International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 07 Issue: 06 | June 2020 www.irjet.net p-ISSN: 2395-0072

AUTOMATED QUESTION GENERATOR SYSTEM USING NLP LIBRARIES

Priti Gumaste[1], Shreya Joshi[2], Srushtee Khadpekar[3], Shubhangi Mali[4]

[1-4]BE Computer Engineering, Sandip Institute of Technology and Research Centre, Nasik, Maharashtra, India.

Abstract - *Automatic generation of questions from text plays an integral role in various domains and is a highly growing topic when one needs to apply automation. The input text is broken into tokens and questions are generated. This paper states a standard methodology for automated question generation. It focuses on the examination of both semantic and syntactic format of a sentence. Various NLP libraries such as Spacy, NLTK are used for tokenization, stemming, lemmatization, punctuation. Part of Speech tagging is one of the most significant parts of natural language processing which helps us to find the proper tag for given text. Using these NLP libraries the proposed system performs effectively utilizing paragraphs of text as input. The system can be used by schools, colleges, coaching institutions for generating Question paper and also by students for self-assessment. The system can also be used by universities to create question paper for students.*

Key Words*: *Automatic Question Generation, Spacy, NLTK, POS tagging, NLP.

1. INTRODUCTION

Humans are curious in nature. We ask questions to gain knowledge and adapt the knowledge in our daily lives. Questions are primary source of learning. Large counts of learners are bad at posing inquiries. Dunlosky and Graesser states that students have an issue with distinguishing their own insight about the knowledge they have. Therefore, they tend to ask fewer questions. Automation of question generation systems can assist learners to discover the levels of their expertise and increase their knowledge by helping them in clarifying their queries. It can also be widely used in various other fields such as forplacement activities. It can be used by scholars for generating questions. Educational institutions can make use of this system making the evaluation system of students more transparent and easy. FAQs can be generated for various applications, courses and for documentations.

Automation of question generation systems help automated question answering systems such as IBM Watson to perform self-training (IBM Watson Ecosystem, 2014).[1] Rather than depending on human specialists to manually define ground truth answers from the queries, automatic question generation systems automatize this process. Intelligent tutoring systems can also utilize this advantage. Depending on human specialists to manually extract questions from study materials takes time and is at times tedious; rather each end user can define its own tutoring framework. At last, automated systems like this help the development of annotated datasets for question answering and reading comprehension. Question answering (QA) is a computer science discipline within the fields of information retrieval and natural language processing (NLP), which is concerned with building systems that automatically answer questions posed by humans in a natural language.

There are three main components in a QA framework that is Question processing, passage selection and answer extraction. There is a vast variety of text data available in the question answering system and so are the problems faced by the researcher and "question understanding" is one of the problems of the QA system. Part-of-speech tagging is first level in any NLP application. The higher the quantity of computational advances required for deciding the parts of speech tag of a sentence, the higher is its intricacy. Named Entity Recognition (NER) is an essential element in NLP. Using Part-of-speech tagger we find the best possible tag for each word in a sentence and understand it properly either it is noun, verb, and adjective and so on. Out of many NLP libraries, NLTK is one of most powerful library, it contains packages which makes machine understand human language and can give appropriate response. Spacy is one of NLPs intelligent libraries, we use it for tokenization, lemmatization.

2. LITERATURE SURVEY

Onur et al suggested a rule based methodology to automatized question generation. The methodology proposed in this paper focuses on analysis of both syntactic and semantic structure of a sentence. This work mainly aims to generate more comprehensive questions by exploiting the semantic roles of words. The framework here proposes a rule based mode to generation of questions from sentence. Reliance based, Named Entity Recognition (NER) based, and semantic role (SRL) marking based layouts/rules are used. For deciding between who and what questions system has proposed a solution by using Chunking [1].

Amruta Umardand et al stated in the paper main idea of question paper generation system. It states various methods that can be approached for the generation of questions, being it a randomized or automated process. Such automation system can be of great help. Database can be added and an administrative level of security can be given to the system, wherein questions once generated can be stored and reused by another author who is willing to create questions based on same or different topics [2].

Aleena et al paper focuses on the implementation idea for a question generation system. The main idea lies in the natural

language understanding of the system, only then can the machine handle and manipulate the data. Preprocessing of data, key phase extraction and NLP are main concerns of the system proposed. In this a way a fast, secure and randomized system can be developed which is beneficial in many aspects including education [3].

Kalpana et al aims to discover answers to questions as: 'What is POS tagging?', Working of POS tagging. POS Tagging is the process of assigning one of the parts of speech to the given word. Parts of speech include nouns, verbs, adverbs, adjectives, pronouns, conjunctions and their sub-categories. Taggers use few sort of data: word references, vocabularies, rules, etc. Dictionaries have category or classifications of a specific word. We may have many words belonging to the same category. For example, escape is both noun and verb. Taggers utilize probabilistic information to resolve this equivocalness. Following the same method POS tagging is done and tags are assigned [4].

Edward Loper et al have introduced a new methodology to a streamlined and adaptable method of sorting out the practical component of computational linguistics. The NLTK toolkit provides an extensible, simple framework for processing of natural language. It covers symbolic and statistical natural language processing. The toolkit is executed as a set of modules; each module defines an alternative data structure or a task. A set of core modules characterize various systems that are used all through the toolkit. Chunk Parsing, Probabilistic parsing are few of many modules that are part of NLTK. Many of the modules present in the toolkit provide us with good instances of what projects should resemble, clean code structure, and thorough documentation [5].

Ankita, K. A. et al used intricacy of POS tagging lies in the number of computational levels required for determining POS tags for a sentence. This paper centers on the number of comparisons made by Hidden Markov Model for POS labeling and the complexity can be reduced. They have also stated an additional method using Bloom Filter for NER (Named Entity Recognition.) In spite of the fact that there are numerous POS taggers accessible, individuals are yet taking a shot to discover a way which sets aside less effort for execution and the with less number of complications as well. In the HMM bases tagger it doesn't find tag for an individual word, rather it finds tag for a sentence as a whole. It uses transition as well as emission probabilities. The algorithm proposed combines two words as a chunk and calculates the tag for them considering them as single unit [6].

Mohd Husain et al have stated various approaches to the classification of queries. Objective type, fill in the blanks or 'wh' type questions. All of them can be generated by extracting appropriate data from the text by integration and conversion. Descriptive and factual questions can be developed using the same. Although we mainly focus on 'wh' type questions in our paper [7].

Pranita Jadhav et al have stated various steps when developing a system which generates fill in the blank type of questions out of many classifications of automated generation of questions. The advantage of this AGM is to speed up the process of evaluation in the education domain. Using Euclidean distance method, POS and tokenization, they have proposed the system [8].

Bowen Xu et al formative study indicates that developers need some automated answer generation tools to extract a succinct and diverse summary of potential answers to their technical questions from the sheer amount of information in Q&A discussions. To meet this need, we propose a three stage framework for automated generation of answer summary. Our user studies demonstrate the relevance, usefulness and diversity of our automated generated answer summaries [9].

Priti et al stated in there paper there paper the generation of questions using bloom's taxonomy. Feature extraction and Stanford pos tagging are the main elements in the Pre-processing of data [10].

Surbhi et al stated system wherein questions generated were stored in database and using randomization, random questions could be picked as output [11].

Sheetal et al have stated various reviews of different papers which have proposed many different algorithms for questions generation using Semantic role labeler or NLP. Much more work can be done in the same field proposing more complex methodologies [12].

3. SPACY

Whenever we are working with a huge amount of text, we will eventually want to know more about the text. Questions like: What does the words mean in the sentence, How do they act together to give a meaningful sentence? , Which texts are similar to each other and so on. Spacy is specifically built to process and help us understand large volumes of text. Spacy framework is written in Cython, and is a quite fast library. It provides access to its techniques and functions which are instructed by AI/machine learning models. In its package spaCy contains different models which contain the information about vocabularies, trained vectors, syntaxes and entities. It provides features for many natural language tasks, can be used to build information extraction systems. These models are to be loaded into our code to access them. Following is an example of loading the default package "english-core-web":

```
>>>import spacy
>>>nlp=spacy.load ("en")
```

4. NLTK

Natural Language Processing is manipulation of information, textual content or speech through any program or device. An analogy is that people have interaction, understand every other perspective, and respond with an appropriate solution. In NLP, this interplay, understanding, the reaction is made via a computer instead of a human. NLTK stands for Natural Language Toolkit. This toolkit is one of the most remarkable NLP libraries which incorporate programs to make machines perceive human language and respond to it with a suitable reaction. It provides different libraries for text processing, classification, tokenization, stemming, and tagging, labeling, parsing, and semantic reasoning. Tokenization in NLP is more robust. It consists of breaking the given sentence into tokens and punctuations before processing the data. Tokenization is done as shown:

```
>>>text="I was absent yesterday!"
>>>tokens=nltk.word_tokenize (text)
>>>tokens
['I', 'was', 'absent', 'yesterday', '!']
```

5. POS TAGGING

POS tagging is the first step in any NLP based application. Tagging is a sort of category that may be described as the computerized assignment of description to the tokens. Here the descriptor is known as tag, which can also represent one of the component-of-speeches, semantic statistics and so on. Now, if we talk about Part-of-Speech (POS) tagging, then it may be interpreted as a method of assigning parts of speech to the given word. It is generally called POS tagging. In easy phrases, we can say that POS tagging is a venture of labelling every phrase in a sentence with its appropriate label of speech. We already know that elements of speech encompass nouns, verb, adverbs, adjectives, pronouns, conjunction and their sub-categories. Taggers use various data of the form: dictionaries, lexicons, rules, and so on. [2] Most of the POS tagging falls underneath Rule Base POS tagging, Stochastic POS tagging and Transformation based tagging. Tag set is a collection of tags used for a particular task. Every tagger will be given a standard tag set. The tag set may be coarse such as NN (Noun), VB (Verb), JJ (Adjective), RB (Adverb), IN (Preposition), and CC (Conjunction) and so on. [2] The following is an example on how tokenization takes place, where words are tokenized based on if they are proper noun/personal noun, etc.

```
>>>import nltk
>>> nltk.pos_tag (nltk.word_tokenize("Hey, how are you doing?"))
[('Hey', 'NNP'), (',', ','), ('how', 'WRB'), ('are', 'VBP'), ('you', 'PRP'), ('doing', 'VBG'), ('?', '.')]
```

6. PROPOSED APPROACH

Initially the text is extracted from a file, or it can be user input too. All those sentences are separated by delimiter ',' and '.'. All these grouped texts are tagged with appropriate labels may it be noun, verb or punctuation marks using POS tagging and NER with the help of spacy core models. They are stored in the form of tuples. All these tuples are passed to clause function of NLP toolkit which extracts semantic role info (Subject, Object), syntactical info (type of noun, verb). After collecting all the parts of speech, the main work in forming a sentence is of chunking, it groups the words into meaningful chunks. The main of chunking is to group into noun phrases. We combine the parts of speech tags with regular expressions. It is done by tagging a noun with an adverb or adjective which are related to them. By trying out various combinations sentences in the form of questions are formed.

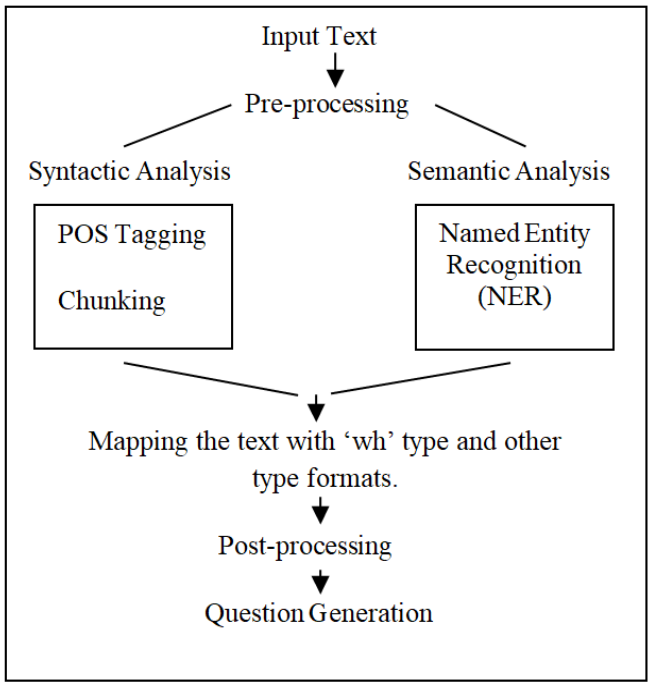

Fig 1 - Flowchart of proposed system.

7. RESULT

The sentences were given as input to the system to check the accuracy, and then the generated questions were compared with incorrect questions generated by the system, also considering questions generated by human proficient in English. In this way all the attributes of the confusion matrix were calculated, using which the accuracy, precision and recall are calculated and are graphically represented:

Table 1: Performance measures calculations of the proposed system.

Sr. No	Number of sentences	TP	TN	FP	FN	Accuracy	Precision	Recall
1	0	0	0	0	0	0	0	0
2	1	1	0	1	0	50.00%	0.50	1.00
3	3	5	1	1	2	66.66%	0.83	0.71
4	5	4	3	3	0	70.00%	0.57	1.00
5	10	9	2	2	3	68.75%	0.81	0.75
6	15	13	4	4	3	70.83%	0.76	0.81
7	20	13	5	5	3	72.00%	0.72	0.81
8	25	17	6	4	2	79.31%	0.81	0.89
9	30	14	9	6	1	76.66%	0.70	0.93
10	35	21	7	4	3	80.00%	0.84	0.87

Performance Measures

1. Accuracy:

Accuracy measures how close a given value is to the truth value. Being one of the performance measuring metrics, it is calculated as follows:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

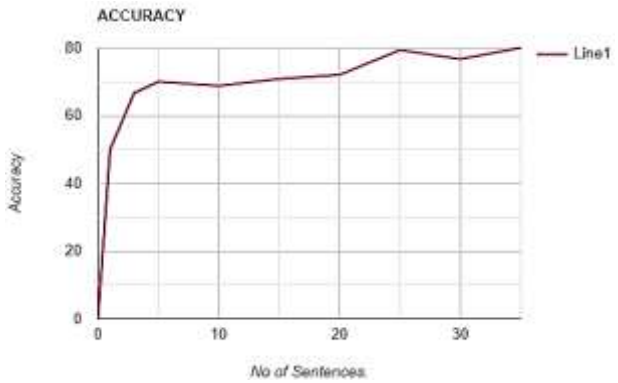

Chart 1 - Graphical representation of Accuracy of the system.

Hence above result shows that system works with 70.46 % of accuracy which can be further improved.

2. Precision:

Precision depicts how closely the measured values are to each other.

$$Precision = \frac{TP}{TP + FP}$$

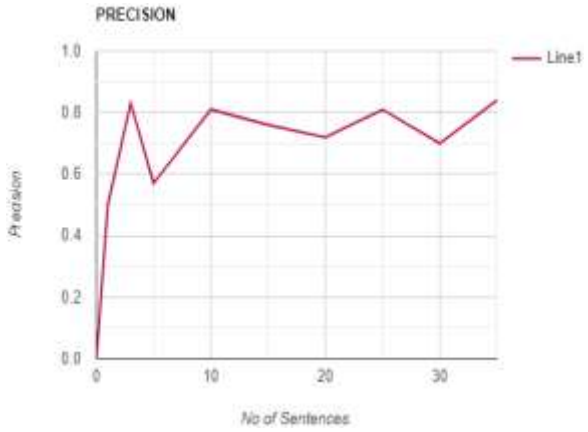

Chart 2- Graphical representation of Precision of the system.

3. Recall

Recall refers to the total relevant results correctly identified by our system.

$$Recall = \frac{TP}{TP + FN}$$

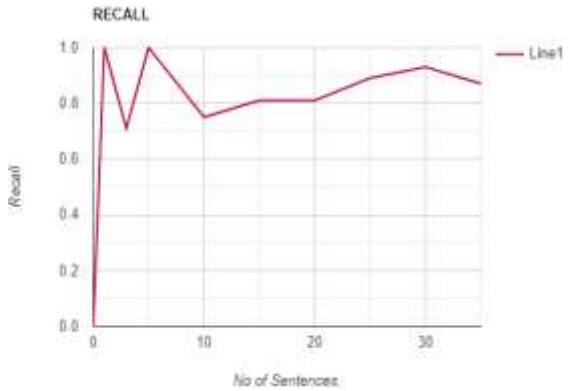

Chart 3 - Graphical representation of Recall of the system.

8. CONCLUSION

While developing the system, we collected and studied various methodologies adopted by papers for automation in Question Generation. The system takes paragraph as an input in textual format and generates questions whose answers are in the paragraph. The text given as input is pre-processed before generating questions. Pre-processing includes syntactic analysis and semantic analysis. Syntactic analysis includes the POS tagging and Chunking. Named Entity Recognition (NER) is carried out in Semantic Analysis. The pre-processing is followed by mapping appropriate 'wh' question word with the text and questions are generated. Future work of this includes generation of questions with increased accuracy; questions generated can be stored in a

database and reused, bloom's taxonomy can be applied to the questions in order to create questions of various difficult levels. The input can be taken after reading text from a PDF document as well. Education is the area with very less automation and this paper can definitely be a building step for future automation.

ACKNOWLEDGEMENT

We would like to take this opportunity to thank our project guide Prof. Gokul Patil sir for his indispensable support, suggestions and giving us all the help and guiding us time to time. We are really grateful to him for his kind support.

REFERENCES

[1] Onur KEKL –"Automatic Question Generation Using Natural Language Processing Techniques.",July 2018.
https://pdfs.semanticscholar.org/ec5e/fc74351f0339e34b91b965f99624aedf9200.pdf

[2] Amruta Umardand, Ashwini – "A survey on Automatic Question Paper Generation System", International Advanced Research Journal in Science, Engineering and Technology (IARJSET), Jan 2017.
https://www.researchgate.net/publication/313812481_A_Survey_on_Automatic_Question_Paper_Generation_System

[3] Aleena, Vidya – "Implementation of Automatic Question Paper Generator System", International Research Journal of Engineering and Technology (IRJET), Feb 2019.
https://www.irjet.net/archives/V6/i2/IRJET-6I297.pdf

[4] Kalpana B. Khandale1, Ajitkumar Pundage, C. Namrata Mahender – "Similarities In Words Using Different Pos Taggers.", IOSR Journal of Computer Engineering (IOSR-JCE),(PP 51-55).
https://www.iosrjournals.org/iosr-jce/papers/Conf.17003/Volume-1/10.%2051-55.pdf?id=7557

[5] Edward Loper and Steven Bird – "Nltk: The Natural Language Toolkit.", July 2002.
https://www.researchgate.net/publication/220482883_NLTK_the_Natural_Language_Toolkit

[6] Ankita, K. A. Abdul Nazeer – "Part-Of-Speech Tagging And Named Entity Recognition Using Improved Hidden Markov Model And Bloom Filter", International Conference on Computing, Power and Communication Technologies (GUCON), 2018.
https://ieeexplore.ieee.org/document/8674901

[7] Mohd Shahid Husain – "Automatic Question generation from Text", International Journal for Innovations in Engineering, Science and Management (IJIESM), April 2015.
https://www.researchgate.net/publication/276886742_Automatic_Question_Generation_from_Text

[8] Pranita Jadhav, Manjushree Laddha – "An Automatic Gap Filling Questions Generation using NLP", International Journal of Computer Science & Engineering Technology (IJCSET), Aug 2017.
http://www.ijcset.com/docs/IJCSET17-08-08-039.pdf

[9] Bowen Xu, Zhenchang Xing, Xin Xia, Davi Lo - "Answer Bot: Automated Generation of Answer Summary to Developers' Technical Questions".
http://www.mysmu.edu/faculty/davidlo/papers/ase17-answerbot.pdf

[10] Priti G, Shreya J, Srushtee, Subhangi – "Automated Question Generator System: A Review", International Journal of Engineering Applied Science and Technology (IJEAST), Dec 2019, Vol. 4, Issue 8, Pages 171-176; ISSN No. 2455-2143.
DOI: 10.33564/IJEAST.2019.v04i08.027

[11] Surbhi, Abdul, Shrutika, Kavita – "Question Paper Generator System", International Journal of Computer Science Trends and Technology (IJCST), Oct 2015.
http://www.ijcstjournal.org/volume-3/issue-5/IJCST-V3I5P28.pdf

[12] Sheetal, Dr. Y R Ghodasara –"Literature Review of Automatic Question Generator Systems", International Journal of Scientific and Research Publications (IJSRP), Jan 2015.
http://www.ijsrp.org/research-paper-0115/ijsrp-p3757.pdf

BIOGRAPHIES

Priti Gumaste
BE Computer Engineering,
Sandip Institute of Technology and research Centre, Nasik, Maharashtra.

Shreya Joshi
BE Computer Engineering,
Sandip Institute of Technology and research Centre, Nasik, Maharashtra.

Srushtee Khadpekar
BE Computer Engineering,
Sandip Institute of Technology and research Centre, Nasik, Maharashtra.

Shubhangi Mali
BE Computer Engineering,
Sandip Institute of Technology and research Centre, Nasik, Maharashtra.