

Automatic Question Generator Using Natural Language Processing

Puneeth Thotad^{1*}, Shanta Kallur², Sukanya Amminabhavi³

^{1, 3} Department of Master of Computer Applications, KLE Institute of Technology, Hubballi-27, India

². Department of Computer Science and Engineering, KLE Institute of Technology, Hubballi-27, India Corresponding author:

shanta06kallur@gmail.com (Shanta S. Kallur)

DOI: 10.47750/pnr.2022.13.S10.330

Abstract

The Automatic Question Generator is intended to generate new questions from the text that are natural language, semantically accurate, and syntactically cohesive. In contrast to other natural language-generating tasks like summarization and paraphrasing, answers are crucial for questions. High-quality distractors and effective questions are used in the construction of multiple-choice questions. With the use of this system, educators can create multiple-choice assessment questions that have correct answers and distractors. An educator can quickly assess a student's comprehension of the subject. This technique allows students to evaluate their own understanding level of the subject. This model is useful for creating test papers for evaluation in the educational sector. By simply copying or pasting one or more paragraphs, teachers can generate questions on their subjects. Python programs that work with data from natural human language can be written with a Natural language tool kit. This tool kit introduces text processing libraries containing functions for tokenization, parsing, lemmatization, chunking, POS tagging, and stemming. Text is used by many natural language processing techniques, such as topic modeling, to identify important information. After that, a list of questions is created based on the texts that were extracted as being significant or instructive. Different ways to question generating typically provide questions that are factual in nature, such as who, when, where, why, and what. A program for natural language processing aids in the understanding of language and spoken language by machines. The method breaks down the content into its component pieces, interprets the language's meaning, chooses the relevant actions, and finally presents the content to the user in a language they can understand.

Keywords: Machine learning, supervised learning, SpaCy, Natural language processing, POS tagging, Artificial Intelligence.

1. Introduction

An Automatic Question Generator using Natural language processing (NLP) generates relevant, syntactically, and semantically accurate questions based on various input formats such as text, a structured database, or knowledge bases. NLP is a subfield of Artificial Intelligence (AI) that aims to facilitate human-computer interaction using natural language that recognizes and understands human spoken and written language. It is the ultimate objective of NLP to read, analyze, understand, and make sense of written or textual information in natural languages in a meaningful manner. With the help of NLP, this project aims to set questions for the computer-based examination, educators, and students who are preparing for competitive exams [1].

The purpose is to improve the method of setting MCQs and modifying them, along with creating a viable question bank that academicians and learners can use. As a result, you will ensure that the MCQ included appropriate questions and options. These questions will align with the learning objectives and significance of the topics discussed in the tutorial material [2]. Automatic question generator can be applied to a range of domains including Massive

open online courses, setting objective questions, search engines, automated help systems, chatbots (e.g. for customer interaction), and health care for analyzing mental health. However, it requires time and effort to manually create meaningful and relevant questions[3].

Setting questions is a challenging undertaking for both teachers and students who are preparing for competitive exams. The current approach entails manually setting the questions, which takes a lot of time and human effort. Consequently, there is an increasing need for a system that can easily, quickly, and with minimal human effort develop questions. Manually creating test papers and quizzes take a lot of time from teachers, professors, and tutors. Similarly, students spend considerable time self-analysis of their knowledge.

The remaining part of the paper comprises four sections. Section 2 gives necessary background knowledge of works carried out by various researchers on NLP. Section 3 describes the methodology adopted. Section 4 represents the implementation details, and the conclusion of the work is given in section 5.

2. Literature Review

To realize the state-of-the-art work in NLP and its application in education sector analysis using machine learning models, a literature survey is conducted, and the following are the gist of the articles related to the proposed work.

A rule-based methodology for automating the generation of questions is proposed by Onur et al. The paper proposes a method that considers both the syntactic and semantic structure of a sentence. The purpose of this paper is mainly to generate more comprehensive questions based on word semantic roles. This paper proposes to generate questions using an existing sentence by following a rule-based model. Specifically, reliance-based, named entity recognition (NER)-based, and semantic role (SRL) marking-based layouts/rules are utilized. To decide between who and what questions, the system has proposed using Chunking [4].

Aleena et al's paper describe an implementation for a question generation system. The main idea is to ensure the system understands natural language so it can process and manipulate the data. The proposed system focuses on pre-processing of data, key phrase extraction, and natural language processing. This method can be used to develop a fast, secure, and randomized system that is advantageous in many aspects, including education. As input, this paper proposed an automatic question paper generation system that accepts text, documents, or PDF files. The proposed system removes stop words and uses Natural Language Processing. A TF-IDF algorithm is used for key phrase extraction, and the existence of terms is checked on the wiki. Using the WordNet tool, create triplets for question paper generation as well as conduct input clarity checks [5].

Priti et al., their paper discusses the generation of questions based on Bloom's taxonomy. The Pre-processing of data is mainly concerned with feature extraction and Stanford pos tagging. An analysis of syntactic structures and semantic structures is part of pre-processing. The POS tagging and Chunking are included in the syntactic analysis. It is carried out in the semantic analysis process to recognize named entities. Afterward, the text is pre-processed, appropriate 'WH' question words are mapped to it, and questions are generated [6].

D. R. CH and S. K. Saha have proposed a survey on automatic multiple-choice question generation from the text. Questions are generated by reading articles from the database. For the text summarization and frequency count of words, an NLP-based summarizer is used, while pattern matching is used for selecting the keywords. They have used wordnets, pattern matching, domain ontologies, and semantic analysis to generate distractors. The creation of a general workflow for automatic MCQ generation is the main topic of this study [7].

Ankita, K. A. et al. describe the intricacy of POS tagging as the number of computational levels required for determining POS tags. The focus of this paper is on how to reduce the complexity of POS labeling by using Hidden Markov Models. The authors have likewise recommended Named Entity Recognition (NER). Even though there are numerous POS taggers out there, individuals are still searching for a way that requires less effort to accomplish and is less likely to create complications, as well. HMM, bases taggers find tags in sentences, rather than in individual words [8].

To summarize, the researchers have worked on NLP applications on question paper generation and multiple choice questions. There is a scope to develop a model keeping in consideration the object-based learning [9] and blooms taxonomy [10] and draw the questions. A web-based application is developed to generate the multiple choice questions for the text/ paragraph given as input. Assess the score of the test and give the results and display right answers to participants.

3. Proposed methodology

Many teachers, professors, and tutors (academicians) spend a significant amount of time preparing test papers and quizzes manually. Similarly, students spend a substantial amount of time on self-analysis (self-calibration). In addition, students rely on their mentors to help them with their self-analysis. This has led to working on the NLP area, which currently has a large scope for improvement. Security is also a big concern for them, as well as the lack of teaching staff in any institute makes creating the papers difficult. In conclusion, the system proposes an Automatic Question Generator that stores the data, provides fast operations, and provides high security.

The proposed system aims to boost the effectiveness of the existing one. Every constraint of the current system can be bypassed with this system. This reduces manual work and provides proper security and includes user interactions. The Educator provides inputs in text format, and it is processed by the NLP system so the system is identified as a layered architecture. The NLP system summarizes, extracts, and analyzes the user input text and generates different types of questions like multiple choice, question, and answers, and fill in the blanks. The Educator can upload these questions to students as an assessment. The main purpose of the proposed system is to reduce the time consumed by the setting of assessment questions like MCQ and take self-assessment to know how much the subject is understood. The system will generate questions with meaningful distractors and correct answers. Students can view these questions generated by educators along with answers, and answer to the questions like an assessment for practice [11].

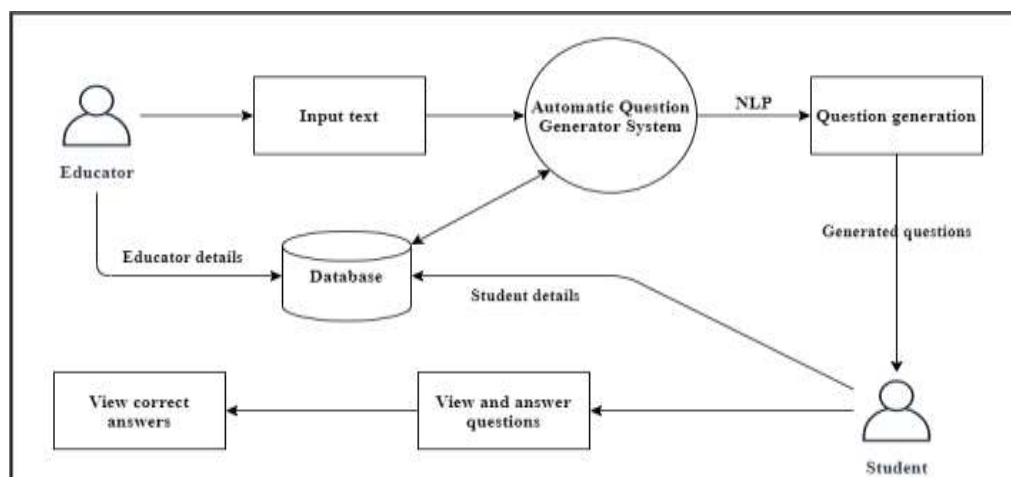


Figure 1: Proposed methodology

The System Architecture of the Automatic Question Generator in Figure 1 represents the relationship between the principal elements of the system by using blocks and arrows as shown in the above figure. The above diagram represents a block diagram of an Automatic question generator. In this process Educator and Student need to provide valid credentials to log in to the system. The user details are then stored in the database. Educators need to provide the text as the input to generate questions. Questions are generated based on text using Natural Language Processing technology. The generated questions are stored in the database and later uploaded to students. Students can practice those questions and answer them. Students can also ask queries and view the results of answered and unanswered questions.

4. Model design

4.1. SpaCy

SpaCy provides Python APIs for Natural Language Processing (NLP) that is free and open source. Data processing and analysis using it are becoming increasingly popular in NLP. There are large volumes of unstructured textual data, so processing and interpreting this data are crucial. Data must be represented in a way that can be understood by computers to achieve this. A natural language processing system can help you achieve this. The NLP models included in spaCy can be used to perform the majority of commonly occurring

NLPtasks. A number of tasks can be performed by SpaCy, it includes text tokenization, POS tagging, named entity recognition (NER), lemmatization, and word vectorization [12].

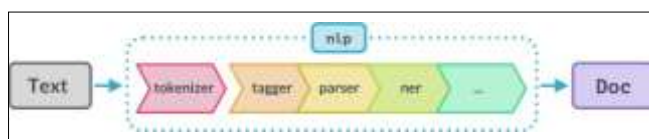


Figure 2: SpaCy Architecture

Figure 2 represents the architecture of spaCy library it involves Tokenizer, Tagger, Parser, and NER, etc. The tokenization process involves breaking up a text into tiny segments, known as tokens. Depending on the target, tokenization can produce tokens of sentences at the document level, tokens of words at the sentence level, or tokens of characters at the word level. The POS tags provided by Spacy include NOUN, PUNCT, ADJ, ADV, etc. With the help of statistical models and a trained pipeline, spaCy can categorize tokens based on labels or tags. A syntactic dependency parser is included in spaCy, as well as a powerful API for navigating the tree. This parser is also used to detect sentence boundaries, and to iterate over phrases containing nouns, or chunks. Identifying and classifying named entities is a task of Named Entity Recognition, a subset of NLP. Taking the raw text and categorizing the named entities into organizations, persons, places, time, money, etc. The entity types are basically identified and segmented into various classes according to their characteristics.

4.2. Natural Language Processing Toolkit (NLTK)

With NLTK, one may create Python programs that operate with data from natural human language. It is one of the most popular Python platforms. Programming language models are introduced in a useful way by NLTK. Tokenization, parsing, lemmatization, chunking, POS tagging, and stemming are all included in the NLTK text processing libraries. NLTK is a massive toolkit for natural language processing (NLP), designed to assist with all aspects of the methodology. NLTK enables them to split sentences into paragraphs, split up words, recognize the words' parts of speech, highlight the main points, and even help the machine understand the content [13] [14].

4.3. Sense2vec

A neural network model called Sense2vec [15] uses large corpora of words to generate vector space representations of them. The Sense2vec algorithm is an extension of the infamous Word2VEC algorithm [16], [17]. Instead of tokenizing words, it embeds "senses". The concept of a sense is the combination of a word with a label, i.e. a statement representing the circumstances in which a word is used. POS Tags, Polarity, Entity Names, Dependency Tags, etc. can be used for this type of label. Neural word representations obtained with Word2vec are unable to encode context, despite being able to capture complex semantic and syntactic relationships among words. A contextually keyed word vector (i.e., one vector for each word sense) is the best way to solve this problem while disambiguating word senses. Contextually keyed vectors are the solution to this problem with Sense2vec. As a simple but powerful variation of word2vec, sense2vec was developed. Syntactic dependency parsing is improved while calculating representations of word senses is significantly reduced in terms of computational overhead. This package contains a sense2vec model that integrates seamlessly with spacy.

```

[15/Jan/2022 21:49:49] "GET /notebook/generate_questions/ HTTP/1.1" 200 8906
C:\anaconda3\lib\site-packages\django\db\models\fields\__init__.py:2534: RuntimeWarning: DateTimeField Question
Fraper.generated_date received a naive datetime (2022-06-25 21:43:10.298216) while time zone support is active
  warnings.warn(
Research methodology is a way of explaining how a researcher intends to carry out their research.
It's a logical, systematic plan to resolve a research problem.
A methodology details a researcher's approach to the research to ensure reliable, valid results that address t
heir aims and objectives.

Naming model for generation
Sense2vec_distractors successful for word : research methodology
Sense2vec_distractors successful for word : aim
Sense2vec_distractors successful for word : objectives
Sense2vec_distractors successful for word : approach
Sense2vec_distractors successful for word : results
Sense2vec_distractors successful for word : plan
Experimental Design, Research methodology, Research Methods, Methodology
aim, books, attempts, aimed
New Camps, map control, Other objective, objectives
Aims, approach, confront, heart
results, Positive Result
being, plan, Near future, start

```

Fig 3. Question Generation Result

5. Evaluation and Performance analysis

After entering the text paragraph, click on Generate question the code will run on the command prompt as shown in Fig 3. Here for example entered a small paragraph, the paragraph will be split into sentences to generate questions for each sentence. There are three types of questions are generated multiple-choice, fill-in-the-blank, and Boolean-type questions. The sentence split is done using the NLTK package. The Sense2vector package is used to generate valid distractors with the correct answer for questions generated. It extracts keywords from each sentence and finds related words to generate distractors from that paragraph only. The questions are formed based on the keywords extracted from the paragraph. Boolean-type questions are also generated with the values Yes/ No.

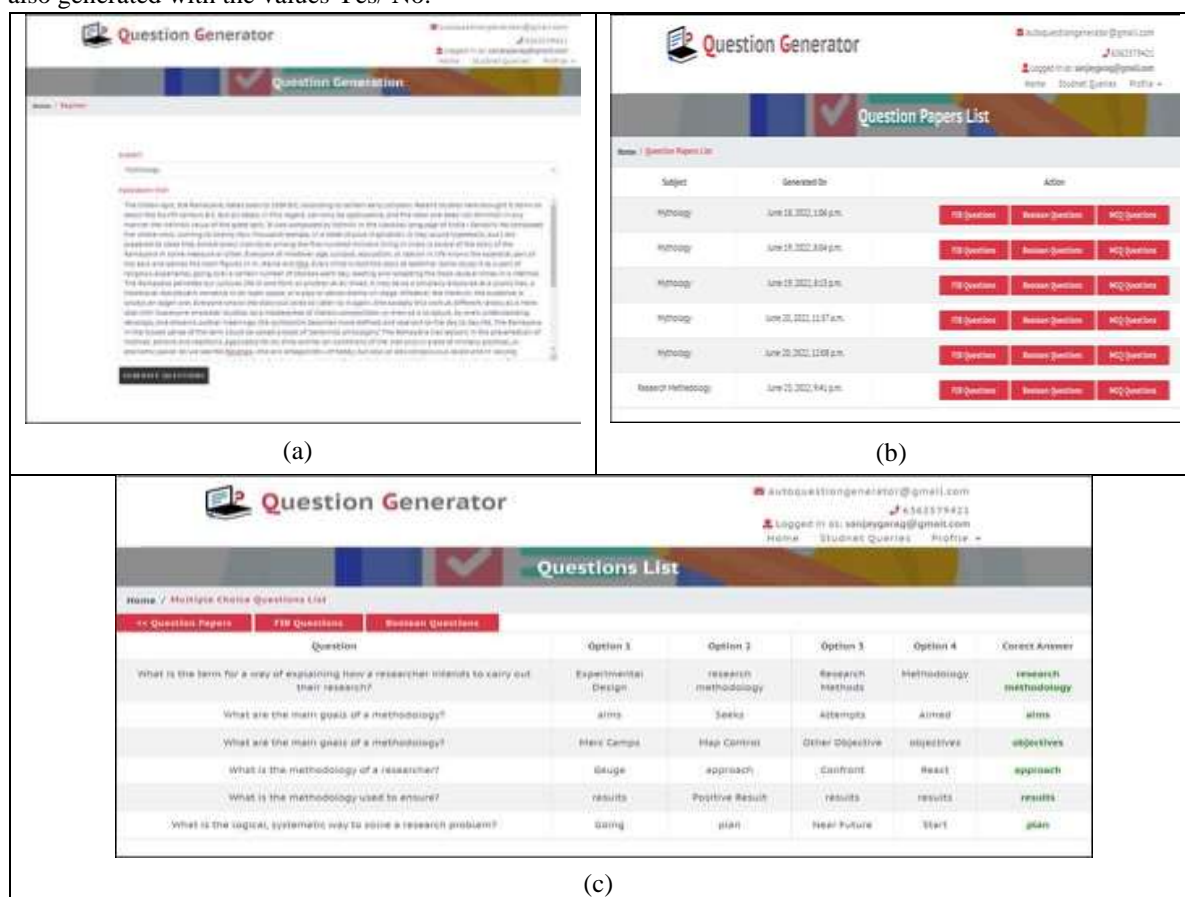


Fig. 3. (a) Question Generation Page, (b) Question types, Fig. 3 (c) Questions list

6. Results and Discussion

The educator should select a subject from the dropdown which wants to generate questions. Paragraph text should be provided, multiple paragraphs can be provided. By clicking on Generate questions the question is generated and stored in the database, and it redirects to the question paper list page to view generated questions. The list of question papers with the subject and generated time and date is displayed. Educators can view the question by clicking on the button for different question types. The questions paper list consists of question types like MCQs, Fill in the blanks, and Boolean-type questions. The question column consists of questions and options consisting of one correct answer with other three distractors. The correct answer column displays the correct answer. Figures 3, 5, & 6 show the process of question generation.

7. Conclusion and Future Scope

The generation of questions from text is useful in numerous application domains, but manual generation is time-consuming and costly. In addition to being a web-based and desktop-based application system, it offers several features mainly for producing MCQs, Boolean type, and fill-in-the-blank questions. The goal of this application is to describe an automatic question generator built on NLP. This system helps students to self-

analyze their knowledge about the subject. The application can be used in the educational field, and it is helpful in generating test papers for assessment. The generated questions are displayed on the student side with questions and answers and like an assessment for practice test for students. It is helpful in competitive exams by generating questions and practicing them.

This project can be further extended to generate paraphrasing questions, descriptive questions, and question paper generation for different types of questions. It is also possible to add additional automated answer copy-checking systems to verify the answers and provide the results to the students.

REFERENCES

- [1] T. B. Shahi and C. Sitaula, "Natural language processing for Nepali text: a review," *Artif. Intell. Rev.*, vol. 55, no. 4, pp. 3401–3429, 2022, doi: 10.1007/s10462-021-10093-1.
- [2] D. Meurers, "Natural Language Processing and Language Learning," in *The Encyclopedia of Applied Linguistics*, Blackwell Publishing Ltd, 2012. doi: 10.1002/9781405198431.wbeal0858.
- [3] N. Patil, K. Kumari, D. Ingale, P. Patil, and A. R. Uttarkar, "A Survey on Automatic Multiple Choice Questions Generation from Text," *Int. J. Sci. Res. Eng. Trends*, vol. 7, no. 3, pp. 1997–1999, 2021.
- [4] O. KEKLIK, "AUTOMATIC QUESTION GENERATION USING NATURAL LANGUAGE PROCESSING TECHNIQUES," 2018. Accessed: Nov. 20, 2022. [Online]. Available: <http://openaccess.iyte.edu.tr/bitstream/handle/11147/6938/T001801.pdf?sequence=1&isAllowed=y>
- [5] M. R. S. M. S. MPhil and G. K., "Automatic Question Paper Generator System," *Int. J. Trend Sci. Res. Dev.*, vol. Volume-3, no. Issue-3, pp. 138–139, 2019, doi:10.31142/ijtsrd21646.
- [6] P. 'Gumaste, S. 'Joshi, S. 'Khadpekar, and S. 'Mali, "Automated Question Generator System Using NLP," *Int. Res. J. Eng. Technol.*, vol. 7, no. 6, pp. 4568–4572, 2020, [Online]. Available: <https://www.irjet.net/archives/V7/i6/IRJET-V7I6848.pdf>
- [7] D. R. Ch and S. K. Saha, "Automatic Multiple Choice Question Generation from Text: A Survey," *IEEE Transactions on Learning Technologies*, vol. 13, no. 1. Institute of Electrical and Electronics Engineers, pp. 14–25, Jan. 01, 2020. doi: 10.1109/TLT.2018.2889100.
- [8] Ankita and K. A. Abdul Nazeer, "Part-of-speech tagging and named entity recognition using improved hidden markov model and bloom filter," in *2018 International Conference on Computing, Power and Communication Technologies, GUCON 2018*, Mar. 2019, pp. 1072–1077. doi: 10.1109/GUCON.2018.8674901.
- [9] K. Ellinghaus, B. Marsden, U. McIlvenna, F. Moore, and J. Spinks, "Object-based learning and history teaching: the role of emotion and empathy in engaging students with the past," *Hist. Aust.*, vol. 18, no. 1, pp. 130–155, 2021, doi: 10.1080/14490854.2021.1881911.
- [10] M. T. Chandio, S. M. Pandhiani, and S. Iqbal, "Bloom's Taxonomy: Improving Assessment and Teaching-Learning Process," *J. Educ. Educ. Dev.*, vol. 3, no. 2, p. 203, Dec. 2016, doi: 10.22555/joeed.v3i2.1034.
- [11] D. Khurana, A. Koli, K. Khatter, and S. Singh, "Natural language processing: state of the art, current trends and challenges," *Multimed. Tools Appl.*, pp. 1–32, Jul. 2022, doi: 10.1007/s11042-022-13428-4.
- [12] N. Colic and F. Rinaldi, "Improving spacy dependency annotation and pos tagging web service using independent NER services," *Genomics and Informatics*, vol. 17, no. 2, 2019, doi: 10.5808/GI.2019.17.2.e21.
- [13] E. Loper and S. Bird, "NLTK," pp. 63–70, 2002, doi: 10.3115/1118108.1118117.
- [14] V. N. Gudivada and K. Arbabifard, "Open-Source Libraries, Application Frameworks, and Workflow Systems for NLP," in *Handbook of Statistics*, vol. 38, Elsevier B.V., 2018, pp. 31–50. doi: 10.1016/bs.host.2018.07.007.
- [15] A. Trask, P. Michalak, and J. Liu, "sense2vec - A Fast and Accurate Method for Word Sense Disambiguation In Neural Word Embeddings," Nov. 2015. doi: 10.48550/arxiv.1511.06388.
- [16] L. Ma and Y. Zhang, "Using Word2Vec to process big text data," in *Proceedings - 2015 IEEE International Conference on Big Data, IEEE Big Data 2015*, Dec. 2015, pp. 2895–2897. doi: 10.1109/BigData.2015.7364114.
- [17] B. Jang, I. Kim, and J. W. Kim, "Word2vec convolutional neural networks for classification of news articles and tweets," *PLoS One*, vol. 14, no. 8, p. e0220976, Aug. 2019, doi: 10.1371/journal.pone.0220976.