

Question Generation System Using NLP

¹Mrs D R Nanda Devi, ²Bitla Sharon, ³Janagam Sindhu, ⁴Kothakapu Arpitha Reddy, ⁵Atiki Likhith

¹Assistant Professor, Dept. of Computer Science and Engineering, G.Narayanamma Institute of Technology and Science, Hyderabad, Telangana, India.

^{2,3,4,5}B.Tech Students, Dept. of Computer Science and Engineering, G.Narayanamma Institute of Technology and Science, Hyderabad, Telangana, India.

Abstract: Manually creating interesting and relevant questions for subjective and objective tests is a time-consuming and tough task for teachers. This technique is useful in Natural Language Processing, but it is a demanding process. This application generates questions that are both subjective and multiple-choice. In summary, the BERT approach is used, and then subsequent phrases are mapped to generate MCQs. The question paper is created by randomly picking questions from the Bloom's Taxonomy. Parts of speech tagging (POS Tagging) is used to generate questions. T5_base is an NLP-based system that creates questions for objective and multiple-choice questions. This tool assists teachers in question generation, saving them time and effort. The development of question papers helps to address the manpower shortfall.

Key Words: MCQs, BERT, Wordnet, PKE, Deep learning, POS, T5_base

1. INTRODUCTION

All universities, colleges, and schools now offer online courses. Exams are a crucial instrument for assessing pupils' knowledge. Examiners are largely responsible for creating their test materials. It can be challenging to search through textbooks for new questions that are relevant and take the least amount of time on the exam. Both objective and subjective questions are acceptable. With the aid of NLP, the objective questions are automatically generated. The system receives the text as input, which is subsequently condensed using the BERT algorithm. Bidirectional Encoder Representation from Transformers, or BERT Xlnet. The Python Keyword Extractor (PKE) is now used to choose the keywords from the condensed text, and it maps each keyword to a sentence and relevant distractors. The bloom taxonomy level is also used to automatically build the subjective questions. The text is input into the system and then compressed using different T5_base approaches. Questions are generated from the summarised material based on sentence selection and POS tagging with the level of Bloom's taxonomy. By establishing this computerized programme, we can make teaching easier. Knowing the pertinent questions to ask about the given text input might be quite time-efficient. As a result, our objective is to develop a system that can generate a wide range of logical queries from the supplied text input. Currently, this can only be finished by individuals.

2. LITERATURE SURVEY

There are numerous studies on creating multiple-choice questions and creating subjective questions using various methods. A system of "Automatic generation of multiple-choice questions for eassessment" has been proposed by Santhanavijayan et al. They've created MCQs for their proposed system using an ontology-based method and

firefly-based preference learning. In order to make it possible to compose questions, they employed a web corpus. Hypernyms and hyponyms, two similarity measures, are used to generate the distractors. The system also generates analogy-based test questions to gauge pupils' linguistic proficiency. As well as Hiroshi Nakagawa, Ayako Hoshino Machine learning automatically produces questions in the study "A realtime multiple-choice question generation for language testing: A preliminary study."

Automatic Cloze Question Generation, or CQG, creates a list of cloze questions—sentences containing one or more blank spaces—from an English article as input. Sentence selection, keyword selection (choosing a potential blank), and distractor selection (choosing potential substitutes for the blank) are important CQG components. Before generating domain-specific distractor phrases using the model's knowledge base, potential sentences are first selected, followed by keywords selected using NER. Each sentence, keyword, and distractor selection is manually analyzed using the algorithm. automatic question generation You might think of the system as content selection and question generation that makes use of discourse cues. Identification of discourse markers and major discourse relations, such as casual, temporal, contrast, consequence, etc., are the main objectives. The type of Wh-question (such as why, where, which, and when) is determined by performing syntax transformations on (seven) discourse connectives after identifying suitable material for question framing. Semantics and syntax are assessed for the system.

3. PROPOSED DESIGN

In the proposed system, questions are generated from summarized text which is given as input. After

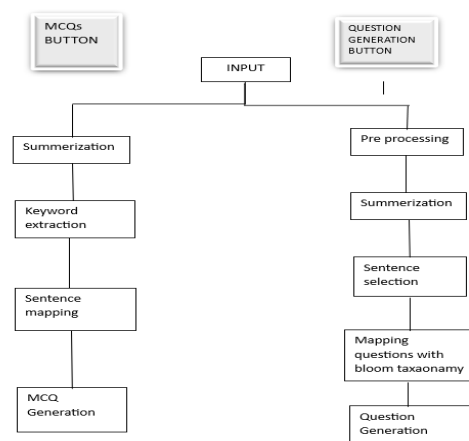


Fig-1:-System Architecture

summarization of text, there are different steps involved to create MCQs and questions. The different stages involved for generating questions are shown in Fig- 1.

I. 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.

II. PREPROCESSING

In both mcq question generation and subjective question generation both the input undergoes into preprocessing.

It transforms text into a more digestible form so that machine learning algorithms can perform better. Generally, there are 3 main components tokenization, Stemming, lemmatization.

Stemming is the elementary rule-based process of removal of inflectional forms from a token. The token is converted into its root form. For example, the word 'troubled' is converted into 'trouble' after performing stemming. Tokenization is about splitting strings of text into smaller pieces, or "tokens". Paragraphs can be tokenized into sentences and sentences can be tokenized into words.

Lemmatization is the process of converting a word to its base form, e.g., "caring" to "care".

III. DATASETS USED:

The wordnet dataset is utilized in both the Mcq generating and the subjective question generation processes. To create structured semantic associations between words, Wordnet is a sizable lexical database for the English language that is freely and openly accessible.

It is one of the earliest and most widely used lemmatizers and also has lemmatization capabilities.

Download Wordnet through NLTK in python console:
import nltk

```
nltk.download('wordnet')
```

You must run the lemmatize() function on a single word and construct an instance of the WordNetLemmatizer() to lemmatize.

```
nltk import from nltk.stem WordNetLemmatizer import
# Initialise the WordNet Lemmatizer
print(lemmatizer.lemmatize("bats")) >>> bat
lemmatizer.lemmatize("words") = WordNetLemmatizer()
#Lemmatize Single Word
```

The striped bats are hanging onto their feet for safety.

Define the sentence to be lemmatized. Tokenize the sentence by breaking it up into words.

```
words_list = nltk.word_tokenize(sentence) print(words_list)
# ['The', 'striped', 'bats', 'are', 'hanging', 'on', 'they', 'are', 'for', 'best']
```

IV. SUMMERISATION

After preprocessing, multiple-choice questions are made from the text's summary using the BERT XLnet approach.

The procedure of developing subjective questions involves POS tagging of the prepared material before summarization.

V. PART-OF-SPEECH TAGGING :

The act of assigning the proper Part-of-Speech to each token (word) in a corpus of text is known as POS Tagging. One of

the most useful parts of the Natural Language Toolkit (NLTK) is the POS tagger. Once the text has been read, the tokens are labeled with the appropriate part of speech (such as a noun, verb, adjective, etc.). A tag is used to specify each grammatical component.

We focus on NN, NNS, NNP, NNPS, VB, VBN, VBD, PRP, and PRPP while creating questions.

Instance: >>>bring in (nltk.pos_tag (nltk.word_tokenize) print)What are you doing right now?

Following POS tagging. After summarizing the input, there are four stages needed to create multiple-choice questions:

A. KEYWORD EXTRACTION:

The input for choosing the keywords is the summarized text. The answer to the query itself is the keyword. From the sentence, the keywords are chosen. Because there is not much information in all the words in the sentence.

By using the Python module YAKE, keywords are extracted. YAKE: Yet Another Keyword Extraction.

There are five essential parts to this algorithm:

a. casing

The word's case is taken into account by this function. Capitalized terms and acronyms like "NASA" are given more weight.

b. Word Position

This feature provides words that are present at the start of the document additional weight. It is based on the presumption that pertinent keywords are frequently more concentrated at the start of a document.

c. Word Frequency

The word frequency is calculated using this feature and is normalized by one standard deviation from the mean.

d. Word Relatedness to Context

A word's relationship to its context is measured using this attribute

e. Word Different Sentence

The frequency of a candidate word appearing in various sentences is measured by this attribute. A word with a higher score is used frequently throughout several sentences.

B. COMBINED WORD SCORE

S(W) is the score of the total 5 components:

$$\text{score}(w) = d * b a + (c / d) \text{score}(w) = d * b a + (c / d) + (e / d)$$

where, a = casing, b = position, c = frequency, d = relatedness, e = different

C. SENTENCE MAPPING

Following the evaluation of candidate keywords, the top-scoring candidates are chosen as keywords. Once a keyword has been chosen, a sentence is mapped for each keyword, extracting the word from the associated sentence from the text that has been summarised.

D. GENERATING DISTRACTORS

The most important phase in the creation of automated MCQs is the generation of distractors. The caliber of the created distractors has a significant impact on how tough MCQs are. A good deterrent is one that closely resembles the key but is not the key. Therefore, the Wordnet technique is employed to generate distractions.

A component of the NLTK corpus, WordNet is a lexical database for the English language developed at Princeton.

The words in the WordNet network are linked together by linguistic relationships.

These linguistic relationships include meronyms, holonyms, hypernyms, and hyponyms.

output: Multiple choice questions were successfully generated. After completing all the procedures, questions of the fill-in-the-blank variety are created by mapping the keyword to the appropriate sentence.

Additionally, wordnet is used to produce the right distractors.

The summarised text processes the following steps to generate subjective questions:

VI. ALGORITHMS

To create questions, the T5 method is employed. T5 is an encoder-decoder model that has been pre-trained on a variety of tasks that are both supervised and unsupervised and are each translated into a text-to-text format. The six levels of the questions are generated using various t5 algorithms. Based on the summarization range, each T5 algorithm depicts the taxonomy level of the bloom.

T5 is available in many sizes:

- t5-small
- t5-base
- t5-large
- t5-3b
- t5-11b.

We have different types of t5-base algorithms

1. t5-base-e2e-qg
2. t5-base-question-generator
3. t5-base-finetuned-question-generation-ap
4. T5_qgen-squad_v2

a. T5-base-e2e-qg

model="valhalla/t5-basee2e-qg", nlp = pipeline("e2e-qg")
 NLP(text)

b. T5-base-question-generator

An example of a model that can create questions from text is T5-base-question-generator. To provide high-quality questions based on the provided text, the model has been specifically tweaked, or fine-tuned. This demonstrates that it has mastered the art of picking out crucial information from texts and creating questions that make sense and are pertinent to the subject matter.

Although it can be utilized with single-word or short-phrase replies, the model works best with answers that are complete sentences. 512 tokens are the maximum length of a sequence. The following format should be used for inputs: The answer text is in this box. Context text is in this box.

c. t5-base-finetuned-question-generation-ap

A model that has been trained to create questions from text is referred to by the technical term "T5 base-finetuned-question-generation-ap." This specific model, referred to as T5-base, has been optimised for a particular purpose associated with autonomous question generating. The word "ap" in the model's name could stand for the particular dataset or application for which it was created. In plainer language, this model, which has been tailored for a certain use case, may automatically produce questions based on textual content.

d. T5_qgen-squad_v2

An artificial intelligence (AI) model called T5_qgen-squad_v2 has been particularly trained to create questions based on given text using the SQuAD v2 dataset. SQuAD v2 is a set of reading comprehension questions based on actual situations.

VII. QUESTION GENERATION

An artificial intelligence (AI) model called T5_qgen-squad_v2 has been particularly trained to create questions based on given text using the SQuAD v2 dataset. SQuAD v2 is a set of reading comprehension questions based on actual situations..

VIII. BLOOM TAXONOMY

There are six levels in the Bloom's taxonomy framework. Automated question paper generation uses the Bloom's taxonomy levels to generate questions of different types of cognitive skills.

Remembering: identify, list, define, recall, label, name, recognize, state, select.

Understanding: explain, summarize, describe, interpret, paraphrase, compare, contrast, illustrate

Applying: solve, use, apply, demonstrate, show, implement, calculate.

	1	1	1	1	1	1
	e	e	e	e	e	e
	v	v	v	v	v	v
	e	e	e	e	e	e
Bloom taxonomy /no of lines	1	1	1	1	1	1
lines 0	1	2	3	4	5	6
lines 10	0	9	8	8	7	5
lines 15	1	1	1			
	3	1	1	9	8	7
lines 20	1	1	1	1	1	1
	6	6	5	4	3	2

Table-1: Performance of various levels

Analyzing: analyze, compare, contrast, differentiate, categorize, deconstruct, examine, break down.

Evaluating: evaluate, judge, critique, justify, assess, argue, defend.

Creating: reate, design, develop, produce, invent, geerate, compose, formulate

Graph: comparision of Bloom Taxonomy level

OUTPUT: After processing all the steps subjective questions are generated according to different levels baes on Bloom's Taxonomy.

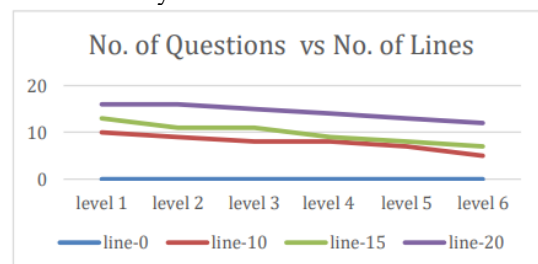


Fig-2:- Performance Analysis using a Graph

The table shows the number of questions generated based on number of lines of input. we can see that as level increases the number of questions generated are decreased gradually. The graph above is used to predict how questions are generated based on input lines. It is plotting for the table 1 to show that comparison levels.

4. CONCLUSIONS:

Questions that are both subjective and objective are successfully generated. The paper's primary goal is to close the technological and labor divide by automating the process of creating test questions. To propose an AQGS system using NLTK in Python, several techniques and methodologies used by existing articles were researched and assessed. The system accepts text passages as input and processes them through pre-processing, summarization, and POS tagging to produce a question paper with multiple-choice questions (MCQs) and long answers based on Bloom's Taxonomy. With the suggested system, the problem of manually creating questions is resolved. The suggested system uses NLP to generate automated queries, reducing the need for human participation while saving time and money.

FUTURE ENHANCEMENT:

Answer evaluation integrate with the question generation system. An Answer evaluation module can be integrated to evaluate and score the test answers submitted by students by calculating semantic similarity with the correct answer.

REFERENCES:

- [1] 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.
- [2] Narendra, A., Manish Agarwal and Rakshit shah, "Automatic Cloze-Questions Generation." RANLP, 2013.
- [3] Agarwal, Manish & Shah, Rakshit & Mannem, Prashanth, Automatic question generation using discourse cues, 2011, pp. 1-9.
- [4] Eldesoky, Ibrahim. (2015). Semantic Question Generation Using Artificial Immunity. *IJ. Modern Education and Computer Science*. 7. 1-8. 10.5815/ijmecs.2015.01.01.
- [5] Deokate Harshada G., Jogdand Prasad P., Satpute Priyanka S., Shaikh Sameer B., Automatic Question Generation from Given Paragraph, *IJSRD - International Journal for Scientific Research & Development* | Vol. 7, Issue 03, 2019 | ISSN (online): 2321-0613.
- [6] Kalpana B. Khandale, Ajitkumar Pundage, C. Namrata Mahender, Similarities in words Using Different Pos Taggers, *IOSR Journal of Computer Engineering (IOSR-JCE)* e-ISSN: 2278-0661, p-ISSN: 2278-8727, PP 51-55.
- [7] D.R Nanda Devi Automatic Question Generation System *International Research Journal of Engineering and Technology* Vol. 08, Issue 11, pp:375-377, November 2021, IF:7.529 Google Scholar 2395-0072.