Automatic Question and Answer Generation from Course Materials.

A.S.M Nibras, M.F.F Mohamed, I.S.M Arham, A.M.M Mafaris, M.P.A.W Gamage

Sri Lanka Institute of Information Technology

Abstract – This paper presents a Question and Answer Generating System based on the approach of Natural Language Processing. As an examiner needs to ask suitable and good questions and prepare correct answers. The potential benefits this system may assist examiners meeting appropriate questions and answers. When a text input is given, this system deeply analyzes the text content and generates a set of questions and suitable answers. This system functionally divided into four main steps namely, identifying key phrases and words, forming proper question, validating generated questions and Find answers for validated question. When analyzing text, a named entity recognizer and a part of speech are applied on each of these sentences to extract necessary information. Also classify the sentences based on their subject, verb, object and preposition for determining the possible type of questions to be generated. Once questions are generated, for the further accuracy, system using a custom developed algorithm for validating the questions. Then, suitable answer for the question will be generated. This paper provides an overview of Question and Answer Generation and a discussion on possible approaches found in the recent research as a guide to the development of an automated system.

Key words: Natural Language Processing, Natural language tool kit, Part of speech tag, Named entity recognition, Information extraction

1. Introduction

Exams plays an important role in teaching and learning process at all level of education. Proper assessment activities help to sustain learners' interest in learning the domain knowledge [1]. Traditionally conducting exam is the successful method to identify and improve the knowledge of the student. Exam is an easy tool to regularly assess a student's capability. Exams promote competition among students. They work harder to improve their knowledge and skills. In this way, they learn more. The important task of conducting exam paper is preparing questions and answers. Preparing questions manually is very difficult task. Examiner has to go through the whole context and come up with questions. According to our studies, most examiners saying that this is very time-consuming task and as a human, they can be a chance to omit some important content.

Based on these problems we have come up with a solution "Automated Question and Answer Generation System". This system is using Natural language processing approach to analyze the text sentence by sentence and generate all possible questions. These questions will be validated by considering various English grammar factors. Once questions are validated, answers for the relevant question will be picked from the text input. From the user's perspective, user has to input text material, all possible questions with answer will be generated and user has to pick relevant questions from the list.

2. Related Works

There are some Question Generation systems are implemented by many researchers. The main flow of all projects is analyzing the text materials and generating different type of questions such as essay type questions, Multiple Choice Question(MCQ), gap-fill multiple choice questions

I. Revup: Automatically generating questions from educational texts

RevUp is Automatically generates gap-fill multiple choice questions from online text. The system analyzes online text and finds most important sentences, then select the main gap-phrases from the selected sentences and choose distractors that are semantically and syntactically similar to the gap-phrase and have contextual fit to the gap-fill question [4].

II. Question Generation as a Competitive Undergraduate Course Project

Some groups of students in Carnegie Mellon University(USA) created Question and Answer Generation projects for them under graduate Natural Language Processing course. Each group tried with different techniques of Natural Language Processing(NLP) such as Language Modeling, Part of Speech Tagging, Named Entity Recognition, Parsing etc. And they allowed for students to use any programming language and any existing NLP components to complete their projects [3]. They created command line interface for question generation program.

When we compare our Automatic Question and Answer Generating System with the above-mentioned systems, our system has some additional functions to generate good quality question without having grammar errors. The first described system generating gap fill questions not WH questions and they not generate answers for the questions. But our system generates WH questions and generating answers as well. And, the both above described projects are not validate the generating questions. So that questions may have grammatical errors. But our system validates the generated systems and add ranking for the questions. And we generate answers for the questions which gets high rank.

3. Methodology

A collection of several methods and approaches are been used in the research to provide an accurate and effective of results. This section describes the approaches that has been used in this system. For the better understand, this has divided into four sub sections and explained.

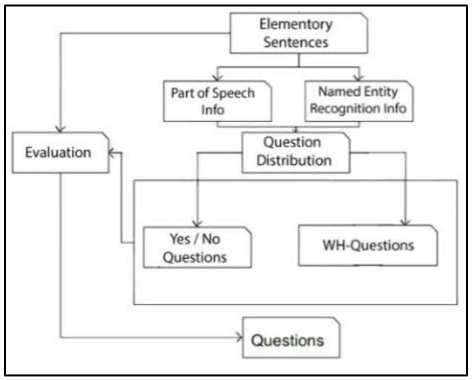


Figure:01 – Question Generation Architecture

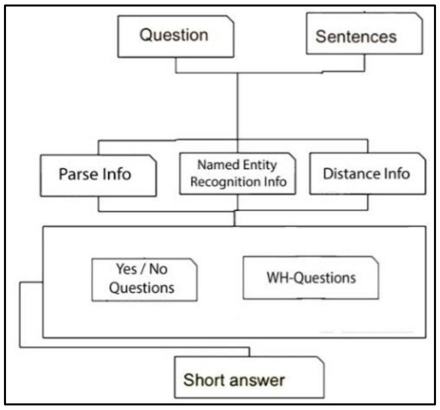


Figure: 02 – Answer Generation Architecture

I. Filtering the text content and identifying the key phrases, words that should be used to generate appropriate questions.

The entry point of the system is this stage. When user input text content in order to generate questions and answers, the text content will be put into the process of Information Extraction. Information Extraction is the task of automatically extracting structured information from unstructured and/or semi-structured documents. In most of the cases this activity concerns processing human languages texts by means of Natural language processing. The goal of Information extraction is to identify the Named entities such as the names of persons, organizations, location, date, time...etc. There are several steps to identify the named entities [2]. First step is sentence segmentation. Here, text content will be tokenized into sentence and proceed to next step word tokenization. Here sentences are further tokenized into words and proceed to next step Part of speech tagging. A part-of-speech tagger, or POS-tagger, processes a sequence of words, and attaches a part of speech tag to each word. For English language, there are set of universal part-of-speech tags available for example, ADJ – adjective, ADV- adverb, DET – determiner, NOUN – noun, PRON – pronoun, VERB – verb …etc. So, when we tokenize sentence into words, our system can map these words with respective part of speech tag using Natural language tool kit. The basic technique here we used for entity detection is Chunking, in simple sentence, Chunking is a term referring to the process of taking individual pieces of information (chunks) and grouping them into larger units. In our system, we use part of speech tags as chunks and group them by using custom grammar. Grammar is defining with regular expressions.

sentence = [("the", "DT"), ("little", "JJ"), ("yellow", "JJ"), ("cat", "NN"), ("ate", "VBD"), ("the", "DT"), ("rat", "NN")]

grammar = "NP: {<DT>?<JJ>*<NN>}"

With the help of this grammar, we can group each piece into a large whole set, there we can avoid unnecessary words from the input text content. At the end of this chunking process we can get the NP-Noun Phrases of the text. Final step of Information Extraction is to identify the Named Entities. We have to find suitable named entity for these noun phrases. We can achieve this with the help of approaches of Natural language processing. Each noun phrases are marked with respective named entities. For the better accuracy, grammar will be construct much better in order to identify all possible noun phrases. These named entities will be matched with relevant noun phrase and stored. Similarly, verbs with it tense will be identified with part of speech tag and custom written grammar (using Regular expressions). The important factor in verb is identifying tense. Different grammar will be used for different types of verb tense. All these different grammars are based on different part of speech tags. These are the set of part of speech verb tags. VB – Verb base form, VBD –past tense, VBG – present participle, VBN – past participle, VBP – non 3rd person singular present, VBZ - 3rd person singular present. From these part of speech tags system can understand the tense of the verb.

Once system identified all these parts of speech tag, noun phrases, named entities, verbs with it tense all of these information will map with the relevant sentence of the source text content and pass it the step 2 of the process.

II. Forming proper questions using the identified key phrases from the context.

In "Automation of question generation from sentences" [6] system, By using syntactic information they simplify the process by extracting elementary sentences from the complex sentences. In the next phase, based on the sentences subject, verb, object and preposition they classify the sentences to determine the possible type of questions that will be generated.

In "Using Automatic Question Generation to Evaluate Questions Generated by Children" [7] system, they have worked on the syntactical parse tree that is created for each sentence. Grammar based approach could be more time consuming, but their algorithm works on the parse tree for each sentence. They feed into their system questions from the Web Questions dataset [8] which creates a rule by itself used to generate the questions.

In our proposed approach, the question generation component takes as input a declarative sentence and produces as output a set of possible questions. It looks for the answer phrases that may be targets for WH-movement and converts them into question phrases. In the current system, answer phrases can be the following.

- Noun phrases (e.g. Mahinda Rajapaksa)
- Prepositional phrases (e.g. in 2007)
- Subordinate clauses (e.g. that Mahinda Rajapaksa was the 6th Sri Lanka President)

The system does not generate questions from all of the phrases of these types. There are certain constraints on the set of possible answer phrases. Also, note that the system does not generate questions about verb phrases (e.g.,What happened to Mahinda Rajapaksa in 2007?). From the possible answer phrases, the system generates questions with the following question words: Who, what, where, when, whose, and how many.

The system could be extended to detect and transform other types of phrases to produce other types of questions (e.g., how, why, and what kind of) since the transformation from answer to question is achieved through a composition of general-purpose rules. This would allow, for example, the addition of a relatively simple rule to generate what kind of questions by building off of the existing rules for subject-auxiliary inversion, verb decomposition, etc. The system also generates yes-no questions, as discussed below.For each sentence, many questions may be produced: there are often multiple possible answer phrases in a particular sentence, and multiple question phrases for each answer phrase. For example, from "John met Lecturer", we generate "Who met Lecturer?" "Who did John meet?" and "Did John meet Lecturer?"This question generation component aims to over generate grammatical, though not concise or specific, questions. The transformation rules in this component, which encode a substantial amount of linguistic knowledge, are applied in the following order:

1. Mark phrases that cannot be answer phrases, due to WH-movement constraints.
2. Select an answer phrase, and generate a set of question phrases for it.
3. Decompose the main verb.
4. Invert the subject and auxiliary verb.
5. Remove the answer phrase and insert one of the question phrases at the beginning of the main clause.
6. Post-process to ensure proper formatting.

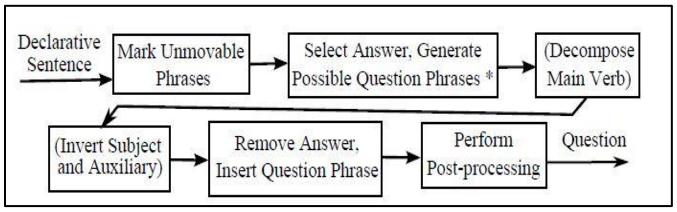

Declarative Sentence
Mark Unmovable Phrases
Select Answer, Generate Possible Question Phrases *
(Decompose Main Verb)
(Invert Subject and Auxiliary)
Remove Answer, Insert Question Phrase
Perform Post-processing
Question

Figure: 03

Figure:03 illustrates the question generation process. We use the term "question phrase" to refer to the phrase at the start of a question that includes a WH word (e.g., who, how many).There are two important special cases. First, when generating yes-no questions, there is no answer phrase to remove nor question phrase to insert. Second, when generating questions for which the answer phrase was the subject of the declarative sentence, decomposition of the main verb and subject-auxiliary inversion are not necessary. The subject is removed and replaced by a question phrase in the same position (e.g., John met Lecturer becomes Who met Lecturer?). Additionally, the main verb may need to be altered to agree with the question phrase if the subject is a first- or second person pronoun.

Marking Unmovable Phrases

A set of Tregex expressions is used to mark the phrases in an input tree which cannot be answer phrases due to constraints on WH-movement. Each parse tree node subject to a movement constraint is renamed by extracting the node's current label and adding a special prefix marker ("UNMOVABLE-") to its label. Then, in the following step of answer phrase selection, these nodes are skipped so that the system does not generate questions for them.

Generating Possible Question Phrases

After marking unmovable phrases, the system iterates over the possible answer phrases, generating possible questions for each one. To generate the question phrases, the system annotates the source sentence with a set of high level semantic types using the super sense tagger. For a given answer phrase, the system uses these high-level semantic tags along with the syntactic structure to generate a set of one or more possible question phrases, each of which is used to generate a final question sentence. Recall that answer phrases can be noun phrases (labeled "NP"), prepositional phrases (labeled "PP"), or subordinate clauses (labeled "SBAR"). For noun phrases, the system uses the conditions to decide what question phrases to generate. For subordinate clause answer phrases, the system only extracts the question phrase what (e.g., what did John notice? from John noticed that the floor was wet). For prepositional phrases, the system extracts the object of the preposition (a noun phrase) and uses the conditions to generate WH phrases from it.

Decomposition of the Main Verb

In order to perform subject-auxiliary inversion, if an auxiliary verb or modal is not present, the system decomposes the main verb into the appropriate form of do and the base form of the main verb. It then modifies the tree structure of the verb phrase accordingly. For example, John saw Lecturer becomes John did see Lecturer before transformation into Who did John see? Rather than *Saw John Lecturer?
If an auxiliary verb is already present, however, this decomposition is not necessary (e.g., John has seen Lecturer could lead to Who has John seen?). In such cases, the main verb phrase includes the auxiliary verb and a nested verb phrase containing the base form of

the main verb. The system identifies main verbs that need to be decomposed with the Tregex expression.

Subject-Auxiliary Inversion

The system performs subject-auxiliary inversion either when the answer phrase is a non-subject noun phrase or when the question to be generated is a yes-no question. For example, if generating questions from the sentence Burr shot Hamilton, subject-auxiliary inversion would be performed when generating

Removing Answers and Inserting Question Phrases

In the next step, the system removes the selected answer phrase, and for each possible question phrase generated from the answer phrase, it inserts that question phrase into a separate copy of the current tree to produce a new candidate question. The question phrase is inserted as a child of the "SBARQ" node under the root node (as shown in Figure 3), following any leading sentence-level adjunct phrases. Note that if the question is a yes-no question, then this step is not necessary. If the answer phrase is a prepositional phrase, then only the object of the preposition is moved. This leads to questions such as What is the building near? which we judged to be more natural than questions such as Near what is the building? The only exception is where and when questions, for which the preposition is also removed (e.g., to generate Where did John run? rather than in where did John run? from the input sentence John ran in the park.).

Post-processing

Some additional post-processing mechanisms are necessary to ensure proper formatting and punctuation. Sentence-final periods are changed to question marks. Additionally, the output is de-tokenized to remove extra whitespace symbols (e.g., spaces preceding punctuation symbols).

III. Validating Generated Questions and Ranking them by focus on Linguistic Factors

The Question Generation function will generate more than one question from a content. So, the function may be generating wrong questions. But the system should give the good meaningful questions for the user. So, the system should validate the generated questions.

There are several stages to validate the generated questions. Such as,

1. Check the correctness of added auxiliary verbs.
2. Check the correctness of used WH word.
3. Check other grammatical errors.
4. Add ranking according to the identified errors.

We have implemented algorithms for the above validate stages with the usage of Natural Language Processing.

Check the correctness of added auxiliary verbs

Auxiliary (or Helping) verbs are used together with a main verb to show the verb's tense or to form a negative or question. The most common auxiliary verbs are **have**, **be**, and **do**. If we look of common WH questions, the first word will be the WH word, the second word will be the auxiliary verb and the third word will be pointing the human or object.

For example;

What did you watch on TV last week?

Here, the second word is the auxiliary verb(*did*). The auxiliary verb depends on noun (the third word). If the noun represents human (like, I, you, we) or noun plural then the auxiliary should be *do*, *did*, *have* etc. otherwise the auxiliary verb should be *does*, *did*, *has* etc.

So according to this concept, our implemented algorithm will tokenize the question into words using NLTK [1] and it will check the noun and the auxiliary verb whether the auxiliary verb is correctly added or not.

Check the correctness of used WH word.

WH-question is a term in generative grammar for a question that is formed with an interrogative word (what, who, whom, whose, which, when, where, why, or how) and that expects an answer with information other than "yes" or "no". The WH word of a question depends on the noun and the verb/auxiliary verb of the question. So, when we check whether the added WH word is correct or not, we should analyze the noun and the verb of the question.

Based on this common English grammar rule, our implemented algorithm first tokens the question into single words and store it into an array and it will tag the tokens by using POS tag [1] and filtered out each word of the question as type of grammar such as NN (noun singular), NNS (noun plural), VBD (verb past tense) etc. [1]. But there is no any grammar rule to find the WH word of the question. Because some questions may be grammatically correct if we added different WH words.

For example;

a) *What is that?*
b) *Who is that?*
c) *Where is that?*

Here, the questions a, b and c are grammatically correct and give correct meanings as well. There for there is no any grammatical rule in English. So, we have implemented a common set of rules to check the suitable WH word of the question. The rule may not be support for all the WH questions, but we can check most of the common simple WH questions. Our Automatic Question and Answer Generating System will generate simple WH questions.

Rule 1

If the last word of the question is noun(NN) or noun plural(NNS) and the auxiliary verb of the question is Verb, non-3rd person singular present(VBP) or Verb, past tense(VBD) or "Does" then the most suitable WH words are "What", "Why" and "When".

For example;

a) *What do we wear on our foot?*
b) *Why does student go to school?*
c) *When did you last see a doctor?*

Rule 2

If the last word of the question is noun(NN) or noun plural(NNS) and the auxiliary verb of the question is Verb, 3rd person singular present(VBZ) then the most suitable WH word is "Who".

For example

a) *Who takes care of animals at the zoo?*

Rule 3

If the last word of the question is verb(VB) then there is a high probability for the WH word to be "Where"

a) *When did you go on your last vacation?*

According to the above three rules we have implemented an algorithm to check and compare with the added question's WH word whether it is correct WH word or not.

Check other grammatical errors

Other than the Auxiliary Verb and WH word, there may be some other grammatical errors in the generated questions. Such as, there may be wrong verb/auxiliary verb added to the question when we check it with the noun.

For example;

a) *What did you **watched** on TV last week?*
b) *Why do we **has** teeth? =>*

In the example a, the verb is in past tense but auxiliary verb is "*did". So, the verb should be in present simple(*watch*).* And in the question b, there is a Personal Pronoun(*we*). So, helping verb should be "have". To check these kind of grammar errors of the question we use Language Check module which is available in python. Using that module, we check the question and find out the grammatical error and it will suggest some suitable words for replace the error. We take those suggestions and find out most suitable word to replace the error and we display the question without those errors.

Add ranking according to the identified errors.

According to the above three methodologies to find out the errors in the question, we add rankings for those questions. We add ranking out of 10. If a question has only the auxiliary verb error then we reduce 3 out of 10. If a question has only the WH word error then we reduce 4 out of 10. And if a question has other grammatical mistakes then we reduce 3 out of 10. Likewise, the ranking will be reducing according to the available mistakes. If a question gets 10 out of 10 then the question does not have any errors. That will be a good question. And that question will be passing to the other section to find out the answer.

IV. Generating Suitable Answers for the Questions

The answer generation component takes two inputs that are declarative sentences and the list of questions. With the help of both inputs the most suitable answer will be generated.

First step the type of the question has to be identified in order to decide the way of analyzing the sentences. The system the reads the questions one by one and parsing the questions to the Stanford parser and identify the initial labels. By analyzing the initial labels, we can determine the type of the question whether that is a binary question or 'WH questions.

Binary questions represent only yes or no questions. Which means these kinds of questions can be answered with yes or no.

'WH' question means the question words starting with 'WH' letters (Example: What, When, Why, Who).

Extract the keyword sentence

Then have to invert the subject in the question and also have to remove the question word to convert the question to its predicate format. Because the predicate format is more likely to be present in the declarative sentences.

Once the format is converted to its predicate format can use the co-sine similarity checking for all the sentences similar to the converted sentence. The highest matching sentence will be retrieved to generate the answer.

Process the answer generation

The final task of generating the answer is to extract the exact answer from the highest matching sentence so as to be able to present the answer like" Maithiripala Sirisena" to the question "Who is the current president of Sri Lanka?"

The algorithm has been applied to the answer extraction task is based on **answer type pattern extraction.**

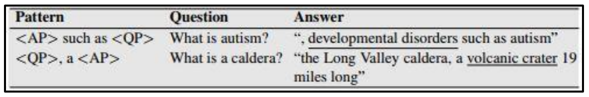
In the **pattern extraction** methods for the answer generation we use the information about the expected answer type together with regular expression patterns. For example, as I mentioned the above example in that case the answer type is a HUMAN. We run the answer type or named entity tagger on the particular candidate sentence and return the entity which is labeled with the type of HUMAN.

In the following examples, the underlined named entities are extracted from the sentence.

"How tall is Pidutalagala Mountain?"
The height of Pidutalagala Mountain is 2524 meter
"Who is the current president of Sri Lanka?"
Maithiripala Sirisena is a Sri Lankan politician who is the 7th and current president of Sri Lanka.

But this is not suitable for some questions such as DEFINITION questions, those questions may not have a particular named entity type. Then instead of using answer types we use hand written regular expression patterns to generate the answer.

These patterns are also useful in cases in which a passage contains multiple examples of the same named entity type. Figure 1.1 shows some patterns from [01] for the question phrase (QP) and answer phrase (AP) of definition questions.



Pattern	Question	Answer
<AP> such as <QP>	What is autism?	", developmental disorders such as autism"
<QP>, a <AP>	What is a caldera?	"the Long Valley caldera, a volcanic crater 19 miles long"

Figure: 03 - Some answer-extraction patterns for definition questions [5]

4. Future Work

This system has fulfilled the fundamental problem that has raised during the research study. Still there are some limitations exists due to lack of time duration and other considerable issues. During the testing, system has performed as we expected. But there are paths where this research work can be extended. The main and important problem that we have faced during this research was accuracy of the results. Since we are dealing with human language, there may be lack accuracy due to complex of the user input. Accuracy can be improved in the future. Other than the accuracy, this research work is done for English language using the approach of Natural language processing. This approach supports other language like French, German, Dutch, Russian, Spanish…etc. As a future work, this research can be extending to other above-mentioned languages as well.

References

[1] Diyana Syamsiyah Rozali, Fadzil Hassan and Norshuhani Zamin "A Survey on Adaptive Qualitative Assessment and Dynamic Questions Generation Approaches" Information Technology (ITSim), 2010 International Symposium in, Volume:3

[2] Jing Jiang "Information Extraction from Text" Singapore Management University – 2012

[3] Noah A. Smith, Michael Heilman, Rebecca Hwa, "Question Generation as a Competitive Undergraduate Course Project" University Pittsburgh, PA, 2008

[4] Girish Kumar, Rafael E. Banchs, Luis F. D'Haro "RevUP: Automatic Gap-Fill Question Generation from Educational Texts" 2015

[5] Pasca, M. (2003). Open-Domain Question Answering from Large Text Collections. CSLI Publications, Stanford University

[6] Ali, Husam, Yllias Chali, and Sadid A. Hasan. "Automation of question generation from sentences." Proceedings of QG2010: The Third Workshop on Question Generation. 2010

[7] Chen, Wei, and Jack Mostow. "Using Automatic Question Generation to Evaluate Questions Generated by Children." The 2011 AAAI Fall Symposium on Question Generation. 2011.

[8] Jonathan Berant, Andrew Chou, Roy Frostig, Percy Liang. Semantic Parsing on Freebase from Question-Answer Pairs. Empirical Methods in Natural Language Processing (EMNLP), 2013