International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 08 Issue: 09 | Sep 2021 www.irjet.net p-ISSN: 2395-0072

# Automated Question Generation from Scanned Documents

**Pranjal Rane[1], Ritvik Patil[2], Ojas Natu[3], Prahlad Pore[4], Pranay Sridhar[5]**

[1]*Student, Dept. of Computer Engineering, Vishwakarma Institute of Technology, Maharashtra, India*
[2]*Student, Dept. of Computer Engineering, Vishwakarma Institute of Technology, Maharashtra, India*
[3]*Student, Dept. of Computer Engineering, Vishwakarma Institute of Technology, Maharashtra, India*
[4]*Student, Dept. of Computer Engineering, Vishwakarma Institute of Technology, Maharashtra, India*
[5]*Student, Dept. of Computer Engineering, Vishwakarma Institute of Technology, Maharashtra, India*

---

**Abstract -** Student assessment has been an integral and a critical aspect of the teaching and learning process of today's time. Teachers assess the performance of students mainly based on different kinds of tests. So it is necessary for the teachers to make sure that the test covers all areas of the course content. Moreover, teachers spend a lot of time creating a question bank for a test by reading textbooks word by word and framing questions. This time could be better invested into teaching students and improving the overall quality of education imparted by the educational institutions. This is why tools which allow teachers to upload images and get a question bank come in very handy and resourceful.

***Key Words*: OCR, Tokens, POS Tagging, Classification, NLP, Grammar Checking, Question Generation**

## 1. INTRODUCTION

In this project, 'Automated Question generation tool from Scanned Documents', we present a smart question generating system. This tool allows the user to upload a scanned document and generate different types of questions depending upon the content uploaded by the user. We will be achieving this using Image Processing and Natural Language Processing. Generally, teachers or administrators generate the question paper manually using software like MS Word which requires a lot of time and effort. This project eliminates human efforts and saves resources and time. Our application allows educational institutions to prepare questions for examination and for students to practice. The basic idea is to perform Optical Character Recognition(OCR) on the scanned documents to generate text from them. Further the sentences go through processing, to find the important words in them and generate questions around them.

## 2. LITERATURE REVIEW

OCR (optical character recognition), also known as text recognition, is used to distinguish printed or handwritten text characters inside digital images of physical documents, such as a scanned paper document.[1]

Tesseract supports unicode (UTF-8), and recognizes more than 100 languages.[2] On the output side of things, Tesseract can give output in various popular formats like: HTML, PDF, Tab Separated values, plain-text and invisible-text-only PDF. The results given by Tesseract are better when the image quality is higher.

Peter Norvig's blog post on setting up a simple spell checking algorithm is used to develop a pure python spell checking library.[3]

This spell checking requires finding all possible permutations with an edit distance between 0 and 2 from the original word. For doing so, the Levenshtein Distance Algorithm is used.[4]

Later, all permutations (like transportation, deletions, replacement, etc) are compared to known words within a word frequency list. The words that are found to have a high occurrence in the frequency list are more likely to be the correct results.

The advantage of using this library is the availability of plenty of algorithms that help in the learning process.

One can compare among different variants of outputs. NLTK is used to perform parts of speech tagging and is used to assign grammatical information of each word of the sentence.[5]

There are more than a few algorithms suggested for the task of automatic question generation. These algorithms typically follow the flow consisting of the following steps: 'Splitting Sentences based on punctuation', 'Tokenization', 'Classification of tokens' and finally 'Question generation' based on the prominent tokens. However, the quality of questions and problems like 'pronoun identification and replacement' are not accounted for. The paper "Algorithm for Generating Questions from Text."[10] mentions the latter in detail and the algorithm mentioned has some amount of human intervention involved.

The publication 'A Systematic Review of Automatic Question Generation for Educational Purposes'[11] from Springer provides a systematic review of different approaches tried so far in the domain of automatic question generation. Even after extensive research in the field, accuracy remains the main problem to be dealt with and various different approaches are being tried to find an algorithm with a higher accuracy than the existing ones.

## 3. METHODOLOGY

The method is divided into 3 sub-parts:-

### 3.1 Optical Character Recognition :

Optical Character Recognition starts with a connected component analysis.[6] In this phase the outlines of the components are stored. The outlines are then gathered together, by nesting into Blobs. The blobs are then organized into text lines, which are then analysed for proportional text.

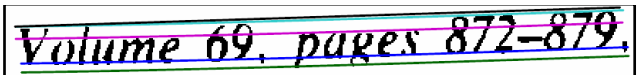



**Fig - 1 :** Curved Fitted Baseline

The text lines are then broken into words differently based on the kind of character spacing. Then the recognition is done using two passes. The first pass is done to recognize each word one by one. Each satisfactory word obtained from the previous phase is then passed further on to an adaptive classifier in the form of training data.

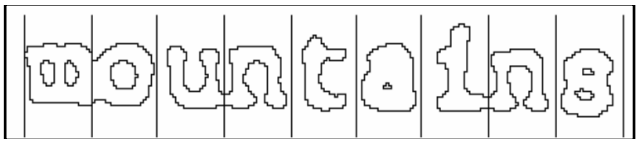


**Fig - 2 :** *Fixed Pitch Chopped Word*

The adaptive classifier then gets a chance to more accurately recognize text that is going to be coming further when we go on reading down the page. At times, it may be that the adaptive classifier has learnt something useful too late to make contributions near the top of the page. So, a second pass is done so that words that were not recognized well enough are recognized again.

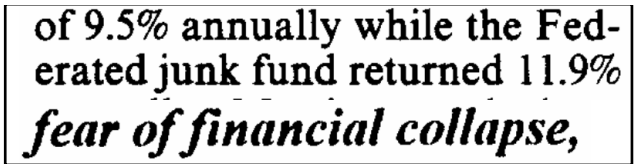



**Fig - 3** : Word Spacing

A final phase resolves fuzzy spaces, and locates the small and capital text.

### 3.2 Spell Check :

The output of the above OCR model is a text file containing the recognized text from a given image. This text file needs to be processed in order to remove all the discrepancies so as to avoid any difficulty while generating questions in the later part.

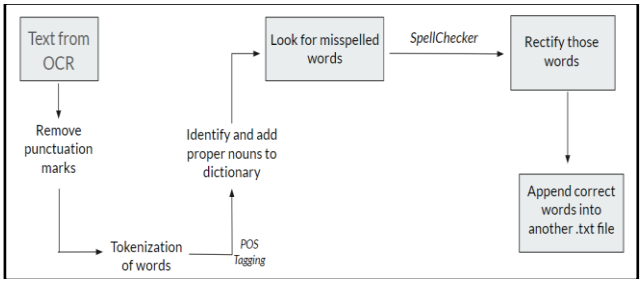



**Fig - 4 :** Spell Check Flow

In every piece of text, there are certain characters which have almost no significance while generating questions from that text. A good example of such characters is the punctuations present within a text. So the text obtained from the OCR model is processed to remove punctuation marks present within the text.

Each "entity" that is a part of the text is split up. Whenever a sentence gets "tokenized" into words, each word within the sentence is referred to as a 'token'. [7] These tokens are then stored in a list. This is essential to analyze each word individually.

To ensure maximum accuracy while generating questions, we must rectify misspelled words present within the text. To do so, we have used the check() function from the 'language_check' library. But before using the check() function we need to remove/mask proper nouns from the text since those would always be considered as misspelled words by the check() function. To identify proper nouns, POS tagging has been used with the help of 'nltk' library. POS tagging can be defined as the process of constructing a word in a corpus to a corresponding part of a speech tag, based on its context and definition. [5]

```
[('His', 'PRP$'), ('name', 'NN'), ('is', 'VBZ'), ('Pranjal', 'NNP')]
```

**Fig - 5 :** POS Tagging

After POS tagging, words except the proper nouns are separated out by checking their tag. Then these words are looked for spelling mistakes using spellchecker.

Now that misspelled words are identified, the last steps would be to rectify these words. To rectify these words, the check() function from the 'language_check' library is used. In the end, these corrections are appended to the original text file obtained from the OCR model along with the masked proper nouns.

### 3.3 Question Generation:

Once the text has been corrected for spelling errors, it is then sent to the Question generation phase.

The Question generation works on a very basic assumption that, 'words from the text would be great answers for questions'. So our major efforts will have to be to decide which words or phrases are good enough to become answers.

The solution proposed is to generate multiple possible answers from text, by splitting this complex problem into the following simpler and smaller steps:

1. Recognise unique keywords from the input text and use them as answers to the questions.
2. Replace these keywords from the sentence with blank space and use it to form the base for the question.
3. Transform the sentences containing blank spaces into a more question-like sentence.
4. Generate incorrect answers, by finding multiple distractor words that are similar to the answer.

Before we could start generating questions, it is very important to understand more about how questions are made and what kind of words are it's answers. For this we used the SQuAD 1.0 dataset which has about 1,00,000 questions generated from Wikipedia articles along with answers to the questions. [8]

To decide the likeliness of a word to be an answer binary classification was employed on each word from the text This is where the 'spaCy' library really helped us and provided the functionality of word tagging. [9]

To continue with the binary classification step we had to create the entire dataset. We created a dataset by extracting each non-stop word from the paragraphs of each question in the SQuAD dataset and added some additional features to it like:

1. Which part of speech is it?
2. Is it a Named entity?
3. Are only alpha characters used?
4. Shape i.e. whether given characters are only alpha characters, or digits or punctuation marks (xxxx, dddd, Xxx X. Xxxx)
5. Word count
6. And the label 'isAnswer' - this label signifies whether the word extracted from the paragraph is the same and also if it is in the same place as the answer of the SQuAD questions.

The next step is to train the binary classification on the above created dataset. We decided to use scikit-learn's Gaussian Naive Bayes algorithm to classify each word whether it's an answer [12]. The results were very good. At a quick glance, the algorithm was able to successfully classify most of the words as answers. The ones it didn't classify were in fact considered to be unfit to be answers.

One of the biggest advantages of Naive Bayes is that it returns the probability for each word whether it is fit to be an answer or not. We've used this to order the words from the most likely answer to the least likely.

By now we were able to classify which words in a sentence could be possible answers for questions created using that sentence. The next step is to generate distractors (incorrect answers) for the question. This was done using the gensim library and the Stanfords Glove dataset. The Glove Dataset consists of word embeddings and cosine similarity between different words.

```
distractions = model.most_similar(positive=['oxygen'], topn=10)
for i in range (len(distractions)):
    print (distractions[i], "\n")
```

```
('hydrogen', 0.63267982006073)

('nitrogen', 0.6251460313796997)

('helium', 0.5435217022895813)

('nutrients', 0.5369840860366821)

('breathing', 0.5023170709609985)

('chlorine', 0.4946938157081604)

('monoxide', 0.4911428689956665)

('dioxide', 0.4911195635795593)

('ammonia', 0.49079084396362305)

('carbon', 0.4836854338645935)
```

**Fig - 6 :** Distraction Generation

Most of the words generated as distractors were just fine and could easily be mistaken for the correct answer. But some of the generated distractors were not appropriate. Since we didn't have a dataset with incorrect answers we fell back on a more classical approach. We cleaned the results by removing the words that didn't belong to the same part of speech or the same named entity as the answer, and added some more context from the question.

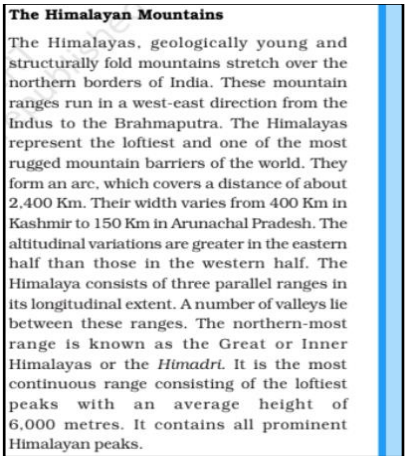
Now that we have trained a model we could make a pickle file for this and load this to make predictions. It will tell us the probability of words to be answers in the input text. We can then replace the words with high probability with a blank and form a 'Fill in the Bank' type question. Following this we can add three distractors (incorrect answers) generated above and form a 'Multiple Choice Question'. We can replace the blank with a distractor or with the correct answer to form a 'True or False' type question. In this way we can generate three different kinds of questions using a single approach.

## 4. RESULTS

Three types of questions are being generated by the system:

1. Multiple Choice Questions (MCQ) (Fig - 8)
2. Fill in the blanks Questions (Fig - 9)
3. True or False Questions (Fig - 10)

The automatic question generation system considers each sentence of the paragraph and tries to create all possible questions considering the most prominent keywords.

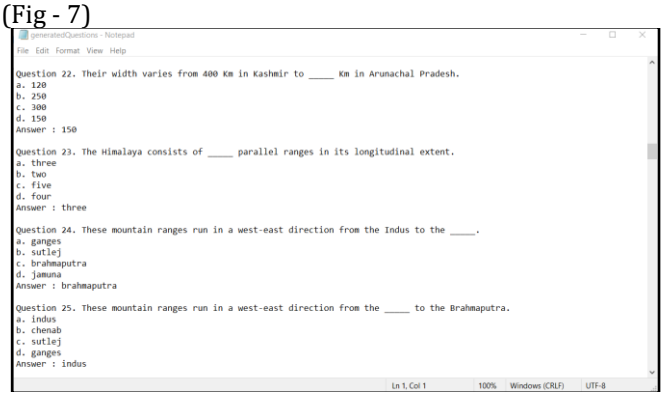


**The Himalayan Mountains**

The Himalayas, geologically young and structurally fold mountains stretch over the northern borders of India. These mountain ranges run in a west-east direction from the Indus to the Brahmaputra. The Himalayas represent the loftiest and one of the most rugged mountain barriers of the world. They form an arc, which covers a distance of about 2,400 Km. Their width varies from 400 Km in Kashmir to 150 Km in Arunachal Pradesh. The altitudinal variations are greater in the eastern half than those in the western half. The Himalaya consists of three parallel ranges in its longitudinal extent. A number of valleys lie between these ranges. The northern-most range is known as the Great or Inner Himalayas or the *Himadri*. It is the most continuous range consisting of the loftiest peaks with an average height of 6,000 metres. It contains all prominent Himalayan peaks.

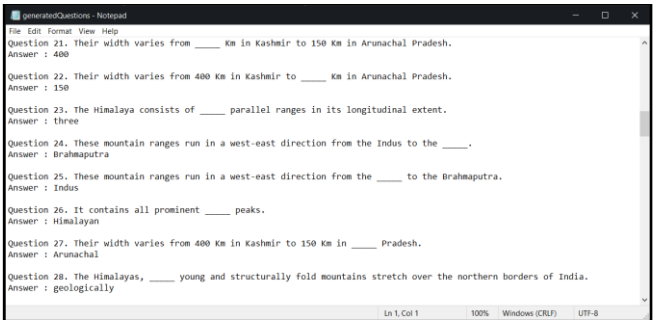
**Fig - 7 :** Input Text

The following figures show the different questions generated by the system on the text from Class 9 NCERT Geography textbook, Chapter 2. Physical Features of India. (Fig - 7)



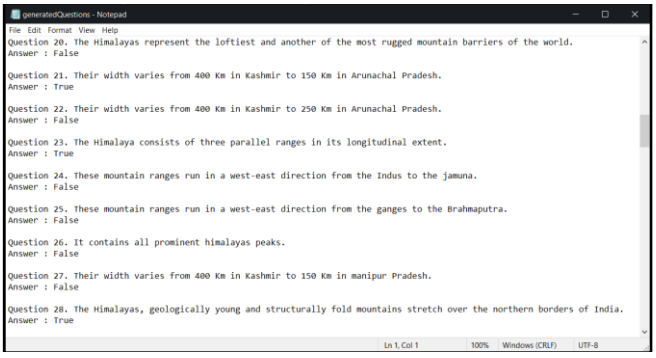
```
generatedQuestions - Notepad
File  Edit  Format  View  Help
Question 22. Their width varies from 400 Km in Kashmir to ____ Km in Arunachal Pradesh.
a. 120
b. 250
c. 300
d. 150
Answer : 150

Question 23. The Himalaya consists of ____ parallel ranges in its longitudinal extent.
a. three
b. two
c. five
d. four
Answer : three

Question 24. These mountain ranges run in a west-east direction from the Indus to the ____.
a. ganges
b. sutlej
c. brahmaputra
d. jamuna
Answer : brahmaputra

Question 25. These mountain ranges run in a west-east direction from the ____ to the Brahmaputra.
a. indus
b. chenab
c. sutlej
d. ganges
Answer : indus
```

**Fig - 8 :** Generated MCQ Questions



```
generatedQuestions - Notepad
File  Edit  Format  View  Help
Question 21. Their width varies from ____ Km in Kashmir to 150 Km in Arunachal Pradesh.
Answer : 400

Question 22. Their width varies from 400 Km in Kashmir to ____ Km in Arunachal Pradesh.
Answer : 150

Question 23. The Himalaya consists of ____ parallel ranges in its longitudinal extent.
Answer : three

Question 24. These mountain ranges run in a west-east direction from the Indus to the ____.
Answer : Brahmaputra

Question 25. These mountain ranges run in a west-east direction from the ____ to the Brahmaputra.
Answer : Indus

Question 26. It contains all prominent ____ peaks.
Answer : Himalayan

Question 27. Their width varies from 400 Km in Kashmir to 150 Km in ____ Pradesh.
Answer : Arunachal

Question 28. The Himalayas, ____ young and structurally fold mountains stretch over the northern borders of India.
Answer : geologically
```

**Fig - 9 :** Generated Fill in the Blanks Questions



```
generatedQuestions - Notepad
File  Edit  Format  View  Help
Question 20. The Himalayas represent the loftiest and another of the most rugged mountain barriers of the world.
Answer : False

Question 21. Their width varies from 400 Km in Kashmir to 150 Km in Arunachal Pradesh.
Answer : True

Question 22. Their width varies from 400 Km in Kashmir to 250 Km in Arunachal Pradesh.
Answer : False

Question 23. The Himalaya consists of three parallel ranges in its longitudinal extent.
Answer : True

Question 24. These mountain ranges run in a west-east direction from the Indus to the jamuna.
Answer : False

Question 25. These mountain ranges run in a west-east direction from the ganges to the Brahmaputra.
Answer : False

Question 26. It contains all prominent himalayas peaks.
Answer : False

Question 27. Their width varies from 400 Km in Kashmir to 150 Km in manipur Pradesh.
Answer : False

Question 28. The Himalayas, geologically young and structurally fold mountains stretch over the northern borders of India.
Answer : True
```

**Fig - 10 :** Generated True or False Questions

## 5. CONCLUSION

In this paper, we presented one possible approach to generate questions automatically from a given text. As discussed above, different algorithms are used to overcome the different phases in the automatic question generation system. It is highly probable there can be other efficient methods to go around with this problem, but with this method, we can systematically create questions with high accuracy, i.e. the questions created will make perfect sense.

## 6. ACKNOWLEDGEMENT

The successful completion of our project required the guidance along with the assistance from various people and we are extremely privileged to have got that. We sincerely thank our project guide Prof. Virendra Pawar who took keen interest in our project work.

## REFERENCES

[1] What is OCR : https://www.necc.mass.edu/wp-content/uploads/accessible-media-necc/uncategorized/resources/What-is-OCR.pdf#:~:text=OCR%20stands%20for%20%22Optical%20Character,accessible%20electronic%20version%20with%20text.

[2] Tesseract Documentation : https://tesseract-ocr.github.io/

[3] How to write a Spelling Corrector: https://norvig.com/spell-correct.html

[4] The Levenshtein Distance Algorithm : https://dzone.com/articles/the-levenshtein-algorithm-1

[5] Categorizing and POS Tagging with NLTK Python : https://medium.com/@muddaprince456/categorizing-and-pos-tagging-with-nltk-python-28f2bc9312c3

[6] J. Memon, M. Sami, R. A. Khan and M. Uddin, "Handwritten Optical Character Recognition (OCR): A Comprehensive Systematic Literature Review (SLR)," in IEEE Access, vol. 8, pp. 142642-142668, 2020, doi: 10.1109/ACCESS.2020.3012542.

[7] What is Tokenization in NLP? : https://www.analyticsvidhya.com/blog/2020/05/what-is-tokenization-nlp/

[8] The Quick Guide to SQuAD : https://towardsdatascience.com/the-quick-guide-to-squad-cae08047ebee

[9] SpaCy Documentation : https://spacy.io/usage

[10] "Algorithm for Generating Questions from Text." ukdiss.com. 11 2018.

[11] Kurdi, G., Leo, J., Parsia, B. et al. A Systematic Review of Automatic Question Generation for Educational Purposes. Int J Artif Intell Educ 30, 121–204 (2020).

[12] Kaviani, Pouria & Dhotre, Sunita. (2017). Short Survey on Naive Bayes Algorithm. International Journal of Advance Research in Computer Science and Management. 04.

## BIOGRAPHIES

**Pranjal Rane**

Computer Engineering Student - Vishwakarma Institute of Technology, Pune

**Ritvik Patil**

Computer Engineering Student - Vishwakarma Institute of Technology, Pune

**Ojas Natu**

Computer Engineering Student - Vishwakarma Institute of Technology, Pune

**Prahlad Pore**

Computer Engineering Student - Vishwakarma Institute of Technology, Pune

**Pranay Shridhar**

Computer Engineering Student - Vishwakarma Institute of Technology, Pune