entire corpus to arrive at a gauge of the interpretation's general quality. BLEU score is mathematically defined as:

$$BLEU = min(1, exp(1 - \frac{ref - length}{op - length}))(\prod_{i=1}^{4} precision_i)^{1/4}$$

With

$$precision_i = \frac{\sum_{snt \epsilon Cand-Corpus} \sum_{i \epsilon snt} min(m^i_{cand}, m^i_{ref})}{w^i_t = \sum_{snt' \epsilon Cand-Corpus} \sum_{i' \epsilon snt'} m^{i'}_{cand}}$$

where

$m^i{}_{cand}$ is the count i-gram in candidate matching the reference translation

$m^i{}_{ref}$ is the count of i-gram in the reference translation

$w^i{}_t$ is the total number of i-grams in candidate translation

5. RESULTS AND ANALYSIS

5.1. Results

1. ***Evaluating efficiency of Extractive summarizer*** Summary generated on input text using extractive summarizer is base for MCQ generation. So this summary is compared with other online summarizers i.e. Sassbook extractive summarizer. Sassbook summarizer is a very popular tool for researchers for generating summaries. ROUGE score is used as an evaluation metric for showing the efficiency of our summarizer. 5 passages are taken from the standard dataset SQuAD and their summary is generated using our extractive summarizer. These generated summaries are compared with online summarizer-generated summaries. ROUGE score is between 0 to 1. A score closer to 1 means generated summary is more identical to the online summarizer-generated summary. Table 2 shows the comparison between the ROUGE score for the summary generated for one of the passages from the SQuAD dataset by our extractive summarizer vs another online summarizer.

Table-2: Similarness between our extractive summarizer and Sassbook online summarizer

Score	Sassbook Extractive
ROUGE-1	0.807
ROUGE-2	0.774
ROUGE-3	0.789
ROUGE-4	0.789
ROUGE-L	0.804

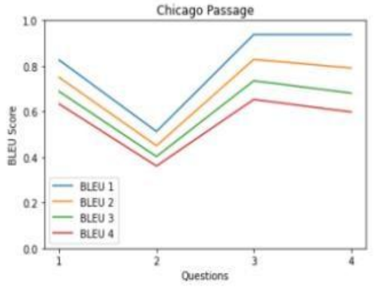
2. ***Evaluating efficiency of Question generation system***

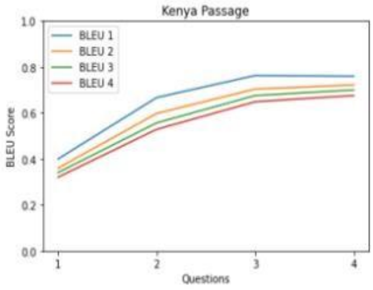
- Qualitative analysis using BLEU score As mentioned in the dataset section, we are using our own dataset for checking the quality of the system-generated MCQ. The own dataset contains questions that are collected from respondents through the survey. Table 3 shows the similarity between the model-generated questions and human-generated questions for 5 passages from the SQuAD dataset. Table 3 values presented in graphical format in Chart 1-5.

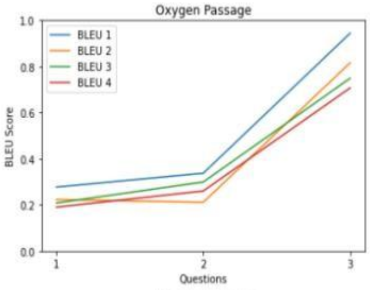
- Qualitative analysis using rating survey In addition to the above experiment, the survey is conducted to rate the questions generated by our system. Respondents are undergraduate and postgraduate students. A total of 25 responses are gathered. Table 4 shows the results of this experiment. This can be presented as a graph as shown in Chart 6.

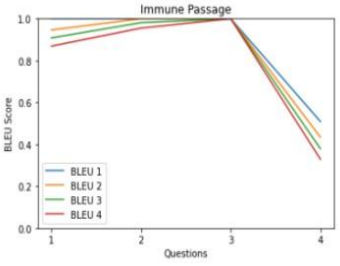
Table-3: BLEU score shows the similarity between model generated questions with human-generated questions

BLEU Score	Questions	BLEU1	BLEU2	BLEU3	BLEU4
Chicago	Q1	0.83	0.75	0.69	0.63
	Q2	0.51	0.45	0.40	0.40
	Q3	0.94	0.83	0.74	0.65
	Q4	0.94	0.80	0.68	0.60
Kenya	Q1	0.4	0.36	0.34	0.32
	Q2	0.67	0.60	0.56	0.53
	Q3	0.76	0.70	0.67	0.65
	Q4	0.76	0.72	0.70	0.67
Oxygen	Q1	0.30	0.22	0.21	0.19
	Q2	0.34	0.21	0.30	0.26
	Q3	0.95	0.82	0.75	0.71
Immune	Q1	1.00	0.95	0.91	0.87
	Q2	1.00	1.00	0.98	0.95
	Q3	1.00	1.00	1.00	1.00
	Q4	0.51	0.43	0.38	0.33
Construction	Q1	0.49	0.45	0.42	0.40
	Q2	1.00	0.93	0.88	0.84
	Q3	1.00	0.93	0.88	0.84
	Q4	1.00	0.95	0.92	0.89
	Q5	1.00	1.00	1.00	1.00
	Q6	1.00	1.00	0.96	0.90
	Q7	0.97	0.90	0.84	0.78









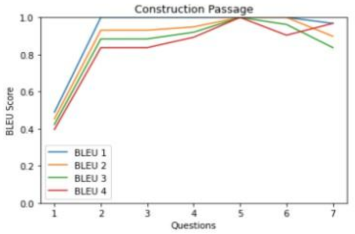


Chart 1-5: Similarity between model-generated questions and human-generated questions

Table-4: Average rating given to the model generated questions by survey respondents

Ques on Technical psg	Average rating given by the 25 respondents (out of 5)	Ques on Literature psg	Average rating given by the 25 respondents (out of 5)
Q1	3.10	Q1	3.00
Q2	3.01	Q2	3.15
Q3	2.95	Q3	2.95
Q4	2.95	Q4	3.20
Q5	2.70	Q5	3.00
Q6	3.30	Q6	3.30
Q7	3.05	Q7	2.75
Q8	2.95	Q8	2.95
Q9	2.95		
Q10	3.15		

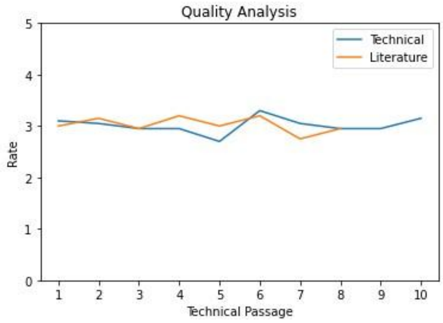


Chart-6: Average rating given by respondents on model generated questions

- Quantitative analysis for coverage of human vs model generated questions

As mentioned in the experimental setup, the experiment is conducted to see the types of questions generated by humans in comparison with our system. Table 5 shows the results of this experiment.

- Quantitative analysis for coverage of the different sizes of input

As mentioned in the experiments section different sizes of input are given to the system to see how no. of questions depends on input size. Also to see more questions formed on direct input passage than summary input. Table 6 shows significant differences between no. of questions formed on summary and direct input.

Table-5: Coverage on types of question human-generated vs model-generated

Passage	#types of ques generated by human	#types of ques generated by model
Chicago	6	4
Kenya	8	4
Oxygen	9	3
Immune	4	4
Construction	5	7

Table-6: Coverage on different input size

Types of input	#ques formed on input	#ques formed on summary
Single passage	20	8
Multi passage	20	9
Multi-page	20	9
Book	20	12

- Quality of distractors

Table 7 shows a number of unique distractors generated using Wordnet and Conceptnet for the questions generated on 5 SQuAD passages.

Table-7: Coverage on distractors

Passage	#unique distractors generated
Chicago	47
Kenya	40
Oxygen	46
Immune	17
Construction	53

5.2. Analysis

1. *Evaluating efficiency of Extractive summarizer*

According to Table 2, it is observed that the ROUGE score for sassbook extractive is closer to 1 which means sassbook also uses a similar extractive summarization technique i.e. clustering. It shows that our summarizer generates a good summary for the input passage which is used as a base for question generation. Also, an extractive summarizer is used for a wide range of passages. It does not depend on a specific domain like abstractive, because it does not require any type of training on some specific domain dataset.

2. *Evaluating efficiency of Question generation system*

- Qualitative analysis using BLEU score
  From graphs in Table 3, it is observed that most of the questions have a BLEU score greater than 0.6. The average BLEU score over the dataset is 0.66. It means our system generates human-like MCQs by achieving gold standards.

- Qualitative analysis using rating survey
  The survey is taken among graduate students. The average rating for system-generated questions is Also, this system works fine for both technical and literature-based input passages. Here technical passage is from an online technical blog and the literature passage is from the novel. So we claimed that our system can generate questions on any type of input from any domain hence it is domain-independent.

- Quantitative analysis for coverage of human vs model generated questions
  From Table 5 it is observed that for some passages human generates more questions. This is because respondents have generated questions on the overall passage where the system generates the questions on a summary to achieve the objective of holding paragraph context. This result will get more clear in the next analysis.

- Quantitative analysis for coverage of the different sizes of input
  From the values from Table 6, it is observed that the system always generates more questions on input passage compared to summary. In this table, the highest number of questions is always 20 because in the implementation of this model we are considering the top 20 keywords from a passage in the keyword extraction stage. There can be a large number of questions. But in summary, will get less number of questions than input. Because there is a loss of information while building the summary. This loss of information can be compromised by holding paragraph-level context.

- Quality of distractors
  From Table 7 it is observed that the system generates a good amount of distractors for comparatively small passages. As the input size grows summary will be more so the number of unique distractors will be formed in large numbers. So our system simplifies the tedious

task of generating options for MCQs. Also, these options are similar in meaning to the original answers which can be used to confuse the one who gave the exam.

6. CONCLUSION

MCQ generator system considers paragraph level context for question generation task which improves performance over existing systems which only consider sentence level context. The implemented system is domain independent which generates questions for any input irrespective of domain, but it limits the number of questions because of the loss of information while generating a summary. A BERT extractive summarizer is used to generate a summary for input passage which is compared with an online summarizer which gives a ROUGE score of around 0.80. Due to the unavailability of standard datasets, we have built our own dataset by collecting questions through a survey. These questions are compared with model-generated questions using the BLEU score to see if our system is reaching gold standards and building human-like questions. The system achieved a BLEU score of 0.66. In the future, we'll be able to create various kinds of question types.

REFERENCES

[1] J. C. Brown, G. A. Frishkoff, and M. Eskenazi, "Automatic question generation for vocabulary assessment," in Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing - HLT '05, (Vancouver, British Columbia,Canada), pp. 819–826, Association for Computational Linguistics, 2005

[2] "Read-me for Kilgarriff's BNC word frequency lists."

[3] Y.-H. Chan and Y.-C. Fan, "BERT for Question Generation," in Proceedings of the 12th International Conference on Natural Language Generation, (Tokyo, Japan), pp. 173–177, Association for Computational Linguistics, 2019.

[4] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, "SQuAD: 100,000+ Questions for Machine Comprehen- sion of Text," in Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, (Austin, Texas), pp. 2383–2392, Association for Compu- tational Linguistics, 2016.

[5] X. Du, J. Shao, and C. Cardie, "Learning to Ask: Neu- ral Question Generation for Reading Comprehension," in Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), (Vancouver, Canada), pp. 1342–1352, Association for Computational Linguistics, 2017.

[6] Y. Zhao, X. Ni, Y. Ding, and Q. Ke, "Paragraph-level Neural Question Generation with Maxout Pointer and Gated Self-attention Networks," in Proceedings of the 2018 Conference on Empirical Methods in Natural Lan- guage Processing, (Brussels, Belgium), pp. 3901–3910, Association for Computational Linguistics, 2018.

[7] D. Miller, "Leveraging BERT for Extractive Text Summa- rization on Lectures," June 2019. arXiv:1906.04165
[cs,eess, stat]

[8] Boudin, "Unsupervised Keyphrase Extraction with Mul- tipartite Graphs," in Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers), (New Orleans, Louisiana), pp. 667–672, Association for Computational Linguistics, 2018.

[9] G. A. Miller, "WordNet: a lexical database for English," Communications of the ACM, vol. 38, pp. 39–41, Nov.1995.