Automated MCQ Generator using Natural Language Processing

Pritam Kumar Mehta[1], Prachi Jain[2], Chetan Makwana[3], Dr. C M Raut[4]

[1-3]*Student, Dept. of Computer Engineering, Datta Meghe College of Engineering, Navi Mumbai, India*
[4]*Prof., Dept. of Computer Engineering, Datta Meghe College of Engineering, Navi Mumbai, India*

Abstract - *Examinations and Assessments are undergoing a tremendous revolution. Universities, colleges, and other educational institutes are increasingly shifting towards online examinations. The pattern of assessment is majorly shifting towards the objective assessment i.e. MCQ based, it is very hard to construct and requires a considerable amount of time for setting numerous questions. There's a growing need for a cost-effective and time-efficient automated MCQ generation system. In this paper, the text is first summarized using the BERT algorithm, and accordingly sentence mapping is done for generating MCQs. In order to generate choices for the questions, distractors are generated using wordnet (A lexical database for English). As the BERT algorithm has much better performance over other legacy methods as well as it can process a large amount of data in less time, it will enhance the speed of generating MCQs from given text.*

Key Words: MCQs, BERT, Wordnet, PKE, Deep learning

1. INTRODUCTION

All institutes, colleges, and schools have been switched to online learning. Assessment is an essential tool to test the knowledge of the students. And the pattern of the assessment has changed from subjective based to objective based i.e. Multiple Choice Questions (MCQs).

So the problem is, it is very difficult for the teachers to set the questions as well as for the students who are preparing for competitive exams. The current method involves the setting of questions manually which requires a lot of human intervention and time. So there is a growing need for a system that can create questions with ease and less amount of time and requires less human effort.

This paper tells about a system that generates questions automatically. In Automated MCQ Generator, questions are generated automatically with the help of NLP. The text of any domain is provided as input to the system which is then summarized using the BERT algorithm. BERT (Bidirectional Encoder Representation from Transformers) is a deep learning-based technique for natural language processing, a pre-trained model from Google. Now the keywords are selected from the summarized text using the python keyword extractor (PKE) and accordingly mapping of a keyword is done with a sentence. This keyword will be one of the options of MCQ. Now the main task is generating relevant distractors. Distractors are generated using the wordnet approach. Wordnet is an API used to get the correct sense of the word. So the good and relatable distractors are generated.

This system solves the problem of manual creation of questions and reduces time consumption and cost.

2. Literature Review

We have studied several research papers on multiple-choice question generation using different approaches. Santhanavijayan et al. have proposed a system of "Automatic generation of multiple-choice questions for e-assessment" [1]. In their proposed system, they have used fireflies-based preference learning and ontology-based approach to generate MCQs. They have used a web corpus to make it feasible to create questions. The distractors are generated using similarity metrics such as hypernyms and hyponyms. The system also creates analogy questions to test the verbal ability of the students.

Ayako Hoshino and Hiroshi Nakagawa "A real-time multiple-choice question generation for language testing: A preliminary study" [2] is based on machine learning to generate questions automatically. They implement machine learning algorithms, such as Naive Bayes and KNearest Neighbors, to create questions on English grammar and vocabulary from online news articles. They designed a system that can receive user input in the form of an HTML file and turns it into a quiz session.

Deepshree S. Vibhandik et al. have proposed a system, "Automatic / Smart Question Generation System for Academic Purpose" [4] in which the Automatic Question Generation system generates specific trigger questions and multiple-choice questions from student's literature review papers. To facilitate the generation of specific trigger questions, the system extracts key concepts from student's papers using the Lingo algorithm. Also, to bring out the generation of multiple-choice questions, the system pulling out abbreviations from student's review papers using the regular expression pattern matching techniques.

D. R. CH and S. K. Saha have proposed "Automatic Multiple Choice Question Generation From Text: A Survey," [3]. In this paper, articles from the database are used to generate questions. NLP-based summarizer is used for text summarization and frequency count of words, and pattern matching techniques are used for key selection. For generating distractors, they have used the wordnet, pattern matching, domain ontology and semantic analysis.

3. Proposed System

In this section, the detailed step-by-step process of creating MCQs is discussed. In the proposed system, questions are generated from summarized text which is given as input. After summarization of text, there are different steps involved to create questions and distractors. The different stages involved for generating MCQs are shown in Fig- 1.

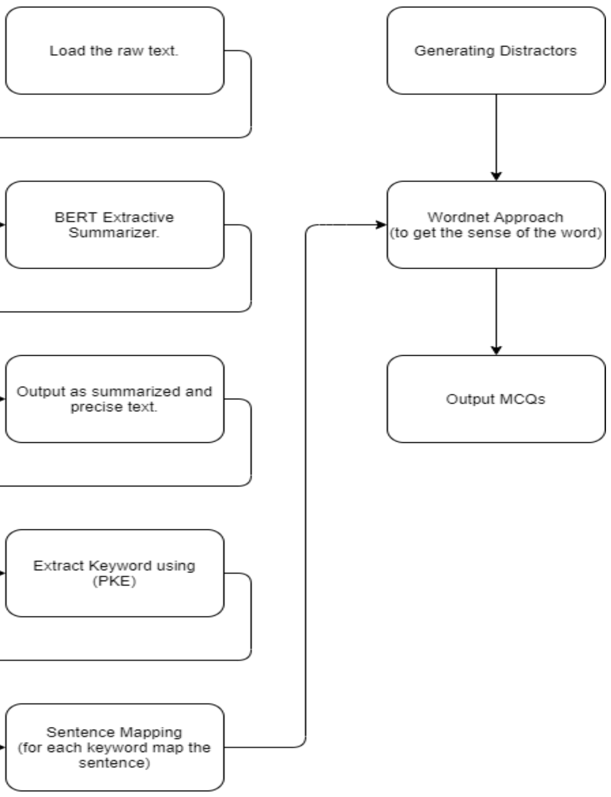

Load the raw text.
BERT Extractive Summarizer.
Output as summarized and precise text.
Extract Keyword using (PKE)
Sentence Mapping (for each keyword map the sentence)
Generating Distractors
Wordnet Approach (to get the sense of the word)
Output MCQs

Fig -1: Proposed System Architecture

3.1 Load raw text

The first step is to load raw text i.e. input text of any domain for which the questions to be generated.

3.2 Summarizer

Each sentence is not capable of generating questions. Only the sentences that contain a questionable fact can act as a candidate for creating MCQs. Therefore, sentence selection plays a crucial role in the automatic MCQ generation task. Hence for summarizing the text, BERT Algorithm is used.

BERT (Bidirectional Encoder Representations from Transformers) is a neural network-based technique for natural language processing. It is a pre-trained open-sourced model from Google. It helps computers to understand the language a bit more as humans do. The input text is summarized using the BERTSUM model, which is fine-tuned BERT for extractive summarization. The architecture of BERTSUM is shown in Fig- 2.

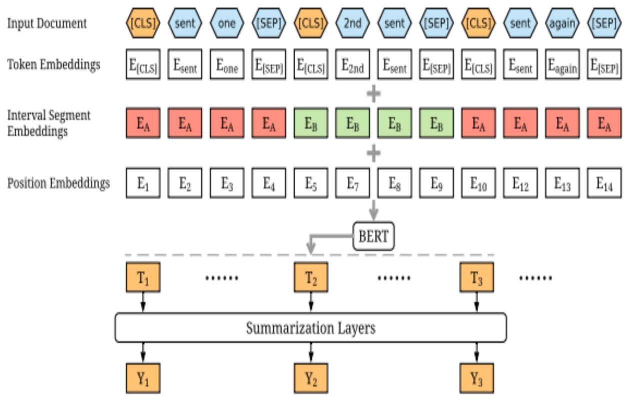


Fig.-2: Overview Architecture of BERTSUM model

In the BERTSUM model, at the start of each sentence, a [CLS] token is added, and between every two sentences, a [SEP] token is added to separate the sentences. Here, a [CLS] token is added to collect the preceding sentence context. Now each word of the sentence is tokenized using token embeddings. There is also a difference in segment embeddings. In the BERTSUM model, each sentence is assigned an embedding of Ea or Eb depending on whether the sentence is even or odd. If the sequence is [s1, s2, s3] then the segment embeddings are [Ea, Eb, Ea]. This way, all sentences are embedded and sent into further layers. BERTSUM assigns scores to each sentence that represents how much value that sentence adds to the overall document. So, [s1, s2, s3] is assigned [score1, score2, score3]. The sentences with the highest scores are then collected and rearranged to summarise the input text. In the output, the precise and summarized text is generated.

3.3 Keyword Extraction

After summarizing the text, keywords are selected from the sentence. This keyword will be the answer to the question. Since all the words in an informative sentence cannot serve as the key proper keyword selection is required. The extraction of keyword is done by python library RAKE.

Rapid Automatic Keyword Extraction (RAKE) is a well-known keyword extraction method that uses a list of stopwords and phrase delimiters to detect the most relevant words or phrases in a piece of text. This algorithm has mainly three components:

3.3.1 Candidate Selection

In candidate selection, all possible words, phrases and terms from the summarized text get extracted that can potentially be a keyword. Consider the following example where the text is,

"Keyword extraction is not that difficult after all. There are many libraries that can help you with keyword extraction. Rapid automatic keyword extraction is one of those."

The first thing this method does is split the text into a list of words and remove stopwords from that list. After splitting, two lists are generated as follows:

stopwords = [is, not, that, there, are, can, you, with, of, those, after, all, one]

delimiters = [".", ","]

The above list of words cannot be part of the candidate key. Now the remaining words will be considered as candidate words.

candidate_words = [keyword, extraction, difficult, many, libraries, help, rapid, automatic]

3.3.2 Word Co-occurrence Matrix

After getting the candidate words, the word co-occurrence matrix is built by this method. In this matrix, each row shows the number of times that a given candidate word co-occurs with every other candidate word in the candidate phrases. For the above-mentioned text example, the matrix looks like this:

	keyword	extraction	difficult	many	libraries	help	rapid	automatic
keyword	3	3	0	0	0	0	1	1
extraction	3	3	0	0	0	0	1	1
difficult	0	0	1	0	0	0	0	0
many	0	0	0	1	1	0	0	0
libraries	0	0	0	1	1	0	0	0
help	0	0	0	0	0	1	0	0
rapid	1	1 RAKE Matrix	0	0	0	0	1	1
automatic	1	1	0	0	0	0	1	1

3.3.3 Scoring and selecting words

After creating a word co-occurrence matrix, RAKE calculates the keyword score. This score of candidate word is computed by taking the degree of a word in the matrix (i.e. the sum of the number of co-occurrences the word has with any other content word in the text), as the word frequency (i.e. the number of times the word appears in the text) which is given by,

$$K = deg(t)/freq(t)$$

For e.g. for word "extraction" degree is 8 and frequency is 3 so we get the score (K) as,

$$K = 8/3 = 2.66$$

If we were to compute the keyword score for each words in our example, we will get as following:

Word	Degree Score
keyword	2.66
extraction	2.66
difficult	1.0
many	2.0
libraries	2.0
help	1.0
rapid	4.0
automatic	4.0

If we were to compute the keyword score for phrases we will get as following:

keyword extraction	5.33
many libraries	4.0
rapid automatic keyword extraction	13.33

After candidate keywords are scored, the top T scoring candidates are selected as keywords (where T is the number of keywords that has be extracted). For the above example, the method will give top 3 keywords which are rapid automatic keyword(13.33), keyword extraction(5.33), and many libraries(4.0).

After the keyword is selected sentence is mapped for each keyword i.e. for each of the keyword corresponding sentences that has the word from the summarized text is extracted.

3.4 Generation of distractors

Generating distractors is the most crucial step in the generation of automated MCQs. The difficulty of MCQs highly relies on the quality of distractors produced. A good distractor is one that is very similar to the key but not the key itself. So for generating distractors Wordnet approach is used.

WordNet is a lexical database for the English language, which was created by Princeton, and is part of the NLTK corpus. In the WordNet network, the words are connected by linguistic relations. These linguistic relations includes hypernym, hyponym, meronym, holonym, etc. WordNet stores synonyms in the form of synsets (Nouns, verbs, adjectives and adverbs are grouped into sets of cognitive synonyms) where each word in the synset shares the same meaning. Basically, each synset is a group of synonyms. Each synset has a definition associated with it. Relations are stored between different synsets. This Lesk algorithm is based on the assumption that words in a given "neighborhood" (section of text) will tend to share a common topic. A simplified version of the Lesk algorithm compares the dictionary definition of an ambiguous word with the terms contained in its neighbourhood.

For example, if there is a sentence "The bat flew into the jungle and landed on a tree" and a keyword is "bat", we automatically know that here we are talking about the mammal bat that has wings, not a cricket bat or baseball bat. Although we humans are good at it, the algorithms are not very good at distinguishing one from another. This is called word sense disambiguation (WSD). In the wordnet, "bat" may have several senses, one for a cricket bat, one for flying mammal etc. So the function get_wordsense tries to get the correct sense of the word given in the sentence. Once the sense of the word is identified the get_distractors_wordnet function is called to get the distractors. This function tries to get the distractors with the help of hypernyms and hyponyms of the key.

Hypernym - Hypernym is a word that names a broad category that includes other words.

Ex: animal is hypernym of dog.

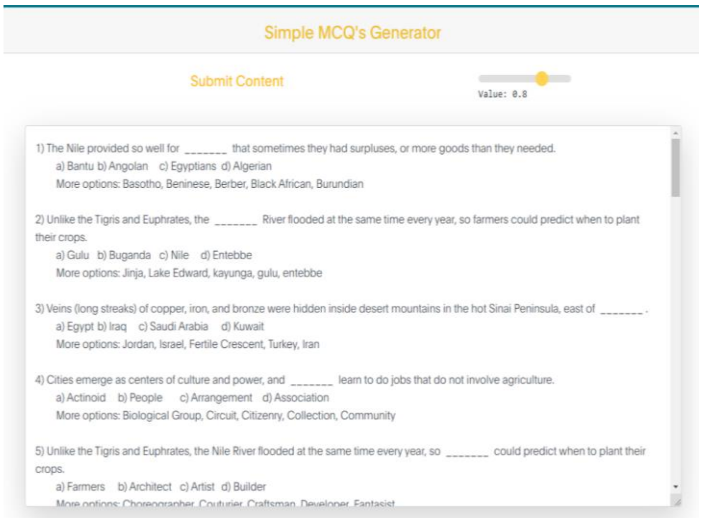
Hyponym - A word of more specific meaning than a super-ordinate term applicable to it.

Ex: car is a hyponym of vehicle.

Now all the possible hypernyms of the key and the corresponding hyponyms for each of the hypernym are found. These hyponyms are considered potential distractors. Then, the potential distractors are ranked, and the final distractors are chosen based on their rank. For the word 'bat' as the key, the hypernym for the bat is - an animal of which hyponyms can be an eagle and other birds which can be good distractors for the key. The ranks given to the potential distractors are based on whether it occurs in the text extracted from the summarization. The potential distractors that exist in the extracted text and having the same part of speech structure as the key have a higher rank than those with a different structure. This is because distractors with the same structure as the key tend to confuse the test takers more. Any three potential distractors with higher ranks are chosen at random as the final distractors.

3.5 output

After processing all the steps fill in the blanks type question are generated by mapping the keyword to the corresponding sentence. And appropriate distractors are generated through wordnet. Fig -3. shows the output of the proposed system.



1) The Nile provided so well for _______ that sometimes they had surpluses, or more goods than they needed.
a) Bantu b) Angolan c) Egyptians d) Algerian
More options: Basotho, Beninese, Berber, Black African, Burundian

2) Unlike the Tigris and Euphrates, the _______ River flooded at the same time every year, so farmers could predict when to plant their crops.
a) Gulu b) Buganda c) Nile d) Entebbe
More options: Jinja, Lake Edward, kayunga, gulu, entebbe

3) Veins (long streaks) of copper, iron, and bronze were hidden inside desert mountains in the hot Sinai Peninsula, east of _______.
a) Egypt b) Iraq c) Saudi Arabia d) Kuwait
More options: Jordan, Israel, Fertile Crescent, Turkey, Iran

4) Cities emerge as centers of culture and power, and _______ learn to do jobs that do not involve agriculture.
a) Actinoid b) People c) Arrangement d) Association
More options: Biological Group, Circuit, Citizenry, Collection, Community

5) Unlike the Tigris and Euphrates, the Nile River flooded at the same time every year, so _______ could predict when to plant their crops.
a) Farmers b) Architect c) Artist d) Builder
More options: Choreographer, Couturier, Craftsman, Developer, Fantasist

Fig -3: Output MCQs

4. Results

The results and accuracy of the proposed system are comparatively higher than other systems. For the summarizer, we have used the BERTSUM model, which outperforms the other models. As shown in Table- 1 [8], all BERT-based models outperformed previous state-of-the-art models by a large margin. BERTSUM with Transformer achieved the best performance on all three metrics. The BERTSUM with LSTM model does not have an obvious influence on the summarization performance than the Classifier model. So among all the models, we have chosen BERTSUM + Transformer model, for the text summarization which has the best score compared to all the previously proposed systems.

Table -1: Test set results on the CNN/DailyMail dataset using ROUGE F1

Model	ROUGE-1	ROUGE-2	ROUGE-L
PGN*	39.53	17.28	37.98
DCA*	41.69	19.47	37.92
LEAD	40.42	17.62	36.67
ORACLE	52.59	31.24	48.87
REFRESH*	41.0	18.8	37.7
NEUSUM*	41.59	19.01	37.98
Transformer	40.90	18.02	37.17
BERTSUM+Classifier	43.23	20.22	39.60
BERTSUM+Transformer	**43.25**	**20.24**	**39.63**
BERTSUM+LSTM	43.22	20.17	39.59

For generating distractors, the wordnet approach is used. We evaluate the quality of wordnet by examining the validity of its constituent synonyms and its hypernym-hyponym pairs [9]. The hypernymy validation procedure was tested on the set of all direct hypernyms for noun synsets in the Princeton WordNet (v2.1). There are a total of 79297 hypernym-hyponym pairs constituting this set. The number of pairs validated by the algorithm at different steps of the approach is summarized in Table 2. In all, the algorithm was able to validate 56203 out of 79297 noun hypernym relation pairs in the Princeton Wordnet, giving a validation percentage of 70.88%. Validation of the hypernym-hyponym relation for a pair of synsets is a strong indicator of hypernym relation between them. However, the failure to validate a synset pair is not a definitive indicator of erroneous construction and has to be treated as a flag for human inspection.

Table -1: Results of Hypernym Validation

Rule	*Number of hypernym-hyponym pairs validated*	*Percentage of total pairs*
(Rule 1)	16,934	21.35%
(Rule 2)	37,145	46.84%
(Rule 3)	2,124	02.68%
Total	56203	70.88%

5. Conclusion & Future scope

Multiple Choice Questions (MCQs) are generated successfully. Efficient questions are produced with good quality distractors. The problem of manually creating questions is solved with the proposed system. The proposed system creates automated questions with the help of NLP that reduces human intervention and it is a cost and time-effective system. And the accuracy of the distractor generated is reasonably high. This system not only helps teachers with E-assessments but also helps students who are preparing for competitive exams. Students can test their ability to solve the questions and can also check their understanding of the concepts.

Since our proposed system is based on Google's BERT Model, the accuracy of the system will increase in the future as the performance of the model is improved and as the research in the field of NLP is trying to reach the human level every day.

ACKNOWLEDGEMENT

Motivation and guidance are the keys to success. I want to extend my thanks to all the sources of motivation. We would like to grab this opportunity to thank **"Dr. S.D.Sawarkar"**, principal, for the encouragement and support he has given for our project. We express our deep gratitude to **"Prof. A. P. Pande"**, the head of the department, who has been the constant driving force behind the completion of this project. We wish to express our heartfelt appreciation and a deep sense of gratitude to our project guide **"Dr. C M Raut"**, for his encouragement, invaluable support, timely help, lucid suggestions, and excellent guidance, which helped us to understand and achieve the project goal. His concrete directions and critical views have significantly helped us in the successful completion of this work.

REFERENCES

[1] Santhanavijayan, A., Balasundaram, S.R., Hari Narayanan, S., Vinod Kumar, S., and Vignesh Prasad, V. (2017) 'Automatic generation of multiple-choice questions for e-assessment', Int. J. Signal and Imaging Systems Engineering, Vol. 10, Nos. 1/2, pp.54–62.

[2] Ayako Hoshino and Hiroshi Nakagawa (2005) " A real-time multiple-choice question generation for language testing: A preliminary study", EdAppsNLP 05: Proceedings of the second workshop on Building Educational Applications Using NLP.

[3] D. R. CH and S. K. Saha, "Automatic Multiple Choice Question Generation From Text: A Survey," in *IEEE Transactions on Learning Technologies*, vol. 13, no. 1, pp. 14-25, 1 Jan.-March 2020, doi: 10.1109/TLT.2018.2889100.

[4] Deepshree S. Vibhandik, Rucha C. Samant "Automatic / Smart Question Generation System for Academic Purpose", International Journal of Emerging Trends & Technology in Computer Science (IJETTCS), Volume 4, Issue 4, July - August 2015.

[5] Susanti, Y., Tokunaga, T., Nishikawa, H. *et al.* "Automatic distractor generation for multiple-choice English vocabulary questions", *RPTEL* **13,** 15 (2018). https://doi.org/10.1186/s41039-018-0082-z.

[6] Mojitha Mohandas1, Aishwarya Chavan2, Rasika Manjarekar3, Divya Karekar4 (2015)"AUTOMATED QUESTION PAPER GENERATOR SYSTEM" International Journal of Advanced Research in Computer and Communication Engineering Vol. 4, Issue 12, December 2015 ISSN (Online) 2278-1021 ISSN (Print) 2319 5940.

[7] A. Y. Satria and T. Tokunaga, "Automatic generation of English reference question by utilising nonrestrictive relative clause", Proc. 9th Int. Conf. Comput. Supported Edu., pp. 379386, 2017.

[8] Yang Liu,"Fine-tune BERT for Extractive Summarization", arXiv:1903.10318v2 [cs.CL] 5 Sep 2019.

[9] Raghuvar Nadig, J. Ramanand, Pushpak Bhattacharyya, "Automatic Evaluation of Wordnet Synonyms and Hypernyms", Proceedings of ICON-2008.

[10] Rahim, T. N. T. A., Aziz, Z. A., Rauf, R. H. A., & Shamsudin, N. (2017). Automated exam question generator using genetic algorithm. 2017 IEEE Conference on e-Learning, e-Management and e-Services (IC3e).