Automated question and test-paper generation system

Neeti Vyas, Harsh Kothari, Akshay Jain and Abhijit R. Joshi*

Department of Information Technology,
Dwarkadas J. Sanghvi College of Engineering,
Mumbai, India
Email: neetivyas1806@gmail.com
Email: harshkothari4444@gmail.com
Email: akshay58.aj@gmail.com
Email: abhijit.joshi@djsce.ac.in
*Corresponding author

Abstract: This paper focuses on a very simple yet an issue of utmost importance in today's fast-paced life. Today everyone desires to get their work done on a click, so why not automate the time-consuming process of question generation. A text document is given as input and our system will generate possible set of questions on it. The categories of questions range from MCQ questions, objective questions, factual and inference-based questions. NLP-based tools are used for automating the process of question generation. Here focus is on three key areas, first being POS tagging followed by pronoun resolution and summarisation. Once the text is resolved and summarised, then questions are generated based on it. In this system, one just needs to give the text document as input and gets the questions on the go. This system will be very beneficial for school teachers and students.

Keywords: question generation; test-paper generation; NLP; discourse connective; distractor; pronoun resolution; summarisation; pos tagging; factual and inference-based questions; MCQ questions; objective questions.

Reference to this paper should be made as follows: Vyas, N., Kothari, H., Jain, A. and Joshi, A.R. (2022) 'Automated question and test-paper generation system', *Int. J. Computer Aided Engineering and Technology*, Vol. 16, No. 3, pp.362–378.

Biographical notes: Neeti Vyas is a final year student pursuing engineering in the field of information technology from Dwarkadas J. Sanghvi College of Engineering, affiliated to Mumbai University, India. She has keen interest in the field of automating the existing processes and doing research work on it. The field of artificial intelligence and machine learning is one of the areas of her interest. Drawing automated solution for manual day to day processes and trying to solve them on a click is her area of focus. Exploring cloud related solutions on platforms like Google Cloud Platform and Amazon Web Services is also the domain on which she researches.

Harsh Kothari is a final year student pursuing engineering in the field of information technology from Dwarkadas J. Sanghvi College of Engineering, affiliated to Mumbai University, India. He has keen interest in researching and implementing various possible applications in artificial intelligence and cloud computing world. Automating tedious tasks is his hobby. In artificial

intelligence, he has mainly worked in natural language processing. He likes researching on services offered by Amazon Web Services and Google Cloud Platform.

Akshay Jain is a final year student pursuing engineering in the field of information technology from Dwarkadas J. Sanghvi College of Engineering, affiliated to Mumbai University, India. He has interest in transforming manual processes to automated ones. DevOps is an area in which he is currently working, using Jenkins tool to automate the manual process. He also likes exploring the cloud solutions like Amazon Web Services and big data like Hadoop ecosystem.

Abhijit R. Joshi completed his PhD from NMIMS University, Mumbai, India in the area of e-learning. Currently, he is working as a Professor at Dwarkadas J. Sanghvi College of Engineering, Mumbai, India. He has more than 20 years of teaching experience in the field of computer engineering and information technology, in various capacities at graduation as well as post-graduation level in disparate universities, namely, SNDT, Mumbai, NMIMS and Kingston (UK).

1 Introduction

Questioning gives an answer for the curiosity within mankind but generating good questions is an art. Generating questions for a lengthy text document is a tedious and time-consuming process. If this process could be automated, it could save a lot of time and efforts. In this computerised world, any tedious manual process could be automated. A lot of data is available on World Wide Web, but the questions for the available data need to be generated after reading the content. Questions are not available directly like any text information. For generating questions, a huge amount of time needs to be spent on reading the text, analysing the content and then deciding, which are the suitable questions for the content to test human knowledge. Thus, questioning is an important tool to assess the amount of information grasped and understood by the person. Automated generation of answers for the given question is available through various applications but the reverse process of generating questions is a tedious process and still not supported and available through applications. Along with generating questions on the go, if the final test paper generation could also be automated, then it would be like an added advantage. But as question generation is still an area of research in NLP, automation of the question generation is a bit complex. So, here we present a system that would help us achieve the above said benefits to some extent. We have focused on the automation of question generation and then finally generation of the final test paper. Automating the process of question generation would also promote paper-less use of technology, which would further ensure better security. Here we are targeting generation of questions, for which the user just needs to input the text document on which he/she wants to generate the questions. Once the text document is given as input to the system, the user gets the generated questions on the go. We have provided an option to select the questions the user wants in the final test-paper. Thus, helping in generation of the final test paper. The test paper can then be downloaded and used. This reduces a lot of effort for test paper generation for the teachers and also huge variety of papers can be generated.

The rest of the paper is organised as follows. Section 2 gives a gist of the existing work, which focuses on the work carried out in this area and the work left to be carried out. In Section 3, the approach followed to realise the system is discussed. The major challenges in the design and implementation of the system is highlighted along with the design and implementation of our system to address those challenges. Section 4 gives a walkthrough of the system from user's view that provides an idea of the effectiveness of the system. The paper ends with conclusion and a lot of ideas to carry the work forward.

2 Literature review

This section highlights the research carried out by the authors. It starts with the survey conducted by the authors to analyse the need of the system in today's generation. This is followed by a brief preface to the work carried out in this area along with the areas left to be targeted. The section ends with observations on the existing work carried out in this area.

2.1 Survey

A pre-survey was conducted to analyse the need of automated question generation system in today's world. We had collected the results from school teachers and students. From the survey results, we were able to get deeper insights into the area of question generation and its use in today's world. Through online survey we obtained the following results:

- 53.4% students feel that most questions are repetitive and lack curiosity
- also 67.2% students feel that it would be very beneficial for exam preparation
- mostly all teachers find that setting different questions according to student's IQ level is the biggest challenge.

2.2 Existing system study and observations related to them

1 *Generation of factual questions using over generating transformations and ranking:* Pichponreay et al. (2016) have discussed generation of factual questions. They used optical character recognition (OCR) technique to extract the contents of any given document into plain text form. The system extracts declarative sentences from the plain text and generates questions on it. Then the questions are ranked to select the appropriate ones. If we have a sentence like, 'Ram shifted to Mumbai in 1997'. Questions generated by the system are as follows:

   - Where did Ram shift in 1997?
   - When did Ram shift to Mumbai?
   - What shifted to Mumbai in 1997?

   Looking at the above questions, there is an error in the last one as it should be 'who' and not 'what'. This system cannot deal with named entity segregation.

2 *Generating questions from a paragraph using named entity recognition:* Swali et al. (2016) have generated mostly factual (WH type) questions by selecting first the

important sentences and then generated questions on those sentences. Here, they have resolved the problem of named entity recognition. But if a sentence likes: 'Ram is a good boy. He did not go to school because he was ill' is given, then the question would be: 'Why he did not go to school?' which is very vague. Since 'he' does not specify the person 'Ram' in this case. So, even though it takes into account discourse connective like 'because' it cannot generate an accurate question.

3 *Generation of multiple choice questions (MCQ) from a paragraph:* Araki et al. (2016) focus on generating MCQ questions by using co-reference resolution and paraphrase detection techniques. They have used the relationship between various entities and events during the process of question generation. The major drawback of their work is the system will not generate MCQ for any random text because of predefined data source to generate the distractor but they are not dynamic.

4 *Generation of descriptive and factual questions from a given set of sentences:* Jethwani et al. (2015) focus on breaking down the paragraph and then generate questions based on it. This system is capable of generating descriptive questions along with factual ones. It can handle some complex sentences, not all. This system uses a set of sentences as input to generate question, e.g., Sita is a good girl but Sita always fights. So, in this case, the system divides this sentence into:

   a Sita is a good girl.

   b Sita always fights.

   Using these two sentences, the system generates the questions for (a) would be 'Who is a good girl?' and the question for (b) would be 'Describe Sita?'.

   The issues of this technique are in handling the complex sentences having more than one discourse due this it cannot take input as a whole document. Also, it cannot generate other than factual type of questions.

5 *Generation of questions with the help of discourse (conjunctions) connective approach and domain dependent distractor-based approach for MCQ (cloze) questions:* Shah (2012) discussed about the methods of question generation, one based on discourse connectives and other based on distractor selection. The method based on discourse connectives focuses on generating questions for a set of selected discourse connectives. Here the author has considered the following discourse connectives they are as follows: 'since', 'although', 'because', 'for e.g.', 'for instance', 'as a result' and 'when'. The distractor selection method for cloze questions is based on a domain dependent approach, which makes use of a knowledge base to generate questions with suitable distractors. If the text does not contain the above-mentioned discourse connectives, then the system will not generate any kind questions.

None of the systems discussed above deal with the generation of different categories of questions like factual, MCQ and objective type of questions together. The existing system discussed earlier catering to specific type of questions only and having certain issues as pointed out previously. In our project work, we are considering all these issues in the development of the system, which will be dealt with all types of questions. Now, let us see our approach to implement the system.

3 Approach

The system takes a text document as input and generates questions as per the user requirement. The system generates three types of questions, which are: 'MCQ-type', 'fill in the blank type' and 'answer the following type' of questions.

The system converts input document to string format. The generated string is then segmented into sentences and the sentences are further segmented into words. Words are then tagged into their respective parts of speech using POS tagging. Words that form a compound word are combined into a single token. For e.g., words like United, States, of, America are combined into a single word, i.e., 'United States of America' and then a token is assigned for this word. Once the words are combined, then named entity recognition is performed on them. Further, pronoun resolution is applied on the text, which is then summarised. The purpose of pronoun resolution is to generate a better summarised text helping in generating meaningful questions.

The pronouns mapping to their respective proper nouns that occur in the previous sentences is achieved by following backward chaining approach. The discourse connectives like 'but', 'because', 'since', etc. are identified using basic grammar rules. After identifying the discourse connectives, the arguments for the question formation are identified. The suitable W-H word is attached to the argument for question formation, e.g., consider the sentence, 'we should work hard because it gives us great results'. In this case, the system identifies discourse and here it is 'because'. Then the argument suitable for question formation is extracted. Here there are two arguments:

arg1 'we should work hard'

arg2 'it gives us great results'.

For discourse like, 'because' arg1 is selected and the W-H word for this discourse is 'why'. So, the system generates the question and it is 'Why should we work hard?'. This approach of discourse connectives is used for generation of factual and same is followed for generating inference-based questions.

For objective type questions, important words are identified from the document with the help of summarisation and then questions are generated on them. Here, important words mean the ones those occurring more frequently and proper nouns.

For MCQ type of questions, POS tagging approach and data dictionary are used for better distractor (option) selection. The distractors are selected dynamically for every time for generated question on the given input text avoiding the repetition of same distractors (options). This helps in overcoming the rote-learning by students and promotes students to understand the concepts instead.

3.1 Major challenges

In the realisation of the system the major challenges are:

1 *Generating sensible questions appropriate for the given content:* the text consists of many sentences; selection of appropriate sentences for question generation is important. The sentences consist of lot of named entities. Some named entities may refer to living things and some might denote non-living things. There might be some named entities which can be confused with its category. So, the question generating system might sometimes ask 'what-type question' with living thing and 'who-type

question' for non-living thing, e.g., there can be a girl named 'Jasmine' and also there is a flower named 'jasmine'. So, asking a question like – 'What is jasmine?' for a girl and 'Who is Jasmine?' for flower will sound incorrect and insensible.

2 *Understanding the meaning of the phrase and word with respect to the given context:* sometimes the text consists of phrases which denote different meaning for different content, so making questions after understanding the meaning of the phrase with respect to content is challenging. Also, there are words which have multiple meanings, so referring to the meaning of the word with respect to the given context is challenging, e.g., the word 'watch' has two meanings. When we say, 'Watch out for any errors', here 'watch' refers to verb, whereas when we say, 'She is wearing a watch', here 'watch' refers to a noun. When we construct a question, we look for the verb in the sentences to frame it. In this case, for the second sentence, one cannot consider watch as noun and generate a question like 'What is watch wearing?'. It becomes an insensible question.

3 *Appropriate mapping of question-type with the particular sentence:* question types are important during construction of a question for complex sentences containing discourse connectives, namely, 'because', 'since', etc. When we select a sentence on the basis of the discourse connective, we assign it a question type such as 'why', 'what', etc. Different discourse connectives refer to different question type. Finding the exact question type for an unknown discourse connective is a challenge. If there is more than one discourse in the sentence, which question type it should refer to while generating a question.

4 *Selection of distractors which are neither irrelevant nor too easy to recognise in cloze (MCQ-type) questions:* distractors are the options that one can provide in MCQ type questions. Distractor selection is a challenging task. The selected distractors should be very relevant to the context of the given sentence, so that they are difficult for the user to guess. The distractors should not be some stop words or irrelevant words. They should also not be too easy and obvious to guess.

5 *Deciding the relevance and importance of a particular question:* the system will generate a lot of questions from declarative sentence approach or based on discourse approach for every sentence that contains a discourse connective. So ranking the questions according to their relevance with respect to the particular text is challenging. We also need to avoid questions which are repetitive but asked in different sense.

6 *Generating questions which are grammatically sound and correct:* questions generated from the system should not only be syntactically correct but also meaningful. The adherence to the grammar rules is not the only requirement. The questions should also be meaningful. So, designing a domain model that follows legal and meaningful questions is a challenging task. Developing a rich knowledge base for inference-based questions proves to be a challenge.

7 *Making the system more adaptive:* developing the system in a way so that it becomes adaptable to future changes and requirements. Considering the future scope of generating questions on text based on any language is considered in this case. Implementing and designing the framework in a way that it would be able to

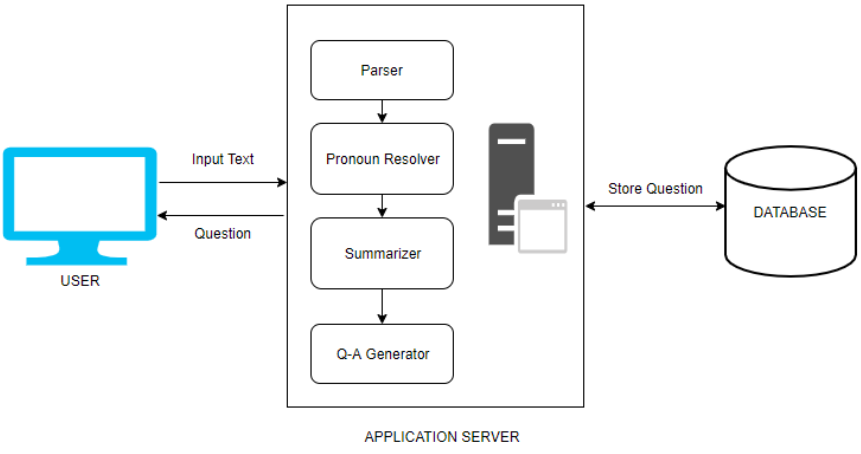
generate questions for text based on any language. The framework of the system is constrained to its language rules. So, incorporating the different language rules and integrating it into one framework is a challenging task. Different versions and adaptability with different software tools and changes in needs from time-to-time.

Let us see now the architecture of our system.

3.2 Design and implementation

The basic architecture of the system is shown in Figure 1. It describes the various components of the system and the interconnection between these components along with their role.

Figure 1 System architecture (see online version for colours)



- *User:* user can be anyone, who wants to avail the benefits of our system-automated question generation. User inputs the text on which question needs to be generated. The text will be then sent to the next component i.e. application server for further processing.
- *Application server:* this component is responsible for generating the questions for the given input text. It generates questions with the help of various sub modules present within it. The sub modules are the parser, the pronoun resolver, the summariser and the Q-A generator. Application server first accepts the text sent by the user and performs sentence and word segmentation with the help of 'spacy' library. Then it passes the segmented words to the parser.
- *Parser:* each segmented sentence is further processed by the parser. The parser tags the segmented words to named entities. Parts of speech of the individual words of the sentences are tagged with the help of the parser.

  E.g., Car is a vehicle.

  POS tagged sentence: Car (NN), is (VBZ), a (DT), vehicle (NN).

Here NN stands for noun, VBZ stands for helping verb, DT stands for determiner. This POS tagged sentences are then sent to pronoun resolver for co reference resolution.

- *Pronoun resolver:* this module receives the POS tagged sentences from the parser. The pronoun resolver replaces pronoun with its preceding proper noun to generate better summary of the input text. It uses the 'neural-coref' library, which is built on top of 'spacy' for pronoun resolution. The pronoun resolved sentences are then sent to the summariser module.
- *Summariser:* since the paragraph contains many sentences it is important to consider only the important sentences for generating questions. The summariser condenses the text so that question will be generated on only important content of the text. It eliminates the unnecessary sentences leaving a text containing only the important sentences on which questions are to be generated. The summariser is implemented using genism. The summarised text is then passed to the Q-A generator module.
- *Q-A generator:* this module applies various discourse connective rules, distractor selection rules and named entity recognition rules for generation of question.

  E.g., on identifying the discourse connective like 'because' in the sentence, it targets the arg1 of the sentence for the question generation purpose. If a sentence like: 'Ram missed the bus because he was late' is given, then the system identifies the discourse 'because'. Further it splits the sentence into its two arguments:

  arg1 'Ram missed the bus'

  arg2 'he was late'.

  Since for a discourse like 'because' for question formation arg1 is considered and the question type for because is 'why'. So, the final question generated is 'Why did Ram miss the bus?'.

  On encountering a named entity which belongs to person category it generates a W-H question starting with 'WHO', e.g., if a sentence like: 'Ram is an intelligent boy' is given, then the system tags the entity 'Ram' as 'person' so the question type for a person would be 'who' and the generated question is 'Who is an intelligent boy?'. For a sentence like: 'Ram stays in Mumbai' is given, then the entity 'Mumbai' is tagged as 'place' and the question-type for place is 'where'. The generated question is: 'Where does Ram stay?'. For distractors, data dictionary is used to find the suitable options for a given sentence. Here, discourse, named entity and distractor are identified and after that sentence formation rules of grammar are applied for question generation.
- *Database:* the generated questions from the Q-A module are then stored in the database for future retrieval.

4 Results and analysis

In this section, we start with the walkthrough of the system as a user and then analyse the system to get deeper insights into the results. Then results and their analysis are provided

to get an idea of the effectiveness of the system. Experiments were carried out for testing the smart question generator with following types of text:

- text containing figures of speech like poems
- news articles
- general descriptive texts
- analytical texts.

4.1 Walkthrough of the system

When the user navigates to the smart question generation website, they are directed on the Homepage as seen in Figure 2. The homepage provides the user the get started option through which the user can navigate to the textbook section page. Also, the textbooks can be accessed through the textbook option.

Figure 2 Homepage (see online version for colours)

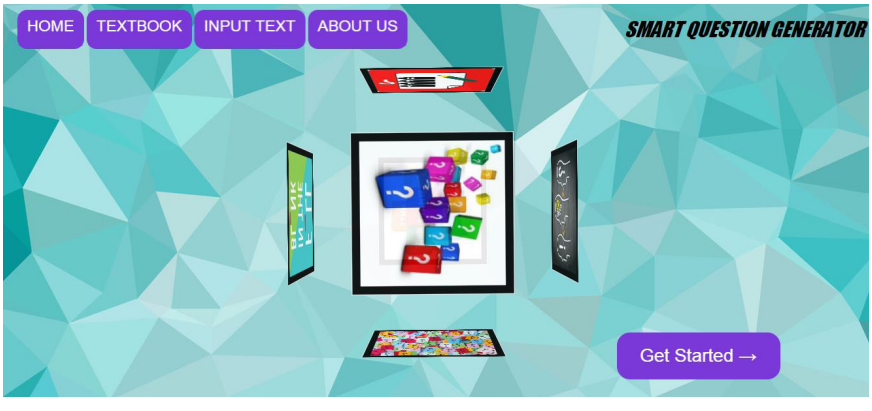
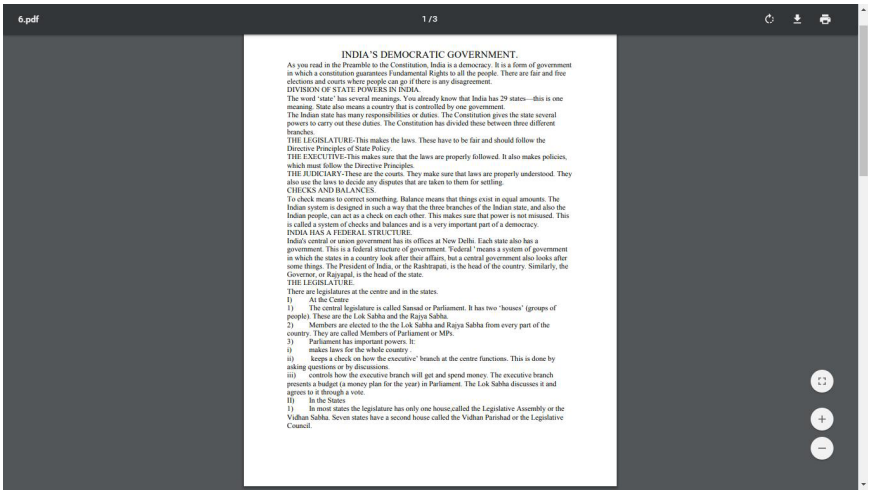


Figure 3 Textbook section page (see online version for colours)

After clicking on get started or textbook option, the user navigates on the textbook section page as shown in Figure 3. The back option takes the user back to the Homepage as shown in Figure 2.

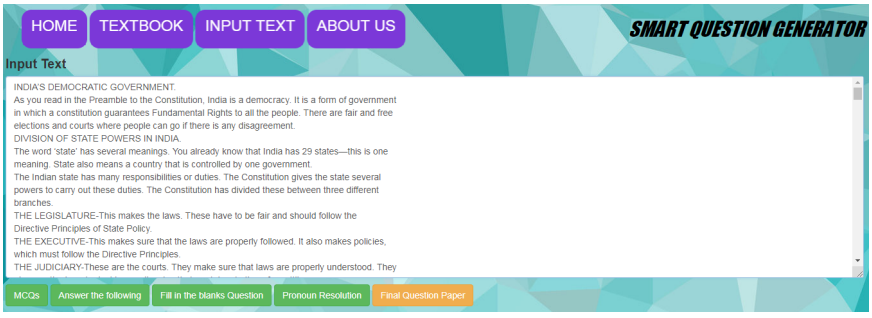
Here, social studies textbook of grade 5 ICSE is available. The user can select any desired chapter on which he/she wants to generate questions. On clicking the chapter, the full text option opens the PDF of that chapter as seen in Figure 4.

Figure 4 PDF of the selected chapter



The user then needs to select the text on which he wants to generate the questions. Just for the ease of use and demonstration purpose we have made use of textbook of grade 5. The user can select any text on which he desires to generate questions and he selects it from any source. The only limitation here is the text should not contain figures, mathematical formulas and it should be grammatically correct. After selecting the text, the user needs to click the input text option to navigate to the input text page as shown in Figure 5.

Figure 5 Input text page for providing the text you want to generate questions on (see online version for colours)

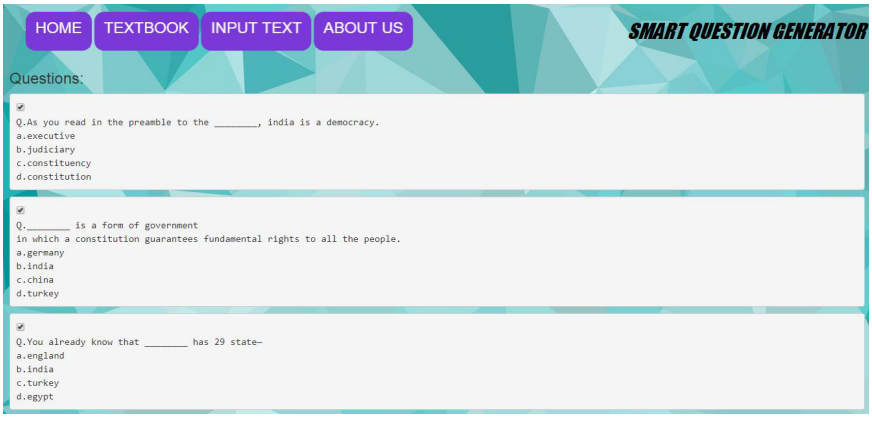


Here the user is provided with three options to generate questions on. The 'MCQ' options helps to generate MCQ type of questions from the given text document. The 'Answer the following' option helps to generate factual and descriptive questions. The 'Fill in the blank' option helps to generate objective type of questions. Further, a 'Pronoun

resolution' option is provided to generate a pronoun resolved summary for given text document.

After clicking on MCQ option, MCQ questions are generated as shown in the Figure 6.

Figure 6 MCQ questions are generated by clicking on MCQ option (see online version for colours)

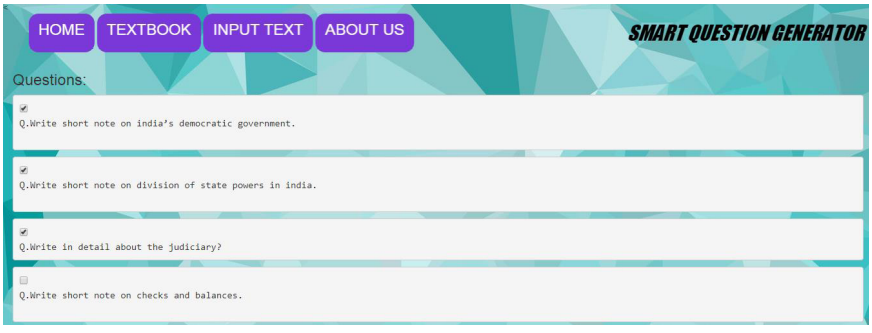


Here an option to edit the questions and also to select the ones he/she wants to include in his/her final paper is provided. In Figure 6, questions 1, 2, 3 are selected. Figure 7 displays the selected and edited question of Figure 6.

Figure 7 The selected MCQ questions for the final test paper along with an edit option to change them (see online version for colours)

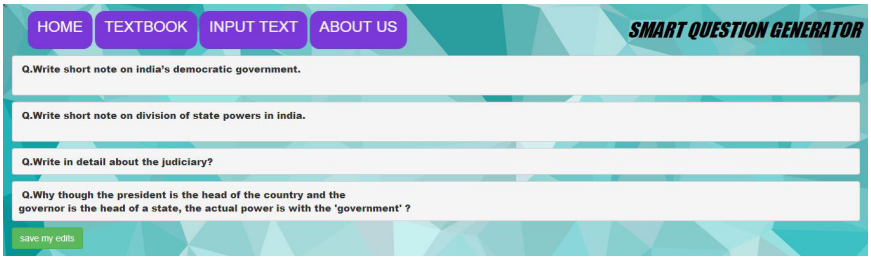
On clicking on 'Answer the following' option, factual and descriptive questions are generated as shown in Figure 8.

Figure 8 Factual and descriptive questions generated by clicking on answer the following option (see online version for colours)



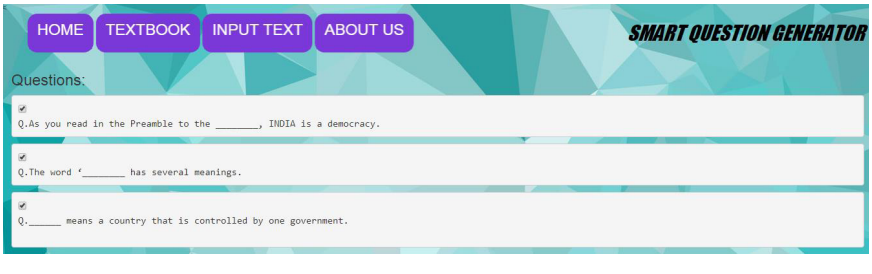
Here an option to edit the questions and also to select the ones he/she wants to include in his/her final paper is provided. Figure 9 displays the selected and edited question of Figure 8.

Figure 9 The selected factual and descriptive questions to include in test paper along with an edit option to save changes (see online version for colours)



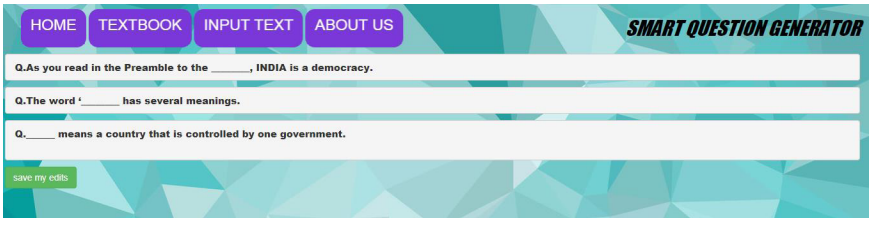
After clicking on 'Fill in the blank' option of Figure 5, objective questions are generated as shown in Figure 10.

Figure 10 Objective questions generated by clicking on FIB option (see online version for colours)



In this case also an option is provided to edit the questions and also to select the ones he/she wants to include in his/her final paper. In Figure10, questions 1, 2, 3 are selected. Figure 11 displays the selected and edited question of Figure 10.

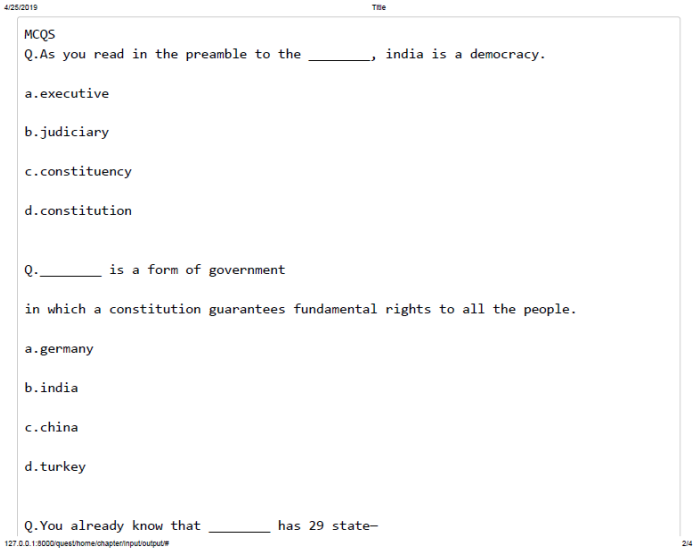
Figure 11 The selected objective questions to include in test paper along with an edit option to save changes (see online version for colours)



On clicking the 'Final Question Paper' option as seen in Figure 5, the final question paper is displayed which is shown in Figure 12.

Figure 12 The final test paper generated from the selected questions (see online version for colours)

Figure 13 The PDF of the generated test paper



One can save the PDF of this paper as seen in Figure 13 and can use it for test and examination purpose.

4.2 Observations on the results

Smart question generator takes a text document as input and generates questions based on it, without the need to read the whole document. A very simple system but with a great impact on easing out the work of the teachers. Our system has touched upon the research domain currently going on in the field of question generation. The results were carried out on a wide variety of text input. We have conducted test on the generation of MCQ questions, objective questions and subjective questions. Also, we have provided pronoun resolution module for further inference of the text. Based on the tests conducted, our system was able to generate wide range of questions such as MCQ, objectives and factual (subjective) questions. If the text contains some grammatically incorrect sentences, then those were displayed incorrectly. Some distractors of MCQ appear to be quite simple to decipher. Our system cannot target the areas of generating questions on poems and diagram-based text documents, it fails in generating questions in such cases. Technical subjects and table-based text questions also remains an area to be touched upon.

4.3 Analysis

Our system correctly generates questions like MCQ, subjective and objective on grammatically correct text documents. Our system is not able to generate questions on poems, technical subjects, diagram-based text document and table-based questions as said earlier. To ensure the usefulness of the system and correctness of the output generated by it, we had carried out surveys. Two surveys were conducted pre-survey before implementing our system and post-survey after development of the final product.

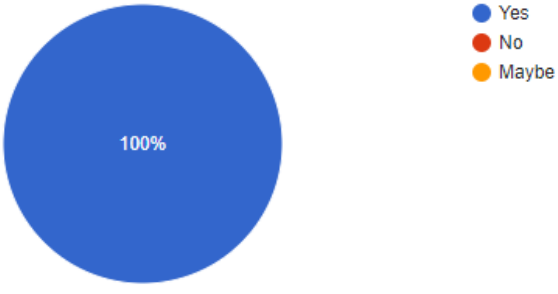
We mentioned earlier about the reason of pre-survey. Post-survey was carried out for the purpose of our system verification. School teachers were asked to give the text document as input to our system for generating questions.

From the survey results (see Figures 14, 15 and 16) we can say that our system is very apt in today's education system. Also, they would like to use our system in their schools.

The following questions were asked to the school teachers for which we obtained the following results:

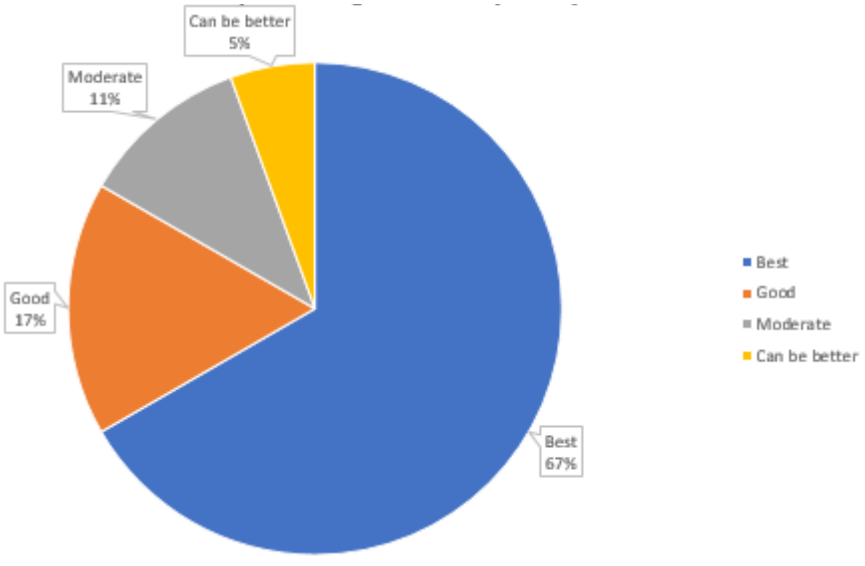
Question 1 Would a system like this be beneficial to be used by your school?

Figure 14 Beneficial for school (see online version for colours)



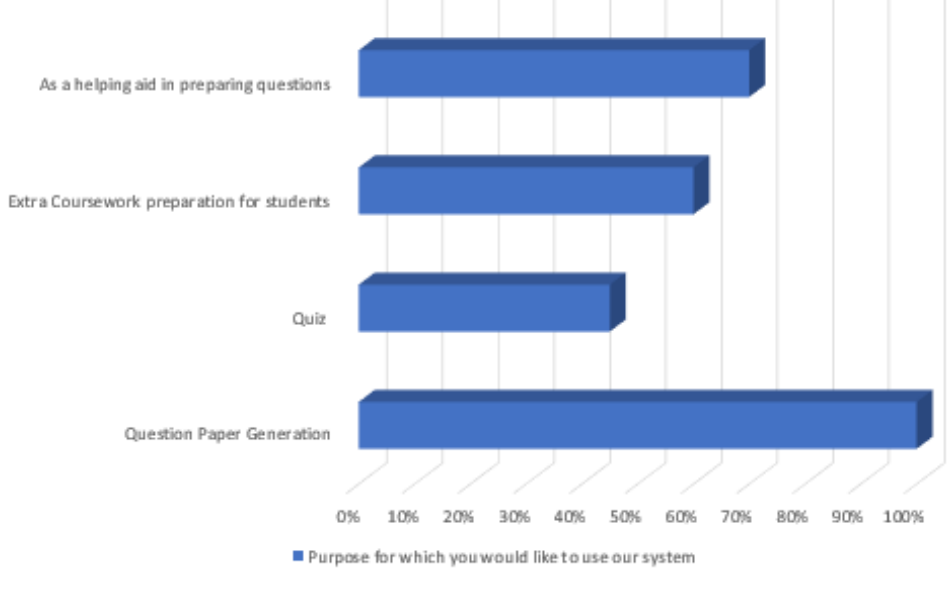
Question 2 Rate the questions generated by our system on the scale of [1 (best) – 4 (can be better)]

Figure 15 Rating the questions generated by our system (see online version for colours)



Question 3 The purpose for which you would like to use our system.

Figure 16 Purpose for which the teacher would use our system (see online version for colours)



5 Conclusions and future scope

The section provides the system in a gist and ideas to carry out the future work in this area.

5.1 Conclusions

The smart question generator system accepts text as input and generates meaningful questions. To do so the system first resolves the pronoun and summarises the text, and then generates questions on this summarised text. Our system not only generates factual questions but also generates inference-based, MCQ and descriptive questions. The system generates questions only on theory subjects like English and social science. Our system can assist both teachers and students. By generating good questions through our system, the teacher can then use those questions to evaluate student's performance and students can use it for examination preparation. The current process of teaching-learning can be made more interesting and dynamic by using smart question generator keeping in mind that the assessment for learning. Finally, we can conclude that our system is like an answer to all the question generation issues.

5.2 Future scope

Our system is based on text-based question generation approach. Keeping this notion in mind, our system is developed. Our system is tested for wide variety of text inputs and suitable MCQ, objective, subjective and inference-based questions are generated on the given text document.

The focus has been on developing a powerful framework for active and interesting teaching-learning process. This has been implemented and we have results showing the same. However, a lot of work remains to be done to complete this journey. Some of these are, generating questions on diagrams because it will require image processing on given image or diagram. Also, our system will not generate question for text which consist of mathematical equation, because it requires all formulas and notation to be stored in knowledge base. To generate such type of question it will also require lot of data to be processed and trained. Also, our system cannot generate analytical questions, e.g., if there is a graph, pie chart and pie graph present in the document, it will not generate questions on them. Generation of technical interview questions based on technical subjects like Python, DBMS, data structures, etc. is considered as a future scope of our project. Our system can only generate questions on English language. The scope can be extended to include languages like French, Spanish, Indian languages, etc.

References

Araki, J., Rajagopal, D., Sankaranarayanan, S., Holm, S., Yamakawa, Y. and Mitamura, T. (2016) 'Generating questions and multiple-choice answers using semantic analysis of texts', *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pp.1125–1136.

Jethwani, H., Husain, M.S. and Akbar, M. (2015) 'Automatic question generation from text', *International Journal for Innovations in Engineering, Science and Management*, Vol. 3, No. 4, pp.27–31.

Pichponreay, L., Kim, J-H., Choi, C-H., Lee, K-H. and Cho, W-S. (2016) 'Smart answering chatbot based on OCR and overgenerating transformations and ranking', *2016 Eighth International Conference on Ubiquitous and Future Networks (ICUFN)*.

Shah, R. (2012) *Automatic Question Generation using Discourse Cues and Distractor Selection for Cloze Questions*, MS by research thesis, Language Technology and Research Center (LTRC) International Institute of Information Technology Hyderabad – 500032, India.

Swali, D., Palan, J. and Shah, I. (2016) 'Automatic question generation from paragraph', *International Journal of Advance Engineering and Research Development*, Vol. 3, No. 12, pp.73–78.