Automatic Question Generation and Evaluation

Parth Panchal, Janak Thakkar, Veerapathiramoorthy Pillai, Prof. Shweta Patil
Computer Engineering Department
Shah and Anchor Kutchhi Engineering College
Mumbai, India
{parth.panchal_2017, janak.thakkar, veerapathiramoorthy.pillai, shweta.patil}@sakec.ac.in

DOI: 10.51201/JUSST/21/05203
http://doi.org/10.51201/JUSST/21/05203

Abstract: *Generation of questions from an extract is a very tedious task for humans and an even tougher one for machines. In Automatic Question Generation (AQG), it is extremely important to examine the ways in which this can be achieved with sufficient levels of accuracy and efficiency. The ways in which this can be taken ahead is by using the Natural Language Processing (NLP) to process the input and to work with it for AQG. Using NLP with question generation algorithms the system can generate the questions for better understanding of the text document.The input is pre-processed before actually moving in for the question generation process. The questions formed are first checked for proper context satisfaction with the context of the input to avoid invalid or unanswerable question generation. It is then preprocessed using various NLP based mechanisms like tokenization, named entity recognition(NER) tagging, parts of speech(POS) tagging, etc. The question generation system consists of machine learning classification-based Fill in the blank(FIB) generator that also generates multiple choices and a rule-based approach to generate Wh type questions. It also consists of a question evaluator where the user can evaluate the generated question. The results of these evaluations can help in improving our system further. Also, evaluation of Wh questions has been done using BLEU score to determine whether the automatically generated questions resemble closely to the human generated ones. This system can be used in various places to help ease the question generation and also at self-evaluator systems where the students can assess themselves so as to determine their concept understanding. Apart from educational use, it would also be helpful in building chatbot based applications. This work can help improve the overall understanding of the level to which the concept given is understood by the candidate and the ways in which it can be understood more properly. We have taken a simple yet effective approach to generate the questions. Our evaluation results show that our model works well on simpler sentences.*

Keywords:**AQG, Question Generator, Automatic Question Generator, MCQ questions,Wh questions**

1. Introduction

Whenever you tell someone that you have read some text, they may ask you to describe it or summarize it. Or firstly, they might ask a few questions whether you remember the basic facts in the text. Then they are assured that you have studied the whole text. These questions might be very naive to ask to an adult but would help a lot for school children. These basic questions may be asked by an elementary school teacher to the students who are still learning to read. For them, such basic questions might be a bit tougher and would help in evaluating their understanding. Generating such questions and authoring such assessments might be a time consuming and effortful task for the teachers. In this research, we are working on automating such question generation tasks.

Our system accepts text documents as an input from the user, generates the questions based on various parts and parameters present in the text document itself. The system performs some pre-processing, and finds the words/phrases that have the potential to be the answers for the questions that would be generated. These answers would then be used to form questions based on the context in which they are used in the document. The goal of the system is to form syntactically and semantically correct questions from these answers, which are answerable in context to the document uploaded by the user. The questions which are important and answerable can be further used to generate a self-

evaluation quiz for the user to evaluate one's understanding of the topic or concept. The system then evaluates the questions generated with the help of learning parameters as well as human evaluation of the questions based on evaluation metrics.

The process of Automatic Question Generation (AQG) systems is a very challenging problem in NLP. Mannem [22] defined it as the task of generating syntactically correct, semantically sound and relevant questions from various input formats such as text, a structured database or a knowledge base. As our system is used by a general set of public, we have avoided use of domain specific knowledge base. Instead, we have focused on creating a general model purely based on lexical and syntactic phenomena. Apart from educational purposes, AQG can also be used in virtual assistants and various dialogue systems. One example for such a system is a travel assistant, where AQG will gather specifics by questioning the customer. It may also be used in the health care system to analyze a patient's mental health. Due to such widespread applications in conversational AI systems and educational sectors, there has been a growing interest in the field.

2. Literature survey

Recently, Question Generation (QG) has got enormous attention from various researchers due to its widespread applications. QG systems can be used in educational fields as well help in chat bot-based applications. There are many types of questions and researchers proposed various taxonomies for organizing the questions.There are many types of questions. but classification based on the answer is important. Bloom[4] defined taxonomies based on education objectives into complexity and specificity. He defined six levels of cognitive learning on how the questions must be formed. The six levels are remembering, understanding, applying, analyzing, evaluating and creating. But from a question generation point of view, the questions are complex to generate based on these levels.

Similar to Bloom's, Rus and Graesser [14] divided questions into two categories, deep and shallow questions. Here shallow questions focus on facts like who,when, etc. from simpler factual sentences and deep questions include conceptual level understanding of questions like why, how, and what-if from complex sentences or a passage containing two or more sentences. Our main focus is on generating shallow questions that are simple to answer.

A more recent paper from Heilman and Smith [1] put forward various challenges that we might face while generating questions as well as the answering part. They were syntactic, lexical and discourse challenges. These challenges put light on the difficulty of generating syntactically and semantically correct questions within the context of the input text. More recently neural network-based approach has been used to solve the question generation. Serban [9] worked on generating questions from a triplet of subject, relation and object. Using the relation, questions were generated

Learning to ask paper [12] proposes a Seq2Seq model with attention for question generation from text. Zhao [17] proposed maxout pointer network to keep track of word coverage. It generated questions from paragraphs. It used a complex deep learning approach to generate the questions. Kumar [7] generated answers from text using Pointer Networks by Vinyals [11] and encodes answers in the question decoder. Pointer Networks was used to identify the answer phrase in the text, which is then encoded and the attention decoder generates questions. It also implemented a copy [15] and coverage mechanism [16] to improve the generated question. However, these systems are very complex to execute as it requires a lot of learning on major datasets. There also has been research to use autoencoders to implement QG. There also have been machine learning systems that use classification techniques to identify keywords and generate questions. We have used a similar classification technique to identify the answer keyword.

As we have to evaluate our generated questions, whether the generated questions are good or not, we need an evaluation metric. But there have been a number of evaluation

metrics for sentences, but not for questions in particular. BLEU[3] score is one of the earliest evaluation metrics. It uses n-gram comparison with the original sentence and generates a score. There are also other metrics like METEOR[6], ROUGE[5] that have been used to evaluate the generated question. ROUGE was used to evaluate the summary generated but can also be applied in question generation. ROUGE stands for Recall-Oriented Understudy for Gisting Evaluation. It is essentially a set of metrics for evaluating automatic summarization of texts as well as machine translations. It works by comparing an automatically produced summary or translation against a set of reference summaries.

Apart from these, human evaluation plays an important role in evaluating such systems. Most of the QG systems have used human evaluation to evaluate the generated questions. We have also used human evaluation in our system

3. Architecture

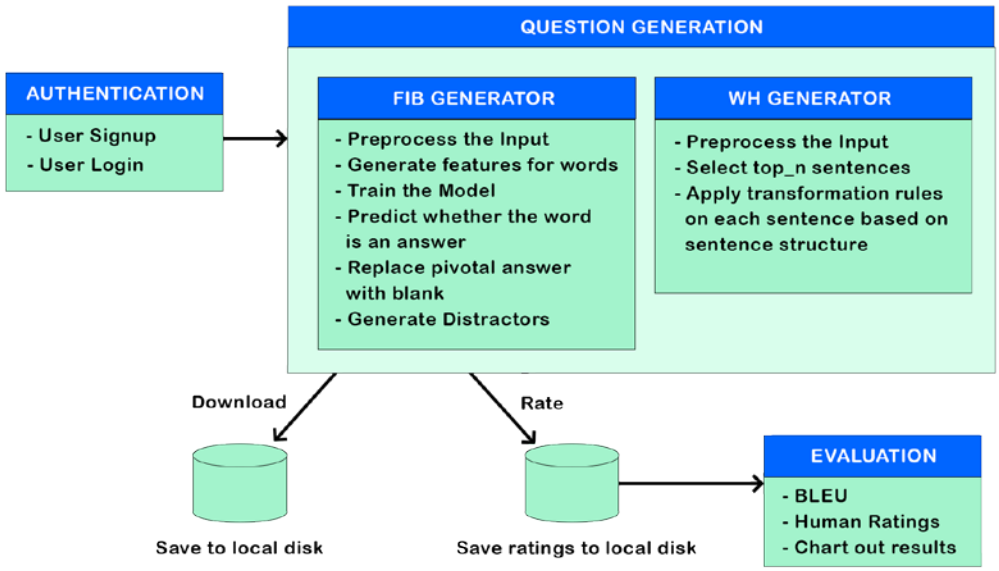

Figure 1: Architecture of system

We build a system that accepts a text document from the user and then generates questions. The document may contain a passage or an extract of any topic. The main architecture is mainly divided into three modules:

Authentication - As the user has to create an account to use the QG system, the user must sign up to create an account. Authentication is done by Django authentication or the user can use his Google account as well to log in to the system.

Question Generator - After logging into the system the user has to upload a text document. The document cannot contain any images or non-textual special characters. The user is provided with the option of downloading or rating the generated question. Based on the choice, the generated question file or the ratings will be saved in the local disk.

Also, the user has to choose the maximum number of questions to be generated, although it cannot exceed the number of sentences in a text document. The questions are of two types, namely Fill in the blank(FIB) and Wh type of questions.

The FIB model uses machine learning techniques to generate FIB questions. It identifies the keyword using classification techniques. The keyword is then replaced with a blank line and the remaining sentence is used as the FIB question. It also generates multiple wrong answers or distractors using the keyword so that it can be used as MCQ. The Wh question first takes out the top sentences, preprocesses the text and then

generates questions based on the type of sentences. Sentences are classified into NER tags based, discourse marker based and non discourse marker based algorithms. It applies various transformation rules to generate the questions based on the structure of the sentences.

Evaluation - Once the questions are generated, if the user wants to evaluate the questions generated, they can rate the generated questions based on various criteria like answerability, grammatical correctness of the question. This would help in improving the system. Apart from this we have also implemented BLEU scores to compare our questions to human generated ones.

This whole system performs well for simpler sentences. However, there is room for improvement in the rules and algorithms used, as some of the questions generated are inaccurate based on the semantic and syntactic correctness. It still has to improve the transformation rules to be able to generate questions for complex sentences.

The preprocessing part contains various preprocessing techniques used in NLP. They are:

Tokenization - Tokenization is used to divide the strings into a list of substrings. Sentence tokenizer can be used to find a list of sentences. It is a problem in NLP for deciding where the sentences begin and end. We have used sent_tokenizer which is an instance of PunktSentenceTokenizer from the nltk.tokenize.punkt module. This has been pretrained and works well with various European languages.

POS tagging - Parts of Speech tagging is a piece of software that reads text and assigns parts of speech to each word, such as nouns, adjectives,verbs etc. However, general computational applications use more fine grained POS tags like ‘noun-plural’.We have used Java implementation of log linear part of speech taggers.

There are mainly two types of taggers: rule based and stochastic

1. Rule based: It uses hand written rules to distinguish tag ambiguity
2. Stochastic: These are either HMM based or cue based, using decision trees or maximum entropy models

It explains how a word is used in the sentence. There are eight main parts of speech namely nouns, adjectives,verbs,pronouns,adverbs, prepositions, conjunctions and interjections. It is a supervised learning approach using features like previous word, next word etc. NLTK has a function pos_tag. Sentences are tokenized and then passed to nltk.pos_tag function which would return each word alongwith its POS tags

NER tagging - It is an information extraction process that labels words into various semantic categories such as person, date, location, etc. NER tagging can be done by rule based approach or machine learning approach. Rule based approach requires expert linguists to define rules whereas machine learning approaches allows us to train models and hence needs a lot of data

4. Methodology

The question generation part of the system generates two types of questions. They are FIB based questions and Wh type questions. They have their own individual architectures.

4.1 FIB based questions

The general idea here is that we can create a fill-in-the-blanks type of question from an input sentence by just converting the answer word into a blank. This task is done in three simple steps as follows:

1. Search for pivotal answer from the input sentences and use these words/phrases as answers to the questions.
2. Replace the pivotal answer chosen with a blank space
3. Generate incorrect options to be used as multiple-choice options

We have used SQUAD [2] 1.0 dataset which contains about 100,000 questions generated on Wikipedia articles. Intuitively, the task of selecting a probable answer is very much similar to tagging a word as spam or not spam [18]. Hence, we decided to use binary classification on each input word to tag it whether it is an answer or not. For this task, each non-stop word from the paragraphs of SQUAD dataset were extracted and we added some features on them like POS tag, shape, word count, NER tag, etc. and a label 'isAnswer'.

Using the data generated from the previous step, we used scikit-learn's Gaussian Naive Bayes [19] algorithm to train a model that would tag each word as whether it can be a pivotal answer or not. The advantage of using Naïve Bayes is that it also gives us the probability of each word, which will be used to choose the most probable pivotal answer. The distractors are generated using pre-trained word-embeddings [20] and cosine-similarity [21]. This will generate words that will be used as the multiple-choice options.

Once the model is trained, we save the model to use it later for user inputs. After the user uploads the document. The content is split into sentences and various preprocessing is done to clean the text. We feed these sentences to the model that we saved earlier to predict the pivotal answers. The generated results are then formatted and displayed to the user.

Example Input Sentence: "The fourth planet from the Sun in the Earth's solar system is Mars, which is sometimes called the Red Planet."

Pivotal Answer chosen: Mars
Options generated: Moon, Jupiter, Saturn

4.2 Wh type Questions

For generation of Wh questions, the sentences are filtered as the entire text cannot be used to generate questions. We use the top sentences to generate our questions. To identify the top sentences we use NLTK's Textrank algorithm. This algorithm takes out the most important sentences present in the text. The general preprocessing part involves tokenizing the uploaded text document. The words are then tagged using POS tagging and NER tagging.The question generation procedures are further classified based on sentence structures. Each of these have different transformation rules and algorithms. They are classified into:

- Named Entity Recognition based algorithm
- Discourse Marker based algorithm
- Non Discourse marker based algorithm

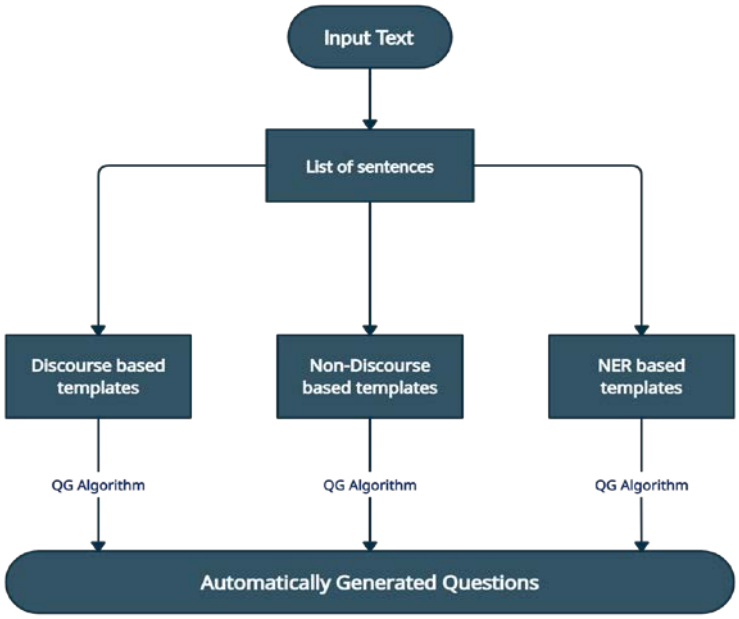

Figure 2: Wh type questions classification

Named Entity Recognition based algorithm: These are used in declarative[1] or assertive sentences or Yes/no sentences. We use parts of speech (POS) tagging to find the nouns and then are run through Named Entity recognition to identify the type of named entity. Named entity recognition identifies these named entities in text and classifies it into predefined categories such as names of persons, organizations, locations, events, expression of times,numerical value, etc. Each of these named entities are mapped to a Wh word that would be used during question generation. For example, if the named entity is a name, then the appropriate question word would be Who. Similarly, if the named entity is time, the question word would be When.

After identification of the named entity, we have made rules to generate the question. According to the rules, if the named entity is present in the beginning of the sentence, we can simply replace the named entity with the appropriate Wh word to form the question.
We find the auxiliary verb in the sentence if the named entity is not present in the beginning. Iterating over the POS tags, we append each word into the question part until we find the first non-auxiliary verb. If the word after this non-auxiliary verb is a noun,we discard it. If it is not a noun, we include it in the question part. We then rearrange the question parts, satisfying the grammatical syntax. After rearrangement the Wh word is prepended and the question mark is added at the end to generate the whole question
For example: "Mahatma Gandhi was born on 2nd October,1869"
Question generated: Who was born on 2nd October,1869?

In the above example, "Mahatma Gandhi" is identified as the named entity. Since it is the name of a person, the associated question word will be "Who". As the named entity is present at the start of the sentence, it is simply replaced with the question word and question mark is added at the end to generate the question.

Discourse Marker based algorithm: In this type of question generation, we first identify discourse markers like 'because', 'although' etc. For every discourse marker present in the sentence, we define a question word associated with it. Then we process the sentence by converting it into lowercase and removing full stop. We further classify the sentence into two types, one having an auxiliary verb and one without an auxiliary verb. Using the POS and NER tags generated in the general preprocessing part, we identify:
If sentence has an auxiliary verb:

- We append each word into the question part until we find the first non-auxiliary verb in the sentence.
- Split the question part across the auxiliary verb and place the auxiliary verb at the start of the question part.
- If the question type is of Yes/no, we return the formed question part with a question mark appended, else prepend the question word at the start of the question part and return.

If the sentence does not contain any auxiliary verb:

- We define various combinational rules to generate an appropriate auxiliary verb. We identify which verb will be appropriate to fulfill the subject verb agreement. The auxiliary verbs like do, does and did may be included in the question part
- We stem the verb present in the sentence using any stemmer or a lemmatizer
- Rearrange the question word so as to fulfill all the grammatical rules
- Prepend the question word, as identified the discourse marker's mapping
- Append the question mark and return the generated question

For example:
Sentence 1: "They were sad because they lost the battle"
Question generated: Why were they sad?
Sentence 2: "I think he played well because of his training and practice"
Question generated: Why do you think he played well?

Non-Discourse Marker based algorithm:Sentences that do not contain any discourse marker fall into this category such as assertive or declarative sentences. Here, we simply use the appropriate auxiliary verb as the starting word of the question and the other procedure of generating the combination from POS tags is the same as in discourse markers based templates. Question words like who, how, etc. are not used in the question generation. The question generated may be a simple Yes or No question.
For example:
Sentence 1: "Yes, I read the research paper"
Question generated: Did you read the research paper?
Sentence 2: "No, I was not playing chess?"
Question generated: Were you not playing chess?

5. Experimental Setup

To evaluate the quality of the generated question and the performance of the whole system. We have used two distinct approaches

- Automated Evaluation of Wh Questions using BLEU.
- Human Evaluation of FIB & Wh Questions.

For both the cases we used a dataset of 100 questions from a wide range of topics.

5.1 Automated Evaluation of Wh questions using BLEU scores

We created a dataset of 100 Wh Questions based on various topics according to the above description.BLEU score[3] or the Bilingual Evaluation Understudy is a metric for comparing a candidate translation of the text to one or more reference translations. Generally, BLEU is used to rate machine translations, however it can be used for our system to rate the syntactical structure for the generated Wh questions. This is the reason why BLEU score could not be used in FIB as there is no syntactical change in the original sentence. We used this metric to rate our Wh Questions where the system-generated question was the candidate sentence, the sentence from which the Wh question was generated was the first reference and a human-crafted question based on reference1 was the second reference. We ran the NLTK's BLEU score algorithm for each of the 100 sentences from the generated dataset. The results were then plotted.

5.2 Human Evaluation of FIB & Wh Questions.

Since the BLEU metric only looks for syntactical similarity with the reference sentences, it does not always correspond well with human judgements. Hence, we decided to evaluate the questions manually using human evaluators. The human evaluation is applied to both FIB and Wh questions. For this, a dataset of 200 questions were generated, out of which 100 were FIB questions and the rest Wh questions. We used 5 human evaluators to evaluate the questions based on various parameters. These evaluators were college going students with fair enough knowledge about English language and grammar. The system asks the user to rate the questions in a range of [0.0–1.0] based on four parameters:

- Grammatical: Whether the question generated is grammatically correct?
- Answerability: Whether the question generated is answerable from the input text?
- Difficulty: What is the difficulty level of the question?
  For FIB Question, it can be the difficulty of choosing the correct answer from the options(distractors) that are generated by the system.
- Context: Does the generated question revolve around the central idea of the input text?

All the ratings are saved in an excel file which will be used to plot the graphs and analyze the results.

6. Results

The summary of results that were generated is as follows:

Table 1: Evaluation results

	FIB Questions (Total 100)	Wh Questions (Total 100)
Correct	59	49
Incorrect	41	51
Success Rate	59%	49%

6.1 Automatic Evaluation using BLEU

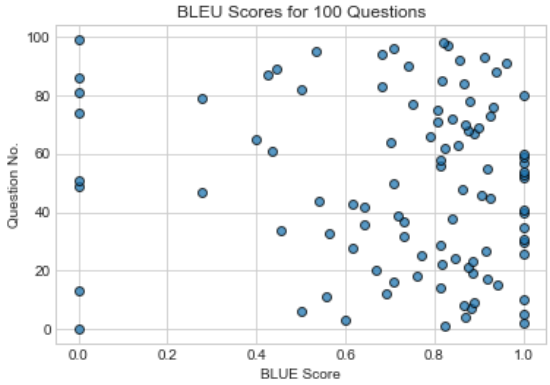

Figure 3: Scatter plots of BLEU scores

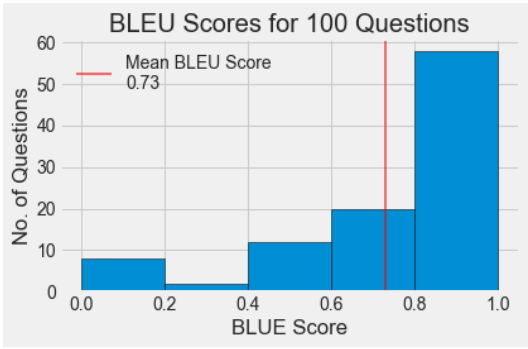

Figure 4: Histogram of BLEU scores

The higher BLEU scores indicate the better results and shows that the system has performed well. Although this does not ensure the generated question is answerable or not as discussed earlier. The BLEU metric is used for comparing the candidate sentence with the reference one so as to check its

syntactic similarity with the previous one. This score in our project gives us the result so as to check the path at which system is progressing along with maintaining the similarity with the human generated questions. The analysis also shows that the quality of questions generated is best when the input sentences are simple in structure.

6.2 Human Evaluation

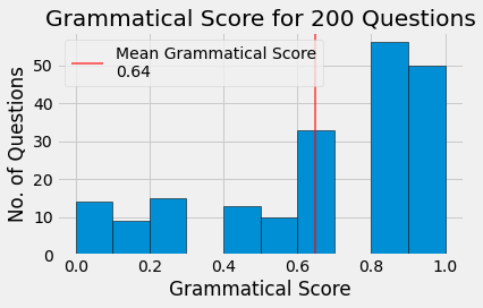

Figure 5: Human generated grammatical scores

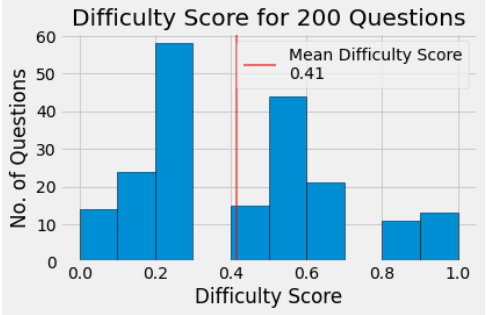

Figure 7: Human generated difficulty scores

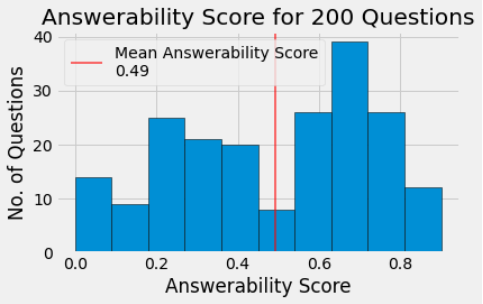

Figure 6: Human generated answerability scores

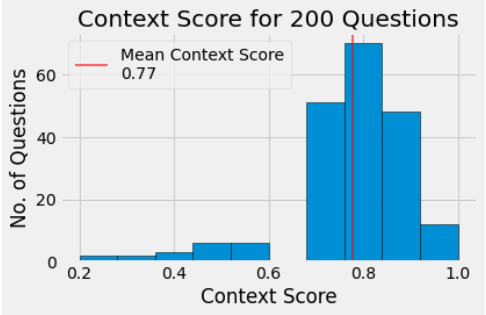

Figure 8: Human generated Context scores

Human Evaluations show the actual performance of the system. We can see that the grammatical structure of most of the questions is proper with few exceptions. The difficulty score is not up to the mark since the answers for most of the questions were easily guessable from the given set of options. Answerability is also average due to some questions being improperly formed which would mean that they cannot be used in a real scenario. However, the context score shows that the TextRank algorithm used in generating Wh questions successfully selected contextually relevant sentences from the input document and hence the generated questions revolve around the central idea of the document.

7. Conclusion

According to the analysis of the system that we conducted we can conclude that the system gives better results when the input sentences are assertive. Overall quality of the questions that were generated turned out to be good with a few exceptions. The system was generating contextually relevant questions for the input text.

Since the FIB model is trained using SQUAD dataset [2], it may sometimes select the wrong word as an answer. This can be avoided by training the model on a domain specific dataset which would add some intelligence to the system. The distractors can also be improved by having a domain specific dataset to work with. Another improvement for the FIB model can be to add some more additional features to the words such as TF-IDF score, cosine-similarity to the title, information on whether the pivotal answer is present

at the start, middle or end of the sentence, details about the words surrounding it and much more.

For the Wh questions, the system would sometimes generate irregularly formed questions. This is due to the fact that we are using a rule-based approach for Wh questions and we have not defined rules to parse sentences with complex structure. There were also some shortcomings in the NER of the Spacy library which caused some entities to not be tagged properly resulting in generating an incorrect or poor question. Also, all types of Wh questions are not generated and rules for those could be added to get a variety of questions.

Further, having a neural network-based architecture would also benefit the system in terms of generating questions that would resemble questions generated by humans. This would also be able to parse complex sentences as defining rules for such sentences would be difficult. The great thing about the approach that we used is that it's a simple proof of concept for the question generation task which could be improved upon with the latest trends in NLP.

8. Reference

[1] *Michael Heilman and Noah A. Smith.* ***(2010)****. Good question! statistical ranking for question generation. In HLT-NAACL, pages 609–617. The Association for Computational Linguistics.*

[2] *Pranav Rajpurkar ,JianZhang, Konstantin Lopyrev, and Percy Liang.* ***(2016)****. Squad: 100, 000+ questions for machine comprehension of text. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas,USA, November1-4,2016, pages 2383–2392.*

[3] *Kishore Papineni, Salim Roukos, Todd Ward, and WeiJing Zhu.* ***(2002)****. Bleu: a method for automatic evaluation of machine translation. In ACL, pages 311– 318. ACL.*

[4] *Bloom, B.S.* ***(1956)****. Taxonomy of educational objectives: The classification of educational goals.*

[5] *Chin-Yew Lin.* ***(2004)****. Rouge: A package for automatic evaluation of summaries. Text Summarization Branches Out.*

[6] *Michael Denkowski and Alon Lavie, "Meteor Universal: Language Specific Translation Evaluation for Any Target Language", Proceedings of the EACL 2014 Workshop on Statistical Machine Translation,* ***(2014).***

[7] *Vishwajeet Kumar, Kireeti Boorla, Yogesh Meena, Ganesh Ramakrishnan, and Yuan-Fang Li.* ***(2018)****. Automating reading comprehension by generating question and answer pairs. In 22nd Pacific-Asia Conference on Knowledge Discovery and Data Mining.*

[8] *Vishwajeet Kumar, Yuncheng Hua, Ganesh Ramakrishnan, Guilin Qi, Lianli Gao, and Yuan-Fang Li.* ***(2019)*** *Difficult-controllable multi-hop question generation from knowledge graphs.*

[9] *Serban, I., Sordoni, A., Bengio, Y., Courville, A., & Pineau, J.* ***(2016)****. Building End-To-End Dialogue Systems Using Generative Hierarchical Neural Network Models. Proceedings of the AAAI Conference on Artificial Intelligence.*

[10] *Preksha Nema, Akash Kumar Mohankumar, Mitesh M. Khapra, Balaji Vasan Srinivasan, Balaraman Ravindran , Let's Ask Again: Refine Network for Automatic Question Generation, EMNLP* ***(2019).***

[11] *Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly.* ***(2015)****. Pointer networks. In Advances in Neural Information Processing Systems, pages 2692–2700.*

[12] *Nan Duan,Duyu Tang, Peng Chen, Ming Zhou, Proceedings of the* ***(2017)*** *Conference on Empirical Methods in Natural Language Processing, pages 866–874.*

[13] *Chin-Yew Lin.* ***(2004)****. Rouge: A package for automatic evaluation of summaries. In Proc .ACL workshop on Text Summarization Branches Out, page 10.*

[14] *Roger Azevedo, Amy Witherspoon, Arthur Graesser, Danielle McNamara, Amber Chauncey, Emily Siler, Zhiquiang Cai, Vasile Rus, Mihai Lintean,* ***(2009)*** *Analyzing Self-Regulated Learning in a Tutoring System for Biology, Frontiers in Artificial Intelligence and Applications, pages 635 - 637*

[15] *Jiatao Gu, Zhengdong Lu, Hang Li, and Victor OK Li****. (2016)****. Incorporating copying mechanism in sequence-to-sequence learning. In ACL, volume 1, pages 1631–1640*

[16] *Zhaopeng Tu, Zhengdong Lu, Yang Liu, Xiaohua Liu, and Hang Li.* ***(2016)****. Modeling coverage for neural machine translation. In ACL 2016, pages 76–85. The Association for Computer Linguistics.*

[17] *Yao Zhao, Xiaochuan Ni, Yuanyuan Ding, and Qifa Ke.* ***(2018)****. Paragraph-level neural question generation with maxout pointer and gated self-attention networks.*

[18] *Q. Luo, B. Liu, J. Yan and Z. He, "Research of a Spam Filtering Algorithm Based on Naïve Bayes and AIS," 2010 International Conference on Computational and Information Sciences,* ***(2010)****, pp. 152-155.*

[19] *Rish, Irina.* ***(2001)****. An Empirical Study of the Naïve Bayes Classifier. IJCAI 2001 Work Empir Methods Artif Intell. 3.*

[20] *Tomas Mikolov, Kai Chen, Greg Corrado, Jeffrey Dean, "Efficient Estimation of Word Representations in Vector Space",* ***(2013)***

[21] *Rahutomo, Faisal & Kitasuka, Teruaki & Aritsugi, Masayoshi.* ***(2012)****. Semantic Cosine Similarity.*

[22] *Prashanth Mannem, Rashmi Prasad and Aravind Joshi.* ***(2010)****. Question Generation from Paragraphs at UPenn: QGSTEC System Description, Proceedings of QG2010: The Third Workshop on Question Generation*