A major project Final Report on

**Automatic Question Generator**

Submitted in Partial Fulfillment of the Requirements for

The Degree of Software Engineering

Under Pokhara University

Submitted by:

Rajendra Shrestha(15792)

Ritesh Adhakari (15765)

Sandesh Koju(15784)

Yaman Maharjan(15772)

Under the supervision of

Er Subash Manandhar

Date:

25/11/2019

****

# ACKNOWLEDGEMENT

We would like to express our special thanks of gratitude to our project supervisor **Er. Subash Manandhar** who provided us the guidance to complete our project on web based Automated Question Generator application, which helped us in doing a lot of research, also we came to know a lot about new things, we are really thankful to them.

Finally, we would like to thank the management body of ‘Nepal College of Information and Technology’ and classmates for helping us tackle the obstacles to complete the report within the given timeframe.

Rajendra Shrestha (15792)

Ritesh Adhikari (15765)

Sandesh Koju (15784)

Yaman Maharjan(15772)

# ABSTRACT

Texts with potential educational value are becoming available through the Internet (e.g., Wikipedia, news services). However, using these new texts in classrooms introduces many challenges, one of which is that they usually lack practice exercises and assessments. Here, this application addresses part of this challenge by automating the creation of a speciﬁc type of assessment item. This application have focused on automatically generating questions. Its goal is to create an automated system that takes text input and generate questions as output. This system AQG somehow managed to generate questions . In the first phase it consists of NLP processing which include sentence splitter to split sentence, NER for categorizations of various names and entities. Also, PoS tagger for finding parts of speech. Dependency parser and SRL helps in finding dependencies of words and semantic roles of each word respectively .The output of each step is transferred to Question Generator in second phase of process design. Main verb decomposition , subject and auxiliary verb inversion and possible questions are created which are processed and valid question as severed as output to user.

*Keywords*: Automted Question Generator(AQG), NLP, SRL, NER, PoS Tagger

Table of Contents

[ACKNOWLEDGEMENT i](#_Toc25618979)

[ABSTRACT ii](#_Toc25618980)

[LIST OF FIGURES v](#_Toc25618981)

[LIST OF TABLES vi](#_Toc25618982)

[LIST OF ABBRIVIATIONS vii](#_Toc25618983)

[CHAPTER 1: 1](#_Toc25618984)

[INTRODUCTION 1](#_Toc25618985)

[1.1 Introduction 1](#_Toc25618986)

[1.2 Problem Statement 1](#_Toc25618987)

[1.3 Objectives 1](#_Toc25618988)

[1.4 Scope and Limitations 2](#_Toc25618989)

[1.4.1 Scope 2](#_Toc25618990)

[1.4.2 Limitation 2](#_Toc25618991)

[1.5 Report Organization 2](#_Toc25618992)

[CHAPTER 2: 4](#_Toc25618993)

[REQUIREMENT ANALYSIS AND FEASIBILITY STUDY 4](#_Toc25618994)

[2.1 Literature review / Related Work / Existing Work: 4](#_Toc25618995)

[2.1.1 Basic Concepts: 5](#_Toc25618996)

[2.2 Requirement Collection 10](#_Toc25618997)

[2.2.1 Functional Requirement 10](#_Toc25618998)

[2.2.2 Non-functional Requirement 11](#_Toc25618999)

[2.3 Feasibility Study 12](#_Toc25619000)

[2.3.1 Technical Feasibility 12](#_Toc25619001)

[2.3.2 Operational Feasibility 12](#_Toc25619002)

[2.3.3 Economic Feasibility 12](#_Toc25619003)

[2.3.4 Schedule Feasibility 12](#_Toc25619004)

[2.4 Structuring System Requirements 13](#_Toc25619005)

[2.4.1. Data Flow Diagram 13](#_Toc25619006)

[CHAPTER 3: 16](#_Toc25619007)

[SYSTEM DESIGN 16](#_Toc25619008)

[3.1 System Design 16](#_Toc25619009)

[3.1.3 Input Output Design 17](#_Toc25619010)

[3.1.4 Process Design 18](#_Toc25619011)

[CHAPTER 4: 20](#_Toc25619012)

[IMPLEMENTATION AND TESTING 20](#_Toc25619013)

[4.1 Implementation 20](#_Toc25619014)

[4.1.1 Tools used 20](#_Toc25619015)

[4.1.2 Listing and Description of Major Classes 21](#_Toc25619016)

[4.2 Testing 22](#_Toc25619017)

[4.2.1 Application Inputs 22](#_Toc25619018)

[4.2.2 Question Generation Task 23](#_Toc25619019)

[4.2.3 Correctness Analysis of Generated Questions 24](#_Toc25619020)

[CHAPTER 5: 27](#_Toc25619021)

[CONCLUSION AND FUTURE ENHANCEMENT 27](#_Toc25619022)

[5.1 Conclusion 27](#_Toc25619023)

[5.2 Future Enhancement 27](#_Toc25619024)

[REFERENCE 28](#_Toc25619025)

# LIST OF FIGURES

**Fig No. Fig Name Page Number**

Figure 1.1 Diagram representing report organization ………………………………………..3

Figure 2.1 Use case diagram to represent Automated Question Generator………………....10

Figure 2.2 Project Scheduling…………………………..……………………………………12

Figure 2.3 Context Level Diagram of Automated Question Generator…………………..….12

Figure 2.4 Level 1 Diagram of Automated Question Generator………………………..……13

Figure 3.1 Automated Question Generator (paragraph interface) ………………………......14

Figure 3.2 Automated Question Generator (question interface) ………………………........14

Figure 3.3 Automated Question Generator (working flowchart) ……………………………15

Figure 3.4 Level 2 DFD of Automated Question Generator.…………………...………...…16

Figure 4.1 A Comparison Chart …………………………..…………………………………20

Figure 4.2 The graph of percentage of Accuracy………………………………………........22

# LIST OF TABLES

**Table No. Table Name Page Number**

Table 2.1 Alphabetical list of part-of-speech tags used in the Penn Treebank ……….…..…6

Table 2.2 Mapping of Answer Phrase to Question Phrase……………………………………9

Table 4.1 Input Texts …………………………………………………………………..…...19

Table 4.2 Number of Generated Questions by Human and the AQG System……..…….....20

Table 4.3 Correctness Analysis of the Generated Questions from the ‘Bangladesh’ Texts ...20

Table 4.4 Correctness Analysis of the Generated Questions from the ‘United Nation’

Texts…………………………………………………………………………….….21

Table 4.5 Correctness Analysis of the Generated Questions from the ‘Pashupatinath’

Texts……………………………………………………………...........................21

# LIST OF ABBRIVIATIONS

|  |  |
| --- | --- |
| **AI** | Artificial Intelligence |
| **AQG** | Automated Question Generator |
| **DFD** | Data Flow Diagram |
| **ER** | Entity Relation |
| **IE** | Information Extraction |
| **ML** | Machine Learning |
| **NER** | Name Entity Recognition |
| **NLP** | Natural Language Processing |
| **NLU** | Natural Language Understanding |
| **PoS** | Parts of Speech |
| **SRL** | Semantic Role Labeler |
| **UI** | User Interface |

# CHAPTER 1:

# INTRODUCTION

## 1.1 Introduction

In today’s world, time is a major concern. Any product that can effectively reduce time and power consumption is accepted and appreciated. Natural language processing (NLP) is concerned with programming machines to understand and generate the natural language. NLP consists of two components: natural language understanding (NLU) and natural language generation (NLG).Thus this application is presenting a Question Generator System that can reduce time consumption by replacing the conventional method of question generation system which needs lesser man power. In this system natural language processing (NLP) transformation is done. The project is based on the automatic question generation system where the user gives the sentence or the paragraph and the system generates the questions from the given paragraph. This proposed system aims at the generation of questions as well as reducing the man power and time required for the same.

## 1.2 Problem Statement

Selection of the question from any given text has been one of the difficult task in recent time .Making a set of question from any text according to their weight is a problem and also will be more time consuming for any person where person has to read all the paragraph numbers of times. Although reading a paragraph multiple of times, [4] creating questions involves all the information about the paragraph is a bit time consuming. Also gist of a writing may not be proper.

## 1.3 Objectives

The objective of the system is summarized as, to:

* Reduce conventional method of paper based question creation.
* Reduce manpower and time consumption during question generation.
* Generate question from the given sentence/paragraph.
* Understand the processing of a sentence / word.

## 1.4 Scope and Limitations

### 1.4.1 Scope

Today’s system are all based on well suited web based application. The system, developing can be used in any Universities/Schools/Colleges for generation of question from the text. They can also be used for the study for their study purpose. This system can be applicable for companies for easy generation of questions for interviews for any texts[6]. It can be helpful for understanding the NLP process easily along with fragments. Therefore, this system has very good scope.

### 1.4.2 Limitation

Different limitations of this project are:

* User should provide information/text each time they use application.
* It is only applicable for English Language.
* Since it is just the model, operating it might take some time (< 1 minutes).
* Questions might not be exactly as those thought of.

## 1.5 Report Organization

Altogether the project is divided into five different chapters, each representing different development phase of the project. The chapters can be discussed briefly as follow:

**Chapter 1:** It deals with the introductory part of the project and explains about what the project is, how it came in idea, the main objective of the project that are planned to be achieved after the completion, its scope and limitations.

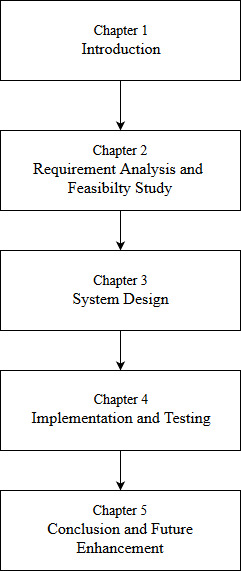
**Chapter 2:** This chapter is all about the research process carried out to do the project. It consists the background idea for the project as well as the study process that are required for this application to be popular among the users.

**Chapter 3:** Chapter 3 deals with the designing phase. System design consists of database design, interface design, and process design that are carried out to build this application.

**Chapter 4:** It is all about implementation and testing of this application. Various tools are used for implementation and testing is carried out using various test cases.

**Chapter 5:** Finally, this chapter is about conclusion ad future which contains idea about what we achieved in the end of the project and what ideas can be added to it.

This can be shown as:



*Figure 1. 1 Diagram representing report organization*

# CHAPTER 2:

# REQUIREMENT ANALYSIS AND FEASIBILITY STUDY

The team started the project with limited understanding of what would be needed complete the entire project, so a requirement analysis was carried out. The following summary of the functional requirements was made:

The application allows user to simply insert paragraph of their own interest and generate the question related to the paragraph the user has inputted.

## 2.1 Literature review / Related Work / Existing Work:

Now the world has a great eye on the field of Natural Language Processing (NLP). Someone do researches on text processing, others on text understanding, text summarizing, finding answers for given questions, and generating questions, etc. All of these tasks are a great deal with NLP. Some of the works that are done earlier on question generation are listed below

The paper on "An Integrated Automated Paperless Academic Module for Education Institutes‟ has stated the importance and working of switching from Paper based systems to Paperless Systems.[1]

Kaur et al. described a system that first collects the corpus of data or paragraph from the encyclopedia to make the questions and find the exact answers. Plamondon et al. developed a system where the question must be asked in English, the document collection was in English and the answer extraction was performed in English. Filho et al. [4] developed a system where they tried to classify the questions in only four types (“who”, “where”, “when” and “how many” questions).

Hoque et al. [5] proposed a framework for generating questions and corresponding answers considering the documents of two different languages- Bengali and English. But this can only generate simple ‘Wh’ questions.

The question processing module converts natural language question queries for the document retriever. The process ranges simply returning the user's question as the query to employing question analysis to generate complex structured queries.

Liu et al. automatically generated questions to support students as they write literature reviews (e.g., What is the research question formulated by X?). They automatically extract and classify citations from a student’s review, and then use templates to generate questions to provide feedback (e.g., about potentially missing content in the student’s work).

Piwek and Stoyanchev discuss some preliminary work on creating dialogues from monologues by converting statements into questions. They aim to use QG to create more engaging and effective ways of presenting instructional content.[5]

Mostow and Chen explore the generation of questions about the intentions and mental states of characters in narrative ﬁction. They use the automatically generated question to scaffold and encourage the generation of questions by beginning readers, since it can be a useful strategy for learning to read (National Institute of Child Health and Human Development, 2000). A number of papers have explored automatic techniques for generating vocabulary questions.

Heilman and Eskenazi apply distributional similarity techniques to extract multiple-choice related word questions from large text corpora. (e.g., Which set of words are most related in meaning to “reject”? A. pray, forget, remember; B. accept, oppose, approve ...).

Also, Pino et al. and Skory and Eskenazi explore the generation of cloze questions from large text corpora for vocabulary assessment.

### 2.1.1 Basic Concepts:

In this module, basically, we have followed the procedure described by Heilman et al. which is explained in this section. Here this procedure is modified slightly. This module deals with the Question Generator (QG) for English that defines a two steps process for question generation:

1. NLP Transformation and
2. Question Creation.

In step (i), the text sentences are transformed into simpler straightforward declarative statements. This is done by applying some syntactical and grammatical operations. It has operations for transforming complex sentences into simple sentences and resolution of pronouns. Here we have developed some NLP tools to analyze the input sentences. The NLP tool is used to auto sentence split, tokenize, and parse sentence resulting in Penn Treebank style. We have also used the Parts of Speech (POS) tagger which labels the words of a sentence as their POS. It also labels the proper nouns with their semantic classes (often just person, organization, and location). All of these are implemented in the NLP tool. In step (ii), these generated sentences are processed to generate questions by following some operations (Wh-movement, subject-auxiliary inversion, main verb decomposition , unused word marking ,validating questions etc.).

#### i) NLP Transformation

* **PoS Tagger**

A PoS tagger is a software tool that labels words as one of several categories to identify the word's function in a given language. In the English language, words fall into one of eight or nine parts of speech. Part-of-speech categories include noun, verb, article, adjective, preposition, pronoun, adverb, conjunction and interjection. PoS taggers use algorithms to label terms in text bodies. These taggers make more complex categories than those defined as basic PoS, with tags such as “noun-plural” or even more complex labels[3].

* **Penn Treebank**

Penn Treebank, a corpus consisting of words of American English is being used in our project.. It is often created on top of a corpus that has already been annotated with part-of-speech tags. Also it is enhanced with semantic or other linguistic information. It can be created completely manually, where linguists annotate each sentence with syntactic structure, or semi-automatically.

*Table2.1: Alphabetical list of part-of-speech tags used in the Penn Treebank*

|  |  |  |
| --- | --- | --- |
| 1. | CC | Coordinating conjunction |
| 2. | CD | Cardinal number |
| 3. | DT | Determiner |
| 4. | EX | Existential *there* |
| 5. | FW | Foreign word |
| 6. | IN | Preposition or subordinating conjunction |
| 7. | JJ | Adjective |
| 8. | JJR | Adjective, comparative |
| 9. | JJS | Adjective, superlative |
| 10. | LS | List item marker |
| 11. | MD | Modal |
| 12. | NN | Noun, singular or mass |
| 14. | NNP | Proper noun, singular |
| 15. | NNPS | Proper noun, plural |
| 13. | NNS | Noun, plural |
| 16. | PDT | Predeterminer |
| 17. | POS | Possessive ending |
| 18. | PRP | Personal pronoun |
| 19. | PRP$ | Possessive pronoun |
| 20. | RB | Adverb |
| 21. | RBR | Adverb, comparative |
| 22. | RBS | Adverb, superlative |
| 23. | RP | Particle |
| 24. | SYM | Symbol |
| 25. | TO | *to* |
| 26. | UH | Interjection |
| 27. | VB | Verb, base form |
| 28. | VBD | Verb, past tense |
| 29. | VBG | Verb, gerund or present participle |
| 30. | VBN | Verb, past participle |
| 31. | VBP | Verb, non-3rd person singular present |
| 32. | VBZ | Verb, 3rd person singular present |
| 33. | WDT | Wh-determiner |
| 34. | WP | Wh-pronoun |
| 35. | WP$ | Possessive wh-pronoun |
| 36. | WRB | Wh-adverb |

* **Lemmatization**

Lemmatization is the grouping together of different forms of the same word. In search queries, lemmatization allows end users to query any version of a base word and get relevant results. Because search engine algorithms use lemmatization, the user is free to query any inflectional form of a word and get relevant results. Lemmatization is an important aspect of NLU and NLP and plays an important role in big data analytics and AI. Complex algorithms use the rules of linguistic morphology, in context with a particular language's vocabulary, to group words used in speech and writing by inflected forms.

* **NER**

Named entity recognition (NER) is a sub-task of information extraction that seeks out and categorizes specified entities in a body or bodies of texts. NER is also known simply as entity identification, entity chunking and entity extraction. NER is used in many fields in AI including NLP and machine learning. Information extraction depends on NER to find targeted information, using models that function based on grammar or statistical models. NER recognizes entities first as one of several categories such as people, locations, organizations, expressions, percentages and monetary values. Categories are abbreviated: location (LOC), persons (PER) and organizations (ORG), etc. Once the information category is recognized, an information extraction utility extracts the named entity’s related information and constructs a machine-readable document from it that other tools can further process to extract meaning.

* **Dependency Parser**

A dependency parser analyzes the grammatical structure of a sentence, establishing relationships between "head" words and words which modify those heads. This parser builds a parse by performing a linear-time scan over the words of a sentence. At every step it maintains a partial parse, a stack of words which are currently being processed, and a buffer of words yet to be processed. The parser continues to apply transitions to its state until its buffer is empty and the dependency graph is completed[2].

* **SRL**

In natural language processing, SLR is the process that assigns labels to words or phrases in a sentence that indicate their semantic role in the sentence, such as that of an agent, goal, or result. It consists of the detection of the semantic arguments associated with the predicate or verb of a sentence and their classification into their specific roles.

**ii) Question Generation**

After completing the NLP transformation, we have followed the following steps:

1. Answer phrase selection and generation of question phrases for the selected answer phrase
2. Main verb decomposition
3. Subject-auxiliary inversion
4. Replacement of answer phrase with question phrase and placing at the beginning of the sentence.
5. Post-process to ensure proper formatting.

The proposed system selects a noun phrase (NP) or prepositional phrase (PP) as answer phrase. The algorithm for this is shown in ‘The Answer Phrase Selection Process’.

|  |
| --- |
| **Proc** selectAnsPhrase(*tree*)  **begin**  **if** *tree.firstChild.label* **is** NP **or** *PP*  **then**  **store** *ansPhrase* **as** *tree.firstChild*  **end if**  selectAnsPhrase(*tree.firstChild*)  **end Proc** |

*The Answer Phrase Selection Process*

If an auxiliary verb or modal is not present, the system changes the main verb into the appropriate form of do and the base form of the main verb and the algorithm is supplied to main verb decomposition. Now, the subject-auxiliary inversion is needed to generate grammatically correct questions. In questions, the auxiliary verb is located before the subject. So, it needs to identify the subject and auxiliary verb and invert them. Now, the remaining steps are ‘Answer Phrase Removal’ and ‘Question Phrase Insertion’[7]. In this step, it follows table 3.1. The table specifies the question phrase for each selected answer phrase.

**if** lexeme(tree.label) is not equal to tree.label

**then** aux is equal to tree

**else**

aux is equal to “do”

**else**

aux is equal to “do”

insert aux before tree

**end if**

**else if**

tree.label is MD

aux is equal to tree

**end if**

return

**end if**

decomposeMainVerb(tree.firstChild)

**end Proc**

**Proc** decomposeMainVerb(tree) **begin**

**if** tree.label is VP **then**

**if** tree.hasChild.label is VBZ **then** aux is “does”

replace tree with lexeme(tree.label) insert aux before tree

**else if** tree.hasChild.label is VBD

**then**  aux is “did

replace tree with lexeme(tree.label)

insert aux before tree **else if** tree.label is VB

**then**

*The Method of Main Verb Decomposition*

*Table 2.2: Mapping of Answer Phrase to Question Phrase*

|  |  |  |
| --- | --- | --- |
| **WH Word** | **Conditions** | **Examples** |
| Who | Person or personal pronoun (I, he, herself, them, etc.) | Shyam , he, etc. |
| What | Object (not person or time) | Mountain, book, etc. |
| Where | Location proceded by the preposition (on, in, etc.) | in Nepal |
| When | Time, month, year, day or date | Wednesday |
| Whose | Noun with a possessive (’s or ’) | Ram’s book |

## 2.2 Requirement Collection

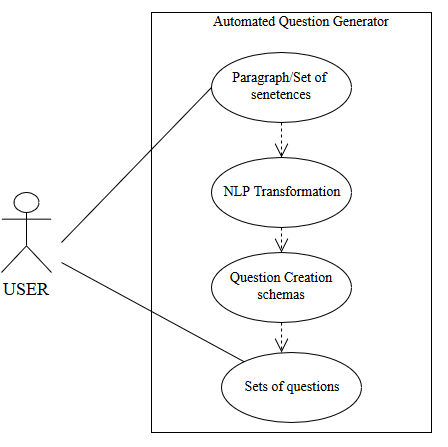
The requirements that is both the functional and non-functional requirements were analyzed.

### 2.2.1 Functional Requirement

Functional requirements describe what the system should do.

**User Function:**

This project will generate the question based on the inputted paragraph using text mining techniques and NLP. The process can be shown as

: 

*Figure 2.1 : Use case diagram to represent Automated Question Generator*

Use case diagram above shows the graphical overview of the actors involved in the system, different functions performed by actors, interaction with the system within environment.

Here, user input paragraph to the system. System then undergoes several NLP transformation and question creation schemas to generate question from that paragraph provided by user.

### 2.2.2 Non-functional Requirement

**i. User Friendly:**

Users with basic knowledge of using internet can use this application. Students/Teachers can insert either a text or paragraph to generate questions.

**ii. Easy Access:**

Automated Question Generator is a web application. Thus, it can be accessed anytime from anywhere with internet connection with the help of web browser.

**iii. Speed of Application:**

Since it is only a model, it may take some time to process which depends on speed of device performance.

## 2.3 Feasibility Study

### 2.3.1 Technical Feasibility

This web application is technically feasible. The software and hardware requirement for the development of this application are not many and already available as free as open source. The work for the project is done with the current equipment and existing software technology.

### 2.3.2 Operational Feasibility

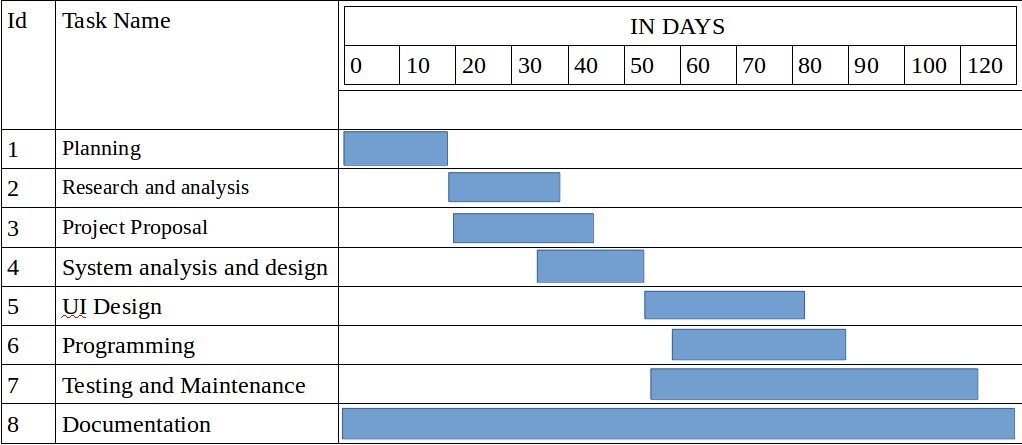
The application will be initially implemented as a web application, so any user with a computer and an internet connection can and use the application. All the features and operations that will be implemented in our project are possible to implement and thus operationally feasible

### 2.3.3 Economic Feasibility

The cost regarding the development of the system is almost minimum. All the Software that will be used during the development process are from the open source community. It can be implemented in almost any computer devices of today’s standard.

### 2.3.4 Schedule Feasibility

With the number of resources and platforms used, the project seems to be completed within the estimated time period. So the system is schedule feasible as per the academic schedule.



*Figure 2.2: Project Scheduling*

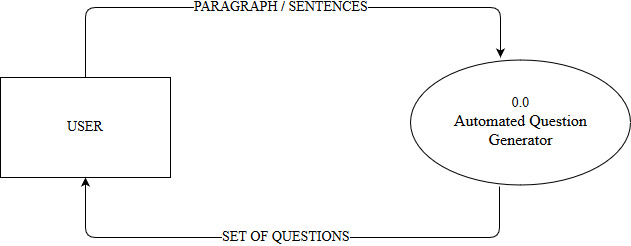
## 2.4 Structuring System Requirements

Another important phase of the system development life cycle is the system design. During this phase, for the interpretation of the findings of the study and analysis, the following designs and diagrams have been developed and thoroughly reviewed.

ER diagram and DFD are used in order to describe the overall features of the secure chat application.

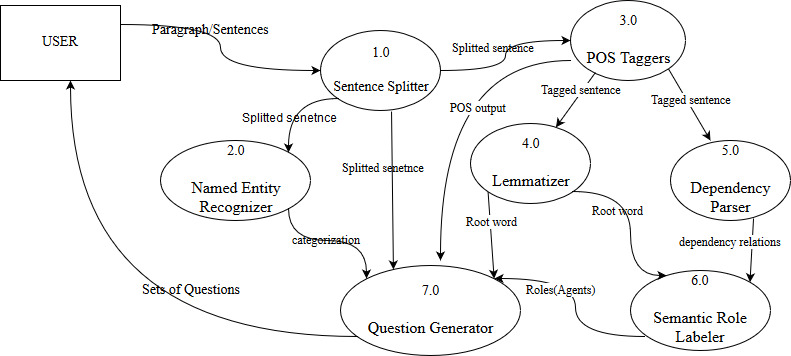
### 2.4.1. Data Flow Diagram

**Context level diagram:**



*Figure 2.3: Context Level Diagram of Automated Question Generator*

Usually user who uses the application must input a paragraph or sentence. The inputted paragraph/ sentence is the served to AQG for further processing which generates Sets of questions as output back to the user.

**Level-1 DFD:**  

*Figure 2.4: Level 1 Diagram of Automated Question Generator*

Level-1 diagram consist of:

* User as an entity. They are the one who uses the application.
* Various processes:
* **Process 1: Sentence Splitter**

The initial sentence is splitted.

* **Process 2: Named Entity Recognizer**

Labels sequence of words in text which are the names of things.

* **Process 3: POS taggers**

Parts of speech are tagged.

* **Process 4: Lemmatizer**

Identify the origin of word

* **Process 5:Dependency Parser**

Analyze grammatical structure of object.

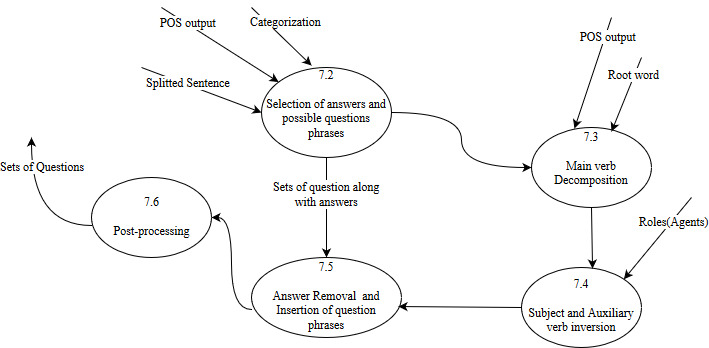
* **Process 6: Semantic Role Labeler**

Indicate role of word in sentence.

* **Process 7:Question Generator**

Generates questions on basis of various fragmentations and validation techniques

**Level-2 DFD:**



*Figure 3.4: Level 2 DFD of Automated Question Generator (Question generator)*

Here, the seventh process is further broken into several sub-processes which helps to explain the working mechanism of the application in detail. The output of NLP transformation are marked and used in each steps. At first unused words are marked which is done through output of NER sentence splitter and POS tagger. Then, possible questions which may be one, more or none with their answers are mapped. Main verb decomposition is done after that, which is done by using root words, roles agent and POS tagged output. Now subject and auxiliary verb are inverted based on dependency relations and roles of each agents. Finally, answers are removed and only valid questions are selected and given as output to the user.

# CHAPTER 3:

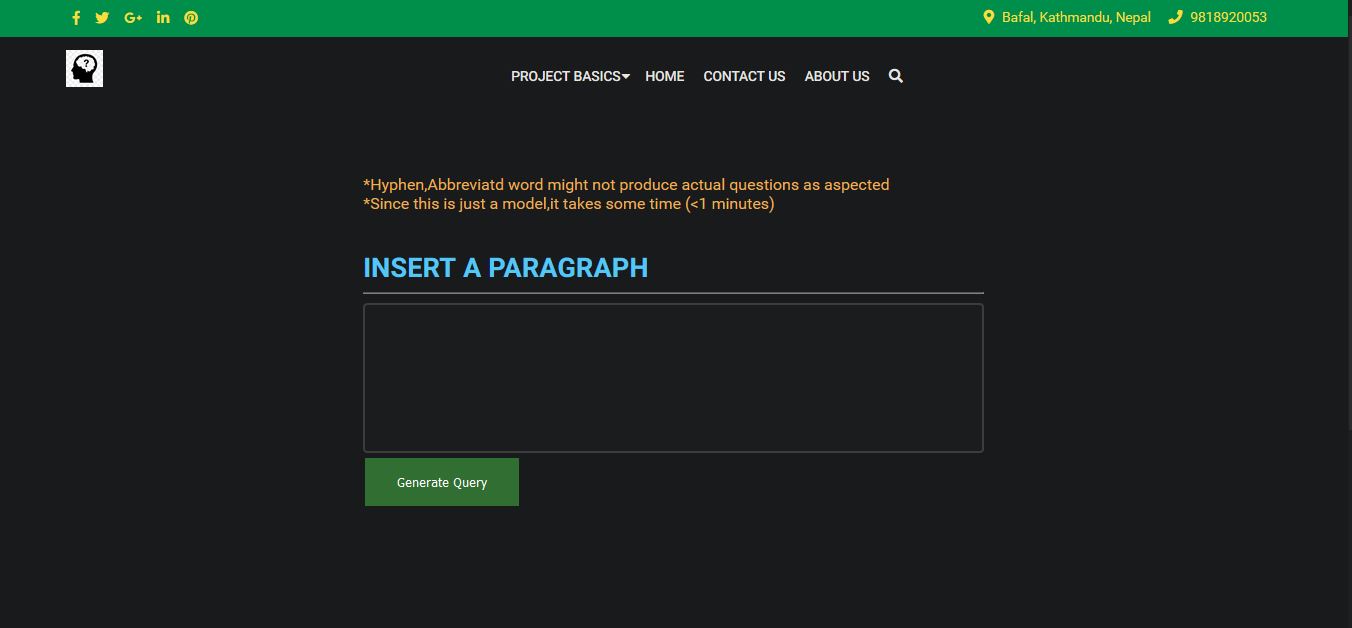
# SYSTEM DESIGN

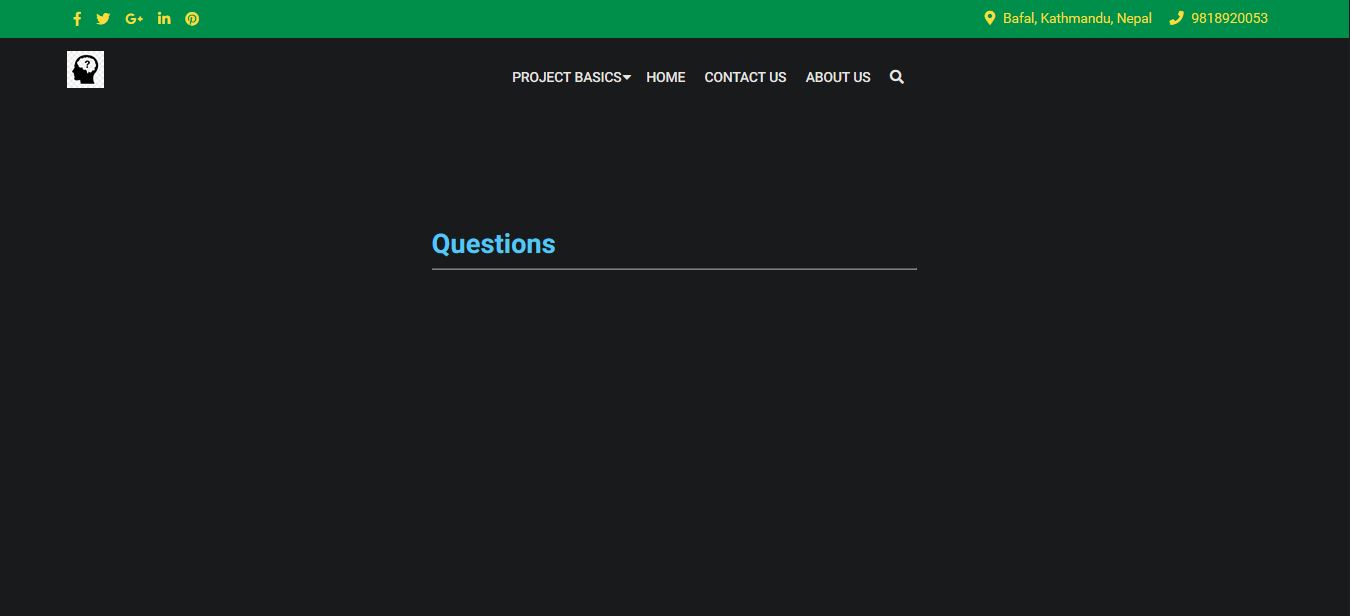
## 3.1 System Design

System design is the process of defining the architecture, modules, interfaces, and data for a system to satisfy specified requirements.

**3.1.2 Interface Design**

Interface of our application is designed using Django framework. This application comprises of basic three interfaces. Home interface is the main page that appears while loading the application initially. Paragraph interface that provides a form layout that accepts a paragraph or set of sentences. Question interface which generates the set of question generated after completion. The interfaces looks like:

 *Figure 3.1: Automated Question Generator (paragraph interface)*

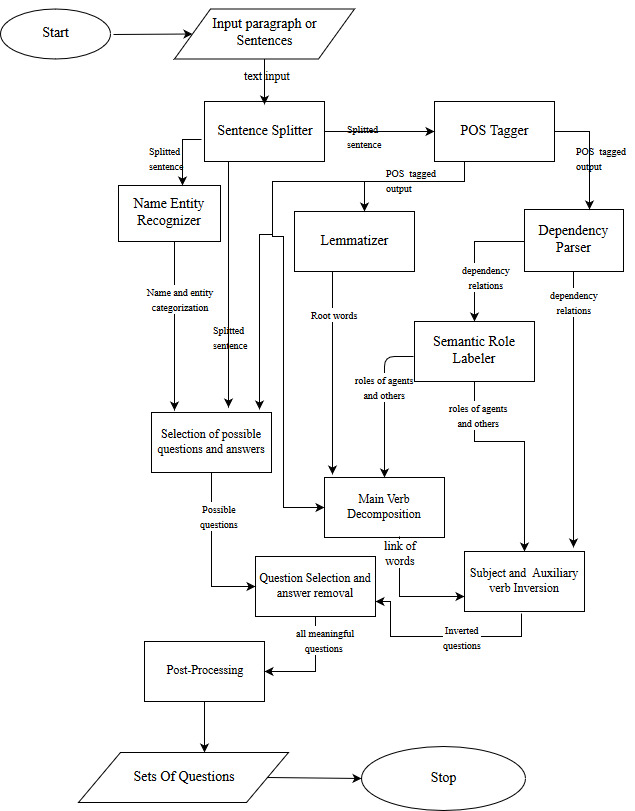


*Figure 3.2: Automated Question Generator (question interface)*

### 3.1.3 Input Output Design

Input design is the link between the system and the user input interface. It comprises of form template that accepts text input called paragraph insertion and output layout that shows the generated question.

### 3.1.4 Process Design

****

*Figure 3.3: Automated Question Generator (working flowchart)*

In order to use this application user should input a paragraph in English language .The first above phase consists of NLP processing which include sentence splitter to split sentence, NER for categorizations of various names and entities. Also, POS tagger for finding parts of speech. Dependency parser and SRL helps in finding dependencies of words and semantic roles of each word respectively .The output of each step is transferred to Question Generator in lower section of process design. Unused word marking, main verb decomposition, subject and auxiliary verb inversion and possible questions are created which is processed and valid question as severed as output to user.

# CHAPTER 4:

# IMPLEMENTATION AND TESTING

## 4.1 Implementation

Various development technologies and tools have been used for the application development.

### 4.1.1 Tools used

Automated question generator is a simple web based application where all the user interfaces are designed using PyCharm and Visual S. Major application tools used for developing the application is categorized into two groups. This can be explained as:

#### 4.1.1.1 Front End Tools:

* **Draw.io**

Draw.io is a website that provides online diagramming tool. It was used to create various diagrams such as ER diagram, DFD, and use-case diagram which helps in understanding the overall flow of the application.

* **Python**

Python is an interpreted, high-level, general-purpose programming language. Python has a design philosophy that emphasizes code readability, notably using significant whitespace. It provided constructs that enable clear programming on both small and large scales. It was used to develop the machine learning model in this project.

* **HTML**

HTML is the standard markup language for creating Web pages. HTML stands for Hyper Text Markup Language. HTML describes the structure of Web pages using markup. HTML elements are the building blocks of HTML pages. It was used to develop the web pages as output of the machine learning model.

* **CSS**

CSS stands for Cascading Style Sheets. CSS describes how HTML elements are to be displayed on screen, paper, or in other media. It is used to maintain the layout of the final web pages

* **Bootstrap**

Build responsive, mobile-first projects on the web with the world’s most popular front-end component library. Bootstrap is an open source toolkit for developing with HTML, CSS, and JS.

#### 4.1.1.2 Back End Tools

* **PIP**

Pip is an application-level package manager for Python that provides a standard format for managing dependencies and required libraries.

* **Django**

Django is the backend web framework for python that helps build web apps much easier and quicker. It was used to support the web-app as backend.

* **GitHub**

GitHub is a version controlling system. GitHub was used for code sharing among the group members.

* **NLTK**

NLTK is a leading platform for building Python programs to work with human language data. It provides easy-to-use interfaces to over 50 corpora and lexical resources such as WordNet, along with a suite of text processing libraries for classification, tokenization, stemming, tagging, parsing, and semantic reasoning, wrappers for industrial-strength NLP libraries, and an active discussion forum.

* **SpaCy**

SpaCy is an open-source software library for advanced Natural Language Processing, written in the programming languages Python. It features convolutional neural network models for part-of-speech tagging, dependency parsing and named entity recognition, as well as API improvements around training and updating models, and constructing custom processing pipelines.

4.1.2 Listing and Description of Major ClassesVarious major classes used can be given as:

* Paragraph From

This class is used to insert a paragraph/sentence for question generation.

* Question Form

This is the class for numbering the question generated using the application.

## 4.2 Testing

In order to verify and validate the application some tests were performed. They are shown below:

### 4.2.1 Application Inputs

In this section, the application was provided by some text inputs. A list of some of these texts is given in Table 4.1.

*Table 4.1: Input Texts*

|  |  |
| --- | --- |
| **Title of the Text** | **Contents** |
| Bangladesh | Bangladesh is a country in South Asia. It shares land borders with India and Myanmar. Nepal, Bhutan and China are located near Bangladesh but do not share a border with it. The country's maritime territory in the Bay of Bengal is roughly equal to the size of its land area. Bangladesh is the world's eighth most populous country. Dhaka is its capital and largest city. Chittagong has the country's largest port. |
| United Nations | The United Nations is an intergovernmental organization tasked to promote international cooperation and to create and maintain international order. A replacement for the ineffective League of Nations, the organization was established on 24 October 1945 after World War II with the aim of preventing another such conflict. At its founding, the UN had 51 member states; there are now 193. The headquarters of the UN is in Manhattan, New York City, and is subject to extraterritoriality. The organization is financed by assessed and voluntary contributions from its member states. |
| Pashupatinath | Pashupatinath is more than just a religious destination. It is a combination of religion, art, and culture. It offers peace and devotion. The temple, spread across 246 hectors of land abounds in temples and monuments. Hundreds of rituals are performed here every day. The temple premises is an open museum. This national treasure was designated a UNESCO World Cultural Heritage Site in 1979.This temple is an important destination for art historians. It displays a variety of temple design some of which are Dome style, Pagoda style, Shikhara style and so on. Additionally there are varieties of statues and sculptures around the complex. There are statues made out of stone, metal, and wood. The door and pillars around the temple area are carved in beautiful shapes of God and griffins. |

### 4.2.2 Question Generation Task

The number of generated questions by human and the system for the texts listed in table 4.1 is provided in table 4.2. On the basis of the number of generated questions, we have evaluated the performance of the developed system. A comparison chart between human and the system on question generation is also shown in Figure 4.1.

|  |  |  |
| --- | --- | --- |
| **Title of the Text** | **No. of Questions Generated by Human** | **No. of Questions Generated by the AQG System** |
| Bangladesh | 20 | 25 |
| United Nations | 20 | 12 |
| Pashupatinath | 28 | 12 |

*Table 4.2: Number of Generated Questions by Human and the AQG System*

*Figure 4.1: A Comparison Chart between Human and the AQG System on Question Generation*

The average number of questions generated by human is (20+20+28) / 3 = 22.67 and the developed system is (25+12+12) / 3 = 16.33. So, the performance of the system on question generation is (13.33/ 22) \* 100% = 72.35% compared to the human.

### 4.2.3 Correctness Analysis of Generated Questions

The generated questions are not always correct (grammatically or syntactically). Here we have analyzed the system on 10 randomly selected questions generated from the texts listed in Table 4.1. The correctness analysis of the selected questions is given in Table 4.3, Table 4.4, and Table 4.5.

*Table 4.3: Correctness Analysis of the Generated Questions from the ‘Bangladesh’ Texts*

|  |  |
| --- | --- |
| **Questions Generated by the AQG System** | **Correctness** |
| What is Bangladesh in South Asia? | Ok |
| Who is a country in South Asia? | Ok |
| What it shares l N with India and Myanmar? | No |
| What do Nepal, Bhutan and China not share with it? | Ok |
| What are Bhutan and China located near Bangladesh but do not share a border with it? | No |
| Who do not share a border with it? | Ok |
| What is roughly equal to the size of its land area? | Ok |
| Who has the country’s largest port? | Ok |
| What is its capital and largest city? | Ok |
| What is the world's eighth most populous country? | Ok |

*Table 4.4. Correctness Analysis of the Generated Questions from the ‘United Nation’ Texts*

|  |  |
| --- | --- |
| **Questions Generated by the AQG System** | **Correctness** |
| Who tasked to promote international cooperation and to create and maintain international order? | Ok |
| Who maintain international order? | Ok |
| What was the organization established on 24 October 1945 after World War II with the aim of preventing another such conflict? | Ok |
| What did the Un had ; there are now 193? | Ok |
| Who was established on 24 October 1945 after World War Ii with the aim of preventing another such conflict? | No |
| What is the organization financed by assessed and voluntary contributions from its member states? | Ok |
| Who preventing another such conflict? | Ok |
| Who is financed by assessed and voluntary contributions from its member states? | Ok |
| Who is subject to extraterritoriality? | Ok |
| How much was the organization established on October 1945 after World War Ii with the aim of preventing another such conflict? | No |

*Table 4.5: Correctness Analysis of the Generated Questions from the ‘Pashupatinath’ Texts*

|  |  |
| --- | --- |
| **Questions Generated by the AQG System** | **Correctness** |
| What is more than just a religious destination? | Ok |
| What it o N and devotion? | Ok |
| Who offers peace and devotion? | Ok |
| What are Hundreds of rituals performed? | Ok |
| Who are performed here every day? | No |
| Who is an open museum? | No |
| When this national treasure was designated a Unesco World Cultural Heritage Site? | Ok |
| What was this national treasure designated in 1979? | Ok |
| What is this temple for art historians? | Ok |
| Who are carved in beautiful shapes of God and griffins? | Ok |

The percentage of accuracy of the generated questions from the texts listed in table 3 is 80%, 90%, and 80% respectively. These accuracy values are also plotted in Figure 4.2 to get the accuracy graph of the AQG system.

*Figure 4.2: The graph of percentage of Accuracy*

# CHAPTER 5:

# CONCLUSION AND FUTURE ENHANCEMENT

## 5.1 Conclusion

The web application was successfully developed to generate questions from the English text served as output. As it can be seen question generation after reading a text time and again is tedious task and it spends some time. But with the help of this application, question from a paragraph or set of sentences can be easily created and drafted .Doing so save some time and pressure.

To achieve this facility, user just have to input a sentence .The input sentence passes through the sets of NLP transformation and question generation schemas to give valid question to the user based on the text that is sent as input.

To conclude, the project was successfully developed for generating automatic questions based on the text provided by the user which help in reducing time taken for paragraph understanding and question generation after analyzing.

## 5.2 Future Enhancement

Main objective of this project was to reduce time taken for analyzing and generating questions from the text provided by the user. Although, the application generate the questions for simple sentences the web application needs more of the features that needs to be added to improve its performance.

Some of them can be given as:

* Mathematical and regular expression can be improved for the system.
* Comma, hyphen can be added to the system.
* Various paragraphs questions with non-repetition can be made.
* Abbreviations can be introduced in the system.
* Response time can be reduced in future.
* Questions along with answer can be generated.
* Inference rules can be implemented for deductive question generation.
* Yes/No questions can be added further more.

# REFERENCE

[1] https://www.researchgate.net/figure/Preprocessing-Flow-Chart\_fig1\_228944445

-Preprocessing Flow Chart

[2] https://www.enotes.com/homework-help/how-you-parse-sentence-example-parse-this- sentence-576636

-How to parse sentence with example

[3] https://en.wikipedia.org/wiki/Named-entity\_recognition

-Named Entity Recognition

[4] https://en.wikipedia.org/wiki/Syntactic\_Structures

-Syntactic Structures

[5] https://www.britannica.com/topic/Syntactic-Structures

-Syntactic Structures

[6] https://towardsdatascience.com/a-review-of-named-entity-recognition-ner-using-automatic-summarization-of-resumes-5248a75de175

- A review of named entity recognition using automatic summarization

[7] https://www.quora.com/How-does-Stanford-NER-work

- How does Stanford-NER WORK?